Expressiveness and Closure Properties for Quantitative Languages*

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

[1]Institute of Science and Technology (IST), Austria
[2]Université Libre de Bruxelles (ULB), Belgium
[3]École Polytechnique Fédérale de Lausanne (EPFL), 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 framework is called the *automata-theoretic approach* to model checking [24].

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 [22, 19] and ω-series for infinite words [9, 13, 16]. In [5], 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 [1, 2]. Quantitative languages can also be used to specify and verify reliability requirements: if a spe-

*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.

†Postdoctoral researcher of the F.R.S.-FNRS.

cial 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 [6]. The discounted sum can be used to specify that failures happening later are less important than those happening soon [10].

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 [5], 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 [21]. 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 [5]. 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.[1] 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 LimAvg- and Disc-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 [5]. 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). Due to lack of space, we omit the details of some proofs that can be found in [4].

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 [23, 15, 7, 25, 2]. It is therefore natural to use the

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

same functions in the linear-time context of languages and automata.

Weighted automata with discounted sum have been considered in [14], 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 [11], over trees [12], or using finite lattices [17], but none of these works has addressed the expressiveness questions and closure properties for quantitative languages that are studied here.

The lattice automata of [20] 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 $r = q_0\sigma_1q_1\sigma_2\dots$ 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 $\gamma(r) = v_0v_1\dots$ 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. $w \in \Sigma^\omega$): $L_A(w) = \sup\{\mathsf{Val}(\gamma(r)) \mid r$ is a run of A 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

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

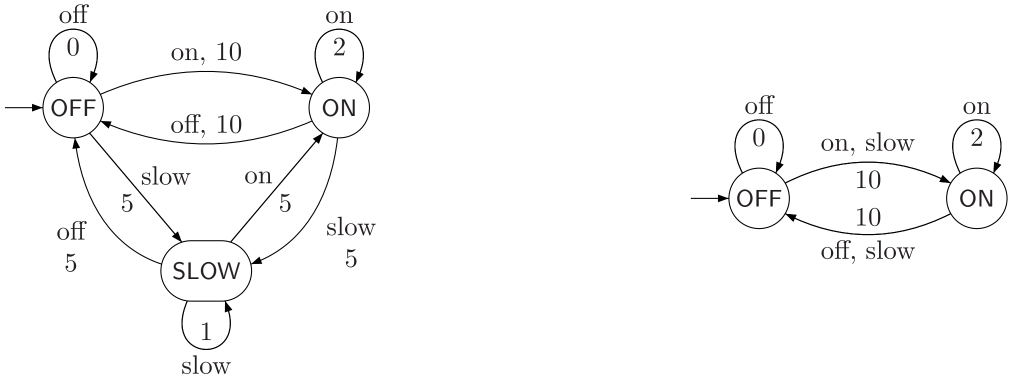
Given an infinite sequence $v = v_0v_1\dots$ of rational numbers, define

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

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.

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 $\substack{\text{D}\\\text{N}}$ (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 automata, we use the classical acronyms 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 [5].



(a) Limit-average automaton A.

(b) Limit-average automaton B.

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

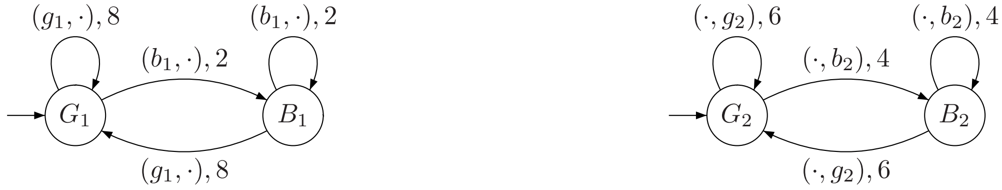
Composition. Given two quantitative languages L and L' over Σ, and a rational number c, we denote by $\max(L, L')$ (resp. $\min(L, L')$, $L+L'$, $c+L$, and cL) the quantitative language that assigns $\max\{L(w), L'(w)\}$ (resp. $\min\{L(w), L'(w)\}$, $L(w) + L'(w)$, $c + L(w)$, and $c \cdot L(w)$) to 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 Figure 1(b) specifies the maximal power 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 Figure 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 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

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 [5] 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



(a) 100 dollars invested in bank A_1.

