

Expressiveness and Closure Properties for Quantitative Languages

Krishnendu Chatterjee¹, Laurent Doyen², and Thomas A. Henzinger²

¹ IST, Austria

² EPFL, Lausanne, Switzerland

Abstract. Weighted automata are nondeterministic automata with numerical weights on transitions. They can define quantitative languages L that assign to each word w a real number $L(w)$. In the case of infinite words, the value of a run is naturally computed as the maximum, limsup, liminf, limit average, or discounted sum of the transition weights. We study expressiveness and closure questions about these quantitative languages.

We first show that the set of words with value greater than a threshold can be non- ω -regular for deterministic limit-average and discounted-sum automata, while this set is always ω -regular when the threshold is isolated (i.e., some neighborhood around the threshold contains no word). In the latter case, we prove that the ω -regular language is robust against small perturbations of the transition weights.

We next consider automata with transition weights 0 or 1 and show that they are as expressive as general weighted automata in the limit-average case, but not in the discounted-sum case.

Third, for quantitative languages L_1 and L_2 , we consider the operations $\max(L_1, L_2)$, $\min(L_1, L_2)$, and $1 - L_1$, which generalize the boolean operations on languages, as well as the sum $L_1 + L_2$. We establish the closure properties of all classes of quantitative languages with respect to these four operations.

1 Introduction

A boolean language L can be viewed as a function that assigns to each word w a boolean value, namely, $L(w) = 1$ if the word w belongs to the language, and $L(w) = 0$ otherwise. Boolean languages model the computations of reactive programs. The verification problem “does the program A satisfy the specification B ?” then reduces to the language-inclusion problem “is $L_A \subseteq L_B$?”, or equivalently, “is $L_A(w) \leq L_B(w)$ for all words w ?”, where L_A represents all behaviors of the program, and L_B contains all behaviors allowed by the specification. When boolean languages are defined by finite automata, this elegant framework is called the *automata-theoretic approach* to model-checking [VW86].

In a natural generalization of this framework, a cost function assigns to each word a real number instead of a boolean value. For instance, the value of a word (or behavior) can be interpreted as the amount of some resource (e.g., memory

consumption, or power consumption) that the program needs to produce it, and a specification may assign a maximal amount of available resource to each behavior, or bound the long-run average available use of the resource.

Weighted automata over semirings (i.e., finite automata with transition weights in a semiring structure) have been used to define cost functions, called formal power series for finite words [Sch61,KS86] and ω -series for infinite words [CK94,DK03,ÉK04]. In [CDH08], we study new classes of cost functions using operations over rational numbers that do not form a semiring. We call them *quantitative languages*. We set the value of a (finite or infinite) word w as the maximal value of all runs over w (if the automaton is nondeterministic, then there may be many runs over w), and the value of a run r is a function of the (finite or infinite) sequence of weights that appear along r . We consider several functions, such as **Max** and **Sum** of weights for finite runs, and **Sup**, **LimSup**, **LimInf**, limit average, and discounted sum of weights for infinite runs. For example, peak power consumption can be modeled as the maximum of a sequence of weights representing power usage; energy use can be modeled as the sum; average response time as the limit average [CCH⁺05,CdAHS03]. Quantitative languages can also be used to specify and verify reliability requirements: if a special symbol \perp is used to denote failure and has weight 1, while the other symbols have weight 0, one can use a limit-average automaton to specify a bound on the rate of failure in the long run [CGH⁺08]. The discounted sum can be used to specify that failures happening later are less important than those happening soon [dAHM03].

The *quantitative language-inclusion problem* “Given two automata A and B , is $L_A(w) \leq L_B(w)$ for all words w ?” can then be used to check, say, if for each behavior, the peak power used by the system lies below the bound given by the specification; or if for each behavior, the long-run average response time of the system lies below the specified average response requirements. In [CDH08], we showed that the quantitative language-inclusion problem is PSPACE-complete for **Sup**-, **LimSup**-, and **LimInf**-automata, while the decidability is unknown for (nondeterministic) limit-average and discounted-sum automata. We also compared the expressive power of the different classes of quantitative languages and showed that nondeterministic automata are strictly more expressive in the case of limit-average and discounted-sum.

In this paper, we investigate alternative ways of comparing the *expressive power* of weighted automata. First, we consider the cut-point languages of weighted automata, a notion borrowed from the theory of probabilistic automata [Rab63]. Given a threshold $\eta \in \mathbb{R}$, the cut-point language of a quantitative language L is the set of all words w with value $L(w) \geq \eta$, thus a boolean language. We show that deterministic limit-average and discounted-sum automata can define cut-point languages that are not ω -regular. Note that there exist ω -regular languages that cannot be expressed as a cut-point language of a limit-average or discounted-sum automaton [CDH08]. Then, we consider the special case where the threshold η is isolated, meaning that there is no word with value in the neighborhood of η . We argue that isolated cut-point languages have

stability properties, by showing that they remain unchanged under small perturbations of the transition weights. Furthermore, we show that every discounted-sum automaton with isolated cut-point defines an ω -regular language, and the same holds for deterministic limit-average automata. This question is open for nondeterministic limit-average automata. Finally, we consider a boolean counterpart of limit-average and discounted-sum automata in which all transitions have weight 0 or 1. Of special interest is a proof that limit-average automata with rational weights in the interval $[0, 1]$ can be reduced to automata with boolean weights. Therefore, the restriction to boolean weights does not change the class of quantitative languages definable by limit-average automata; on the other hand, we show that it reduces that of discounted-sum automata.

In the second part of this paper, we study the *closure properties* of quantitative languages. It is natural and convenient to decompose a specification or a design into several components, and to apply composition operations to obtain a complete specification. We consider a natural generalization of the classical operations of union, intersection and complement of boolean languages. We define the *maximum*, *minimum*, and *sum* of two quantitative languages L_1 and L_2 as the quantitative language that assigns $\max(L_1(w), L_2(w))$, $\min(L_1(w), L_2(w))$, and $L_1(w) + L_2(w)$ to each word w . The *complement* L^c of a quantitative language L is defined by $L^c(w) = 1 - L(w)$ for all words w .³ The sum is a natural way of composing two automata if the weights represent costs (*e.g.*, energy consumption). We give other examples in Section 2 to illustrate the composition operations and the use of quantitative languages as a specification framework.

We give a complete picture of the closure properties of the various classes of quantitative languages (over finite and infinite words) under maximum, minimum, complement and sum (see Table 1). For instance, limit-average automata are not closed under sum and complement, while nondeterministic discounted-sum automata are closed under sum but not under complement. All other classes of weighted automata are closed under sum. For infinite words, the closure properties of *Sup*-, *LimSup*-, and *LimInf*-automata are obtained as a direct extension of the results for the boolean finite automata, while for *LimAvg*- and *Disc*-automata, the proofs respectively require the analysis of the structure of the automata cycles and properties of the solutions of polynomials with rational coefficients. Note that the quantitative language-inclusion problem “is $L_A(w) \leq L_B(w)$ for all words w ?” reduces to closure under sum and complement because it is equivalent to the question of the non-existence of a word w such that $L_A(w) + L_B^c(w) > 1$, that is an *emptiness* question which is decidable for all classes of quantitative languages [CDH08]. Also note that deterministic limit-average and discounted-sum automata are not closed under maximum, which implies that nondeterministic automata are strictly more expressive in these cases (because the maximum can be obtained by an initial nondeterministic choice).

Related work. Functions such as limit average (or mean payoff) and discounted sum have been studied extensively in the branching-time context of game the-

³ One can define $L^c(w) = k - L(w)$ for any constant k without changing the results of this paper.

ory [Sha53, EM79, Con92, ZP96, CdAHS03]. It is therefore natural to use the same functions in the linear-time context of languages and automata.

Weighted automata with discounted sum have been considered in [DR07], with multiple discount factors and a boolean acceptance condition (Muller or Büchi); they are shown to be equivalent to a weighted monadic second-order logic with discounting. Several other works have considered quantitative generalizations of languages, over finite words [DG07], over trees [DKR08], or using finite lattices [GC03], but none of these works has addressed the expressiveness questions and closure properties for quantitative languages that are studied here.

The lattice automata of [KL07] map finite words to values from a finite lattice. The lattice automata with Büchi condition are analogous to our **LimSup** automata, and their closure properties are established there. However, the other classes of quantitative automata (**Sum**, **limit-average**, **discounted-sum**) are not studied there as they cannot be defined using lattice operations and finite lattices.

2 Quantitative Languages

A *quantitative language* L over a finite alphabet Σ is either a mapping $L : \Sigma^+ \rightarrow \mathbb{R}$ or a mapping $L : \Sigma^\omega \rightarrow \mathbb{R}$, where \mathbb{R} is the set of real numbers.

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

- Q is a finite set of states, $q_I \in Q$ is the initial state, and Σ is a finite alphabet;
- $\delta \subseteq Q \times \Sigma \times Q$ is a finite set of labelled transitions. We assume that δ is *total*, that is for all $q \in Q$ and $\sigma \in \Sigma$, there exists $(q, \sigma, q') \in \delta$ for at least one $q' \in Q$;
- $\gamma : \delta \rightarrow \mathbb{Q}$ is a *weight* function, where \mathbb{Q} is the set of rational numbers. We assume that rational numbers are encoded as pairs of integers in binary.

