# Mean-Payoff Automaton Expressions*

Krishnendu Chatterjee[1], Laurent Doyen[2], Herbert Edelsbrunner[1],
Thomas A. Henzinger[1], and Philippe Rannou[3]

[1] IST Austria (Institute of Science and Technology Austria)
[2] LSV, ENS Cachan & CNRS, France
[3] ENS Cachan Bretagne, Rennes, France

**Abstract.** Quantitative languages are an extension of boolean languages that assign to each word a real number. Mean-payoff automata are finite automata with numerical weights on transitions that assign to each infinite path the long-run average of the transition weights. When the mode of branching of the automaton is deterministic, nondeterministic, or alternating, the corresponding class of quantitative languages is not *robust* as it is not closed under the pointwise operations of max, min, sum, and numerical complement. Nondeterministic and alternating mean-payoff automata are not *decidable* either, as the quantitative generalization of the problems of universality and language inclusion is undecidable.

We introduce a new class of quantitative languages, defined by *mean-payoff automaton expressions*, which is robust and decidable: it is closed under the four pointwise operations, and we show that all decision problems are decidable for this class. Mean-payoff automaton expressions subsume deterministic mean-payoff automata, and we show that they have expressive power incomparable to nondeterministic and alternating mean-payoff automata. We also present for the first time an algorithm to compute distance between two quantitative languages, and in our case the quantitative languages are given as mean-payoff automaton expressions.

## 1 Introduction

Quantitative languages $L$ are a natural generalization of boolean languages that assign to every word $w$ a real number $L(w) \in \mathbb{R}$ 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) needed to produce it, or bound the long-run average available use of the resource. Thus quantitative languages can specify properties related to resource-constrained programs, and an implementation $L_A$ satisfies (or refines) a specification $L_B$ if $L_A(w) \leq L_B(w)$ for all words $w$. This notion of refinement is a *quantitative generalization of language inclusion*, and it can be used to check for example if for each behavior, the long-run average response time of the system lies below the specified average response requirement. Hence it is crucial to identify some relevant class of quantitative languages for which this question is decidable. The

* This research was supported by the European Union project COMBEST, the European Network of Excellence ArtistDesign, the DARPA grant HR0011-05-1-0057, and the NSF grant DBI-0820624.

other classical decision questions such as emptiness, universality, and language equivalence have also a natural quantitative extension. For example, the *quantitative emptiness problem* asks, given a quantitative language $L$ and a threshold $\nu \in \mathbb{Q}$, whether there exists some word $w$ such that $L(w) \geq \nu$, and the *quantitative universality problem* asks whether $L(w) \geq \nu$ for all words $w$. Note that universality is a special case of language inclusion (where $L_A(w) = \nu$ is constant).

Weighted *mean-payoff automata* present a nice framework to express such quantitative properties [4]. A weighted mean-payoff automaton is a finite automaton with numerical weights on transitions. The value of a word $w$ is 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 the long-run average of the weights that appear along $r$. A mean-payoff extension to alternating automata has been studied in [5]. Deterministic, nondeterministic and alternating mean-payoff automata are three classes of mean-payoff automata with increasing expressive power. However, none of these classes is closed under the four pointwise operations of max, min (which generalize union and intersection respectively), numerical complement[4], and sum (see Table 1). Deterministic mean-payoff automata are not closed under max, min, and sum [6]; nondeterministic mean-payoff automata are not closed under min, sum and complement [6]; and alternating mean-payoff automata are not closed under sum [5]. Hence none of the above classes is *robust* with respect to closure properties.

Moreover, while deterministic mean-payoff automata enjoy decidability of all quantitative decision problems [4], the quantitative language-inclusion problem is undecidable for nondeterministic and alternating mean-payoff automata [10], and thus also all decision problems are undecidable for alternating mean-payoff automata. Hence although mean-payoff automata provide a nice framework to express quantitative properties, there is no known class which is both robust and decidable (see Table 1).