(b) 100 dollars invested in bank A_2.

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

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 [21], then generalized to inverse image recognition for semiring automata over finite words (see e.g. [19, 8]). 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 [5, Theorems 13 and 14].

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

Proof sketch. 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 [3] 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 $r_a(w) = \sum_{i|w_i=a} \lambda^i$ 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 $w.b^\omega$). It is easy to show that for all $\lambda > \frac{1}{2}$, if w is ambiguous, then either $w.a$ or $w.b$ is ambiguous. Therefore, there exists an infinite word $w^{\preceq}$ all of whose finite prefixes are ambiguous (and $L_2(w^{\preceq}) = 1$).

The proof proceeds by showing that if there exists a nondeterministic Büchi automaton A for $L_2^{\geq 1}$, then the set of states S_n reached in A after the first n letters of $w^{\preceq}$ must be different for each n, i.e. $n \neq m$ implies $S_n \neq S_m$, and therefore 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 [21].

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 have stability properties, 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-approximation 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.*

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 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 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 $C_1, C_2, \dots, C_k$ 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 [18]). 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 automata in $\mathcal{C}$ whose weights are rational numbers in R.

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 $\text{NL}\textsc{avg}_{[0,1]}$-automaton $A = \langle Q, q_I, \Sigma, \delta, \gamma\rangle$, we construct a $\text{NL}\textsc{avg}_{\{0,1\}}$-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 $\{0, 1, \ldots, n_A - 1\}$). 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 $\text{ND}\textsc{isc}_{[0,1]}$ 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 $\text{ND}\textsc{isc}_{\{0,1\}}$ 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 $\lambda + \lambda^2 + \lambda^3 + \cdots = \frac{\lambda}{1-\lambda}$ 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 $\text{op}(\cdot, \cdot)$ (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 rationals).

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

Table 1(a) summarizes the closure properties of Max-, Last- and Sum-automata under max, min, complement, and sum. Detailed proofs are given in [4].

Since Max- and Last-automata can be determinized, they have the same closure properties as their deterministic counterpart. They are closed under all operations except complementation for Max-automata, as a direct consequence of the same result 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 b).

The cost of every positive closure property in Table 1(a) is polynomial except the complementation of nondeterministic Last-automata which is exponential.

Finally, Sum-automata are not closed under min as the language $L_m = \min\{L_a, L_b\}$ where $L_\sigma(w)$ is the number of occurrences of σ in w (for $\sigma \in \{a, b\}$) is definable as the minimum of two deterministic-Sum languages, but not by a nondeterministic Sum-automaton. Being closed under max but not under min, nondeterministic Sum-automata are not closed under complement. We now proceed with quantitative languages over infinite words, for which the closure properties are summarized in Table 1(b).

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

(a) Finite words

	max	min	comp.	sum
$^{\text{D}}_{\text{N}}$ Sup	✓	✓	×	✓
$^{\text{D}}_{\text{N}}$ Linf	✓	✓	×	✓
DLsup	✓	✓	×	✓
NLsup	✓	✓	✓	✓
DLavg	×	×	×	×
NLavg	✓	×	×	×
DDisc	×	×	✓	✓
NDisc	✓	×	×	✓

(b) Infinite words

Table 1. Closure properties. A class $\mathcal{C}$ of weighted automata is closed under an operator op iff the entry $(\mathcal{C}, \mathrm{op})$ is ✓.

4.2 Closure under max for infinite words

The maximum of two quantitative languages defined by nondeterministic weighted automata can be obtained by an initial nondeterministic choice between the two automata. This observation was also made in [14] for discounted-sum automata. For deterministic weighted automata, a synchronized product can be used for Sup and LimSup, while for LimInf we use the fact that NLinf can be determinized with an exponential blow-up [5].

