Alternating Weighted Automata*

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

[1] Institute of Science and Technology, Austria
[2] Université Libre de Bruxelles (ULB), Belgium
[3] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract. Weighted automata are finite automata with numerical weights on transitions. Nondeterministic weighted automata define quantitative languages L that assign to each word w a real number $L(w)$ computed as the maximal value of all runs over w, and the value of a run r is a function of the sequence of weights that appear along r. There are several natural functions to consider such as Sup, LimSup, LimInf, limit average, and discounted sum of transition weights.

We introduce alternating weighted automata in which the transitions of the runs are chosen by two players in a turn-based fashion. Each word is assigned the maximal value of a run that the first player can enforce regardless of the choices made by the second player. We survey the results about closure properties, expressiveness, and decision problems for nondeterministic weighted automata, and we extend these results to alternating weighted automata.

For quantitative languages L_1 and L_2, we consider the pointwise operations $max(L_1, L_2)$, $min(L_1, L_2)$, $1 - L_1$, and the sum $L_1 + L_2$. We establish the closure properties of all classes of alternating weighted automata with respect to these four operations.

We next compare the expressive power of the various classes of alternating and nondeterministic weighted automata over infinite words. In particular, for limit average and discounted sum, we show that alternation brings more expressive power than nondeterminism.

Finally, we present decidability results and open questions for the quantitative extension of the classical decision problems in automata theory: emptiness, universality, language inclusion, and language equivalence.

1 Introduction

A classical *language* is a set of infinite words over a finite alphabet Σ, or equivalently a function $L : \Sigma^\omega \to \{0, 1\}$. Either a word w belongs to the language

* 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 Combest, Quasimodo, and Gasics projects, 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.

and then $L(w) = 1$, or w does not belong to the language and then $L(w) = 0$. Languages are natural models of computation for reactive programs: each execution of a program is an infinite sequence of events (or a word), and the set of all executions (or the language) defines the possible behaviors of the program. Finite automata can be used to define languages, and questions about the correctness of programs can be reduced to decision problems on automata, such as emptiness and language inclusion [14, 7].

A *quantitative language* is a function $L : \Sigma^\omega \to \mathbb{R}$, generalizing the classical languages (called *boolean languages* in this paper). A natural interpretation of the value $L(w)$ of a word w is the cost incurred by a program to produce the execution w, for example in terms of energy or memory consumption. Values can also be used to quantify the reliability or the quality of executions, rather than simply classifying them as good or bad. Hence, quantitative languages provide a more accurate model of program computation.

To define quantitative languages, we use *weighted automata*, i.e., finite automata with numerical weights on transitions. To compute the value of a word in a weighted automaton, we need to fix a *mode of branching* and a *value function*. In this paper, we consider four modes of branching (alternating, universal, nondeterministic, and deterministic) and five value functions (Sup, LimSup, LimInf, limit average, and discounted sum). In an *alternating* weighted automaton, the value of an input word is determined by two players playing in rounds, starting in the initial state of the automaton. If the current state is q and the next input letter is σ, the first player (called the maximizer) chooses one transition (q, σ, s) where s is a set of states in which the second player (called the minimizer) then chooses a state q'. The next round starts in q' and the game proceeds for infinitely many rounds, constructing an infinite weighted path whose value is computed as the value function of its weights. The value of the input word is the maximal value of such a path that the maximizer can enforce no matter what choices the minimizer makes. When the choices available to the maximizer are trivial (i.e., in every state q and for every input letter σ, there is exactly one transition (q, σ, s)), the weighted automaton is *universal*, and when the choices available to the minimizer are trivial (i.e., for every transition (q, σ, s), the set s is a singleton), the weighted automaton is *nondeterministic*. A *deterministic* weighted automaton is both universal and nondeterministic. Note that for weighted automata with weights in $\{0, 1\}$, these definitions coincide with the classical finite automata theory [2, 10], and in particular the LimSup- and LimInf-automata can then be viewed as Büchi and coBüchi automata respectively.

