Expressiveness and Closure Properties for Quantitative Languages[⋆,⋆⋆]

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

[1] Institute of Science and Technology (IST), Austria
[2] LSV, ENS Cachan & CNRS, France
[3] 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. The value of a word w is the supremum of the values of the runs over w. 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,

⋆ A preliminary version of this paper appeared in the Proceedings of the 24th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press, 2009.

⋆⋆ This research was supported in part by the Swiss National Science Foundation under the Indo-Swiss Joint Research Programme, by the European Network of Excellence on Embedded Systems Design (ArtistDesign), by the European projects Combest, Quasimodo, and Gasics, by the PAI program Moves funded by the Belgian Federal Government, and by the CFV (Federated Center in Verification) funded by the F.R.S.-FNRS.

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 the behaviors of the program, and L_B contains all behaviors allowed by the specification. When boolean languages are defined by finite automata, this 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 supremum 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,CdHS03]. 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 than deterministic automata in the limit-average and discounted-sum cases.

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 also 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 a value in the neighborhood of η. We argue that isolated cut-point languages are robust, 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 for every limit-average automaton with rational weights in the interval $[0, 1]$ there is an equivalent limit-average automaton 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 the expressive power 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 operators 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.[4] 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 operators 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, (non)deterministic 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 boolean finite automata, while for limit-

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

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

(a) Finite words

	max.	min.	comp.	sum
N/DSUP	✓	✓	×	✓
N/DLINF	✓	✓	×	✓
DLSUP	✓	✓	×	✓
NLSUP	✓	✓	✓	✓
DLAVG	×	×	×	×
NLAVG	✓	×	×	×
DDISC	×	×	✓	✓
NDISC	✓	×	×	✓

(b) Infinite words

Table 1. Closure properties. The meaning of the acronyms is described on p.7.

average and discounted-sum automata, the proofs 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 $L_A(w) + L_B^c(w) > 1$, 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 theory [Sha53,EM79,Con92,ZP96,CdHS03]. 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 $L : \Sigma^+ \to \mathbb{R}$ or a mapping $L : \Sigma^\omega \to \mathbb{R}$, 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*, i.e., for all $q \in Q$ and $\sigma \in \Sigma$, there exists q' such that $(q, \sigma, q') \in \delta$;
- $\gamma : \delta \to \mathbb{Q}$ 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* $\mathsf{Val} : \mathbb{Q}^+ \to \mathbb{R}$ (resp. $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$), we say that the Val-automaton A defines the quantitative language L_A such that for all $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$):

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

We assume that $\mathsf{Val}(v)$ is bounded when the numbers in v are taken from a finite set (namely, the set of weights in A), and since weighted automata are total, every word has at least one run and thus $L_A(w)$ is not infinite.

We consider the following value functions to define quantitative languages (they all satisfy the boundedness assumption above). Given a finite sequence $v = v_1 \dots v_n$ of rational numbers, define

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

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

- $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$;
- $\mathsf{LimSup}(v) = \limsup_{n\to\infty} v_n = \lim_{n\to\infty} \sup\{v_i \mid i \geq n\}$;
- $\mathsf{LimInf}(v) = \liminf_{n\to\infty} v_n = \lim_{n\to\infty} \inf\{v_i \mid i \geq n\}$;

- $\mathsf{LimAvg}(v) = \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i$;
- for $0 < \lambda < 1$, $\mathsf{Disc}_\lambda(v) = \sum_{i=0}^{\infty} \lambda^i \cdot v_i$;

Intuitively for a sequence $v = v_0 v_1 \ldots$ of rational numbers from the finite set V, the Sup function chooses the maximal number that appear in v; the LimSup function chooses the maximal number that appear infinitely often in v; the LimInf function chooses the minimal number that appear infinitely often in v; the LimAvg functions gives the long-run average of the numbers in v; and the Disc_λ gives the discounted sum of the numbers in v. Note that $\mathsf{LimAvg}(v)$ is defined using $\liminf$ and is therefore well-defined; all results of this paper hold also if the limit-average of v is defined instead as $\limsup_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i$. One could also consider the value function $\inf\{v_n \mid n \geq 0\}$ and obtain results analogous to the Sup value function. Note that the classical finite-word acceptance condition of finite automata (defining regular languages) can be encoded by Last-automata with weights in $\{0, 1\}$, while Büchi and coBüchi automata are special cases of respectively LimSup- and LimInf-automata, with weights in $\{0, 1\}$. The class of languages defined by nondeterministic Büchi automata is called ω-regular.

