Quantitative Languages*

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

[1] University of California, Santa Cruz
[2] EPFL, Lausanne, Switzerland

Abstract. Quantitative generalizations of classical languages, which assign to each word a real number instead of a boolean value, have applications in modeling resource-constrained computation. We use weighted automata (finite automata with transition weights) to define several natural classes of quantitative languages over finite and infinite words; in particular, the real value of an infinite run is computed as the maximum, limsup, liminf, limit average, or discounted sum of the transition weights. We define the classical decision problems of automata theory (emptiness, universality, language inclusion, and language equivalence) in the quantitative setting and study their computational complexity. As the decidability of language inclusion remains open for some classes of weighted automata, we introduce a notion of quantitative simulation that is decidable and implies language inclusion. We also give a complete characterization of the expressive power of the various classes of weighted automata. In particular, we show that most classes of weighted automata cannot be determinized.

1 Introduction

The automata-theoretic approach to verification is boolean. To check that a system satisfies a specification, we construct a finite automaton A to model the system and a finite (usually nondeterministic) automaton B for the specification. The language $L(A)$ of A contains all behaviors of the system, and $L(B)$ contains all behaviors allowed by the specification. The language of an automaton A can be seen as a boolean function L_A that assigns 1 (or true) to words in $L(A)$, and 0 (or false) to words not in $L(A)$. The verification problem "does the system satisfy the specification?" is then formalized as the language-inclusion problem "is $L(A) \subseteq L(B)$?", or equivalently, "is $L_A(w) \leq L_B(w)$ for all words w?". We present a natural generalization of this framework: a *quantitative language* L is a function that assigns a real-numbered value $L(w)$ to each (finite or infinite) word w. With quantitative languages, systems and specifications can be formalized more accurately. For example, a system may use a varying amount of some resource (e.g., memory consumption, or power consumption) depending on its behavior, and a specification may assign a maximal amount of available resource to each behavior, or fix the long-run average available use of the resource. The quantitative language-inclusion problem "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 requirement.

* Research supported in part by the NSF grants CCR-0132780, CNS-0720884, and CCR-0225610, by the Swiss National Science Foundation, and by the European COMBEST project.

In the boolean automaton setting, the value of a word w in $L(A)$ is the maximal value of a run of A over w (if A is nondeterministic, then there may be many runs of A over w), and the value of a run is a function that depends on the class of automata: for automata over finite words, the value of a run is true if the last state of the run is accepting; for Büchi automata, the value is true if an accepting state is visited infinitely often; etc. To define quantitative languages, we use automata with weights on transitions. We again set the value of a word w as the maximal value of all 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 [2, 3]. Quantitative languages have also been 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 [6]. Alternatively, the discounted sum can be used to specify that failures happening later are less important than those happening soon [8]. It should be noted that LimSup and LimInf automata generalize Büchi and coBüchi automata, respectively. Functions such as limit average (or mean payoff) and discounted sum are classical in game theory [26]; they have been studied extensively in the branching-time context of games played on graphs [12, 7, 3, 14], and it is therefore natural to consider the same functions in the linear-time context of automata and languages.

We attempt a systematic study of quantitative languages defined by weighted automata. The main novelties concern quantitative languages of infinite words, and especially those that have no boolean counterparts (i.e., limit-average and discounted-sum languages). In the first part, we consider generalizations of the boolean decision problems of emptiness, universality, language inclusion, and language equivalence. The quantitative emptiness problem asks, given a weighted automaton A and a rational number ν, whether there exists a word w such that $L_A(w) \geq \nu$. This problem can be reduced to a one-player game with a quantitative objective and is therefore solvable in polynomial time. The quantitative universality problem asks whether $L_A(w) \geq \nu$ for all words w. This problem can be formulated as a two-player game (one player choosing input letters and the other player choosing successor states) with imperfect information (the first player, whose goal is to construct a word w such that $L_A(w) < \nu$, is not allowed to see the state chosen by the second player). The problem is PSPACE-complete for simple functions like Sup, LimSup, and LimInf, but we do not know if it is decidable for limit-average or discounted-sum automata (the corresponding games of imperfect information are not known to be decidable either). The same situation holds for the quantitative language-inclusion and language-equivalence problems, which ask, given two weighted automata A and B, if $L_A(w) \leq L_B(w)$ (resp. $L_A(w) = L_B(w)$) for all words w. Therefore we introduce a notion of quantitative simulation between weighted automata, which generalizes boolean simulation relations, is decidable, and implies language inclusion. Simulation can be seen as a weaker version of the above game, where the first player has perfect information about the state of the game. In particular, we show that quantitative simulation can be decided in NP $\cap$ coNP for limit-average and discounted-sum automata.

In the second part of this paper, we present a complete characterization of the expressive power of the various classes of weighted automata, by comparing the classes of quantitative languages they can define. The complete picture relating the expressive powers of weighted automata is shown in Fig. 4. For instance, the results for LimSup and LimInf are analogous to the special boolean cases of Büchi and coBüchi (nondeterminism is strictly more expressive for LimSup, but not for LimInf). In the limit-average and discounted-sum cases, nondeterministic automata are strictly more expressive than their deterministic counterparts. Also, one of our results shows that nondeterministic limit-average automata are not as expressive as deterministic Büchi automata (and vice versa). It may be noted that deterministic Büchi languages are complete for the second level of the Borel hierarchy [28], and deterministic limit-average languages are complete for the third level [4]; so there is a Wadge reduction [29] from deterministic Büchi languages to deterministic limit-average languages. Our result shows that Wadge reductions are not captured by automata, and in particular, that the Wadge reduction from Büchi to limit-average languages is not regular. We sketch some details of the most interesting proofs; complete proofs are available in [5].

Other researchers have considered generalizations of languages, but as far as we know, nobody has addressed the quantitative language setting presented here. The lattice automata of [21] map finite words to values from a finite lattice. Roughly speaking, the value of a run is the meet (greatest lower bound) of its transition weights, and the value of a word w is the join (least upper bound) of the values of all runs over w. This corresponds to Min and Inf automata in our setting, and for infinite words, the Büchi lattice automata of [21] are analogous to our LimSup automata. However, the other classes of weighted automata (Sum, limit-average, discounted-sum) cannot be defined using operations on finite lattices. The complexity of the emptiness and universality problems for lattice automata is given in [21] (and implies our results for LimSup automata), while their generalization of language inclusion differs from ours. They define the *implication value* $v(A, B)$ of two lattice automata A and B as the meet over all words w of the join of $\neg L_A(w)$ and $L_B(w)$, while we use $+$ instead of join and define $v(A, B)$ as $\min_w(L_B(w) - L_A(w))$.