We say that A is *deterministic* if for all $q \in Q$ and $\sigma \in \Sigma$, there exists $(q, \sigma, q') \in \delta$ for exactly one $q' \in Q$. We sometimes call automata *nondeterministic* to emphasize that they are not necessarily deterministic.

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) $q_0 = q_I$, and (ii) $(q_i, \sigma_{i+1}, q_{i+1}) \in \delta$ 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 $0 \leq i < |w|$.

Given a *value function* $\text{Val} : \mathbb{Q}^+ \rightarrow \mathbb{R}$ (resp. $\text{Val} : \mathbb{Q}^\omega \rightarrow \mathbb{R}$), we say that the **Val**-automaton A defines the quantitative language L_A such that for all $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$):

$$L_A(w) = \sup\{\text{Val}(\gamma(r)) \mid r \text{ is a run of } A \text{ over } w\}.$$

We consider the following value functions to define quantitative languages. Given a finite sequence $v = v_1 \dots v_n$ of rational numbers, define

- $\text{Last}(v) = v_n$;
- $\text{Max}(v) = \sup\{v_i \mid 1 \leq i \leq n\}$;
- $\text{Sum}(v) = \sum_{i=1}^n v_i$;

Given an infinite sequence $v = v_0 v_1 \dots$ of rational numbers, define

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

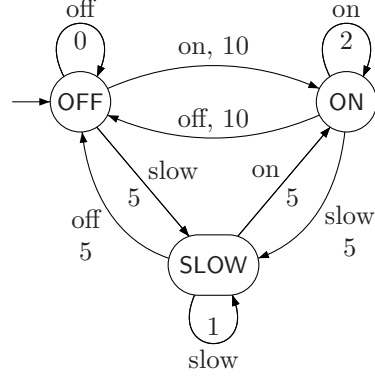
Note that Büchi and coBüchi automata are special cases of respectively LimSup - and LimInf -automata, where all weights are either 0 or 1.

Notations. Classes of weighted automata over infinite words are denoted with acronyms of the form xy where x is either N(ondeterministic), D(eterministic), or $\mathbb{N}yW$ (when deterministic automata have the same expressiveness as non-deterministic automata), and y is one of the following: SUP, LSUP(LimSup), LINF(LimInf), LAVG(LimAvg), or DISC. For Büchi and coBüchi automata, we use the classical acronyms NBW, DBW, NCW, etc.

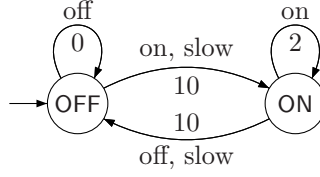
Reducibility. A class \mathcal{C} of weighted automata is *reducible* to a class \mathcal{C}' of weighted automata if for every $A \in \mathcal{C}$ there exists $A' \in \mathcal{C}'$ such that $L_A = L_{A'}$, i.e. $L_A(w) = L_{A'}(w)$ for all (finite or infinite) words w . In particular, a class of weighted automata *can be determinized* if it is reducible to its deterministic counterpart. Reducibility relationships for (non)deterministic weighted automata are given in [CDH08].

Composition. Given two quantitative languages L and L' over Σ , and a rational number c , we denote by $\max(L, L')$ (resp. $\min(L, L')$, $L+L'$, $c+L$, and cL) the quantitative language that assigns $\max\{L(w), L'(w)\}$ (resp. $\min\{L(w), L'(w)\}$, $L(w) + L'(w)$, $c + L(w)$, and $c \cdot L(w)$) to each word $w \in \Sigma^+$ (or $w \in \Sigma^\omega$). We say that $c + L$ is the *shift by c* of L and that cL is the *scale by c* of L . 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. For instance, De Morgan's laws hold (the complement of the max of two languages is the min of their complement, etc.) and complementing twice leave languages unchanged.

Example 1. We consider a simple illustration of the use of limit-average automata to model the power consumption of a motor. The automaton B in Fig. 1(b) specifies the maximal power consumption to maintain the motor on or off, and the



(a) Limit-average automaton A .



(b) Limit-average automaton B .

Fig. 1. Specifications for the power consumption of a motor. A refines B , i.e. $L_A \leq L_B$.

maximal consumption for a mode change. The specification abstracts away that a mode change can occur smoothly with the *slow* command. A refined specification A is given in Fig. 1(a) where the effect of slowing down is captured by a third state. One can check that $L_A(w) \leq L_B(w)$ for all words $w \in \{on, off, slow\}^\omega$. Given two limit-average automata that model the power consumption of two different motors, the maximal, minimal, and the sum of average power consumption are obtained by composing the automata under max, min and sum operations, respectively.

Example 2. Consider an investment of 100 dollars that can be made in two banks A_1 and A_2 as follows: (a) 100 dollars to bank A_1 , (b) 100 dollars to bank A_2 , or (c) 50 dollars to bank A_1 and 50 dollars to bank A_2 . The banks can be either in a good state (denoted G_1, G_2) or in a bad state (denoted B_1, B_2). If it is in a good state, then A_1 offers 8% reward while A_2 offers 6% reward. If it is in a bad state, then A_1 offers 2% reward while A_2 offers 4% reward. The change

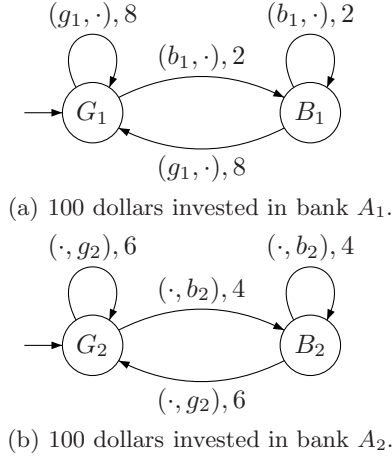


Fig. 2. The discounted-sum automata models of two banks.

of state is triggered by the input symbols b_1, b_2 (from a good to a bad state) and g_1, g_2 (from a bad to a good state). The rewards received earlier weight more than rewards received later due to inflation represented by the discount factor. The automata A_1 and A_2 in Figure 2 specify the behavior of the two banks for an investment of 100 dollars, where the input alphabet is $\{g_1, b_1\} \times \{g_2, b_2\}$ (where the notation (g_1, \cdot) represents the two letters (g_1, g_2) and (g_1, b_2) , and similarly for the other symbols). If 50 dollars are invested in each bank, then we obtain automata C_1 and C_2 from A_1 and A_2 where each reward is halved. The combined automaton is obtained as the composition of C_1 and C_2 under the sum operation.

3 Expressiveness Results for Weighted Automata

The expressive power of weighted automata can be compared by mean of the reducibility relation, saying that a class \mathcal{C} of weighted automata is at least as expressive as a class \mathcal{C}' if every quantitative language definable by some automaton in \mathcal{C} is also definable by some automaton in \mathcal{C}' . The comparison includes boolean languages, considering them as a special case of quantitative languages of the form $L : \Sigma^\omega \rightarrow \{0, 1\}$. It was shown in [CDH08] that a wide variety of classes of quantitative languages can be defined by the different types of weighted automata, depending on the value function and whether they are deterministic or not. This contrasts with the situation for boolean languages where most of the classes of automata define ω -regular languages. In this section, we investigate alternative ways of comparing the expressive power of weighted automata and of classical finite automata. First, we use the cut-point languages of weighted automata to compare with the class of ω -regular languages, and then we use

weighted automata with boolean weights, i.e. all transitions have weight 0 or 1, to compare with general weighted automata.

3.1 Cut-point languages

Let L be a quantitative language over infinite words and let $\eta \in \mathbb{R}$ be a threshold. The *cut-point language* defined by (L, η) is the (boolean) language

$$L^{\geq \eta} = \{w \in \Sigma^\omega \mid L(w) \geq \eta\}.$$

Cut-point languages for finite words are defined analogously. They have been first defined for probabilistic automata [Rab63], then generalized to inverse image recognition for semiring automata over finite words [CM00]. It is easy to see that the cut-point languages of **Max**- and **Last**-automata are regular, those of **Sum**-automata are context-free, and those of **Sup**-, **LimSup**-, and **LimInf**-automata are ω -regular.

We show that the classes of cut-point languages definable by (non)deterministic limit-average and discounted-sum automata are incomparable with the ω -regular languages. The result follows from Theorem 1, and from [CDH08, Theorems 13 and 14].

Theorem 1. *There exists deterministic limit-average and discounted-sum automata whose cut-point language is not ω -regular.*

Proof. Consider the alphabet $\Sigma = \{a, b\}$, and consider the languages L_1 that assigns to each word its long-run average number of a 's, and L_2 that assigns the discounted sum of a 's. Note that L_1 is definable by a deterministic limit-average automaton, and L_2 by a deterministic discounted-sum automaton. It was shown in [Cha07] that the cut-point language $L_1^{\geq 1}$ is complete for the third level of the Borel hierarchy, and therefore is not ω -regular. We show that $L_2^{\geq 1}$ is not ω -regular.