Significance of value functions. The value functions provide natural generalizations of the classical boolean languages, they are complete for different levels of the Borel hierarchy, and they have been well studied in the context of game theory.

1. The Sup value function is the natural quantitative generalization of the reachability condition and is complete for the first level of the Borel hierarchy (Σ_1 complete).
2. The LimSup and LimInf objectives are the natural quantitative generalizations of the classical Büchi and coBüchi conditions. Moreover, the LimSup and LimInf objectives are complete for the second level of the Borel hierarchy, and hence important and canonical quantitative functions (LimSup and LimInf objectives are Π_2 and Σ_2 complete, respectively) (see [Wad84,MP92] for details related to completeness and reducibility of objectives in the Borel hierarchy).
3. The LimAvg and Disc_λ value functions have been studied in many different contexts in game theory. Discounted functions on graph games were introduced in the seminal work of Shapley [Sha53], and have been extensively studied in economics. Discounted conditions have also been studied for discounting the future in systems theory [dAHM03]. The LimAvg function has also been studied extensively in the context of games on graphs: the works of Everett [Eve57], Liggett-Lippman [LL69], Hopfman-Karp [HK66], Ehrenfeucht-Mycielski [EM79], Mertens-Neyman [MN81], Zwick-Paterson [ZP96] have studied different classes of games with LimAvg

objective. Also see the books [FV97,Put94] for applications of discounted and limit-average value functions in the context of games on graphs. Moreover, the LimAvg value function is complete for the third level of the Borel hierarchy (Π_3-complete) [Cha07a].

Hence the value functions considered are classical, canonical, and well-studied in the branching-time framework of games on graphs, and we study them in the linear-time framework of weighted automata.

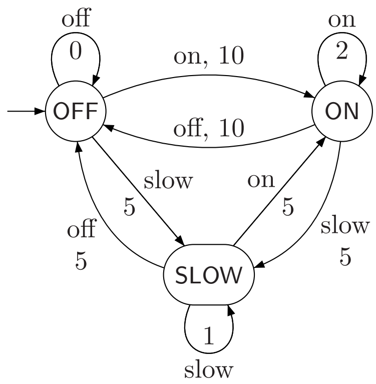
Notation. Classes of weighted automata over infinite words are denoted with acronyms of the form xy where x is either N(ondeterministic), D(eterministic), or $^{\mathrm{N}}\!/_{\mathrm{D}}$ (when deterministic and nondeterministic automata have the same expressiveness), and y is one of the following: Sup, Lsup(LimSup), Linf(LimInf), Lavg(LimAvg), or Disc. For Büchi and coBüchi condition, we use BW and CW respectively.

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].

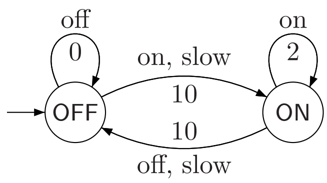
Composition. Given two quantitative languages L and L' over Σ, and a rational number c, we denote by $\max(L, L')$ (resp. $\min(L, L')$, $L+L'$, $c+L$, and cL) the quantitative language that assigns $\max\{L(w), L'(w)\}$ (resp. $\min\{L(w), L'(w)\}$, $L(w)+L'(w)$, $c+L(w)$, and $c \cdot L(w)$) to each 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 energy consumption of a motor. The automaton B in Fig. 1(b) specifies the maximal energy consumption to maintain the motor on or off, and the 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 energy consumption of two different motors, one needs to define composition operations for weighted automata to obtain the maximal, minimal, and sum of the average energy consumption of the motors.

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) Limit-average automaton A.



(b) Limit-average automaton B.

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

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 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.