In classical weighted automata [25, 23] and semiring automata [20], the value of a finite word is defined using the two algebraic operations $+$ and $\cdot$ of a semiring as the sum of the product of the transition weights of the runs over the word. In that case, quantitative languages are called *formal power series*. Over infinite words, weighted automata with discounted sum were first investigated in [11]. Researchers have also considered other quantitative generalizations of languages over finite words [9], over trees [10], and using finite lattices [15]. However, these works do not address the quantitative decision problems, nor do they compare the relative expressive powers of weighted automata over infinite words, as we do here. In [2], a quantitative generalization of languages is defined by discrete functions (the value of a word is an integer) and the decision problems only involve the extremal value of a language, which corresponds to emptiness.

In models that use transition weights as probabilities, such as probabilistic *Rabin automata* [24], one does not consider values of individual infinite runs (which would usually have a value, or measure, of 0), but only measurable sets of infinite runs (where basic open sets are defined as extensions of finite runs). Our quantitative setting is orthogonal to the probabilistic framework: we assign quantitative values (e.g., peak power consumption, average response time, failure rate) to individual infinite behaviors, not probabilities to finite behaviors.

2 Boolean and Quantitative Languages

We recall the classical automata-theoretic description of boolean languages, and introduce an automata-theoretic description of several classes of quantitative languages.

2.1 Boolean Languages

A *boolean language* over a finite alphabet Σ is either a set $L \subseteq \Sigma^*$ of finite words or a set $L \subseteq \Sigma^\omega$ of infinite words. Alternatively, we can view these sets as functions in $[\Sigma^* \to \{0,1\}]$ and $[\Sigma^\omega \to \{0,1\}]$, respectively.

Boolean automata. A *(finite) automaton* is a tuple $A = \langle Q, q_I, \Sigma, \delta \rangle$ where:
- Q is a finite set of states, and $q_I \in Q$ is the initial state;
- Σ is a finite alphabet;
- $\delta \subseteq Q \times \Sigma \times Q$ is a finite set of labeled transitions.

The automaton A is *total* if for all $q \in Q$ and $\sigma \in \Sigma$, there exists $(q, \sigma, q') \in \delta$ for at least one $q' \in Q$. The automaton 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 $0 \leq i < |w|$. When the run r is finite, we denote by $\mathsf{Last}(r)$ the last state in r. When r is infinite, we denote by $\mathsf{Inf}(r)$ the set of states that occur infinitely many times in r. The prefix of length i of an infinite run r is the prefix of r that contains the first i states.

Given a set $F \subseteq Q$ of final (or accepting) states, the *finite-word language* defined by the pair $\langle A, F \rangle$ is $L_A^f = \{w \in \Sigma^* \mid \text{there exists a run } r \text{ of } A \text{ over } w \text{ such that } \mathsf{Last}(r) \in F\}$. The *infinite-word languages* defined by $\langle A, F \rangle$ are as follows: if $\langle A, F \rangle$ is interpreted as a Büchi automaton, then $L_A^b = \{w \in \Sigma^\omega \mid \text{there exists a run } r \text{ of } A \text{ over } w \text{ such that } \mathsf{Inf}(r) \cap F \neq \varnothing\}$, and if $\langle A, F \rangle$ is interpreted as a coBüchi automaton, then $L_A^c = \{w \in \Sigma^\omega \mid \text{there exists a run } r \text{ of } A \text{ over } w \text{ such that } \mathsf{Inf}(r) \subseteq F\}$.

Boolean decision problems. We recall the classical decision problems for automata, namely, emptiness, universality, language inclusion and language equivalence. Given a finite automaton A, the *boolean emptiness problem* asks whether $L_A^f = \varnothing$ (or $L_A^b = \varnothing$, or $L_A^c = \varnothing$), and the *boolean universality problem* asks whether $L_A^f = \Sigma^*$ (or $L_A^b = \Sigma^\omega$, or $L_A^c = \Sigma^\omega$). Given two finite automata A and B, the *boolean language-inclusion problem* asks whether $L_A \subseteq L_B$, and the *boolean language-equivalence problem* asks whether $L_A = L_B$. It is well-known that for both finite- and infinite-word languages, the emptiness problem is solvable in polynomial time, while the universality, inclusion, and equivalence problems are PSPACE-complete [22, 27].

2.2 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:

- $\langle Q, q_I, \Sigma, \delta \rangle$ is a total finite automaton, and
- $\gamma : \delta \to \mathbb{Q}$ is a *weight* function, where $\mathbb{Q}$ is the set of rational numbers.

Given a finite (resp. infinite) run $r = q_0\sigma_1 q_1 \sigma_2 \dots$ of A over a finite (resp. infinite) word $w = \sigma_1\sigma_2 \dots$, let $\gamma(r) = v_0 v_1 \dots$ be the sequence of weights that occur in r, where $v_i = \gamma(q_i, \sigma_{i+1}, q_{i+1})$ for all $0 \le i < |w|$.

Given a *value function* $\mathsf{Val} : \mathbb{Q}^+ \to \mathbb{R}$ (resp. $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$), the Val-automaton A defines the quantitative language L_A such that for all words $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$), we have $L_A(w) = \sup\{\mathsf{Val}(\gamma(r)) \mid r \text{ is a run of } A \text{ over } w\}$.

In sequel we denote by n the number of states and by m the number of transitions of a given automaton. We assume that rational numbers that are given as pairs of integers, encoded in binary. All time bounds we give in this paper assume that the size of the largest integer in the input is a constant p. Without this assumption, most complexity results would involve a factor p^2, as we use only addition, multiplication, and comparison of rational numbers, which are quadratic operations.