Given a finite word $w \in \Sigma^*$, let $r_a(w) = \sum_{i \mid w_i = a} \lambda^i$ be the discounted sum of a 's in w . We say that w is *ambiguous* if $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq r_a(w) < 1$. The ambiguity lies in that some continuations of w (namely $w.a^\omega$) are in $L_2^{\geq 1}$ and some are not (namely $w.b^\omega$). We show that for all $\lambda > \frac{1}{2}$, if w is ambiguous, then either $w.a$ or $w.b$ is ambiguous, which entails that there exists an infinite word w^ω all of whose finite prefixes are ambiguous (and $L_2(w^\omega) = 1$). To do this, assume that $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq r_a(w) < 1$, and show that either $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq r_a(w.a) < 1$ or $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq r_a(w.b) < 1$. Since $r_a(w.a) = r_a(w) + \lambda^{|w|}$ and $r_a(w.b) = r_a(w)$, we have to show that $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq r_a(w) < 1 - \lambda^{|w|}$ or $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq r_a(w) < 1$. This holds if $1 - \frac{\lambda^{1+|w|}}{1-\lambda} < 1 - \lambda^{|w|}$, which is equivalent to $\lambda > \frac{1}{2}$.

Now, we show that if there exists a nondeterministic Büchi automaton A for $L_2^{\geq 1}$, then the set of states S_n reached in A by reading the first n letters of w^ω (which we denote by $w_{[1..n]}^\omega$) should be different for each n , i.e. $n \neq m$ implies

$S_n \neq S_m$. Towards a contradiction, assume that $S_n = S_m$ for $n < m$. Then for all $w' \in \Sigma^\omega$, we have $w_{[1\dots n]}^\preceq \cdot w' \in L_2^{\geq 1}$ if and only if $w_{[1\dots m]}^\preceq \cdot w' \in L_2^{\geq 1}$. In particular, for $w' = w_{[m+1\dots]}^\preceq$, this shows that $L_2(w_{[1\dots n]}^\preceq \cdot w') = 1 = L_2(w_{[1\dots m]}^\preceq \cdot w')$ since $L_2(w^\preceq) = 1$ and $r_a(w_{[1\dots n]}) \leq r_a(w_{[1\dots m]})$. This yields

$$r_a(w_{[1\dots n]}) + \lambda^n \cdot r_a(w') = 1 = r_a(w_{[1\dots m]}) + \lambda^m \cdot r_a(w')$$

that is, by eliminating $r_a(w')$, $\lambda^{m-n}(1 - P(\lambda)) = 1 - Q(\lambda)$ where $P(\lambda) = r_a(w_{[1\dots n]})$ and $Q(\lambda) = r_a(w_{[1\dots m]})$ are polynomials of respective degree $n - 1$ and $m - 1$, and with coefficients in the set $\{0, 1\}$. First, observe that the equation is not identically 0 because the coefficient of the term of degree 0 is not 0 (as the first letter of w^\preceq must be b since a is not ambiguous). Second, every coefficient in the equation is in the set $\{-1, 0, 1, 2\}$, and a classical result shows that if $\frac{p}{q}$ is a solution of a polynomial equation with p and q mutually prime, then p divides the coefficient of degree 0, and q divides the coefficient of highest degree. Therefore, no rational number in the interval $]\frac{1}{2}, 1[$ can be a solution. This shows that $n \neq m$ implies $S_n \neq S_m$, and the automaton A cannot have finitely many states. ■

We note that cut-point languages are not stable under arbitrarily small perturbations of the transition weights, nor of the value of the cut-point. Consider the quantitative languages L_1, L_2 from the proof of Theorem 1. If for instance a limit-average automaton A assigns weight $1 + \epsilon$ to the a 's and 0 to the b 's, its cut-point language $L_A^{\geq 1}$ is clearly not different from $L_1^{\geq 1}$ that assigns to each word its long-run average number of a 's, no matter the value of $\epsilon > 0$. The same holds with respect to L_2 if A is interpreted as a discounted-sum automaton.

In the theory of probabilistic automata, where finite words are assigned a probability of acceptance, the cut-point languages may also be non-regular. Therefore, one considers the special case where the cut-point is isolated, and shows that the cut-point languages are then regular [Rab63].

A number η is an *isolated cut-point* of a quantitative language L if there exists $\epsilon > 0$ such that

$$|L(w) - \eta| > \epsilon \text{ for all } w \in \Sigma^\omega.$$

We show that every discounted-sum automaton with isolated cut-point defines an ω -regular language, and that this also holds for deterministic limit-average automata. We also argue that this notion has stability properties, in that isolated cut-point languages remain unchanged under small perturbations of the transition weights. This follows from a more general result about the robustness of weighted automata.

A class of weighted automata is robust if a small (syntactical) perturbation in the weights of an automaton induces only a small (semantical) perturbation in the values of the words in the quantitative language of the automaton, and the semantical perturbation tends to 0 when the syntactical perturbation tends to 0. To formally define robustness, we need ϵ -approximations of automata, and distance between quantitative languages.

Let $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ be a (nondeterministic) weighted automaton, and let $\epsilon \in \mathbb{R}^{\geq 0}$. We say that a weighted automaton $B = \langle Q', q'_I, \Sigma, \delta', \gamma' \rangle$ is an ϵ -approximation of A if

- $Q' = Q$, $q'_I = q_I$, $\delta' = \delta$, and
- $|\gamma'(q, \sigma, q') - \gamma(q, \sigma, q')| \leq \epsilon$ for all $(q, \sigma, q') \in \delta$.

The sup-distance between two quantitative languages $L_1, L_2 : \Sigma^\omega \rightarrow \mathbb{R}$ is defined by

$$D_{\text{sup}}(L_1, L_2) = \sup_{w \in \Sigma^\omega} |L_1(w) - L_2(w)|.$$

We say that a class \mathcal{C} of weighted automata is *uniformly robust* if for all $\eta \in \mathbb{R}^{>0}$, there exists $\epsilon \in \mathbb{R}^{>0}$ such that for all automata $A, B \in \mathcal{C}$ where B is an ϵ -approximation of A , we have $D_{\text{sup}}(L_A, L_B) \leq \eta$. Note that uniform robustness implies a weaker notion of robustness where a class \mathcal{C} of weighted automata is called *robust* if for all automata $A \in \mathcal{C}$ and for all $\eta \in \mathbb{R}^{>0}$, there exists $\epsilon \in \mathbb{R}^{>0}$ such that for all ϵ -approximation B of A (with $B \in \mathcal{C}$), we have $D_{\text{sup}}(L_A, L_B) \leq \eta$.

Theorem 2. *The classes of (non)deterministic Sup-, LimSup-, LimInf-, LimAvg- and Disc-automata are uniformly robust.*

Proof. Let A, B be two weighted automata with B an ϵ -approximation of A . It is easy to see that for Sup-, LimSup-, LimInf- and LimAvg-automata, the value of a run r of B differs by at most ϵ from the value of the same run in A . Therefore, $D_{\text{sup}}(L_A, L_B) \leq \epsilon$ and we can take $\epsilon = \eta$. For Disc-automata, the value of a run of B differs by at most $\frac{\epsilon}{1-\lambda}$ from the value of the same run in A , where λ is the discount factor. Therefore, we can take $\epsilon = \eta(1 - \lambda)$. ■

As a corollary of Theorem 2, for an isolated cut-point η , the cut-point language $L^{\geq \eta}$ remains unchanged under small perturbations of the transition weights.

Theorem 3. *Let L_A be the quantitative language defined by a weighted automaton A , and let η be an isolated cut-point of L_A . There exists $\epsilon > 0$ such that for all ϵ -approximations B of A , $L_A^{\geq \eta} = L_B^{\geq \eta}$ (where L_B is the quantitative language defined by B).*

Now, we show that the isolated cut-point languages of deterministic discounted-sum and limit-average automata are ω -regular. For nondeterministic automata, the same property holds in the discounted-sum case, but the question is open for limit average.

Theorem 4. *Let L be the quantitative language defined by a Disc-automaton. If η is an isolated cut-point of L , then the cut-point language $L^{\geq \eta}$ is ω -regular.*

Proof. Let λ be the discount factor of the Disc-automaton that defines L . Since, η is an isolated cut-point of L , let $\epsilon > 0$ such that $|L(w) - \eta| > \epsilon$ for all $w \in \Sigma^\omega$. Let $n \in \mathbb{N}$ such that $u_n = \frac{V \cdot \lambda^n}{1 - \lambda} < \epsilon$ where $V = \max_{(q, \sigma, q')} |\delta(q, \sigma, q')|$ is largest weight in A . Consider any run r in A of length n , and let $\gamma(r)$ be the λ -discounted sum of the weights along r . Then, it should be clear that $\gamma(r) \notin [\eta - \epsilon + u_n, \eta + \epsilon - u_n]$, because otherwise, the value of any (infinite) continuation of r would lie in the interval $[\eta - \epsilon, \eta + \epsilon]$, which would be a contradiction. Moreover, if $\gamma(r) \leq \eta - \epsilon + u_n$, then any (infinite) continuation of r has value less than η , while if $\gamma(r) \geq \eta + \epsilon - u_n$, then any (infinite) continuation of r has value greater than η . Therefore, the cut-point language $L^{\geq \eta}$ can be defined by the unfolding up to length n of the Disc-automaton that defines L , in which the states that are reached via a path with value at least $\eta + \epsilon - u_n$ are declared to be accepting, and have a self-loop on Σ . ■

Theorem 5. *Let L be the quantitative language defined by a deterministic LimAvg-automaton. If η is an isolated cut-point of L , then the cut-point language $L^{\geq \eta}$ is ω -regular.*

Proof. Let A be a deterministic LimAvg-automaton, defining the language L . Consider the SCC-decomposition C_1, C_2, \dots, C_k of the underlying graph of A . For each $1 \leq i \leq k$, let m_i and M_i be the minimal and maximal average weight of a cycle in C_i (those values can be computed with Karp's algorithm [Kar78]). It is easy to see that for every $1 \leq i \leq k$, for every $v \in [m_i, M_i]$, there exists a word $w \in \Sigma^\omega$ such that $L(w) = v$. Therefore, since η is an isolated cut-point of L , we have $\eta \notin [m_i, M_i]$ for all $1 \leq i \leq k$. A DBW for $L^{\geq \eta}$ is obtained from A by declaring to be accepting all states q of A such that $q \in C_i$ and $m_i > \eta$. ■

3.2 Boolean weights

We consider weighted automata with boolean set of weights, i.e. all transitions have weight 0 or 1. The aim is to have a boolean counterpart to limit-average and discounted-sum automata, and check if this changes their expressive power. We show that the restriction does not change the class of quantitative languages definable by limit-average automata, but does reduce that of discounted-sum automata.