We survey the results about closure properties, expressiveness, and decision problems for nondeterministic weighted automata [3, 4], and we extend these results to alternating weighted automata. For closure properties, 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 numerical *complement* L^c of a quantitative language L is defined by

$L^c(w) = 1 - L(w)$ for all words w.[4] We give the closure properties of all classes of weighted automata with respect to these four quantitative operations, extending the results of [4]. For expressiveness, we compare the sets of quantitative languages definable by the various classes of weighted automata, and we give a complete picture of their relationships. For decision problems, we consider a quantitative generalization of the classical questions of emptiness, universality, language inclusion, and language equivalence. The quantitative *emptiness* and *universality* problems ask, given a weighted automaton A (defining quantitative language L_A) and a rational number ν, if $L_A(w) \geq \nu$ for some (resp., all) words w. The quantitative *language-inclusion* and *language-equivalence* problems 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. For nondeterministic weighted automata, the quantitative emptiness problem is decidable in polynomial time for every value function, and the quantitative universality, language-inclusion, and language-equivalence problems are PSPACE-complete for all modes of branching of Sup-, LimSup-, and LimInf-automata [3]. We extend these results to alternating weighted automata. The main open question remains the decidability of the universality problem for limit-average and discounted-sum automata.

2 Definitions

While weighted automata have been studied extensively over finite words [12, 9], we focus on weighted automata over infinite words.

Value functions. We consider the following value functions $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$ to define quantitative languages. Given an infinite sequence $v = v_0 v_1 \dots$ of rational numbers, define

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

Alternating weighted automata. An *alternating weighted automaton* over a finite alphabet Σ is a tuple $A = \langle Q, q_I, \Sigma, \delta, \gamma \rangle$, where

- Q is a finite set of states, and $q_I \in Q$ is the initial state;
- $\delta \subseteq Q \times \Sigma \times (2^Q \setminus \{\varnothing\})$ is a finite set of labeled transitions;
- $\gamma : Q \times \Sigma \times Q \to \mathbb{Q}$ is a weight function.

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

We require that A is *total*, that is for all $q \in Q$ and $\sigma \in \Sigma$, there exists $(q, \sigma, s) \in \delta$ for at least one nonempty set $s \subseteq Q$. An automaton A is *universal* if for all $q \in Q$ and $\sigma \in \Sigma$, there exists $(q, \sigma, s) \in \delta$ for exactly one $s \subseteq Q$; it is *nondeterministic* if for all $(q, \sigma, s) \in \delta$, the set s is a singleton; and it is *deterministic* if it is both universal and nondeterministic.

The set of transitions $(q, \sigma, s_i) \in \delta$ from a state q over σ can be described by a boolean formula over Q, namely $\varphi(q, \sigma) = \bigvee_{(q,\sigma,s_i)\in\delta} \bigwedge_{q_j \in s_i} q_j$. For example, the formula $\varphi(q, \sigma) = (q_1 \wedge q_2) \vee (q_3 \wedge q_4)$ corresponds to the transitions $(q, \sigma, \{q_1, q_2\})$ and $(q, \sigma, \{q_3, q_4\})$. In a game interpretation of alternation, two players (the maximizer and the minimizer) are constructing a path in the automaton A while reading the input word. If the current state is q and the next input symbol is σ, then the maximizer (also called the nondeterministic player) chooses a set of states s_i such that $(q, \sigma, s_i) \in \delta$ (i.e., such that the formula $\varphi(q, \sigma)$ is satisfied when true is assigned to every state $q \in s_i$), and the minimizer (also called the universal player) then chooses a state $q' \in s_i$. Thus in the formula $\varphi(q, \sigma)$, disjunctions correspond to nondeterministic choices, and conjunctions correspond to universal choices. The outcome of the game is an infinite weighted path in the automaton, and the value of the input word is the maximal value of such a path that the maximizer can enforce regardless of the choices of the minimizer. We obtain the *dual* of an alternating weighted automaton by exchanging disjunctions and conjunctions in the boolean formulas of the transition relations.