Theorem 7 *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 $O(n_1 \cdot n_2)$; and the deterministic* LimInf-*automata with cost $O((m_1 + m_2) \cdot 2^{n_1+n_2})$.*

For limit average and discounted sum, deterministic automata are not closed under max. The result for discounted sum directly follows from the proof of Theorem 16 in [5] (showing that discounted sum automata cannot be determinized), while for limit-average, we show that there is no DLavg that defines the language $L_m = \max(L'_a, L'_b)$ where L'_a (resp. L'_b) is the language defined by the DLavg that assigns weight 1 (resp. 0) to a's and weight 0 (resp. 1) to b's. Roughly, the argument is based on the analysis of the cycles of any DLavg A that would define L_m, and on the fact that the word $a^{|Q|+k} \cdot b^{2|Q|}$ has value $\frac{2}{3}$ in L_m, for all $k \leq |Q|$ where Q is the set of states of A.

Theorem 8 *The deterministic* LimAvg- *and* Disc-*automata are not closed under* max*.*

4.3 Closure under min for infinite words

Theorem 9 generalizes the closure properties 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 9 *The (non)deterministic* Sup-*automata are closed under* min*, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$; the* *(non)deterministic* LimInf*-automata with cost* $O(n_1 \cdot n_2)$*; the* LimSup*-automata with cost* $O(n_1 \cdot n_2 \cdot 2^{m_1+m_2})$ *for deterministic, and cost* $O(n_1 \cdot n_2 \cdot (m_1 + m_2))$ *for nondeterministic automata.*

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 assign the long-run average number of a's and b's, respectively. There is no nondeterministic* LimAvg*-automaton that defines the language* $min(L_a, L_b)$*.*

Proof. To obtain a contradiction, assume that there exists a NLAVG A with set of states Q for $L_{\min} = min(L_a, L_b)$. We first claim that there must be either an a-cycle or a b-cycle C reachable in A such that the sum of the weights in C is positive. Otherwise, if the sum of the weights of all a-cycles and b-cycles 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 $\alpha = \lceil \beta \rceil$. Consider the word $w = (a^{3\cdot\alpha\cdot|Q|} \cdot b^{3\cdot\alpha\cdot|Q|})^\omega$. For all runs r of A over w, the long-run average of the weights in r is bounded by

$$\frac{2 \cdot \beta \cdot |Q|}{6 \cdot \alpha \cdot |Q|} \leq \frac{1}{3}.$$

The above bound is as follows: a run over $a^{3\cdot\alpha\cdot|Q|}$ can be decomposed into a-cycles (with sum of weight at most 0) and a path of length at most $|Q|$ with sum of weights at most $|Q| \cdot \beta$. A similar argument holds for the segment of $b^{3\cdot\alpha\cdot|Q|}$. Hence $L_A(w) \leq \frac{1}{3}$ which contradicts that $L_{\min}(w) = \frac{1}{2}$. Without loss of generality, we can thus assume that there is an a-cycle C in A with positive sum of weights. Then, consider a word w with finite prefix w_C to reach C, followed by a^ω. The value of w in the automaton A is positive, i.e., $L_A(w) > 0$, while $L_{\min}(w) = 0$. Hence the result follows. ■

As a direct consequence of Lemma 1 and the fact that there exist DLAVG for the languages L_a and L_b in that lemma, we get the first part of Theorem 10.

Theorem 10 *The (non)deterministic* LimAvg*- and* Disc*-automata are not closed under* min*.*

The proof of the second part of Theorem 10 uses the quantitative languages L_a^λ and L_b^λ that assign the λ-discounted sum of a's and b's, respectively. The languages L_a and L_b are definable by DDISC with discount factor λ, but $L_m = \min(L_a^\lambda, L_b^\lambda)$ is not definable by a NDISC (when λ is either rational or non-algebraic). To see this, we show that the maximal value of a word in L_m is $\frac{1}{2(1-\lambda)}$ (essentially because $L_a^\lambda(w) + L_b^\lambda(w) = \frac{1}{1-\lambda}$ for all words w) and that it is actually the value of some word, while there is no lasso-word of the form $w_1 \cdot (w_2)^\omega$ (for finite words w_1, w_2) that has the maximal value. Therefore, an NDISC for L_m cannot exist since the maximal value in such an automaton would be obtained by a lasso-word, as pure memoryless strategies exist in games over finite graphs with the objective to maximize the discounted sum of payoffs [25].