Given a set $R \subseteq \mathbb{R}$, and a class \mathcal{C} of nondeterministic weighted automata, we denote by \mathcal{C}_R the class of all automata in \mathcal{C} whose weights are rational numbers in R .

Theorem 6. *The class of nondeterministic (resp. deterministic) LimAvg-automata with weights in $[0, 1] \cap \mathbb{Q}$ is reducible to the class of nondeterministic (resp. deterministic) LimAvg-automata with weights 0 and 1 only.*

Proof. Given a NLAVG_[0,1]-automaton $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$, we construct a NLAVG_{0,1}-automaton B such that $L_A = L_B$.

First, let $W = \{\gamma(q, \sigma, q') \mid (q, \sigma, q') \in \delta\}$ be the set of weights that occur in A , and let n_A be the smallest integer n such that for all $v \in W$, there exists $e \in \mathbb{N}$ such that $v = \frac{e}{n}$ (i.e., $\frac{1}{n_A}$ is the greatest common divisor of the weights of A). We define $B = \langle Q', q'_I, \Sigma, \delta', \gamma' \rangle$ as follows:

- $Q' = Q \times [n_A]$ (where $[n_A]$ denotes the set $\{0, 1, \dots, n_A - 1\}$). Intuitively, when we reach a state (q, i) in B , it means that the state q was reachable in A and that the sum of the weights to reach q is of the form $k + \frac{i}{n_A}$ for some integer k . In B however, the sum of the weights to reach (q, i) will then be k , and we store in the discrete state the information that the remainder weight is $\frac{i}{n_A}$. Whenever this remainder exceeds 1, we introduce a weight 1 and decrement the remainder.
- $q'_I = (q_I, 0)$;
- for each transition $(q, \sigma, q') \in \delta$ and each value $i \in [n_A]$, the following transitions are in δ' (where $v = \gamma(q, \sigma, q')$):
 - $((q, i), \sigma, (q', j))$ for $j = i + (v - 1) \cdot n_A$ if $\frac{i}{n_A} + v \geq 1$; the weight of such a transition is 1 in γ' ,
 - $((q, i), \sigma, (q', j))$ for $j = i + v \cdot n_A$ if $\frac{i}{n_A} + v < 1$; the weight of such a transition is 0 in γ' .

Note that in the above, $v \cdot n_A$ is an integer and $j \in [n_A]$.

There is a straightforward correspondence between the runs in A and the runs in B . Moreover, if the average weight of a prefix of length n of a run in A is $\frac{S}{n}$, then the average weight of the prefix of length n of the corresponding run in B is between $\frac{S}{n}$ and $\frac{S+1}{n}$. Hence the difference tends to 0 when $n \rightarrow \infty$. Therefore, the value of a run in A is the same as the value of the corresponding run in B , and therefore $L_A = L_B$.

Finally, note that if A is deterministic, then B is deterministic. ■

Theorem 7. *The class of deterministic Disc-automata with rational weights in $[0, 1]$ is not reducible to the class of (even nondeterministic) Disc-automata with weights 0 and 1 only.*

Proof. Given a discount factor $0 < \lambda < 1$, consider the NDISC_[0,1] over $\Sigma = \{a, b\}$ that consists of a single state with a self-loop over a with weight $\frac{1+\lambda}{2}$ and a self-loop over b with weight 0. Let L_λ be the quantitative language defined by this automaton. Towards a contradiction, assume that this language is defined by a NDISC_{0,1} A . First, consider the word ab^ω whose value in L_λ is $\frac{1+\lambda}{2} < 1$. This entails that A cannot have a transition from the initial state over a with weight 1 (as this would imply that $L_A(ab^\omega) \geq 1$). Now, the maximal value that L_A can assign to the word a^ω is $\lambda + \lambda^2 + \lambda^3 + \dots = \frac{\lambda}{1-\lambda}$ which is strictly smaller than $L_\lambda(a^\omega) = \frac{1+\lambda}{2(1-\lambda)}$. This shows that A cannot exist. ■

4 The Closure Properties of Weighted Automata

We study the closure properties of weighted automata with respect to max, min, complement and sum. We say that a class \mathcal{C} of weighted automata is *closed* under a binary operator $\text{op}(\cdot, \cdot)$ (resp. a unary operator $\text{op}'(\cdot)$) if for all $A_1, A_2 \in \mathcal{C}$, there exists $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \text{op}(L_{A_1}, L_{A_2})$ (resp. $L_{A_{12}} = \text{op}'(L_{A_1})$). All closure properties that we present in this paper are constructive: when \mathcal{C} is closed under an operator, we can always construct the automaton $A_{12} \in \mathcal{C}$ given $A_1, A_2 \in \mathcal{C}$. We say that the *cost* of the closure property of \mathcal{C} under a binary operator op is at most $O(f(n_1, m_1, n_2, m_2))$ if for all automata $A_1, A_2 \in \mathcal{C}$ with n_i states and m_i transitions (for $i = 1, 2$ respectively), the constructed automaton $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \text{op}(L_{A_1}, L_{A_2})$ has at most $O(f(n_1, m_1, n_2, m_2))$ many states. Analogously, the *cost* of the closure property of \mathcal{C} under a unary operator op' is at most $O(f(n, m))$ if for all automata $A_1 \in \mathcal{C}$ with n states and m transitions, the constructed automaton $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \text{op}'(L_{A_1})$ has at most $O(f(n, m))$ many states. For all reductions presented, the size of the largest weight in A_{12} is linear in the size p of the largest weight in A_1, A_2 (however, the time needed to compute the weights is quadratic in p , as we need addition, multiplication, or comparison, which are quadratic operations over the rational numbers).

Notice that every class of weighted automata is closed under shift by c and under scale by $|c|$ for all $c \in \mathbb{Q}$. For **Sum**-automata and discounted-sum automata, we can define the shift by c by making a copy of the initial states and adding c to the weights of all its outgoing transitions. For the other automata, it suffices to add c to (resp. multiply by $|c|$) all weights of an automaton to obtain the automaton for the shift by c (resp. scale by $|c|$) of its language. Therefore, all closure properties also hold if the complement of a quantitative language L was defined as $k - L$ for any constant k .

Our purpose is the study of quantitative languages over infinite words. For the sake of completeness, we first give an overview of the closure properties for finite words.

4.1 Closure properties for finite words

We successively consider closure under max, min, complement, and sum for weighted automata over finite words. Table 1(a) summarizes the closure properties of **Max**-, **Last**- and **Sum**-automata.

Theorem 8. *Deterministic Sup- and Last-automata are closed under max, with cost $O(n_1 \cdot n_2)$. Nondeterministic Sup-, Last- and Sum-automata are closed under max, with cost $O(n_1 + n_2)$. Deterministic Sum-automata are not closed under max.*

Proof. For the nondeterministic automata, the result follows from the fact that the max operator can be obtained by an initial nondeterministic choice between two quantitative automata. For deterministic Sup- and Last-automata,

the result follows from the fact that the classes of nondeterministic **Sup**- and **Last**-automata are reducible⁴ to their respective deterministic counterpart. Finally, deterministic **Sum**-automata are not closed under the max operator because the language over $\Sigma = \{a, b\}$ that assigns to each finite word $w \in \Sigma^+$ the number $\max\{L_a(w), L_b(w)\}$ where $L_\sigma(w)$ is the number of occurrences of σ in w (for $\sigma = a, b$) is definable by the max of two deterministic-**Sum** languages, but not by a deterministic **Sum**-automaton (Theorem 4 in [CDH08]). ■

Theorem 9. *Deterministic and nondeterministic **Sup**-automata are closed under min, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$. Deterministic and nondeterministic **Last**-automata are closed under min, with cost $O(n_1 \cdot n_2)$. Deterministic and nondeterministic **Sum**-automata are not closed under min.*

Proof. Given two **Last**-automata A_1 and A_2 (over the same alphabet), we use the classical synchronized product $A_{12} = A_1 \times A_2$, where the weight of a transition in A_{12} is the minimum of the corresponding transition weights in A_1 and A_2 . It is easy to see that $L_{A_{12}} = \min(L_{A_1}, L_{A_2})$. If A_1 and A_2 are deterministic, then so is A_{12} .

The construction for **Sup**-automata is the same as for **Sup**-automata over infinite words given in the proof of Theorem 14.