Formally, a *run* of A over an infinite word $w = \sigma_0\sigma_1 \dots$ is a weighted Q-labelled tree (T, λ, γ') where $T \subseteq \mathbb{N}^*$ is a nonempty prefix-closed set of nodes (i.e., $x \cdot c \in T$ implies $x \in T$ for all $x \in \mathbb{N}^*$ and $c \in \mathbb{N}$), $\lambda : T \to Q$ and $\gamma' : \{(x, x \cdot c) \mid x \cdot c \in T\} \to \mathbb{Q}$ are labelings of the tree such that: *(i)* $\lambda(\epsilon) = q_I$ (where ϵ is the empty sequence) and *(ii)* if $x \in T$ and $\lambda(x) = q$, then there exists a set $s = \{q_1, \dots, q_k\} \subseteq Q$ such that $(q, \sigma_{|x|}, s) \in \delta$ and for all $1 \leq c \leq k$, we have $x \cdot c \in T$ and $\lambda(x \cdot c) = q_c$. Moreover, $\gamma'(x, x \cdot c) = \gamma(q, \sigma_{|x|}, q_c)$.

A *path* in a run $\rho = (T, \lambda, \gamma')$ is a set $\pi \subseteq T$ such that $\epsilon \in \pi$ and for all $x \in \pi$, there exists a unique $c \in \mathbb{N}$ such that $x \cdot c \in \pi$. We denote by $\mathsf{Run}^A(w)$ the set of all runs of A over w, and by $\mathsf{Path}(\rho)$ the set of all paths in a run ρ. We define $\gamma_\rho(\pi) = v_0 v_1 \dots$ such that for all $i \geq 0$, $v_i = \gamma'(x, x')$ where x, x' are the unique nodes of π with $|x'| = |x| + 1 = i + 1$.

Given a value function $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$, we say that the alternating Val-automaton A defines the quantitative language $L_A : \Sigma^\omega \to \mathbb{R}$ such that for all $w \in \Sigma^\omega$:

$$L_A(w) = \sup_{\rho \in \mathsf{Run}^A(w)} \ \inf_{\pi \in \mathsf{Path}(\rho)} \mathsf{Val}(\gamma_\rho(\pi)).$$

The *alternating* $\{0, 1\}$*-automata* are the special case of alternating weighted automata where all transition weights are either 0 or 1. In the case of Sup, LimSup, and LimInf, the $\{0, 1\}$-automata define *boolean languages* $L : \Sigma^\omega \to \{0, 1\}$ that are traditionally viewed as sets of words $\{w \in \Sigma^\omega \mid L(w) = 1\}$. Note that the LimSup- and LimInf- $\{0, 1\}$-automata are the classical Büchi and coBüchi automata respectively. A word is in the boolean language of an alternating Büchi (resp. coBüchi) automaton if there exists a run over that word all of whose paths contain infinitely many 1-weighted edges (resp. finitely many 0-weighted edges).

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 $w \in \Sigma^\omega$. 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 are natural generalizations of respectively the union, intersection and complement operators for boolean languages.

Reducibility. A class $\mathcal{C}$ of alternating weighted automata is *reducible* to a class $\mathcal{C}'$ of alternating weighted automata if for every $A \in \mathcal{C}$ there exists $A' \in \mathcal{C}'$ such that $L_A = L_{A'}$, i.e. $L_A(w) = L_{A'}(w)$ for all words $w \in \Sigma^\omega$. In particular, a class of weighted automata *can be determinized* if it is reducible to its deterministic counterpart. Two classes of weighted automata have the same expressiveness if they are reducible to each other.

Decision problems. We present quantitative generalizations of the classical decision problems in automata theory. Given two quantitative languages L_1, L_2, we write $L_1 \sqsubseteq L_2$ if $L_1(w) \leq L_2(w)$ for all words $w \in \Sigma^\omega$.

Given a weighted automaton A and a rational number $\nu \in \mathbb{Q}$, the *quantitative emptiness problem* asks whether there exists a word $w \in \Sigma^\omega$ such that $L_A(w) \geq \nu$, and the *quantitative universality problem* asks whether $L_A(w) \geq \nu$ for all words $w \in \Sigma^\omega$. 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 $L_A = L_B$. All results presented in this paper also hold for the decision problems defined above with inequalities replaced by strict inequalities.