Quantitative decision problems. We now present quantitative generalizations of the classical decision problems for automata. Given two quantitative languages L_1 and L_2 over Σ, we write $L_1 \sqsubseteq L_2$ if $L_1(w) \le L_2(w)$ for all words $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$). Given a weighted automaton A and a rational number $\nu \in \mathbb{Q}$, the *quantitative emptiness problem* asks whether there exists a word $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$) such that $L_A(w) \ge \nu$, and the *quantitative universality problem* asks whether $L_A(w) \ge \nu$ for all words $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$). Given two weighted automata A and B, the *quantitative language-inclusion problem* asks whether $L_A \sqsubseteq L_B$, and the *quantitative language-equivalence problem* asks whether $L_A = L_B$, that is, whether $L_A(w) = L_B(w)$ for all $w \in \Sigma^+$ (resp. $w \in \Sigma^\omega$). All results that we present in this paper also hold for the decision problems defined above with inequalities replaced by strict inequalities.

Our purpose is the study of the quantitative decision problems for infinite-word languages and the expressive power of weighted automata that define infinite-word languages. We start with a brief overview of the corresponding results for finite-word languages, most of which follow from classical results in automata theory.

Finite words. For finite words, we consider the value functions Last, Max, and Sum such that for all finite sequences $v = v_1 \dots v_n$ of rational numbers,

$$\mathsf{Last}(v) = v_n, \qquad \mathsf{Max}(v) = \max\{v_i \mid 1 \le i \le n\}, \qquad \mathsf{Sum}(v) = \sum_{i=1}^{n} v_i.$$

Note that Last generalizes the classical boolean acceptance condition for finite words. One could also consider the value function $\mathsf{Min} = \min\{v_i \mid 1 \le i \le n\}$, which roughly corresponds to lattice automata [21].

Theorem 1. *The quantitative emptiness problem can be solved in linear time for* Last *and* Max*-automata, and in quadratic time for* Sum*-automata. The quantitative language-inclusion problem is PSPACE-complete for* Last*- and* Max*-automata.*

The complexity of the quantitative emptiness problem for Last and Max-automata is obtained by reduction to reachability in graphs, and for Sum-automata, by reduction to reachability of a cycle with positive value. The quantitative language-inclusion problem is undecidable for Sum-automata [19]. However, the quantitative language-inclusion problem for deterministic Sum-automata can be solved in polynomial time using a product construction. This naturally raises the question of the power of nondeterminism, which we address through translations between weighted automata.

Expressiveness. A class $\mathcal{C}$ of weighted automata *can be reduced* to a class $\mathcal{C}'$ of weighted automata if for every $A \in \mathcal{C}$ there exists $A' \in \mathcal{C}'$ such that $L_A = L_{A'}$. In particular, a class of weighted automata *can be determinized* if it can be reduced to its deterministic counterpart. All reductions that we present in this paper are constructive: when $\mathcal{C}$ can be reduced to $\mathcal{C}'$, we always construct the automaton $A' \in \mathcal{C}'$ that defines the same quantitative language as the given automaton $A \in \mathcal{C}$. We say that the *cost* of a reduction is $O(f(n,m))$ if for all automata $A \in \mathcal{C}$ with n states and m transitions, the constructed automaton $A' \in \mathcal{C}'$ has at most $O(f(n,m))$ many states. For all reductions we present, the size of the largest transition weight in A' is linear in the size p of the largest weight in A (however, the time needed to compute these weights may be quadratic in p).

It is easy to show that Last- and Max-automata can be determinized using a subset construction, while Sum-automata cannot be determinized. Results about determinizable sub-classes of Sum-automata can be found in [23, 18].

Theorem 2 (see also **[23]**)**.** Last- *and* Max*-automata can be determinized in* $O(2^n)$ *time;* Sum*-automata cannot be determinized. Deterministic* Max*-automata can be reduced to deterministic* Last*-automata in* $O(n \cdot m)$ *time; deterministic* Last*-automata can be reduced to deterministic* Sum*-automata in* $O(n \cdot m)$ *time. Deterministic* Sum*-automata cannot be reduced to* Last*-automata; deterministic* Last*-automata cannot be reduced to* Max*-automata.*

Infinite words. For infinite words, we consider the following classical value functions from $\mathbb{Q}^\omega$ to $\mathbb{R}$. Given an infinite sequence $v = v_0 v_1 \dots$ of rational numbers, define

- $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$;
- $\mathsf{LimSup}(v) = \limsup\limits_{n\to\infty} v_n = \lim\limits_{n\to\infty} \sup\{v_i \mid i \geq n\}$;
- $\mathsf{LimInf}(v) = \liminf\limits_{n\to\infty} v_n = \lim\limits_{n\to\infty} \inf\{v_i \mid i \geq n\}$;
- $\mathsf{LimAvg}(v) = \liminf\limits_{n\to\infty} \dfrac{1}{n} \cdot \sum\limits_{i=0}^{n-1} v_i$;
- given a discount factor $0 < \lambda < 1$, $\mathsf{Disc}_\lambda(v) = \sum\limits_{i=0}^{\infty} \lambda^i \cdot v_i$.

For decision problems, we always assume that the discount factor λ is a rational number. 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 $\mathsf{Inf} = \inf\{v_n \mid n \geq 0\}$ and obtain results analogous to the Sup value function.

Notation. Classes of automata are sometimes denoted by acronyms of the form xyW where x is either N(ondeterministic) or D(eterministic), and y is one of the following: B(üchi), C(oBüchi), SUP, LS (LimSup), LI (LimInf), LA (LimAvg), or DI (Disc).

3 The Complexity of Quantitative Decision Problems

We study the complexity of the quantitative decision problems for weighted automata over infinite words.

Emptiness. The quantitative emptiness problem can be solved by reduction to the problem of finding the maximal value of an infinite path in a graph. This is decidable because pure memoryless strategies for resolving nondeterminism exist for all quantitative objectives that we consider [13, 17, 1].

Theorem 3. *The quantitative emptiness problem is solvable in $O(m+n)$ time for* Sup-, LimSup-, *and* LimInf-*automata; in $O(n \cdot m)$ time for* LimAvg-*automata; and in $O(n^2 \cdot m)$ time for* Disc-*automata.*

Language inclusion. The following theorem relies on the analogous result for finite automata.

Theorem 4. *The quantitative language-inclusion problem is PSPACE-complete for* Sup-, LimSup-, *and* LimInf-*automata.*

We do not know if the quantitative language-inclusion problem is decidable for LimAvg- or Disc-automata. The special cases of deterministic automata are easy, using a product construction.