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

We give examples of quantitative languages over $\Sigma = \{a, b\}$ that witness Theorem 11. For DSUP and NSUP, consider the language L_1 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 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 12 *The nondeterministic* LimSup*-automata are closed under complement, with cost* $O(m \cdot 2^{n \log n})$*.*

Nondeterministic limit-average and discounted-sum automata are not closed under complement because they are closed under max, but not under min. The fact that DLAVG is not closed under complement can be proved as follows. Consider the language L_a over alphabet $\Sigma = \{a, b\}$ that assigns the long-run average number of a's. Clearly, L_a is definable by a DLAVG. Notice that $L_a(w.a^\omega) = 1$ and $L_a(w.b^\omega) = 0$ for all $w \in \Sigma^*$. Towards 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 $w = b^{n_1}a^{m_1}b^{n_2}a^{m_2}\ldots$ 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 13 *The (non)deterministic* LimAvg-*automata and the nondeterministic* Disc-*automata are not closed under complement.*

Finally, the fact that $\mathsf{Disc}_\lambda(-v) = -\mathsf{Disc}_\lambda(v)$ for all infinite sequences $v = v_0 v_1 \dots$ of rational numbers establishes the next result.

Theorem 14 *The deterministic* Disc-*automata are closed under complement, with cost $O(n)$.*

4.5 Closure under sum for infinite words

All weighted automata are closed under sum, except DLavg and NLavg. The result for NDisc is also presented in [13].

Theorem 15 *The (non)deterministic* Sup-*automata are closed under sum, with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$; the nondeterministic* LimSup-*automata with cost $O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$; the deterministic* LimSup-*automata and (non)deterministic* LimInf-*automata with cost $O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$; and the (non)deterministic* Disc-*automata with cost $O(n_1 \cdot n_2)$. The (non)deterministic* LimAvg-*automata are not closed under sum.*

For discounted-sum, it suffices to take the synchronized product of the two automata and assign to each joint transition the sum of the weights of the corresponding transitions in the corresponding automata. The construction for the other classes of automata are relatively straightforward. It gives an exponential blow-up in the case of DLinf, NLinf and DLsup.

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

References

[1] 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.

[2] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT*, LNCS 2855, pages 117–133. Springer, 2003.

[3] K. Chatterjee. *Stochastic ω-Regular Games.* PhD thesis, University of California, Berkeley, 2007.

[4] K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure for quantitative languages. Technical Report MTC-REPORT-2008-004, http://infoscience.epfl.ch/record/116423, EPFL, 2008.

[5] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *CSL*, LNCS 5213, pages 385–400. Springer, 2008.

[6] 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.

[7] A. Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.

[8] C. Cortes and M. Mohri. Context-free recognition with weighted automata. *Grammars*, 3(2/3):133–150, 2000.

[9] K. Culik II and J. Karhumäki. Finite automata computing real functions. *SIAM J. Comput.*, 23(4):789–814, 1994.

[10] L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *ICALP*, LNCS 2719, pages 1022–1037. Springer, 2003.

[11] M. Droste and P. Gastin. Weighted automata and weighted logics. *Th. C. Sci.*, 380(1-2):69–86, 2007.

[12] M. Droste, W. Kuich, and G. Rahonis. Multi-valued MSO logics over words and trees. *Fundamenta Informaticae*, 84(3-4):305–327, 2008.

[13] M. Droste and D. Kuske. Skew and infinitary formal power series. In *ICALP*, LNCS 2719, pages 426–438. Springer, 2003.

[14] M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting. In *CIAA*, LNCS 4783, pages 73–84. Springer, 2007.

[15] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.

[16] Z. Ésik and W. Kuich. An algebraic generalization of omega-regular languages. In *MFCS*, LNCS 3153, pages 648–659. Springer, 2004.

[17] A. Gurfinkel and M. Chechik. Multi-valued model checking via classical model checking. In *CONCUR*, LNCS 2761, pages 263–277. Springer, 2003.

[18] R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.

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

[20] O. Kupferman and Y. Lustig. Lattice automata. In *VMCAI*, LNCS 4349, pages 199–213. Springer, 2007.

[21] M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[22] M. P. Schützenberger. On the definition of a family of automata. *Inf. and control*, 4(2-3):245–270, 1961.

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

[24] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *LICS*, pages 332–344. IEEE, 1986.

[25] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.