Finally, for **Sum**-automata, consider the language L_m over $\Sigma = \{a, b\}$ that assigns to each finite word $w \in \Sigma^+$ the value $\min\{L_a(w), L_b(w)\}$ where $L_\sigma(w)$ is the number of occurrences of σ in w (for $\sigma = a, b$). We claim that L_m is not definable by a nondeterministic **Sum**-automaton. Indeed, assume that the **Sum**-automaton A defines L_m . First, every the sum of weights in every reachable cycle of A over a 's must be at most 0. Otherwise, we can reach the cycle with a finite word w_1 and obtain an arbitrarily large value for the word $w_1 a^i$ for i sufficiently large, while for such i the value of $w_1 a^i$ is the number of b 's in w_1 which is independent of i . Analogously, the sum of weights in every reachable cycle of A over b 's must be at most 0. Now, let $\beta = \max_{e \in \delta} \gamma(e)$ be the maximal weight in A , and consider the word $w = a^n b^n$ for $n > 2\beta \cdot |Q|$. Every run of A over a^n (or over b^n) can be decomposed in possibly nested cycles (since A is nondeterministic) and a remaining non-cyclic path of length at most $|Q|$. Hence, the value of any run over w is at most $2\beta \cdot |Q|$. However, the value of w should be n , yielding a contradiction. ■

Theorem 10. *Deterministic **Last**- and **Sum**-automata are closed under complement, with cost $O(n)$. Nondeterministic **Last**-automata are closed under complement, with cost $O(2^n)$. Nondeterministic **Sum** automata, and both deterministic and nondeterministic **Sup**-automata are not closed under complement.*

⁴ We say that a class \mathcal{C} of quantitative automata is *reducible* to a class \mathcal{C}' of quantitative automata if for every $A \in \mathcal{C}$ there exists $A' \in \mathcal{C}'$ such that $L_A = L_{A'}$.

Proof. To define the complement of the language of a deterministic Sum (or Last-) automaton, it suffices to multiply all the weights by -1 , and then shift the language by 1. For the class of nondeterministic Last-automata, the result follows from the fact that it is reducible to its deterministic counterpart.

The negative result for Sup-automata follows from an analogous in the boolean case (consider the language L over $\{a, b\}$ such that $L(a^i) = 0$ for all $i \geq 1$, and $L(w) = 1$ for all words containing the letter b). Finally, according to the proof of Theorem 9, the language $\min(L_a, L_b)$ where $L_\sigma(w)$ is the number of occurrences of σ in w (for $\sigma = a, b$) is not definable by a nondeterministic Sum-automaton. Since $\min(L_a, L_b) = -\max(-L_a, -L_b)$ and (i) $-L_a$ and $-L_b$ are definable by Sum-automata, and (ii) nondeterministic Sum-automata are closed under max (Theorem 8), the language $\max(-L_a, -L_b)$ is definable by a nondeterministic Sum-automaton, and the result follows. ■

Theorem 11. *Every class of weighted automata over finite words are closed under sum. The cost is $O(n_1 \cdot n_2)$ for Last- and Sum-automata, and $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$ for Sup-automata.*

Proof. It is easy to see that the synchronized product of two Last-automata (resp. Sum-automata) defines the sum of their languages if the weight of a joint transition is defined as the sum of the weights of the corresponding transitions in the two Last-automata (resp. Sum-automata).

The construction for Sup-automata is the same as for Sup-automata over infinite words given in the proof of Theorem 24. ■

	max.	min.	comp.	sum
Sup	✓	✓	×	✓
Last	✓	✓	✓	✓
Det. Sum	×	✓	✓	✓
Nondet. Sum	✓	×	×	✓

(a) Finite words

	max.	min.	comp.	sum
\mathbb{R} SUP	✓	✓	×	✓
\mathbb{R} LINF	✓	✓	×	✓
DLSUP	✓	✓	×	✓
NLSUP	✓	✓	✓	✓
DLAVG	×	×	×	×
NLAVG	✓	×	×	×
DDISC	×	×	✓	✓
NDISC	✓	×	×	✓

(b) Infinite words

Table 1. Closure properties.

4.2 Closure under max for infinite words

The maximum of two quantitative languages defined by nondeterministic automata can be obtained by an initial nondeterministic choice between the two

automata. This observation was also made in [DR07] for discounted-sum automata. For deterministic automata, a synchronized product can be used for Sup and LimSup, while for LimInf we use the fact that NLINF is determinizable with an exponential blow-up [CDH08].

Theorem 12. *The nondeterministic Sup-, LimSup-, LimInf-, LimAvg- and Disc-automata are closed under max, with cost $O(n_1 + n_2)$, the deterministic Sup- and LimSup-automata with cost $O(n_1 \cdot n_2)$, the deterministic LimInf-automata with cost $O((m_1 + m_2) \cdot 2^{n_1 + n_2})$.*

Proof Sketch. For all the nondeterministic quantitative automata, the result follows from the fact that the max operator can be achieved with an initial nondeterministic choice between two quantitative automata. For DLINF, the result follows from the reducibility of NLINF to DLINF with an exponential blow-up [CDH08]. We now prove that DLSUP and DSUP are closed under max with cost $O(n_1 \cdot n_2)$. Given two DLSUP (or DSUP) A_1 and A_2 over the same alphabet, we construct the usual synchronized product $A_{12} = A_1 \times A_2$, where the weight of a transition in A_{12} is the maximum of the corresponding transition weights in A_1 and A_2 . It is easy to see that $L_{A_{12}} = \max(L_{A_1}, L_{A_2})$ in both cases.

Theorem 13. *The deterministic LimAvg- and Disc-automata are not closed under max.*

Proof. The fact that DDISC is not closed under max follows from the proof of Theorem 34 in [CDH08], where it is shown that the quantitative language $\max(L_1, L_2)$ cannot be defined by a DDISC, where L_1 (resp. L_2) is the language defined by the DDISC that assigns weight 1 (resp. 0) to a 's and weight 0 (resp. 1) to b 's.

We now show that DLAVG 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 DLAVG for L_a and L_b . We show that $L_m = \max(L_a, L_b)$ cannot be expressed by a DLAVG. By contradiction, assume that A is a DLAVG with set of states Q that defines L_m . Consider any reachable cycle C over a 's in A . The sum of the weights of the cycle must be its length $|C|$, as if we consider the word $w^* = w_C \cdot (a^{|C|})^\omega$ where w_C is a finite word whose run reaches C , the value of w^* in L_m is 1. It follows that the sum of the weights of the cycle C must be $|C|$. Hence, the sum of the weights of all the reachable cycles C over a 's in A is $|C|$.

Consider the infinite word $w_\infty = (a^{|Q|} \cdot b^{2|Q|})^\omega$, and let $w_j = (a^{|Q|} \cdot b^{2|Q|})^j$. Since $L_m(w_\infty) = \frac{2}{3}$, the run of A over w_∞ has value $\frac{2}{3}$. It follows that for all $\varepsilon > 0$, there is an integer j_ε , such that for all $j \geq j_\varepsilon$, we have

$$\frac{\gamma(w_j)}{|w_j|} \geq \frac{2}{3} - \varepsilon$$

where $\gamma(w_j)$ is the sum of the weights of the run of A over w_j . Consider a word \hat{w}_∞ constructed as follows. We start with the empty word \hat{w}_0 and the initial state q_0 of A , and for all $j \geq 0$, we construct (\hat{w}_{j+1}, q_{j+1}) from (\hat{w}_j, q_j) as follows: the state q_{j+1} is the last state of the run of A from q_j over $a^{|Q|} \cdot b^{2|Q|}$. This run has to contain a cycle C_{j+1} over a 's. We set $\hat{w}_{j+1} = \hat{w}_j \cdot a^{|Q|+|C_{j+1}|} \cdot b^{2|Q|}$. Observe that for all $j \geq 1$, the run of A over w_∞ in the segment between w_j and w_{j+1} is identical to the run from q_j to q_{j+1} up to the repetition of the cycle C_{j+1} once more. The word \hat{w}_∞ is the limit of this construction (\hat{w}_j is a prefix of \hat{w}_∞ for all $j \geq 0$). Let $\alpha_j = \sum_{i=1}^j |C_i|$. Since $1 \leq |C_i| \leq |Q|$ we have $j \leq \alpha_j \leq j \cdot |Q|$. Hence we have the following equality: $\frac{\gamma(\hat{w}_j)}{|\hat{w}_j|} = \frac{\gamma(w_j) + \alpha_j}{|w_j| + \alpha_j}$. Hence for all $\varepsilon > 0$, there exists j_ε such that for all $j \geq j_\varepsilon$ we have

$$\begin{aligned} \frac{\gamma(\hat{w}_j)}{|\hat{w}_j|} &\geq \frac{\frac{2}{3} \cdot |w_j| - \varepsilon \cdot |w_j| + \alpha_j}{|w_j| + \alpha_j} \\ &\geq \frac{2}{3} - \varepsilon + \frac{1}{3} \cdot \frac{\alpha_j}{|w_j| + \alpha_j} \\ &\geq \frac{2}{3} - \varepsilon + \frac{1}{3} \cdot \frac{j}{j \cdot (3|Q| + |Q|)} \\ &\geq \frac{2}{3} - \varepsilon + \frac{1}{12|Q|} \end{aligned}$$

Hence we have $L_A(\hat{w}_\infty) \geq \frac{2}{3} + \frac{1}{12|Q|}$. Since $1 \leq |C_i| \leq |Q|$ for all $i \geq 1$, we have $L_m(\hat{w}_\infty) \leq \frac{2}{3}$ which is a contradiction. \blacksquare

4.3 Closure under min for infinite words

The next theorems generalize the closure property under intersection of the boolean languages. The construction of the automaton for the min is a direct extension of the well-known constructions in the boolean case.