Theorem 5. *The quantitative language-inclusion problems $L_A \sqsubseteq L_B$ for* LimAvg- *and* Disc-*automata are decidable in polynomial time when B is deterministic.*

When B is not deterministic, we make the following observation. There exist two LimAvg-automata A and B such that (i) $L_A \not\sqsubseteq L_B$ and (ii) there exist no finite words w_1 and w_2 such that $L_A(w) > L_B(w)$ for $w = w_1 \cdot w_2^\omega$ (the word w is called a *lasso-word*). Consider the two LimAvg-automata A and B shown in Fig. 1, where B is nondeterministic. For all words $w \in \Sigma^\omega$, we have $L_A(w) = 1$. For a lasso-word of the form $w = w_1 \cdot w_2^\omega$, if in w_2 there are more b's than a's, then B chooses q_3' from q_1', and else chooses q_2' from q_1'. Hence for all lasso-words $w = w_1 \cdot w_2^\omega$, we have $L_B(w) \geq 1$. However $L_A \not\sqsubseteq L_B$. Consider the word w generated inductively such that w_0 is the empty word, and w_{i+1} is generated from w_i as follows: (i) first generate a long enough sequence w_{i+1}' of a's after w_i such that the average number of b's in $w_i \cdot w_{i+1}'$ falls below $\frac{1}{3}$; (ii) then generate a long enough sequence w_{i+1}'' of

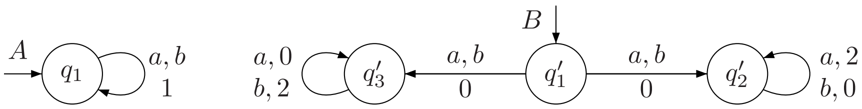

Fig. 1. Two limit-average automata A and B (nondeterministic) such that $L_A \not\sqsubseteq L_B$, but there is no word of the form $w = w_1 \cdot w_2^\omega$ with $L_A(w) > L_B(w)$.

b's such that the average number of a's in $w_i \cdot w'_{i+1} \cdot w''_{i+1}$ falls below $\frac{1}{3}$; and (iii) let $w_{i+1} = w_i \cdot w'_{i+1} \cdot w''_{i+1}$. The infinite word w is the limit of this sequence. For the word w, we have $L_B(w) = 2 \cdot \frac{1}{3} = \frac{2}{3} < 1$, and thus $L_A \not\sqsubseteq L_B$. This observation is in contrast to the case of boolean language inclusion for, e.g., parity automata, where non-inclusion is always witnessed by a lasso-word.

For discounted-sum automata A with weight function γ_1 and B with weight function γ_2, assume that we have a finite word $w \in \Sigma^*$ such that for some run r_1 of A over w and for all runs r_2 of B over w, we have

$$\gamma_1(r_1) + v \cdot \frac{\lambda^{|w|}}{1-\lambda} > \gamma_2(r_2) + V \cdot \frac{\lambda^{|w|}}{1-\lambda}$$

where v (resp. V) is the minimal (resp. maximal) weight in (the union of) A and B. Then, we immediately have $L_A \not\sqsubseteq L_B$, as $L_A(w \cdot w') > L_B(w \cdot w')$ for all words $w' \in \Sigma^\omega$. We say that w is a *finite witness* of $L_A \not\sqsubseteq L_B$. We claim that there always exists a finite witness of $L_A \not\sqsubseteq L_B$. To see this, consider an infinite word w^∞ such that $L_A(w^\infty) = \eta_1$, $L_B(w^\infty) = \eta_2$, and $\eta_1 > \eta_2$. Let r_1 be an (infinite) run of A over w^∞ whose value is η_1. For $i > 0$, consider the prefix of w^∞ of length i. Then, for all runs r_2 of B over w^∞, we have

$$\gamma_1(r_1^i) + V \cdot \frac{\lambda^i}{1-\lambda} \geq \eta_1 \quad \text{and} \quad \gamma_2(r_2^i) + v \cdot \frac{\lambda^i}{1-\lambda} \leq \eta_2$$

where r_1^i and r_2^i are the prefixes of length i of r_1 and r_2, respectively. Then, a prefix of length i of w^∞ is a finite witness of $L_A \not\sqsubseteq L_B$ if

$$\eta_1 - (V - v) \cdot \frac{\lambda^i}{1-\lambda} > \eta_2 + (V - v) \cdot \frac{\lambda^i}{1-\lambda}$$

which must hold for sufficiently large values of i. Thus, we have the following theorem.

Theorem 6. *The quantitative language-inclusion problem for* Disc*-automata is co-r.e.*

Universality and language equivalence. All of the above results about language inclusion hold for quantitative universality and language equivalence also.

4 Quantitative Simulation

As the decidability of the quantitative language-inclusion problems for limit-average and discounted-sum automata remain open, we introduce a notion of *quantitative simulation* as a decidable approximation of language inclusion for weighted automata. The quantitative language-inclusion problem can be viewed as a game of imperfect information, and we view the quantitative simulation problem as exactly the same game, but with perfect information. For quantitative objectives, perfect-information games can be solved much more efficiently than imperfect-information games, and in some cases the solution of imperfect-information games with quantitative objectives is not known. For example, perfect-information games with limit-average and discounted-sum objectives can be decided in NP $\cap$ coNP, whereas the solution for such imperfect-information games is not known. Second, quantitative simulation implies quantitative language inclusion, because it is easier to win a game when information is not hidden. Hence, as in the case of finite automata, simulation can be used as a conservative and efficient approximation for language inclusion.

Language-inclusion game. Let A and B be two weighted automata with weight function γ_1 and γ_2, respectively, for which we want to check if $L_A \sqsubseteq L_B$. The language-inclusion game is played by a *challenger* and a *simulator*, for infinitely many rounds. The goal of the simulator is to prove that $L_A \sqsubseteq L_B$, while the challenger has the opposite objective. The position of the game in the initial round is $\langle q_I^1, q_I^2 \rangle$ where q_I^1 and q_I^2 are the initial states of A and B, respectively. In each round, if the current position is $\langle q_1, q_2 \rangle$, first the challenger chooses a letter $\sigma \in \Sigma$ and a state q_1' such that $(q_1, \sigma, q_1') \in \delta_1$, and then the simulator chooses a state q_2' such that $(q_2, \sigma, q_2') \in \delta_2$. The position of the game in the next round is $\langle q_1', q_2' \rangle$. The outcome of the game is a pair (r_1, r_2) of runs of A and B, respectively, over the same infinite word. The simulator wins the game if $\mathsf{Val}(\gamma_2(r_2)) \geq \mathsf{Val}(\gamma_1(r_1))$. To make this game equivalent to the language-inclusion problem, we require that the challenger cannot observe the state of B in the position of the game.