In this paper, we introduce a new class of quantitative languages that are defined by *mean-payoff automaton expressions*. An expression is either a deterministic mean-payoff automaton, or it is the max, min, or sum of two mean-payoff automaton expressions. Since deterministic mean-payoff automata are closed under complement, mean-payoff automaton expressions form a robust class that is closed under max, min, sum and complement. We show that (a) all decision problems (quantitative emptiness, universality, inclusion, and equivalence) are decidable for mean-payoff automaton expressions; (b) mean-payoff automaton expressions are incomparable in expressive power with both the nondeterministic and alternating mean-payoff automata (i.e., there are quantitative languages expressible by mean-payoff automaton expressions that are not expressible by alternating mean-payoff automata, and there are quantitative languages expressible by nondeterministic mean-payoff automata that are not expressible by mean-payoff automata expressions); and (c) the properties of cut-point languages (i.e., the sets of words with value above a certain threshold) for deterministic automata carry over to mean-payoff automaton expressions, mainly the cut-point language is $\omega$-regular when the threshold is isolated (i.e., some neighborhood around the threshold contains no word). Moreover, mean-payoff automaton expressions can express all examples in the literature of quantitative properties using mean-payoff measure [1, 6, 7].

[4] The numerical complement of a quantitative languages $L$ is $-L$.

| | Closure properties | | | | Decision problems | | | |
|---|---|---|---|---|---|---|---|---|
| | max | min | sum | comp. | empt. | univ. | incl. | equiv. |
| Deterministic | × | × | × | ✓[5] | ✓ | ✓ | ✓ | ✓ |
| Nondeterministic | ✓ | × | × | × | ✓ | × | × | × |
| Alternating | ✓ | ✓ | × | ✓[5] | × | × | × | × |
| Expressions | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Table 1.** Closure properties and decidability of the various classes of mean-payoff automata. Mean-payoff automaton expressions enjoy fully positive closure and decidability properties.

Along with the quantitative generalization of the classical decision problems, we also consider the notion of *distance* between two quantitative languages $L_A$ and $L_B$, defined as $\sup_w |L_A(w) - L_B(w)|$. When quantitative language inclusion does not hold between an implementation $L_A$ and a specification $L_B$, the distance is a relevant information to evaluate how far they are from each other, as we may accept implementations that overspend the resource but we would prefer the least expensive ones. We present the first algorithm to compute the distance between two quantitative languages: we show that the distance can be computed for mean-payoff automaton expressions.

Our approach to show decidability of mean-payoff automaton expressions relies on the characterization and algorithmic computation of the *value set* $\{L_E(w) \mid w \in \Sigma^\omega\}$ of an expression $E$, i.e. the set of all values of words according to $E$. The value set can be viewed as an abstract representation of the quantitative language $L_E$, and we show that all decision problems, cut-point language and distance computation can be solved efficiently once we have this set.

First, we present a precise characterization of the value set for quantitative languages defined by mean-payoff automaton expressions. In particular, we show that it is not sufficient to construct the convex hull $\mathsf{conv}(S_E)$ of the set $S_E$ of the values of simple cycles in the mean-payoff automata occurring in $E$, but we need essentially to apply an operator $F_{\min}(\cdot)$ which given a set $Z \subseteq \mathbb{R}^n$ computes the set of points $y \in \mathbb{R}^n$ that can be obtained by taking pointwise minimum of each coordinate of points of a set $X \subseteq Z$. We show that while we need to compute the set $V_E = F_{\min}(\mathsf{conv}(S_E))$ to obtain the value set, and while this set is always convex, it is not always the case that $F_{\min}(\mathsf{conv}(S_E)) = \mathsf{conv}(F_{\min}(S_E))$ (which would immediately give an algorithm to compute $V_E$). This may appear counter-intuitive because the equality holds in $\mathbb{R}^2$ but we show that the equality does not hold in $\mathbb{R}^3$ (Example 2).

Second, we provide algorithmic solutions to compute $F_{\min}(\mathsf{conv}(S))$, for a finite set $S$. We first present a constructive procedure that given $S$ constructs a finite set of points $S'$ such that $\mathsf{conv}(S') = F_{\min}(\mathsf{conv}(S))$. The explicit construction presents interesting properties about the set $F_{\min}(\mathsf{conv}(S))$, however the procedure itself is computationally expensive. We then present an elegant and geometric construction of $F_{\min}(\mathsf{conv}(S))$ as a set of linear constraints. The computation of $F_{\min}(\mathsf{conv}(S))$ is a new problem in

[5] Closure under complementation holds because LimInfAvg-automata and LimSupAvg-automata are dual. It would not hold if only LimInfAvg-automata (or only LimSupAvg-automata) were allowed.

computational geometry and the solutions we present could be of independent interest. Using the algorithm to compute $F_{\min}(\mathsf{conv}(S))$, we show that all decision problems for mean-payoff automaton expressions are decidable. Due to lack of space, most proofs are given in the fuller version [3].