Notation. We use acronyms to denote classes of weighted automata. The first letter can be A(lternating), N(ondeterministic), U(niversal), or D(eterministic). For $\mathrm{X} \in \{\mathrm{A}, \mathrm{N}, \mathrm{U}\}$ and $\mathrm{Y} \in \{\mathrm{D}, \mathrm{N}, \mathrm{U}\}$ (with $\mathrm{X} \neq \mathrm{Y}$), we use the notation $\substack{\mathrm{X}\\\mathrm{Y}}$ to denote the classes of automata for which the X and Y versions have the same expressiveness. Note that if the expressiveness of alternating and deterministic automata coincide for some class (i.e., X=A and Y=D), then the expressiveness of the four modes of branching is the same. The second part of the acronyms is one of the following: BW(Büchi), CW(coBüchi), Sup, LSup(LimSup), LInf(LimInf), LAvg(LimAvg), or Disc.

3 Closure Properties

We present the closure properties of alternating weighted automata with respect to the pointwise operations max, min, complement and sum.

We say that a class $\mathcal{C}$ of weighted automata is *closed* under a binary operator $\mathrm{op}(\cdot,\cdot)$ (resp. a unary operator $\mathrm{op}'(\cdot)$) if for all $A_1, A_2 \in \mathcal{C}$, there exists $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \mathrm{op}(L_{A_1}, L_{A_2})$ (resp. $L_{A_{12}} = \mathrm{op}'(L_{A_1})$). All closure properties that presented 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), we construct an automaton $A_{12} \in \mathcal{C}$ such that $L_{A_{12}} = \mathrm{op}(L_{A_1}, L_{A_2})$ with at most $O(f(n_1, m_1, n_2, m_2))$ states. We define analogously the cost of closure properties under unary operators. For all reductions presented, the size of the largest weight in A_{12} is linear in the size p of the largest weight in A_1, A_2 (however, the time needed to compute the weights is quadratic in p, as we need addition, multiplication, or comparison, which are quadratic operations over the rationals).

Note that every class of weighted automata is closed under shift by c and under scale by $|c|$ for all $c \in \mathbb{Q}$. For 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 all weights of an automaton to obtain the automaton for the shift by c of its language. Analogously, scaling by factor $|c|$ the weights of an automaton gives the scale by $|c|$ of its language. As a consequence, all closure properties also hold if the complement of a quantitative language L was defined as $k - L$ for any constant k.

Theorem 1. *The closure properties of alternating weighted automata are shown in Table 1.*

For example, according to Theorem 1, every class of alternating and nondeterministic weighted automata is closed under max, and every class of alternating and universal weighted automata is closed under min, all with cost $O(n_1 + n_2)$. This follows from the definition of alternating automata since the maximum and minimum of two quantitative languages can be obtained by an initial (either nondeterministic or universal) choice between the corresponding alternating automata.

The closure properties of nondeterministic weighted automata are established in [4]. The results for universal weighted automata are essentially obtained by duality since (i) if we interpret a universal automaton as a nondeterministic one, and if we replace each weight v by $1 - v$, then we obtain the complement of its quantitative language, and (ii) the maximum of two quantitative languages is the complement of the minimum of their complement.

For complementation, the positive closure results for LimSup- and LimInf-automata are obtained as a direct extension of the complementation results for NBW and UCW [10], and for Disc-automata by dualizing the automaton and replacing every weight v by $1-\lambda-v$ (where λ is the discount factor). The negative results for Sup-, LimSup-, and LimInf-automata follow from a similar result in the case of $\{0, 1\}$-automata. We give the essential argument for showing that alternating LimAvg-automata are not closed under complement.

Consider the alphabet $\Sigma = \{a, b\}$ and the language L_a that assigns to every word $w \in \Sigma^\omega$ the limit-average number of the a's in w. Formally, for an infinite word w, let w_j be its prefix of length j and let w_j^a and w_j^b denote the number of a's and b's in w_j, respectively. Then for $w \in \Sigma^\omega$ we have

$$L_a(w) = \liminf_{n\to\infty} \frac{1}{n} \cdot w_n^a.$$

Let us denote by $\hat{L}_b$ the language $(1 - L_a)$ and assume towards contradiction that there exists an ALAVG A with set Q of states for the language $\hat{L}_b$. Let β be the maximum absolute value of the weights in A. Since $\hat{L}_b(a^\omega) = 0$ and $\hat{L}_b(b^\omega) = 1$, we must have $L_A(a^\omega) = 0$ and $L_A(b^\omega) = 1$. By memoryless determinacy of perfect-information limit-average games [5], it follows that the following assertions hold: (a) it is possible to fix choices of the minimizer in the automaton on the letter a such that in the resulting non-deterministic automaton the sum of weights of all a-cycles C is at most 0; and (b) it is possible to fix choices of the minimizer in the automaton on the letter b such that in the resulting non-deterministic automaton the sum of weights of all b-cycles C is at most $|C|$. We fix the choices for the minimizer as above and consider a word w that consists of sequences of a's and b's of increasing length such that every sequence of a and b is of length at least $10 \cdot |Q| \cdot \beta$ and the long-run average number of b's oscillates between 0 and 1, i.e.