Simulation game. The simulation game is the language-inclusion game without the restriction on the vision of the challenger, that is, the challenger is allowed to observe the full position of the game. Formally, given $A = \langle Q_1, q_I^1, \Sigma, \delta_1, \gamma_1 \rangle$ and $B = \langle Q_2, q_I^2, \Sigma, \delta_2, \gamma_2 \rangle$, a *strategy* τ for the challenger is a function from $(Q_1 \times Q_2)^+$ to $\Sigma \times Q_1$ such that for all $\pi \in (Q_1 \times Q_2)^+$, if $\tau(\pi) = (\sigma, q)$, then $(\mathsf{Last}(\pi_{|Q_1}), \sigma, q) \in \delta_1$, where $\pi_{|Q_1}$ is the projection of π on Q_1^+. A strategy τ for the challenger is *blind* if $\tau(\pi) = \tau(\pi')$ for all sequences $\pi, \pi' \in (Q_1 \times Q_2)^*$ such that $\pi_{|Q_1} = \pi'_{|Q_1}$. The set of *outcomes* of a challenger strategy τ is the set of pairs (r_1, r_2) of runs such that if $r_1 = q_0 \sigma_1 q_1 \sigma_2 \dots$ and $r_2 = q_0' \sigma_1 q_1' \sigma_2 \dots$, then $q_0 = q_I^1$, $q_0' = q_I^2$, and for all $i \geq 0$, we have $(\sigma_{i+1}, q_{i+1}) = \tau((q_0, q_0') \dots (q_i, q_i'))$ and $(q_i', \sigma_{i+1}, q_{i+1}') \in \delta_2$. A strategy τ for the challenger is *winning* if $\mathsf{Val}(\gamma_1(r_1)) > \mathsf{Val}(\gamma_2(r_2))$ for all outcomes (r_1, r_2) of τ.

Theorem 7. *For all value functions and weighted automata A and B, we have $L_A \sqsubseteq L_B$ iff there is no blind winning strategy for the challenger in the language-inclusion game for A and B.*

Given two weighted automata A and B, there is a *quantitative simulation* of A by B if there exists no (not necessarily blind) winning strategy for the challenger in the simulation game for A and B. We note that for the special cases of Büchi and coBüchi automata, quantitative simulation coincides with *fair simulation* [16].

Corollary 1. *For all value functions and weighted automata A and B, if there is a quantitative simulation of A by B, then $L_A \sqsubseteq L_B$.*

Given two weighted automata A and B, the *quantitative simulation problem* asks if there is a quantitative simulation of A by B.

Theorem 8. *The quantitative simulation problem is in NP $\cap$ coNP for* LimSup-*,* LimInf-*,* LimAvg-*, and* Disc-*automata.*

The proof of Theorem 8 is obtained as follows. The quantitative simulation problems for LimSup- and LimInf-automata is reduced to perfect-information parity games; the quantitative simulation problem for LimAvg-automata is reduced to perfect-information limit-average games; and the quantitative simulation problem for Disc-automata is reduced to perfect-information discounted-sum games. All reductions are polynomial time, and the resulting games can all be solved in NP $\cap$ coNP.

5 The Expressive Power of Weighted Automata

We study the expressiveness of different classes weighted automata over infinite words by comparing the quantitative languages they can define. For this purpose, we show the existence and non-existence of translations between classes of finite and weighted automata. We will use the following definition. A class $\mathcal{C}$ of finite automata *can be weakly reduced* to a class $\mathcal{C}'$ of weighted automata if for every $A \in \mathcal{C}$ there exists an $A' \in \mathcal{C}'$ such that $\inf_{w \in L_A} L_{A'}(w) > \sup_{w \notin L_A} L_{A'}(w)$.

5.1 Positive Reducibility Results

We start with the positive results about the existence of reductions between various classes of weighted automata, most of which can be obtained by generalizing corresponding results for finite automata. Our results also hold if we allow transition weights to be irrational numbers.

First, it is clear that Büchi and coBüchi automata can be reduced to LimSup- and LimInf-automata, respectively. In addition, we have the following results.

Theorem 9. Sup-*automata can be determinized in $O(2^n)$ time;* LimInf-*automata can be determinized in $O(m \cdot 2^n)$ time. Deterministic* Sup-*automata can be reduced to deterministic* LimInf-*, to deterministic* LimSup-*, and to deterministic* LimAvg-*automata, all in $O(n \cdot m)$ time.* LimInf-*automata can be reduced to* LimSup- *and to* LimAvg-*automata, both in $O(n \cdot m)$ time.*

The reduction from LimInf- to LimSup-automata (resp. to LimAvg-automata) essentially consists of guessing a position i and a transition weight v such that only weights greater than v are seen after position i. Once the guess is made, all transitions have weight v.

All reducibility relationships are summarized in Fig. 4, where the notation ${}^{\mathrm{D}}_{\mathrm{N}}$yW is used to denote the classes of automata that are determinizable.

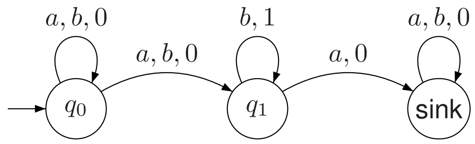

Fig. 2. A nondeterministic limit-average automaton.

5.2 Negative Reducibility Results

We show that all other reducibility relationships do not hold. The most important results in this section show that (i) deterministic coBüchi automata cannot be reduced to deterministic LimAvg-automata, deterministic Büchi automata cannot be reduced to LimAvg-automata, and (ii) neither LimAvg- nor Disc-automata can be determinized. Over the alphabet $\hat{\Sigma} = \{a, b\}$, we use in the sequel the boolean languages L_F, which contains all infinite words with finitely many a's, and L_I, which contains all infinite words with infinitely many a's.

The classical proof that deterministic coBüchi automata cannot reduced to deterministic Büchi automata can be adapted to show the following theorem.