*Related works.* Quantitative languages have been first studied over finite words in the context of probabilistic automata [17] and weighted automata [18]. Several works have generalized the theory of weighted automata to infinite words (see [14, 12, 16, 2] and [13] for a survey), but none of those have considered mean-payoff conditions. Examples where the mean-payoff measure has been used to specify long-run behaviours of systems can be found in game theory [15, 20] and in Markov decision processes [8]. The mean-payoff automata as a specification language have been investigated in [4, 6, 5], and extended in [1] to construct a new class of (non-quantitative) languages of infinite words (the multi-threshold mean-payoff languages), obtained by applying a query to a mean-payoff language, and for which emptiness is decidable. It turns out that a richer language of queries can be expressed using mean-payoff automaton expressions (together with decidability of the emptiness problem). A detailed comparison with the results of [1] is given in Section 5. Moreover, we provide algorithmic solutions to the quantitative language inclusion and equivalence problems and to distance computation which have no counterpart for non-quantitative languages. Related notions of metrics have been addressed in stochastic games [9] and probabilistic processes [11, 19].

## 2 Mean-Payoff Automaton Expressions

**Quantitative languages.** A *quantitative language* $L$ over a finite alphabet $\Sigma$ is a function $L : \Sigma^\omega \to \mathbb{R}$. Given two quantitative languages $L_1$ and $L_2$ over $\Sigma$, we denote by $\max(L_1, L_2)$ (resp., $\min(L_1, L_2)$, $\mathrm{sum}(L_1, L_2)$ and $-L_1$) the quantitative language that assigns $\max(L_1(w), L_2(w))$ (resp., $\min(L_1(w), L_2(w))$, $L_1(w) + L_2(w)$, and $-L_1(w)$) to each word $w \in \Sigma^\omega$. The quantitative language $-L$ is called the *complement* of $L$. The $\max$ and $\min$ operators for quantitative languages correspond respectively to the least upper bound and greatest lower bound for the pointwise order $\preceq$ such that $L_1 \preceq L_2$ if $L_1(w) \leq L_2(w)$ for all $w \in \Sigma^\omega$. Thus, they generalize respectively the union and intersection operators for classical boolean languages.

**Weighted automata.** A $\mathbb{Q}$-*weighted automaton* is a tuple $A = \langle Q, q_I, \Sigma, \delta, \mathsf{wt}\rangle$, where

- $Q$ is a finite set of states, $q_I \in Q$ is the initial state, and $\Sigma$ is a finite alphabet;
- $\delta \subseteq Q \times \Sigma \times Q$ is a finite set of labelled transitions. We assume that $\delta$ is *total*, i.e., for all $q \in Q$ and $\sigma \in \Sigma$, there exists $q'$ such that $(q, \sigma, q') \in \delta$;
- $\mathsf{wt} : \delta \to \mathbb{Q}$ 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.

**Words and runs.** A *word* $w \in \Sigma^\omega$ is an infinite sequence of letters from $\Sigma$. A *lasso-word* $w$ in $\Sigma^\omega$ is an ultimately periodic word of the form $w_1 \cdot w_2^\omega$, where $w_1 \in \Sigma^*$

is a finite prefix, and $w_2 \in \Sigma^+$ is a finite and nonempty word. A *run* of $A$ over an infinite word $w = \sigma_1\sigma_2 \dots$ is an 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 $i \geq 0$. We denote by $\mathsf{wt}(r) = v_0v_1 \dots$ the sequence of weights that occur in $r$ where $v_i = \mathsf{wt}(q_i, \sigma_{i+1}, q_{i+1})$ for all $i \geq 0$.

**Quantitative language of mean-payoff automata.** The *mean-payoff value* (or limit-average) of a sequence $\bar{v} = v_0v_1 \dots$ of real numbers is either

$$\mathsf{LimInfAvg}(\bar{v}) = \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i, \text{ or } \quad \mathsf{LimSupAvg}(\bar{v}) = \limsup_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i.$$

Note that if we delete or insert finitely many values in an infinite sequence of numbers, its limit-averages do not change, and if the sequence is ultimately periodic, then the LimInfAvg and LimSupAvg values coincide (and correspond to the mean of the weights on the periodic part of the sequence). However in general the LimInfAvg and LimSupAvg values do not coincide.