$$\liminf_{n\to\infty} \frac{1}{n} \cdot w_n^b = 0; \qquad \limsup_{n\to\infty} \frac{1}{n} \cdot w_n^b = 1.$$

Any run on a sequence of a's consists of a prefix of length at most Q (with sum of weights at most $|Q| \cdot \beta$), and then nested a-cycles where the sum of weights is at most 0. Similarly, any run on a sequence of b's consists of a prefix of length at most Q (with sum of weights at most $|Q| \cdot \beta$), and then nested b-cycles where the sum of weights is at most the length of the nested cycles. It is then easy to show that $L_A(w) \leq \frac{1}{10}$ while $\hat{L}_b(w) = 1$. Hence, we have a contradiction and the result follows.

Finally, every class of alternating weighted automata is closed under sum, except for LimAvg. Below, we give the proof that alternating LimAvg automata are not closed under sum. Consider the languages L_a and L_b over alphabet $\Sigma = \{a, b\}$ that assigns to each word w the long-run average number of a's and b's in w respectively. Let $L_+ = L_a + L_b$. Assume that L_+ is defined by an ALAVG A with set of states Q (we assume w.l.o.g that every state in Q is reachable). From every state $q \in Q$, the value of the words a^ω and b^ω in A is 1 since $L_+(w_q \cdot a^\omega) = L_+(w_q \cdot b^\omega) = 1$ for all finite words $w_q \in \Sigma^*$. Therefore, by memoryless determinacy of perfect-information limit-average games [5], we can fix a memoryless strategy for the maximizer (in the restriction of A to transitions over a's) such that all paths in the resulting graph have value at least 1. Hence, every cycle in this graph has average weight at least 1. The same result holds for the restriction of A to transitions over b's. Now, we can easily construct an input word $w = a^{n_1} b^{m_1} a^{n_2} b^{m_2} \ldots$ such that $L_a(w) = L_b(w) = 0$, but the maximizer

		max	min	complement	sum
alternating	ASup	$O(n_1 + n_2)$	$O(n_1 + n_2)$	$\times$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	ALsup	$O(n_1 + n_2)$	$O(n_1 + n_2)$	$O(m \cdot n^2)$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	ALinf	$O(n_1 + n_2)$	$O(n_1 + n_2)$	$O(m \cdot n^2)$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	ALavg	$O(n_1 + n_2)$	$O(n_1 + n_2)$	$\times$	$\times$
	ADisc	$O(n_1 + n_2)$	$O(n_1 + n_2)$	$O(n)$	$O(n_1 \cdot n_2)$
universal	USup	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$	$O(n_1 + n_2)$	$\times$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	ULsup	$O(n_1 \cdot n_2)$	$O(n_1 + n_2)$	$\times$	$O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$
	ULinf	$O(n_1 \cdot n_2 \cdot (m_1 + m_2))$	$O(n_1 + n_2)$	$O(m \cdot 2^{n \log n})$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	ULavg	$\times$	$O(n_1 + n_2)$	$\times$	$\times$
	UDisc	$\times$	$O(n_1 + n_2)$	$\times$	$O(n_1 \cdot n_2)$
nondeterm.	NSup	$O(n_1 + n_2)$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$	$\times$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	NLsup	$O(n_1 + n_2)$	$O(n_1 \cdot n_2 \cdot (m_1 + m_2))$	$O(m \cdot 2^{n \log n})$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	NLinf	$O(n_1 + n_2)$	$O(n_1 \cdot n_2)$	$\times$	$O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$
	NLavg	$O(n_1 + n_2)$	$\times$	$\times$	$\times$
	NDisc	$O(n_1 + n_2)$	$\times$	$\times$	$O(n_1 \cdot n_2)$
deterministic	DSup	$O(n_1 \cdot n_2)$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$	$\times$	$O(n_1 \cdot m_1 \cdot n_2 \cdot m_2)$
	DLsup	$O(n_1 \cdot n_2)$	$O(n_1 \cdot n_2)$	$\times$	$O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$
	DLinf	$O((m_1 + m_2) \cdot 2^{n_1+n_2})$	$O((m_1 + m_2) \cdot 2^{n_1+n_2})$	$\times$	$O(n_1 \cdot n_2 \cdot 2^{m_1 \cdot m_2})$
	DLavg	$\times$	$\times$	$\times$	$\times$
	DDisc	$\times$	$\times$	$O(n)$	$O(n_1 \cdot n_2)$