Theorem 10. *Deterministic coBüchi automata cannot be reduced to deterministic* LimSup-*automata.*

Since deterministic LimAvg- and deterministic Disc-automata can define quantitative languages whose range is infinite, while LimSup-automata cannot, we obtain the following result.

Theorem 11. *Deterministic* LimAvg-*automata and deterministic* Disc-*automata cannot be reduced to* LimSup-*automata.*

The next theorem shows that nondeterministic LimAvg-automata are strictly more expressive than their deterministic counterpart. Theorem 13 will show that the expressive powers of LimAvg- and LimSup-automata are incomparable.

Theorem 12. *Deterministic coBüchi automata cannot be weakly reduced to deterministic* LimAvg-*automata, and therefore they cannot be reduced to deterministic* LimAvg-*automata.* LimAvg-*automata cannot be determinized.*

Proof. Consider the language L_F of finitely many a's, which is obviously accepted by a DCW. It is also easy to see that the NLAW shown in Fig. 2 defines L_F. We show that L_F cannot be defined by any DLAW to prove the desired claims. By contradiction, assume that A is a DLAW with set of states Q and the initial state q_I that defines L_F. We assume without loss of generality that every state $q \in Q$ is reachable from q_I by a finite word w_q.

Let $\alpha = \inf_{w \in L_F} L_A(w)$. We claim that all b-cycles (a b-cycle is a cycle in A that can be executed with only b's) must be such that the average of the weights on the cycle is at least α. Indeed, if there is a b-cycle C in A with average weights less than α, then consider a state $q \in C$ and the word $w = w_q \cdot b^\omega$. We have $L_A(w) < \alpha$. Since $w = w_q \cdot b^\omega \in L_F$, this contradicts that $\alpha = \inf_{w \in L_F} L_A(w)$.

We now show that for all $\epsilon > 0$, there exists $w' \notin L_F$ such that $L_A(w') \geq \alpha - \epsilon$. Fix $\epsilon > 0$. Let $\beta = \max_{q,q' \in Q, \sigma \in \{a,b\}} |\gamma(q,\sigma,q')|$. Let $j = \lceil \frac{6 \cdot |Q| \cdot \beta}{\epsilon} \rceil$, and consider the word $w_\epsilon = (b^j \cdot a)^\omega$. A lower bound on the average of the weights in the unique run of A over $(b^j \cdot a)$ is as follows: it can have a prefix of length at most $|Q|$ whose sum of weights is at least $-|Q| \cdot \beta$, then it goes through b-cycles for at least $j - 2 \cdot |Q|$ steps with sum of weights at least $(j - 2 \cdot |Q|) \cdot \alpha$ (since all b-cycles have average weights at least α), then again a prefix of length at most $|Q|$ without completing the cycle (with sum of weights at least $-|Q| \cdot \beta$), and then weight for a is at least $-\beta$. Hence the average is at least

$$\frac{(j - 2 \cdot |Q|) \cdot \alpha - 2 \cdot |Q| \cdot \beta - \beta}{j+1} \geq \alpha - \frac{6 \cdot |Q| \cdot \beta}{j} \geq \alpha - \epsilon;$$

we used above that $|\alpha| \leq \beta$, and by choice of j we have $\frac{6 \cdot |Q| \cdot \beta}{j} \leq \epsilon$. Hence we have $L_A(w_\epsilon) \geq \alpha - \epsilon$. Since $\epsilon > 0$ is arbitrary, and $w_\epsilon \notin L_F$, we have $\sup_{w \notin L_F} L_A(w) \geq \alpha = \inf_{w \in L_F} L_A(w)$. This establishes a contradiction, and thus A cannot exist. The desired result follows. ■

Theorem 13. *Deterministic Büchi automata cannot be weakly reduced to* LimAvg-*automata, and therefore they cannot be reduced to* LimAvg-*automata.*

Proof. We consider the language L_I of infinitely many a's, which is obviously accepted by a DBW.

By contradiction, assume that A is a NLAW with set of states Q and initial state q_I that defines L_I. We assume without loss of generality that every state $q \in Q$ is reachable from q_I by a finite word w_q.

Let $\alpha = \sup_{w \notin L_I} L_A(w)$, and $\beta = \max_{q,q' \in Q, \sigma \in \{a,b\}} |\gamma(q,\sigma,q')|$. We claim that all b-cycles C in A must have average weights at most α; otherwise, consider a state $q \in C$ and the word $w = w_q \cdot b^\omega$, we have $L_A(w) > \alpha$ which contradicts that $\alpha = \sup_{w \notin L_I} L_A(w)$.

We now show that for all $\epsilon > 0$, there exists $w \in L_I$ such that $L_A(w') \leq \alpha + \epsilon$. Fix $\epsilon > 0$. Let $j = \lceil \frac{3 \cdot |Q| \cdot \beta}{\epsilon} \rceil$, and consider the word $w_\epsilon = (b^j \cdot a)^\omega$. An upper bound on the average of the weights in any run of A over $(b^j \cdot a)$ is as follows: it can have a prefix of length at most $|Q|$ with the sum of weights at most $|Q| \cdot \beta$, then it follows (possibly nested) b-cycles[3] for at most j steps with sum of weights at most $j \cdot \alpha$ (since all b-cycles have average weights at most α), then again a prefix of length at most $|Q|$ without completing a cycle (with sum of weights at most $|Q| \cdot \beta$), and then weight for a is at most β. So, for any run of A over $w_\epsilon = (b^j \cdot a)^\omega$, the average weight is at most

$$\frac{j \cdot \alpha + 2 \cdot |Q| \cdot \beta + \beta}{j+1} \leq \alpha + \frac{3 \cdot |Q| \cdot \beta}{j} \leq \alpha + \epsilon$$

[3] Since A is nondeterministic, a run over b^j may have nested cycles. We can decompose the run by repeatedly eliminating the innermost cycles.

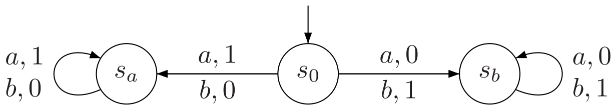

Fig. 3. The nondeterministic discounted-sum automaton N.