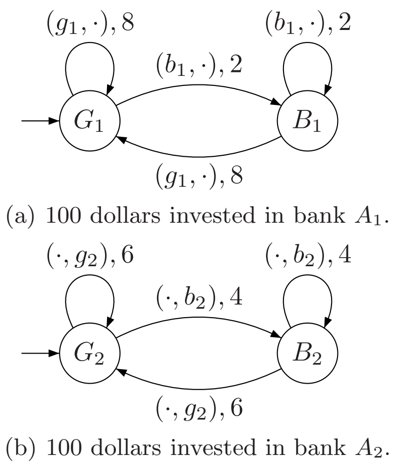


Fig. 2. The discounted-sum automaton models of two banks.

The combined automaton is obtained as the composition of C_1 and C_2 under the sum operator.

3 Expressiveness Results

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 $L : \Sigma^\omega \to \{0, 1\}$. 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 (see e.g. [KS86,CM00]). It is easy to see that the cut-point languages of Max- and Last-automata are regular (they have the same acceptance condition as finite automata), those of Sum-automata are context-free (using a stack to simulate accumulated weights), and those of Sup-, LimSup-, and LimInf-automata are ω-regular (they have the same acceptance condition as Büchi and coBüchi automata).

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. One direction of this result follows from Theorem 1, and the other direction follows from [CDH08, Theorems 13 and 14] where ω-regular languages are given that are not definable as cut-point language of nondeterministic limit-average and discounted-sum automata.

Theorem 1. *There exist 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 [Cha07b] 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 $v_a(w) = \sum_{i|w_i=a} \lambda^{i-1}$ be the discounted sum of a's in w. We say that w is *ambiguous* if $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq v_a(w) < 1$. 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 $\hat{w}$ all of whose finite prefixes are ambiguous (and $L_2(\hat{w}) = 1$). To do this, assume that $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq v_a(w) < 1$, and show that either $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq v_a(w.a) < 1$ or $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq v_a(w.b) < 1$. Since $v_a(w.a) = v_a(w) + \lambda^{|w|}$ and $v_a(w.b) = v_a(w)$, we have to show that $1 - \frac{\lambda^{|w|}}{1-\lambda} \leq v_a(w) < 1 - \lambda^{|w|}$ or $1 - \frac{\lambda^{1+|w|}}{1-\lambda} \leq v_a(w) < 1$. 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 $\hat{w}$ (which we denote by $\hat{w}_{[1\ldots n]}$) 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 continuations $w' \in \Sigma^\omega$, we have $\hat{w}_{[1\ldots n]}.w' \in L_2^{\geq 1}$ if and only if $\hat{w}_{[1\ldots m]}.w' \in L_2^{\geq 1}$ $(\star)$.

In particular, consider the continuations $\hat{w}_{[n+1\ldots]}$ and $\hat{w}_{[m+1\ldots]}$, and for each $i \geq 1$, let $\gamma^i = v_a(\hat{w}_{[1\ldots i]})$ and $K^i = L_2(\hat{w}_{[i+1\ldots]})$. Then, we have $\gamma^i + \lambda^i \cdot K^i = 1$, and thus $\gamma^m + \lambda^m \cdot K^n \leq 1$ iff $K^n \leq K^m$. Since either $K^n \leq K^m$ or $K^m \leq K^n$, we have either $L_2(\hat{w}_{[1\dots m]}.\hat{w}_{[n+1\dots]}) \leq 1$ or $L_2(\hat{w}_{[1\dots n]}.\hat{w}_{[m+1\dots]}) \leq 1$. By $(\star)$, this implies that either $L_2(\hat{w}_{[1\dots m]}.\hat{w}_{[n+1\dots]}) = 1$, or $L_2(\hat{w}_{[1\dots n]}.\hat{w}_{[m+1\dots]}) = 1$, and in both cases since $L_2(\hat{w}) = 1$, we get

$$\frac{1-\gamma^m}{\lambda^m} = \frac{1-\gamma^n}{\lambda^n}.$$

This implies $\lambda^{m-n}(1-P(\lambda)) = 1 - Q(\lambda)$ where $P(\lambda) = v_a(\hat{w}_{[1\dots n]})$ and $Q(\lambda) = v_a(\hat{w}_{[1\dots m]})$ 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 $\hat{w}$ must be b since a is not ambiguous). Second, every coefficient in the equation is in the set $\{-2,-1,0,1,2\}$, 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 thus 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 different from $L_1^{\geq 1}$, 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 argue that isolated cut-point languages are robust, in that they 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 $L_1, L_2 : \Sigma^\omega \to \mathbb{R}$ is defined by