For $\mathsf{Val} \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$, the quantitative language $L_A$ of $A$ is defined by $L_A(w) = \sup\{\mathsf{Val}(\mathsf{wt}(r)) \mid r \text{ is a run of } A \text{ over } w\}$ for all $w \in \Sigma^\omega$. Accordingly, the automaton $A$ and its quantitative language $L_A$ are called LimInfAvg or LimSupAvg. Note that for deterministic automata, we have $L_A(w) = \mathsf{Val}(\mathsf{wt}(r))$ where $r$ is the unique run of $A$ over $w$.

We omit the weight function wt when it is clear from the context, and we write LimAvg when the value according to LimInfAvg and LimSupAvg coincide (e.g., for runs with a lasso shape).

**Decision problems and distance.** We consider the following classical decision problems for quantitative languages, assuming an effective presentation of quantitative languages (such as mean-payoff automata, or automaton expressions defined later). Given a quantitative language $L$ and a threshold $\nu \in \mathbb{Q}$, the *quantitative emptiness problem* asks whether there exists a word $w \in \Sigma^\omega$ such that $L(w) \geq \nu$, and the *quantitative universality problem* asks whether $L(w) \geq \nu$ for all words $w \in \Sigma^\omega$.

Given two quantitative languages $L_1$ and $L_2$, the *quantitative language-inclusion problem* asks whether $L_1(w) \leq L_2(w)$ for all words $w \in \Sigma^\omega$, and the *quantitative language-equivalence problem* asks whether $L_1(w) = L_2(w)$ for all words $w \in \Sigma^\omega$. Note that universality is a special case of language inclusion where $L_1$ is constant. Finally, the *distance* between $L_1$ and $L_2$ is $D_{\sup}(L_1, L_2) = \sup_{w\in\Sigma^\omega} |L_1(w) - L_2(w)|$. It measures how close is an implementation $L_1$ as compared to a specification $L_2$.

It is known that quantitative emptiness is decidable for nondeterministic mean-payoff automata [4], while decidability was open for alternating mean-payoff automata, and for the quantitative language-inclusion problem of nondeterministic mean-payoff automata. From recent undecidability results on games with imperfect information and mean-payoff objective [10] we derive that these problems are undecidable (Theorem 5).

**Robust quantitative languages.** A class $\mathcal{Q}$ of quantitative languages is *robust* if the class is closed under $\max, \min, \mathrm{sum}$ and complementation operations. The closure properties allow quantitative languages from a robust class to be described compositionally. While nondeterministic LimInfAvg- and LimSupAvg-automata are closed under the $\max$ operation, they are not closed under $\min$ and complement [6]. Alternating

$\mathsf{LimInfAvg}$- and $\mathsf{LimSupAvg}$-automata[6] are closed under $\max$ and $\min$, but are not closed under complementation and $\mathrm{sum}$ [5]. We define a *robust* class of quantitative languages for mean-payoff automata which is closed under $\max$, $\min$, $\mathrm{sum}$, and complement, and which can express all natural examples of quantitative languages defined using the mean-payoff measure [1, 6, 7].

**Mean-payoff automaton expressions.** A *mean-payoff automaton expression* $E$ is obtained by the following grammar rule:

$$E ::= A \mid \max(E, E) \mid \min(E, E) \mid \mathrm{sum}(E, E)$$

where $A$ is a *deterministic* $\mathsf{LimInfAvg}$- or $\mathsf{LimSupAvg}$-automaton. The quantitative language $L_E$ of a mean-payoff automaton expression $E$ is $L_E = L_A$ if $E = A$ is a deterministic automaton, and $L_E = \mathrm{op}(L_{E_1}, L_{E_2})$ if $E = \mathrm{op}(E_1, E_2)$ for $\mathrm{op} \in \{\max, \min, \mathrm{sum}\}$. By definition, the class of mean-payoff automaton expression is closed under $\max$, $\min$ and $\mathrm{sum}$. Closure under complement follows from the fact that the complement of $\max(E_1, E_2)$ is $\min(-E_1, -E_2)$, the complement of $\min(E_1, E_2)$ is $\max(-E_1, -E_2)$, the complement of $\mathrm{sum}(E_1, E_2)$ is $\mathrm{sum}(-E_1, -E_2)$, and the complement of a deterministic $\mathsf{LimInfAvg}$-automaton can be defined by the same automaton with opposite weights and interpreted as a $\mathsf{LimSupAvg}$-automaton, and vice versa, since $-\limsup(v_0, v_1, \dots) = \liminf(-v_0, -v_1, \dots)$. Note that arbitrary linear combinations of deterministic mean-payoff automaton expressions (expressions such as $c_1E_1 + c_2E_2$ where $c_1, c_2 \in \mathbb{Q}$ are rational constants) can be obtained for free since scaling the weights of a mean-payoff automaton by a positive factor $|c|$ results in a quantitative language scaled by the same factor.