Theorem 14. *The (non)deterministic Sup-automata are closed under min, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$,*

Proof. Let $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$ be two NSUP. We construct a NSUP $A_{12} = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_{A_{12}} = \min\{L_{A_1}, L_{A_2}\}$. Let $V_i = \{\gamma_i(e) \mid e \in \delta_i\}$ be the set of weights that appear in A_i (for $i = 1, 2$), and define:

- $Q = Q_1 \times V_1 \times Q_2 \times V_2$. Intuitively, we remember in a state (q_1, v_1, q_2, v_2) the largest weights v_1, v_2 seen so far in the corresponding runs of A_1 and A_2 ;
- $q_I = (q_I^1, v_{\min}^1, q_I^2, v_{\min}^2)$ where v_{\min}^i is the minimal weight in V_i (for $i = 1, 2$);
- For each $\sigma \in \Sigma$, the set δ contains all the triples $\langle (q_1, v_1, q_2, v_2), \sigma, (q'_1, v'_1, q'_2, v'_2) \rangle$ such that $v_i \in V_i$, $(q_i, \sigma, q'_i) \in \delta_i$, and $v'_i = \max\{v_i, \gamma(q_i, \sigma, q'_i)\}$, for $i = 1, 2$;

- γ is defined by $\gamma(\langle (q_1, v_1, q_2, v_2), \sigma, (q'_1, v'_1, q'_2, v'_2) \rangle) = \min\{v'_1, v'_2\}$ for each $\langle (q_1, v_1, q_2, v_2), \sigma, (q'_1, v'_1, q'_2, v'_2) \rangle \in \delta$.

If A_1 and A_2 are deterministic, then A_{12} is deterministic. The result for DLSUP follows. ■

Theorem 15. *The (non)deterministic LimInf-automata are closed under min with cost $O(n_1 \cdot n_2)$, and the nondeterministic LimSup-automata with cost $O(n_1 \cdot n_2 \cdot (m_1 + m_2))$.*

Proof. Let $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$ be two NLSUP. We construct a NLSUP $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_A = \min\{L_{A_1}, L_{A_2}\}$. Let $V_i = \{\gamma_i(e) \mid e \in \delta_i\}$ be the set of weights that appear in A_i (for $i = 1, 2$). Let $V_1 \cup V_2 = \{v_1, \dots, v_n\}$ and define

- $Q = \{q_I\} \cup Q_1 \times Q_2 \times \{1, 2\} \times (V_1 \cup V_2)$ (where $q_I \notin Q_1 \cup Q_2$ is a new state). Initially, a guess is made of the value v of the input word. Then, we check that both A_1 and A_2 visit a weight at least v infinitely often. In a state $\langle q_1, q_2, j, v \rangle$ of A , the guess is stored in v (and will never change along a run) and the value of the index j is toggled to $3 - j$ as soon as A_j does visit a weight at least v ;
- For each $\sigma \in \Sigma$, the set δ contains all the triples
 - $(q_I, \sigma, \langle q_1, q_2, 1, v \rangle)$ such that $v \in V_1 \cup V_2$ and for all $i \in \{1, 2\}$, we have $(q_I^i, \sigma, q_i) \in \delta_i$.
 - $(\langle q_1, q_2, j, v \rangle, \sigma, \langle q'_1, q'_2, j', v' \rangle)$ such that $v' = v$, $(q_i, \sigma, q'_i) \in \delta_i$ ($i = 1, 2$), and $j' = 3 - j$ if $\gamma_j(q_j, \sigma, q'_j) \geq v$, and $j' = j$ otherwise.
- γ is defined by $\gamma(q_I, \sigma, \langle q_1, q_2, 1, v \rangle) = 0$ and $\gamma(\langle q_1, q_2, j, v \rangle, \sigma, \langle q'_1, q'_2, j', v' \rangle) = v$ if $j \neq j'$ and v_{\min} otherwise, where v_{\min} is the minimal weight in $V_1 \cup V_2$.

For DLINF, the construction is similar to the one presented in the proof of Theorem 12 for DLSUP, where max is replaced by min. The result for NLINF follows from the fact that NLINF is reducible to DLINF. ■

Theorem 16. *The deterministic LimSup-automata are closed under min with cost $O(n_1 \cdot n_2 \cdot 2^{m_1 + m_2})$.*

Proof. Let $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$ be two DLSUP. We construct a DLSUP $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_A = \min\{L_{A_1}, L_{A_2}\}$. Let $V_i = \{\gamma_i(e) \mid e \in \delta_i\}$ be the set of weights that appear in A_i (for $i = 1, 2$). For each weight $v \in V_1 \cup V_2 = \{v_1, \dots, v_n\}$, we construct a DBW A_{12}^v with accepting edges. The automaton A_{12}^v consists of a copy of A_1 and a copy of A_2 . We switch from one copy to the other whenever an edge with weight at least v is crossed. All such switching edges are accepting in A_{12}^v . The automaton A then consists of the synchronized product of these DBW, where the weight of a joint edge is the largest weight v for which the underlying edge in A_{12}^v is accepting. Formally, let

- $Q = Q_1 \times Q_2 \times \{1, 2\}^m$ where $m = |V_1 \cup V_2|$;
- $q_I = (q_I^1, q_I^2, b_1, \dots, b_m)$ where $b_i = 1$ for all $1 \leq i \leq m$;
- δ contains all the triples $(\langle q_1, q_2, b_1, \dots, b_m \rangle, \sigma, \langle q'_1, q'_2, b'_1, \dots, b'_m \rangle)$ such that $\sigma \in \Sigma$ and
 - $(q_i, \sigma, q'_i) \in \delta_i$ for $i = 1, 2$;
 - for all $1 \leq j \leq m$, we have $b'_j = 3 - b_j$ if $\gamma_{b_j}(q_{b_j}, \sigma, q'_{b_j}) \geq v_j$, and $b'_j = b_j$ otherwise.
- γ assigns to each transition $(\langle q_1, q_2, b_1, \dots, b_m \rangle, \sigma, \langle q'_1, q'_2, b'_1, \dots, b'_m \rangle) \in \delta$ the weight $v = \max(\{v_{\min}\} \cup \{v_j \mid b_j \neq b'_j\})$ where v_{\min} is the minimal weight in $V_1 \cup V_2$.

■

On the negative side, the (deterministic or not) limit-average and discounted-sum automata are not closed under min. The following lemma establishes the result for limit average.

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

Proof. To obtain a contradiction, assume that there exists a NLAVGA (for either L_m or L^*). We first claim that there must be either an a -cycle or a b -cycle C that is reachable in A such that the sum of the weights in C is positive. Otherwise, if for all a -cycles and b -cycles we have that the sum of the weights is zero or negative, then we fool the automaton as follows. Let β be the maximum of the absolute values of the weights in A , and let $\alpha = \lceil \beta \rceil$. Then consider the word $w = (a^{5 \cdot \alpha \cdot |Q|} \cdot b^{5 \cdot \alpha \cdot |Q|})^\omega$. For a run r of A over w , the long-run average of the weights is bounded as follows:

$$\frac{4 \cdot \beta \cdot |Q|}{10 \cdot \alpha \cdot |Q|} \leq \frac{2}{5}.$$

The above bound is as follows: in the run over $a^{5 \cdot \alpha \cdot |Q|}$, there can be a prefix of size at most $|Q|$ with sum of weights at most $|Q| \cdot \beta$, and then there would be a -cycles, and then a trailing prefix of size at most $|Q|$ with sum of weights at most $|Q| \cdot \beta$. Similar argument holds for the segment of $b^{5 \cdot \alpha \cdot |Q|}$. Hence $L_A(w) \leq \frac{2}{5}$, 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 -cycle C such that 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. ■

Theorem 17. *The (non)deterministic LimAvg-automata are not closed under min.*

Proof. The result follows from Lemma 1 and the fact that there exists DLAVG for the languages L_a and L_b of Lemma 1. ■

Finally, we show that discounted-sum automata are not closed under min.

Theorem 18. *The (non)deterministic Disc-automata are not closed under min.*

Proof. Let λ be a non-algebraic number in $] \frac{1}{2}, 1[$. We consider the quantitative languages L_a^λ and L_b^λ that assign the λ -discounted sum of a 's and b 's, respectively. Formally, given a (finite or infinite) word $w = w_0 w_1 \dots \in \Sigma^* \cup \Sigma^\omega$, let

$$r_a(w) = \sum_{i|w_i=a}^{|w|} \lambda^i \quad \text{and} \quad r_b(w) = \sum_{i|w_i=b}^{|w|} \lambda^i$$

be the λ -discounted sum of the a 's (resp. b 's) of w . Then, $L_a^\lambda(w) = r_a(w)$ and $L_b^\lambda(w) = r_b(w)$. These languages are definable by DDISC. We show that the language $L_m = \min(L_a^\lambda, L_b^\lambda)$ is not definable by a NDISC.

Assume towards contradiction that there is a NDISC A for L_m . By Lemma 5 and 6 in [CDH08], there exists an infinite word w^\prec such that $r_a(w^\prec) = r_b(w^\prec)$.

Since $r_a(w^\prec) + r_b(w^\prec) = \frac{1}{1-\lambda}$, we have $L_m(w^\prec) = \frac{1}{2(1-\lambda)}$ and this is the maximal value of a word in $L_m(\cdot)$.