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

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 $A, B \in \mathcal{C}$ such that B is an ϵ-approximation of A, we have $D_{\sup}(L_A, L_B) \leq \eta$. 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-approximations B of A (with $B \in \mathcal{C}$), we have $D_{\sup}(L_A, L_B) \leq \eta$ (here the value of ϵ can depend for instance on the weights of the automaton A).

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, $D_{\sup}(L_A, L_B) \leq \epsilon$ 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 weights.

Corollary 1. *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 a rational $\epsilon > 0$ such that for all ϵ-approximations B of A, we have $L_A^{\geq\eta} = L_B^{\geq\eta}$ (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 3. *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 A 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 $V = \max_{(q,\sigma,q') \in \delta_A} |\gamma(q,\sigma,q')|$ is the largest weight in A. Note that u_n is a bound on the difference between the λ-discounted sum of the weights in any infinite run $\hat{r}$ of A and the λ-discounted sum of the weights in the prefix of length n of $\hat{r}$, and $u_n \to 0$ when $n \to \infty$.

Consider an arbitrary 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 $\gamma(r) \not\in [\eta-\epsilon+u_n, \eta+\epsilon-u_n]$, because otherwise, the value of any (infinite) continuation of r would lie in the interval $[\eta-\epsilon, \eta+\epsilon]$, which would be a contradiction to the fact that η is an isolated cut-point of L. Moreover, if $\gamma(r) \leq \eta - \epsilon + u_n$, then any (infinite) continuation of r has value less than $\eta - \epsilon + 2u_n < \eta + \epsilon$, and therefore 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 (for Büchi condition), and have a self-loop on Σ. ∎

Theorem 4. *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 $C_1, C_2, \ldots, C_k$ 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 $\eta \not\in [m_i, M_i]$ for all $1 \leq i \leq k$. A deterministic Büchi automaton (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 compare the expressive power. We show that the restriction does not change the class of quantitative languages definable by limit-average automata, but does reduce the expressive power of discounted-sum automata.

Theorem 5. *The class of nondeterministic (resp., deterministic)* LimAvg*-automata with rational weights in $[0, 1]$ is reducible to the class of nondeterministic (resp., deterministic)* LimAvg*-automata with weights 0 and 1 only.*

Proof. Given a NLavg $A = \langle Q, q_I, \Sigma, \delta, \gamma\rangle$ with weights in $[0, 1]$, we construct a NLavg B with weights in $\{0, 1\}$ 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 $p \in \mathbb{N}$ such that $v = \frac{p}{n}$ (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 $\{0, 1, \ldots, n_A - 1\}$). 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 $j = i + v \cdot n_A$ if $\frac{i}{n_A} + v < 1$; the weight of such a transition is 0 in γ',
  - $((q, i), \sigma, (q', j))$ 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'.

  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 $n \to \infty$. 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 6. *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 DDisc 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 A with weights in $\{0, 1\}$. 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 $\lambda + \lambda^2 + \lambda^3 + \cdots = \frac{\lambda}{1-\lambda}$ which is strictly smaller than $L_\lambda(a^\omega) = \frac{1+\lambda}{2(1-\lambda)}$. This shows that A cannot exist. ■

4 Closure Properties

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 $\mathrm{op}(\cdot,\cdot)$ (resp. a unary operator $\mathrm{op}'(\cdot)$) if for all $A_1, A_2 \in \mathcal{C}$, there exists $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \mathrm{op}(L_{A_1}, L_{A_2})$ (resp. $L_{A_{12}} = \mathrm{op}'(L_{A_1})$). 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 $L_{A_{12}} = \mathrm{op}(L_{A_1}, L_{A_2})$ 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 $L_{A_{12}} = \mathrm{op}'(L_{A_1})$ 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 in p).

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. Table 1(a) summarizes the closure properties for finite words, and Table 1(b) for infinite words.