Table 1. Closure properties. The cost is given for the positive results, and the negative results are marked by $\times$. For example, given two alternating Sup-automata with n_1 and n_2 states, respectively, there is an alternating Sup-automaton with $O(n_1 + n_2)$ states that defines the max of their quantitative language; and there exist two universal LimAvg-automata such that the max of the their quantitative language cannot be defined by a universal LimAvg-automaton.

has a strategy (essentially to use the memoryless strategies for a^ω and b^ω) such that for all strategies of the minimizer, the outcome path has value arbitrarily close to 1, yielding a contradiction as then $L_A(w) = 1$ while $L_+(w) = 0$.

4 Expressive Power

The expressive power of nondeterministic weighted automata has been studied in detail in [3]. We present these results and extend them to alternating and universal weighted automata. Note that for each value function, the deterministic automata are reducible to the other modes of branching, and all modes of branching are reducible to alternating automata (as a straightforward consequence of the definition).

Theorem 2. *The relative expressive power of alternating weighted automata is as follows: a class $\mathcal{C}$ of alternating weighted automata can be reduced to a class $\mathcal{C}'$ if and only if there exists a path from $\mathcal{C}$ to $\mathcal{C}'$ in the directed graph of Figure 1.*

Note that Theorem 2 also holds if transition weights are irrational numbers. For Sup-automata, the alternating and deterministic automata have the same

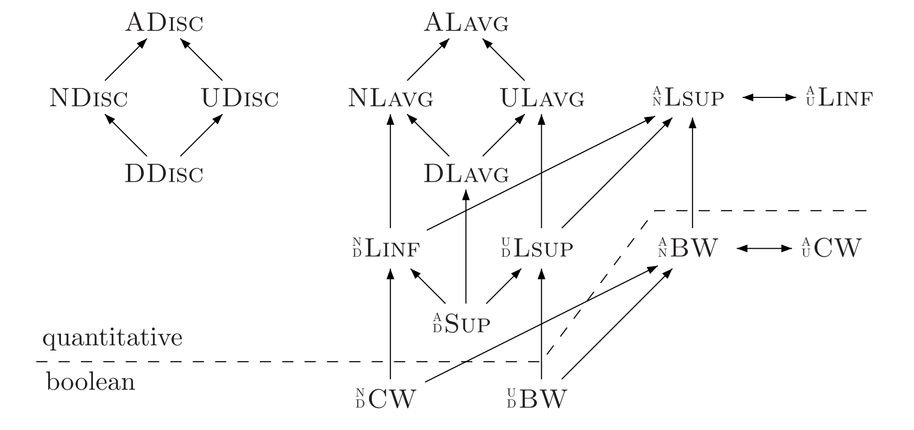

Fig. 1. Reducibility relation. A class $\mathcal{C}$ of automata can be reduced to $\mathcal{C}'$ iff $\mathcal{C} \to^* \mathcal{C}'$.

expressive power, thus we denote this class by ${}^{\mathrm{A}}_{\mathrm{D}}$Sup. For LimInf- and LimSup-automata, the relative expressive power is the same as for $\{0, 1\}$-automata, and the proofs are based on generalization of the constructions for the boolean case [10].

For LimAvg- and Disc-automata, the main result is that nondeterministic automata cannot be determinized [3]. From that and the fact that ALavg and ADisc are closed under max and min while NLavg and NDisc are not closed under min, and ULavg and UDisc are not closed under max, it follows that the alternating automata are reducible neither to nondeterministic automata, nor to universal automata.