Hence we have $L_A(w_\epsilon) \leq \alpha + \epsilon$. Since $\epsilon > 0$ is arbitrary, and $w_\epsilon \in L_I$, we have $\inf_{w \in L_I} L_A(w) \leq \alpha = \sup_{w \notin L_I} L_A(w)$. The desired result follows. ■

None of the weighted automata we consider can be reduced to Disc-automata (Theorem 14), and Disc-automata cannot be reduced to any of the other classes of weighted automata (Theorem 15, and also Theorem 11).

Theorem 14. *Deterministic coBüchi automata and deterministic Büchi automata cannot be weakly reduced to* Disc*-automata, and therefore they cannot be reduced to* Disc*-automata. Also deterministic* Sup*-automata cannot be reduced to* Disc*-automata.*

The proofs of Theorem 14 and 15 are based on the property that the value assigned by a Disc-automaton to an infinite word depends essentially on a finite prefix, in the sense that the values of two words become arbitrarily close when they have sufficiently long common prefixes. In other words, the quantitative language defined by a discounted-sum automaton is a continuous function in the Cantor topology. In contrast, for the other classes of weighted automata, the value of an infinite word depends essentially on its tail.

Theorem 15. *Deterministic* Disc*-automata cannot be reduced to* LimAvg*-automata.*

The next result shows that discounted-sum automata cannot be determinized. Consider the nondeterministic discounted-sum automaton N over the alphabet $\hat{\Sigma} = \{a, b\}$ shown in Fig. 3. The automaton N computes the maximum of the discounted sum of a's and b's. Formally, given a (finite or infinite) word $w = w_0 w_1 \ldots \in \hat{\Sigma}^* \cup \hat{\Sigma}^\omega$, let

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

be the λ-discounted sum of all a's (resp. b's) in w. Then $L_N(w) = \max\{v_a(w), v_b(w)\}$ for all infinite words $w \in \hat{\Sigma}^\omega$. We show that N cannot be determinized for some discount factors λ. The proof uses a sequence of intermediate lemmas.

For $\sigma \in \hat{\Sigma}$, let $\overline{\sigma} = a$ if $\sigma = b$, and $\overline{\sigma} = b$ if $\sigma = a$. We say that an infinite word $w \in \hat{\Sigma}^\omega$ *prefers* $\sigma \in \hat{\Sigma}$ if $v_\sigma(w) > v_{\overline{\sigma}}(w)$.

Lemma 1. *For all* $0 < \lambda < 1$*, all* $w \in \hat{\Sigma}^*$*, and all* $\sigma \in \hat{\Sigma}$*, there exists* $w' \in \hat{\Sigma}^\omega$ *such that* $w \cdot w'$ *prefers* σ *iff* $v_\sigma(w \cdot \sigma^\omega) > v_{\overline{\sigma}}(w \cdot \sigma^\omega)$.

We say that a finite word $w \in \hat{\Sigma}^*$ is *ambiguous* if there exist two infinite words $w'_a, w'_b \in \hat{\Sigma}^\omega$ such that $w \cdot w'_a$ prefers a and $w \cdot w'_b$ prefers b.

Lemma 2. *For all* $0 < \lambda < 1$ *and* $w \in \hat{\Sigma}^*$, *the word* w *is ambiguous iff* $|v_a(w) - v_b(w)| < \frac{\lambda^{|w|}}{1-\lambda}$.

Intuitively, ambiguous words are problematic for a deterministic automaton because it cannot decide which one of the two functions v_a and v_b to choose.

Lemma 3. *For all* $\frac{1}{2} < \lambda < 1$, *there exists an infinite word* $\hat{w} \in \hat{\Sigma}^\omega$ *such that every finite prefix of* $\hat{w}$ *is ambiguous.*

Proof. We construct $\hat{w} = w_1 w_2 \dots$ inductively as follows. First, let $w_1 = a$ which is an ambiguous word for all $\lambda > \frac{1}{2}$ (Lemma 2). Assume that $w_1 \dots w_i$ is ambiguous for all $1 \leq i \leq k$, that is $|x_i| < \frac{\lambda^i}{1-\lambda}$ where $x_i = v_a(w_1 \dots w_i) - v_b(w_1 \dots w_i)$ (Lemma 2). We take $w_{k+1} = a$ if $x_k < 0$, and $w_{k+1} = b$ otherwise. Let us show that $|x_{k+1}| < \frac{\lambda^{k+1}}{1-\lambda}$. We have $|x_{k+1}| = \big||x_k| - \lambda^k\big|$, and thus we need to show that $|x_k| - \lambda^k < \frac{\lambda^{k+1}}{1-\lambda}$ and $-|x_k| + \lambda^k < \frac{\lambda^{k+1}}{1-\lambda}$ knowing that $|x_k| < \frac{\lambda^k}{1-\lambda}$. It suffices to show that

$$\frac{\lambda^k}{1-\lambda} \leq \lambda^k + \frac{\lambda^{k+1}}{1-\lambda} \text{ and } \lambda^k - \frac{\lambda^{k+1}}{1-\lambda} < 0.$$

In other words, it suffices that $1 \leq 1 - \lambda + \lambda$ and $1 - \lambda - \lambda < 0$, which is true for all $\lambda > \frac{1}{2}$. ■

The word $\hat{w}$ constructed in Lemma 3 could be harmless for a deterministic automaton if some kind of periodicity is encountered in $\hat{w}$. We make this notion formal by defining $\mathit{diff}(w) = \frac{v_a(w) - v_b(w)}{\lambda^{|w|}}$ for all finite words $w \in \hat{\Sigma}^*$. It can be shown that if the set $R_\lambda = \{\mathit{diff}(w) \mid w \in \Sigma^*\} \cap (\frac{-1}{1-\lambda}, \frac{1}{1-\lambda})$ is finite, then the automaton N can be determinized [5], where (a, b) denotes the open interval between two reals a and b with $a < b$. Lemma 4 shows that this is also a necessary condition.

Lemma 4. *For all* $0 < \lambda < 1$, *if the set* R_λ *is infinite, then there exists no deterministic* Disc-*automaton* D *such that* $L_D = L_N$.