4.1 Closure properties for finite words

For finite words, we consider closure under max, min, complement, and sum for Max-, Last- and Sum-automata.

Theorem 7. *Deterministic* Max- *and* Last-*automata are closed under* max*, with cost* $O(n_1 \cdot n_2)$*. Nondeterministic* Max-*,* 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 Max- and Last-automata, the result is obtained using a standard synchronized product construction, where the weight of a transition in the product is the maximum of the corresponding transition weights in the two automata. 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 2 in [CDH08]). ■

Theorem 8. *Deterministic and nondeterministic* Max*-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 Max-automata is the same as for Sup-automata over infinite words given in the proof of Theorem 13.

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 with state space Q defines L_m. First, 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 any 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, thus A cannot exist. ■

Theorem 9. *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* Max*-automata are not closed under complement.*

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 Max-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 8, 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 $\min(L_a, L_b) = 1 - \max(1 - L_a, 1 - L_b)$ and (i) $1 - L_a$ and $1 - L_b$ are definable by Sum-automata, and (ii) nondeterministic Sum-automata are closed under max (Theorem 7), the language $\max(1-L_a, 1-L_b)$ is definable by a nondeterministic Sum-automaton, but not its complement and the result follows. ∎

Theorem 10. *Every class of weighted automata over finite words is 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* Max-*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).

We give the construction for two Max-automata $A_1 = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $A_2 = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$. We construct a Max-automaton $A_{12} = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ such that $L_{A_{12}} = L_{A_1} + L_{A_2}$. 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 $\gamma(\langle (q_1, v_1, q_2, v_2), \sigma, (q_1', v_1', q_2', v_2') \rangle) = v_1' + v_2'$ 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 deterministic Max-automata follows. ∎

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 11. *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 $O((m_1 + m_2)^{n_1 + n_2})$.*

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 weighted 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 12. *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 16 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 $\widehat{w}_\infty$ constructed as follows. We start with the empty word $\widehat{w}_0$ and the initial state q_0 of A, and for all $j \geq 0$, we construct $(\widehat{w}_{j+1}, q_{j+1})$ from $(\widehat{w}_j, q_j)$ 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 $\widehat{w}_{j+1} = \widehat{w}_j \cdot a^{|Q|+|C_{j+1}|} \cdot b^{2|Q|}$. 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 $\widehat{w}_\infty$ is the limit of this construction ($\widehat{w}_j$ is a prefix of $\widehat{w}_\infty$ 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: $\frac{\gamma(\widehat{w}_j)}{|\widehat{w}_j|} = \frac{\gamma(w_j)+\alpha_j}{|w_j|+\alpha_j}$. Hence for all $\varepsilon > 0$, there exists j_ε such that for all $j \geq j_\varepsilon$ we have

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

This shows that $\liminf_{j\to\infty} \frac{\gamma(\widehat{w}_j)}{|\widehat{w}_j|} \geq \frac{2}{3} + \frac{1}{12|Q|}$ and thus we have $L_A(\widehat{w}_\infty) \geq \frac{2}{3} + \frac{1}{12|Q|}$. Since $1 \leq |C_i| \leq |Q|$ for all $i \geq 1$, we have $L_m(\widehat{w}_\infty) \leq \frac{2}{3}$ which is a contradiction. ■

4.3 Closure under min for infinite words

The positive results about closure properties under min for quantitative languages generalize the closure properties of boolean languages under intersection. The constructions are straightforward extensions of the standard constructions for finite, Büchi, and coBüchi automata (see e.g. [Var96]).

Theorem 13. *The (non)deterministic* Sup*-automata are closed under* min*, with cost* $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$*,*

Proof. The construction in the proof of Theorem 10 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 $\min\{v_1',v_2'\}$ for each $\langle(q_1,v_1,q_2,v_2),\sigma,(q_1',v_1',q_2',v_2')\rangle \in \delta$. ■

Theorem 14. *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 occur 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 that 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 (i.e., they have weight 1 while all other edges have weight 0). 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 $m = |V_1 \cup V_2|$ (and assume $V_1 \cup V_2 = \{v_1, \dots, v_m\}$);
– $q_I = (q_I^1, q_I^2, b_1, \dots, b_m)$ where $b_i = 1$ for all $1 \le i \le m$;
– 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 $1 \le j \le m$, we have $b_j' = 3 - b_j$ if $\gamma_{b_j}(q_{b_j}, \sigma, q_{b_j}') \ge v_j$, 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 $v = \max(\{v_{\min}\} \cup \{v_j \mid b_j \ne b_j'\})$ where $v_{\min}$ is the minimal weight in $V_1 \cup V_2$.