The maximal value in the automaton A can be obtained for a lasso-word of the form $w_1.(w_2)^\omega$ (where w_1, w_2 are finite words and w_2 is nonempty), as pure memoryless strategies exist in games over finite graphs with the objective to maximize the discounted sum of payoffs. Since the language of A is L_m , the value of $w_1.(w_2)^\omega$ is $\frac{1}{2(1-\lambda)}$, and thus $r_a(w_1.(w_2)^\omega) = r_b(w_1.(w_2)^\omega)$ by a similar argument as above. This last condition can be written as

$$p_a(\lambda) + \frac{\lambda^{n_1} \cdot q_a(\lambda)}{1 - \lambda^{n_2}} = p_b(\lambda) + \frac{\lambda^{n_1} \cdot q_b(\lambda)}{1 - \lambda^{n_2}}$$

for some polynomials p_a, p_b, q_a, q_b and integers $n_1 \geq 0$ and $n_2 > 0$, or more simply as

$$(1 - \lambda^{n_2}) \cdot p(\lambda) + \lambda^{n_1} \cdot q(\lambda) = 0 \tag{1}$$

for some polynomials p of degree $n_1 - 1$ and q of degree $n_2 - 1$, all of whose coefficients are either 1 or -1 . Equation (1) is not identically zero as either (i) $n_1 = 0$ and it reduces to $q(\lambda) = 0$ or (ii) $n_1 > 0$ and then p has degree at least 0 so that the term of degree zero is not null in (1).

Therefore, λ must be algebraic, a contradiction. ■

4.4 Closure under complement for infinite words

Most of the weighted automata are not closed under complement. The next result is a direct extension of the boolean case.

Theorem 19. *The (non)deterministic Sup- and LimInf-automata, and the deterministic LimSup-automata are not closed under complement.*

Proof. The result follows from a similar result for the boolean version of these classes. For DSUP and NSUP, consider the language L_1 over $\Sigma = \{a, b\}$ such that $L_1(a^\omega) = 0$ and $L_1(w) = 1$ for all $w \neq a^\omega$. For DLINF and NLINF, consider the language L_2 over $\Sigma = \{a, b\}$ such that $L_2(\Sigma^*.a^\omega) = 1$ and $L(w) = 0$ for all words w containing infinitely many b 's, and for DLSUP, consider L_3 the complement of L_2 . ■

The next theorem is a positive result of closure under complementation for NLSUP. It reduces to the complementation of nondeterministic Büchi automata.

Theorem 20. *The nondeterministic LimSup-automata are closed under complement, with cost $O(m \cdot 2^{n \log n})$.*

Proof. Let $A = \langle Q, q_0, \Sigma, \delta, \gamma \rangle$ be a NLSUP, and let $V = \{\gamma(e) \mid e \in \delta\}$ be the set of weights that appear in A . For each $v \in V$, it is easy to construct a 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 edges with weight at least v . We then construct for each $v \in V$ a NBW \bar{A}_v (with accepting edges) that accepts the (boolean) complement of the language accepted by A_v . Finally, assuming that $V = \{v_1, \dots, v_n\}$ with $v_1 < v_2 < \dots < v_n$, we construct the NLSUP B_i for $i = 2, \dots, n$ where B_i is obtained from \bar{A}_{v_i} by assigning weight $-v_{i-1}$ to each accepting edges, and $-v_n$ to all the other edges. The complement of L_A is then $\max\{L_{B_2}, \dots, L_{B_n}\}$ which is accepted by a NLSUP by Theorem 12. ■

Theorem 21. *The deterministic Disc-automata are closed under complement, with cost $O(n)$.*

Proof Sketch. It suffices to replace each weight v of a DDISC by $1 - \lambda - v$ (where λ is the discount factor) to obtain the DDISC for the complement. ■

Theorem 22. *The deterministic LimAvg-automata are not closed under complement.*

Proof. Consider the DLAvg A over alphabet $\Sigma = \{a, b\}$ (shown in Fig. 3) that consists of a single self-loop state with weight 1 for a and 0 for b . Notice that $L_A(w.a^\omega) = 1$ and $L_A(w.b^\omega) = 0$ for all $w \in \Sigma^*$. To obtain a contradiction, assume that there exists a DLAvg B whose language is $L_B = 1 - L_A$. For all

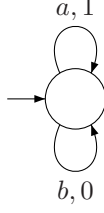


Fig. 3. Deterministic Limit-average Automaton.

finite words $w \in \Sigma^*$, let $L_B^{\text{Avg}}(w)$ be the average weight of the unique (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 $L_B^{\text{Avg}}(w.a^{m_w}) \leq \epsilon$ (since $L_B(w.a^\omega) = 0$). Hence, we can construct a word $w = b^{n_1}a^{m_1}b^{n_2}a^{m_2} \dots$ such that $L_A(w) \leq \epsilon$ and $L_B(w) \leq \epsilon$. Since $L_B = 1 - L_A$, this implies that $1 \leq 2\epsilon$, a contradiction. ■

Theorem 23. *The nondeterministic LimAvg- and Disc-automata are not closed under complement.*

Proof. The fact that NLAVG are not closed under complementation is as follows: it follows from Lemma 1 that the language $L^* = 1 - \max\{L_a, L_b\}$ cannot be expressed as a NLAVG, however, the language $\max\{L_a, L_b\}$ can be expressed as NLAVG by Theorem 12. That NDISC are not closed under complement can be obtained as follows: given $0 < \lambda < 1$, consider the language L_a^λ and L_b^λ that assigns to words the λ -discounted sum of a 's and b 's, respectively. The language L_a^λ and L_b^λ can be expressed as DDISC, and the max of them can be defined by NDISC. Observe that $L_a^\lambda(w) + L_b^\lambda(w) = \frac{1}{1-\lambda}$ for all $w \in \Sigma^\omega$. Therefore, $\min\{L_a^\lambda, L_b^\lambda\} = \frac{1}{1-\lambda} - \max\{L_a^\lambda, L_b^\lambda\}$. Since NDISC is not closed under min (Theorem 18), we immediately obtain that NDISC are not closed under complementation. ■

4.5 Closure under sum for infinite words

All weighted automata are closed under sum, except DLAVG and NLAVG.

Theorem 24. *The (non)deterministic Sup-automata are closed under sum, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$.*

Proof Sketch. The construction in the proof of Theorem 14 can be adapted as follows: define the weight $\gamma(\langle (q_1, v_1, q_2, v_2), \sigma, (q'_1, v'_1, q'_2, v'_2) \rangle)$ as $v'_1 + v'_2$ for each $\langle (q_1, v_1, q_2, v_2), \sigma, (q'_1, v'_1, q'_2, v'_2) \rangle \in \delta$. ■

Theorem 25. *The nondeterministic LimSup-automata are closed under sum, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$.*

Proof Sketch. Given two NLSUP A_1 and A_2 , we construct a NLSUP A for the sum of their languages as follows. Initially, we make a guess of a pair (v_1, v_2) of weights (v_i in A_i , for $i = 1, 2$) and we branch to 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 transition is $v_1 + v_2$. All other transitions (*i.e.* when b is unchanged) have weight $\min\{v_1 + v_2 \mid v_1 \in V_1 \wedge v_2 \in V_2\}$. ■

Theorem 26. *The deterministic LimSup-automata are closed under sum, with cost $O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$.*

Proof. Let $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$ be two DLSUP. We construct a DLSUP $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_A = L_{A_1} + L_{A_2}$. Let $V_i = \{\gamma_i(e) \mid e \in \delta_i\}$ be the set of weights that appear in A_i (for $i = 1, 2$). The automaton A implements the synchronized product of A_1 and A_2 , and keeps one bit $b(v_1, v_2)$ for each pair (v_1, v_2) of weights $v_1 \in V_1$ and $v_2 \in V_2$. For $i = 1, 2$, if $b(v_1, v_2) = i$, then A_i is expected to cross a transition with weight v_i . Whenever this occurs, the bit is set to $3 - i$. The weight of a transition in A is the largest value of $v_1 + v_2$ such that the corresponding bit $b(v_1, v_2)$ has changed in the transition. Formally, we define:

- $Q = Q_1 \times Q_2 \times [V_1 \times V_2 \rightarrow \{1, 2\}]$;
- $q_I = \langle q_I^1, q_I^2, b_I \rangle$ where $b_I(v_1, v_2) = 1$ for all $(v_1, v_2) \in V_1 \times V_2$;
- For each $\sigma \in \Sigma$, the set δ contains all the triples $(\langle q_1, q_2, b \rangle, \sigma, \langle q'_1, q'_2, b' \rangle)$ such that $(q_i, \sigma, q'_i) \in \delta_i$ ($i = 1, 2$), and for all $(v_1, v_2) \in V_1 \times V_2$, we have $b'(v_1, v_2) = 3 - b(v_1, v_2)$ if $\gamma_i(\langle q_i, \sigma, q'_i \rangle) = v_i$ for $i = b(v_1, v_2)$, and otherwise $b'(v_1, v_2) = b(v_1, v_2)$.
- γ is defined by $\gamma(\langle q_1, q_2, b \rangle, \sigma, \langle q'_1, q'_2, b' \rangle) = \max(\{v_{\min} \cup \{v_1 + v_2 \mid b'(v_1, v_2) \neq b(v_1, v_2)\})$ where v_{\min} is the minimal weight in $V_1 + V_2 = \{v_1 + v_2 \mid v_1 \in V_1 \wedge v_2 \in V_2\}$.