Proof. By contradiction, assume that R_λ is infinite and there exists a DDIW D such that $L_D = L_N$. For all $w \in \hat{\Sigma}^*$, let $\mathsf{Post}(w)$ be the (unique) state reached in D after reading w. We show that for all words $w_1, w_2 \in \hat{\Sigma}^*$ such that $\mathit{diff}(w_1), \mathit{diff}(w_2) \in R_\lambda$, if $\mathit{diff}(w_1) \neq \mathit{diff}(w_2)$, then $\mathsf{Post}(w_1) \neq \mathsf{Post}(w_2)$. Therefore D cannot have finitely many states.

We show this by contradiction. Assume that $\mathsf{Post}(w_1) = \mathsf{Post}(w_2)$. Then w_1 and w_2 are ambiguous by Lemma 2 since $\mathit{diff}(w_1), \mathit{diff}(w_2) \in R_\lambda$. For $i = 1, 2$, we thus have by Lemma 1

$$L_N(w_i \cdot a^\omega) = v_a(w_i) + \frac{\lambda^{|w_i|}}{1-\lambda} \quad \text{and} \quad L_N(w_i \cdot b^\omega) = v_b(w_i) + \frac{\lambda^{|w_i|}}{1-\lambda}.$$

On the other hand, since $\mathsf{Post}(w_1) = \mathsf{Post}(w_2)$, there exist $v_1, v_2, K_a, K_b \in \mathbb{R}$ such that for $i = 1, 2$,

$$L_D(w_i \cdot a^\omega) = v_i + \lambda^{|w_i|} \cdot K_a \quad \text{and} \quad L_D(w_i \cdot b^\omega) = v_i + \lambda^{|w_i|} \cdot K_b.$$

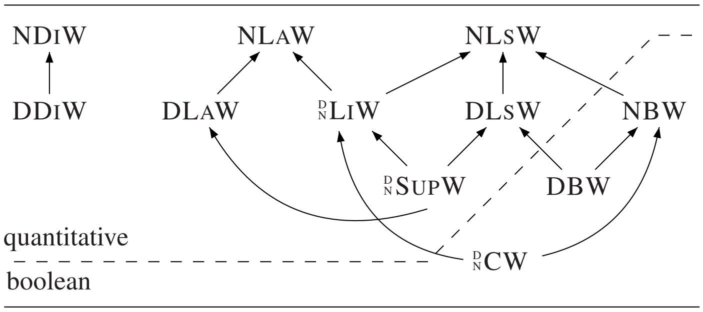

Fig. 4. Reducibility relations: a class $\mathcal{C}$ of automata can be reduced to $\mathcal{C}'$ iff $\mathcal{C} \to^* \mathcal{C}'$.

Since $L_D = L_N$, this entails that $L_D(w_i{\cdot}a^\omega) - L_D(w_i{\cdot}b^\omega) = L_N(w_i{\cdot}a^\omega) - L_N(w_i{\cdot}b^\omega)$, and therefore

$$\frac{v_a(w_1) - v_b(w_1)}{\lambda^{|w_1|}} = K_a - K_b = \frac{v_a(w_2) - v_b(w_2)}{\lambda^{|w_2|}}$$

which yields a contradiction. ■

We are now ready to prove the following theorem.

Theorem 16. Disc-*automata cannot be determinized.*

Proof. Let λ^* be a non-algebraic number in the open interval $(\frac{1}{2}, 1)$. Then, we show that the set R_{λ^*} is infinite, which establishes the theorem by Lemma 4.

By Lemma 2 and Lemma 3, there exist infinitely many finite words $w \in \hat{\Sigma}^*$ such that $\mathit{diff}(w) \in R_{\lambda^*}$. Since λ^* is not algebraic, the polynomial equation $\mathit{diff}(w_1) = \mathit{diff}(w_2)$ cannot hold for $w_1 \neq w_2$. Therefore, R_{λ^*} is infinite. ■

By a careful analysis of the shape of the family of polynomial equations in the above proof, we can show that the automaton N cannot be determinized for any rational value of λ greater than $\frac{1}{2}$ [5].

References

1. D. Andersson. An improved algorithm for discounted payoff games. In *ESSLLI Student Session*, pages 91–98, 2006.
2. 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.
3. A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT*, LNCS 2855, pages 117–133. Springer, 2003.
4. K. Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388:181–198, 2007.
5. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. Technical Report MTC-REPORT-2008-003, http://infoscience.epfl.ch/record/115228, EPFL, 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*. ACM, 2008.
7. Anne Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.
8. L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *ICALP*, LNCS 2719, pages 1022–1037. Springer, 2003.
9. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.
10. M. Droste, W. Kuich, and G. Rahonis. Multi-valued MSO logics over words and trees. *Fundamenta Informaticae*, 84:305–327, 2008.
11. M. Droste and D. Kuske. Skew and infinitary formal power series. In *ICALP*, LNCS 2719, pages 426–438. Springer, 2003.
12. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.
13. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
14. H. Gimbert. *Jeux positionnels*. PhD thesis, Université Paris 7, 2006.
15. A. Gurfinkel and M. Chechik. Multi-valued model checking via classical model checking. In *CONCUR*, LNCS 2761, pages 263–277. Springer, 2003.
16. T.A. Henzinger, O. Kupferman, and S.K. Rajamani. Fair simulation. In *CONCUR*, LNCS 1243, pages 273–287. Springer, 1997.
17. R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.
18. D. Kirsten and I. Mäurer. On the determinization of weighted automata. *Journal of Automata, Languages and Combinatorics*, 10:287–312, 2005.
19. D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. In *ICALP*, LNCS 623, pages 101–112. Springer, 1992.
20. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1986.
21. O. Kupferman and Y. Lustig. Lattice automata. In *VMCAI*, LNCS 4349, pages 199–213. Springer, 2007.
22. A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *FOCS*, pages 125–129. IEEE, 1972.
23. M. Mohri. Finite-state transducers in language and speech processing. *Comp. Linguistics*, 23(2):269–311, 1997.
24. A. Paz. *Introduction to probabilistic automata*. Computer Science and Applied Mathematics. Academic Press, New York, 1971.
25. M. P. Schützenberger. On the definition of a family of automata. *Information and control*, 4:245–270, 1961.
26. L. S. Shapley. Stochastic games. In *Proc. of the National Academy of Science USA*, volume 39, pages 1095–1100, 1953.
27. A. P. Sistla, M. Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.
28. W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
29. W.W. Wadge. *Reducibility and Determinateness of Baire Spaces*. PhD thesis, UC Berkeley, 1984.