■

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 $\gamma_j(q_j, \sigma, q_j') \ge v$, and $j' = j$ otherwise.
– γ is defined by $\gamma(q_I, \sigma, \langle q_1, q_2, 1, v\rangle) = 0$ and $\gamma(\langle q_1, q_2, j, v\rangle, \sigma, \langle q_1', q_2', j', v'\rangle)$ is v if $j \ne j'$ and $v_{\min}$ otherwise, where $v_{\min}$ is the minimal weight in $V_1 \cup V_2$.

For the case of LimInf-automata A_1, A_2, we can use the synchronized product $A_{12} = A_1 \times A_2$, where the weight of a joint transition in A_{12} is the minimum of the corresponding transition weights in A_1 and A_2. It is easy to see that $L_{A_{12}} = \min(L_{A_1}, L_{A_2})$ in both cases, and A_{12} is deterministic when A_1 and A_2 are deterministic. This case is simpler also because for LimInf-automata, deterministic and nondeterministic automata have the same expressive power.

■

On the negative side, the (deterministic or not) limit-average and discounted-sum automata are not closed under min.

Theorem 16. *The (non)deterministic* LimAvg*-automata are not closed under* min.

Proof.

Consider the alphabet $\Sigma = \{a, b\}$, and consider the languages L_a and L_b that assign the long-run average number of a's and b's, respectively. Note that there exist DLAVG for the languages L_a and L_b.

We show that there is no NLAVG for the language $L_m = \min\{L_a, L_b\}$. To obtain a contradiction, assume that there exists a NLAVG A for L_m. 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, $L_m(w) = \frac{1}{2}$, i.e., we have a contradiction. W.l.o.g., we assume that there is an a-cycle C such that the 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 $L_m(w) = 0$. Hence the result follows. ■

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

Theorem 17. *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 $w = w_0 w_1 \cdots \in \Sigma^* \cup \Sigma^\omega$, let

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

be the λ-discounted sum of the a's (resp. b's) of w. Then, $L_a^\lambda(w) = v_a(w)$ and $L_b^\lambda(w) = v_b(w)$. 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 2 and 3 in [CDH08], there exists an infinite word $\hat{w}$ such that $v_a(\hat{w}) = v_b(\hat{w})$.

Since $v_a(\hat{w})+v_b(\hat{w}) = \frac{1}{1-\lambda}$, we have $L_m(\hat{w}) = \frac{1}{2(1-\lambda)}$ 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 $v_a(w_1.(w_2)^\omega) = v_b(w_1.(w_2)^\omega)$ 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 18. *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 19. *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

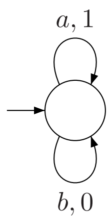


Fig. 3. Deterministic Limit-average Automaton.

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 $1 - v_{i-1}$ to each accepting edges, and $1 - v_n$ to all 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 11. ∎

Theorem 20. *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 21. *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 finite words $w \in \Sigma^*$, let $L_B^{\mathsf{Avg}}(w)$ 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 $L_B^{\mathsf{Avg}}(w.a^{m_w}) \leq \epsilon$ (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 22. *The nondeterministic* LimAvg- *and* Disc-*automata are not closed under complement.*

Proof. The fact that NLAVG are not closed under complementation is as follows. Consider the quantitative language $L^* = 1 - \max\{L_a, L_b\}$ where L_a and L_b assign the long-run average number of a's and b's, respectively. Exactly the same argument as in the proof of Theorem 16 shows that L^* cannot be expressed as a NLAVG, while the language $\max\{L_a, L_b\}$ can be expressed as NLAVG by Theorem 11.