■

Theorem 27. *The (non)deterministic LimInf-automata are closed under sum with cost $O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$.*

Proof. Let $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$ be two DLINF. We construct a DLINF $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_A = L_{A_1} + L_{A_2}$. Let $V_i = \{\gamma_i(e) \mid e \in \delta_i\}$ be the set of weights that appear in A_i (for $i = 1, 2$). The automaton A implements the synchronized product of A_1 and A_2 , and keeps one bit $b(v_1, v_2)$ for each pair (v_1, v_2) of weights $v_1 \in V_1$ and $v_2 \in V_2$. If a transition

in A_i for some $i \in \{1, 2\}$ has weight less than v_i , then the bit $b(v_1, v_2)$ is set to \perp , otherwise is set to \top . The weight of a transition in A is the largest value of $v_1 + v_2$ such that the corresponding bit $b(v_1, v_2)$ is \top . Formally, we define:

- $Q = Q_1 \times Q_2 \times [V_1 \times V_2 \rightarrow \{\top, \perp\}]$;
- $q_I = \langle q_I^1, q_I^2, b_I \rangle$ where $b_I(v_1, v_2) = \perp$ for all $(v_1, v_2) \in V_1 \times V_2$;
- For each $\sigma \in \Sigma$, the set δ contains all the triples $(\langle q_1, q_2, b \rangle, \sigma, \langle q'_1, q'_2, b' \rangle)$ such that $(q_i, \sigma, q'_i) \in \delta_i$ ($i = 1, 2$), and for all $(v_1, v_2) \in V_1 \times V_2$, we have $b'(v_1, v_2) = \top$ if $\gamma_i(\langle q_i, \sigma, q'_i \rangle) \geq v_i$ for $i = 1, 2$, and otherwise $b'(v_1, v_2) = \perp$.
- γ is defined by $\gamma(\langle q_1, q_2, b \rangle, \sigma, \langle q'_1, q'_2, b' \rangle) = \max(\{v_{\min} \cup \{v_1 + v_2 \mid b'(v_1, v_2) = \top\})$ where v_{\min} is the minimal weight in $V_1 + V_2 = \{v_1 + v_2 \mid v_1 \in V_1 \wedge v_2 \in V_2\}$.

The result for NLINF follows from the fact that NLINF is reducible to DLINF. ■

Theorem 28. *The (non)deterministic Disc-automata are closed under sum, with cost $O(n_1 \cdot n_2)$.*

Proof Sketch. It is easy to see that the synchronized product of two NDISC (resp. DDISC) defines the sum of their languages, if the weight of a joint transition is defined as the sum of the weights of the corresponding transitions in the two NDISC (resp. DDISC). ■

Theorem 29. *The (non)deterministic LimAvg-automata are not closed under sum.*

Proof. Consider the alphabet $\Sigma = \{a, b\}$, and consider the DLAVG-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 $L_+ = L_a + L_b$. Assume that L_+ is defined by a NLAVG A with set of states Q (we assume w.l.o.g that every state in Q is reachable).

First, we claim that from every state $q \in Q$, there is a run of A over $a^{|Q|}$ that visit a cycle C^* with average weight 1. To see this, notice that from every state $q \in Q$, there is an infinite run ρ of A over a^ω whose value is 1 (since $L_+(w_q \cdot a^\omega) = 1$ for all finite words w_q). Consider the following decomposition of ρ . Starting with an empty stack, we push the states of ρ onto the stack as soon as all the states on the stack are different. If the next state is already on the stack, we pop all the states down to the repeated state thus removing a simple cycle of ρ . Let C_1, C_2, \dots be the cycles that are successively removed. Observe that the height of the stack is always at most $|Q|$. Let β be the largest average weight of the cycles C_i , $i \geq 1$, and let α_{\max} be the largest weight in A . Assume towards contradiction that $\beta < 1$. Then, for all $n > 0$, the value of the prefix of length n of ρ is at most:

$$\frac{\alpha_{\max} \cdot |Q| + \beta \cdot \sum_{i=1}^{k_n} |C_i|}{n}$$

where k_n is the number of cycles that have been removed from the stack when reading the first n symbols of ρ . Hence, the value of ρ is at most $\beta < 1$, which is a contradiction. Therefore, the average weight of some cycle $C^* = C_i$ is exactly⁵ 1 (there are finitely many different cycles as they are simple cycles). Since the height of the stack is at most $|Q|$, the cycle C^* is reachable in at most $|Q|$ steps.

Second, it can be shown analogously that from every state $q \in Q$, there is a run over $b^{|Q|}$ that visit a cycle C^* with average weight 1.

Third, for arbitrarily small $\epsilon > 0$, consider the word w and the run ρ of A over w generated inductively by the following procedure: w_0 is the empty word and ρ_0 is the initial state of A . We generate w_{i+1} and ρ_{i+1} from w_i and ρ_i as follows: (i) 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 ϵ and we can continue ρ_i and reach within at most $|Q|$ steps (and then repeat k times) a cycle C of average weight 1 and such that the average weight of this run prolonged by $|Q|$ arbitrary transitions is at least $1 - \epsilon$, *i.e.*

$$\frac{\gamma(\rho_i) + k \cdot |C| + 2\alpha_{\min} \cdot |Q|}{|\rho_i| + k \cdot |C| + 2 \cdot |Q|} \geq 1 - \epsilon$$

where α_{\min} is the least weight in A . This is possible since k can be chosen arbitrarily large. Let ρ'_i be the prolongation of ρ_i over w'_{i+1} ; (ii) 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 ϵ and as above, we can construct a continuation ρ''_i of ρ'_i whose average weight is at least $1 - \epsilon$ (even if prolonged by $|Q|$ arbitrary transitions); (iii) the word $w_{i+1} = w_i \cdot w'_{i+1} \cdot w''_{i+1}$ and the run ρ_{i+1} is ρ''_i . The word w and the run ρ are the limit of these sequences. We have $L_a(w) = L_b(w) = 0$ and thus $L_+(w) = 0$, while the value of ρ is at least $1 - \epsilon$, a contradiction. ■

Acknowledgment. We thank Wolfgang Thomas for pointing out the isolated cut-point problem.

References

- [CCH⁺05] A. Chakrabarti, K. Chatterjee, T. A. Henzinger, O. Kupferman, and R. Majumdar. Verifying quantitative properties using bound functions. In *CHARME*, LNCS 3725, pages 50–64. Springer, 2005.
- [CdAHS03] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT*, LNCS 2855, pages 117–133. Springer, 2003.
- [CDH08] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *CSL*, LNCS 5213, pages 385–400. Springer, 2008.
- [CGH⁺08] K. Chatterjee, A. Ghosal, T. A. Henzinger, D. Iercan, C. Kirsch, C. Pinello, and A. Sangiovanni-Vincentelli. Logical reliability of interacting real-time tasks. In *DATE*, pages 909–914. ACM, 2008.

⁵ It cannot be greater than 1 since $L_+(w \cdot a^\omega) = 1$ for all finite words w .

- [Cha07] K. Chatterjee. *Stochastic ω -Regular Games*. PhD thesis, University of California, Berkeley, 2007.
- [CK94] Karel Culik II and Juhani Karhumäki. Finite automata computing real functions. *SIAM J. Comput.*, 23(4):789–814, 1994.
- [CM00] Corinna Cortes and Mehryar Mohri. Context-free recognition with weighted automata. *Grammars*, 3(2/3):133–150, 2000.
- [Con92] Anne Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.
- [dAHM03] L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *ICALP*, LNCS 2719, pages 1022–1037. Springer, 2003.
- [DG07] M. Droste and P. Gastin. Weighted automata and weighted logics. *Th. C. Sci.*, 380(1-2):69–86, 2007.
- [DK03] Manfred Droste and Dietrich Kuske. Skew and infinitary formal power series. In *ICALP*, LNCS 2719, pages 426–438. Springer, 2003.
- [DKR08] Manfred Droste, Werner Kuich, and George Rahonis. Multi-valued MSO logics over words and trees. *Fundamenta Informaticae*, 84(3-4):305–327, 2008.
- [DR07] Manfred Droste and George Rahonis. Weighted automata and weighted logics with discounting. In *CIAA*, LNCS 4783, pages 73–84. Springer, 2007.
- [ÉK04] Zoltán Ésik and Werner Kuich. An algebraic generalization of omega-regular languages. In *MFCS*, LNCS 3153, pages 648–659. Springer, 2004.
- [EM79] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
- [GC03] Arie Gurfinkel and Marsha Chechik. Multi-valued model checking via classical model checking. In *CONCUR*, LNCS 2761, pages 263–277. Springer, 2003.
- [Kar78] R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.
- [KL07] O. Kupferman and Y. Lustig. Lattice automata. In *VMCAI*, LNCS 4349, pages 199–213. Springer, 2007.
- [KS86] Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1986.
- [Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.
- [Sch61] M. P. Schützenberger. On the definition of a family of automata. *Information and control*, 4(2-3):245–270, 1961.
- [Sha53] L. S. Shapley. Stochastic games. In *Proc. of the National Academy of Science USA*, volume 39, pages 1095–1100, 1953.
- [VW86] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *LICS*, pages 332–344. IEEE, 1986.
- [ZP96] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.