When comparing different classes of weighted automata, the most surprising result is probably the fact that the class of DBW (which defines a strict subclass of the ω-regular languages) are not reducible to NLavg, and similarly DCW are not reducible to ULavg.

Finally, note that Disc-automata are incomparable with the other classes of weighted automata. This follows from the property that the value of a path in a Disc-automaton is essentially determined by a finite prefix, in the sense that the values of two paths can be arbitrarily close if 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 path depends essentially on its tail and is independent of finite prefixes.

5 Decision Problems

We study the complexity of the quantitative emptiness, universality, language-inclusion, and language-equivalence problems for alternating weighted automata.

		emptiness	universality	inclusion	equivalence
alternating	ASup	PSPACE-complete	PSPACE-complete	PSPACE-complete	PSPACE-complete
	ALsup	PSPACE-complete	PSPACE-complete	PSPACE-complete	PSPACE-complete
	ALinf	PSPACE-complete	PSPACE-complete	PSPACE-complete	PSPACE-complete
	ALavg	?	?	?	?
	ADisc	co-r.e.	co-r.e.	co-r.e.	co-r.e.
universal	USup	PSPACE-complete	PTIME	PSPACE-complete	PSPACE-complete
	ULsup	PSPACE-complete	PTIME	PSPACE-complete	PSPACE-complete
	ULinf	PSPACE-complete	PTIME	PSPACE-complete	PSPACE-complete
	ULavg	?	PTIME	?	?
	UDisc	co-r.e.	PTIME	co-r.e.	co-r.e.
nondeterm.	NSup	PTIME	PSPACE-complete	PSPACE-complete	PSPACE-complete
	NLsup	PTIME	PSPACE-complete	PSPACE-complete	PSPACE-complete
	NLinf	PTIME	PSPACE-complete	PSPACE-complete	PSPACE-complete
	NLavg	PTIME	?	?	?
	NDisc	PTIME	co-r.e.	co-r.e.	co-r.e.
deterministic	DSup	PTIME	PTIME	PTIME	PTIME
	DLsup	PTIME	PTIME	PTIME	PTIME
	DLinf	PTIME	PTIME	PTIME	PTIME
	DLavg	PTIME	PTIME	PTIME	PTIME
	DDisc	PTIME	PTIME	PTIME	PTIME

Table 2. Complexity results for the quantitative decision problems. The decidability of the problems marked by ? is, to the best of our knowledge, open.

Theorem 3. *Table 2 summarizes the known complexity results for the quantitative decision problems of alternating weighted automata.*

The quantitative emptiness problem for nondeterministic weighted automata can be solved by a 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 [6, 8, 1]. By duality, we get the same results for the quantitative universality problem of universal weighted automata.

The universality problem is known to be PSPACE-complete for finite automata and for NBW [11, 13]. This result extends easily to nondeterministic Sup- and LimSup-automata and to the related problems of quantitative language inclusion and equivalence. The results about expressive power and the duality between LimSup and LimInf, and between nondeterministic and universal modes of branching allow to derive the PSPACE-completeness results of Table 2.

The main open question about decision problems remains the decidability of quantitative universality for LimAvg- and Disc-automata. For Disc-automata, a partial answer is known since the quantitative universality problem is co-recursively enumerable [3].

References

1. D. Andersson. An improved algorithm for discounted payoff games. In *ESSLLI Student Session*, pages 91–98, 2006.
2. A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.
3. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proc. of CSL: Computer Science Logic*, LNCS 5213, pages 385–400. Springer, 2008.
4. K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages. In *Proc. of LICS: Logic in Computer Science*. IEEE Comp. Soc. Press, 2009.
5. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.
6. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
7. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, LNCS 2500. Springer, 2002.
8. R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978.
9. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1986.
10. O. Kupferman and M. Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.*, 2(3):408–429, 2001.
11. A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proc. of FOCS: Foundations of Computer Science*, pages 125–129. IEEE, 1972.
12. M. P. Schützenberger. On the definition of a family of automata. *Information and control*, 4:245–270, 1961.
13. 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.
14. M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. of LICS: Logic in Computer Science*, pages 332–344. IEEE, 1986.