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

Proof. The construction is the same as for Max-automata over finite words given in the proof of Theorem 10. ∎

Theorem 24. *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 $\min\{v_1 + v_2 \mid v_1 \in V_1 \land v_2 \in V_2\}$. ∎

Theorem 25. *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:

- $Q = Q_1 \times Q_2 \times [V_1 \times V_2 \to \{1, 2\}]$;
- $q_I = \langle q_I^1, q_I^2, b_I \rangle$ 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 26. *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 NLinf. We construct a NLinf $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$ 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 $\bot$, 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:

- $Q = Q_1 \times Q_2 \times [V_1 \times V_2 \to \{\top, \bot\}]$;
- $q_I = \langle q_I^1, q_I^2, b_I \rangle$ where $b_I(v_1, v_2) = \bot$ 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 $b'(v_1, v_2) = \bot$.
- 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 DLinf follows from the fact A is deterministic if A_1 and A_2 are deterministic. ■

Theorem 27. *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 28. *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 $\alpha_{\max}$ 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 exactly[5] 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

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

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 $\alpha_{\min}$ 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 *Proc. of CHARME: Correct Hardware Design and Verification Methods*, LNCS 3725, pages 50–64. Springer, 2005.

[CDH08] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proc. of CSL: Computer Science Logic*, LNCS 5213, pages 385–400. Springer, 2008.

[CdHS03] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *Proc. of EMSOFT: Embedded Software*, LNCS 2855, pages 117–133. Springer, 2003.

[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 *Proc. of DATE: Design, Automation and Test*. ACM, 2008.

[Cha07a] K. Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388:181–198, 2007.

[Cha07b] 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. *Information and Computation*, 96:203–224, 1992.

[dAHM03] L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *Proc. of ICALP: International Colloquium on Automata, Languages and Programming*, LNCS 2719, pages 1022–1037. Springer, 2003.

[DG07] M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.

[DK03] M. Droste and D. Kuske. Skew and infinitary formal power series. In *Proc. of ICALP: International Colloquium on Automata, Languages and Programming*, LNCS 2719, pages 426–438. Springer, 2003.

[DKR08] M. Droste, W. Kuich, and G. Rahonis. Multi-valued MSO logics over words and trees. *Fundamenta Informaticae*, 84:305–327, 2008.

[DR07] M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting. In *Proc. of CIAA: Implementation and Application of Automata*, LNCS 4783, pages 73–84. Springer, 2007.

[ÉK04] Zoltán Ésik and Werner Kuich. An algebraic generalization of omega-regular languages. In *Proc. of MFCS: Mathematical Foundations of Computer Science*, LNCS 3153, pages 648–659. Springer, 2004.

[EM79] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.

[Eve57] H. Everett. Recursive games. In *Contributions to the Theory of Games III*, volume 39 of *Annals of Mathematical Studies*, pages 47–78, 1957.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[GC03] A. Gurfinkel and M. Chechik. Multi-valued model checking via classical model checking. In *Proc. of CONCUR: Concurrency Theory*, LNCS 2761, pages 263–277. Springer, 2003.

[HK66] A.J. Hoffman and R.M. Karp. On nonterminating stochastic games. *Management Sciences*, 12(5):359–370, 1966.

[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 *Proc. of VMCAI: Verification, Model Checking, and Abstract Interpretation*, LNCS 4349, pages 199–213. Springer, 2007.

[KS86] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1986.

[LL69] T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *Siam Review*, 11:604–607, 1969.

[MN81] J.F. Mertens and A. Neyman. Stochastic games. *International Journal of Game Theory*, 10:53–66, 1981.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.

[Put94] M.L. Puterman. *Markov Decision Processes*. John Wiley and Sons, 1994.

[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*, 4:245–270, 1961.

[Sha53] L. S. Shapley. Stochastic games. In *Proc. of the National Acadamy of Science USA*, volume 39, pages 1095–1100, 1953.

[Var96] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Proc. of Banff Higher Order Workshop*, LNCS 1043, pages 238–266. Springer, 1996.

[VW86] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proc. of LICS: Logic in Computer Science*, pages 332–344. IEEE, 1986.

[Wad84] W.W. Wadge. *Reducibility and Determinateness of Baire Spaces*. PhD thesis, University of California, Berkeley, 1984.

[ZP96] U. Zwick and M.S. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.