## 3 The Vector Set of Mean-Payoff Automaton Expressions

Given a mean-payoff automaton expression $E$, let $A_1, \dots, A_n$ be the deterministic weighted automata occurring in $E$. The *vector set* of $E$ is the set $V_E = \{\langle L_{A_1}(w), \dots, L_{A_n}(w)\rangle \in \mathbb{R}^n \mid w \in \Sigma^\omega\}$ of tuples of values of words according to each automaton $A_i$. In this section, we characterize the vector set of mean-payoff automaton expressions, and in Section 4 we give an algorithmic procedure to compute this set. This will be useful to establish the decidability of all decision problems, and to compute the distance between mean-payoff automaton expressions. Given a vector $v \in \mathbb{R}^n$, we denote by $\|v\| = \max_i |v_i|$ the $\infty$-*norm* of $v$.

The *synchronized product* of $A_1, \dots, A_n$ such that $A_i = \langle Q_i, q_I^i, \Sigma, \delta_i, \mathsf{wt}_i\rangle$ is the $\mathbb{Q}^n$-weighted automaton $A_E = A_1 \times \dots \times A_n = \langle Q_1 \times \dots \times Q_n, (q_I^1, \dots, q_I^n), \Sigma, \delta, \mathsf{wt}\rangle$ such that $t = ((q_1, \dots, q_n), \sigma, (q'_1, \dots, q'_n)) \in \delta$ if $t_i := (q_i, \sigma, q'_i) \in \delta_i$ for all $1 \leq i \leq n$, and $\mathsf{wt}(t) = (\mathsf{wt}_1(t_1), \dots, \mathsf{wt}_n(t_n))$. In the sequel, we assume that all $A_i$'s are deterministic $\mathsf{LimInfAvg}$-automata (hence, $A_E$ is deterministic) and that the underlying graph of the automaton $A_E$ has only one strongly connected component (scc). We show later how to obtain the vector set without these restrictions.

[6] See [5] for the definition of alternating $\mathsf{LimInfAvg}$- and $\mathsf{LimSupAvg}$-automata that generalize nondeterministic automata.

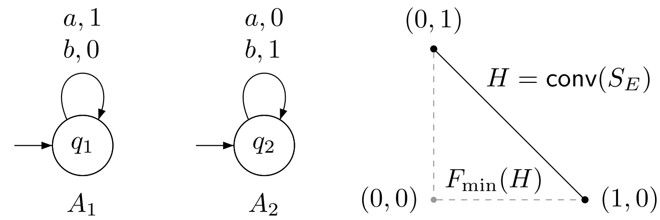


**Fig. 1.** The vector set of $E = \max(A_1, A_2)$ is $F_{\min}(\mathsf{conv}(S_E)) \supsetneq \mathsf{conv}(S_E)$.

For each (simple) cycle $\rho$ in $A_E$, let the *vector value* of $\rho$ be the mean of the tuples labelling the edges of $\rho$, denoted $\mathsf{Avg}(\rho)$. To each simple cycle $\rho$ in $A_E$ corresponds a (not necessarily simple) cycle in each $A_i$, and the vector value $(v_1, \dots, v_n)$ of $\rho$ contains the mean value $v_i$ of $\rho$ in each $A_i$. We denote by $S_E$ the (finite) set of vector values of simple cycles in $A_E$. Let $\mathsf{conv}(S_E)$ be the convex hull of $S_E$.

**Lemma 1.** *Let $E$ be a mean-payoff automaton expression. The set $\mathsf{conv}(S_E)$ is the closure of the set $\{L_E(w) \mid w$ is a lasso-word$\}$.*

The vector set of $E$ contains more values than the convex hull $\mathsf{conv}(S_E)$, as shown by the following example.

*Example 1.* Consider the expression $E = \max(A_1, A_2)$ where $A_1$ and $A_2$ are deterministic $\mathsf{LimInfAvg}$-automata (see Fig. 1). The product $A_E = A_1 \times A_2$ has two simple cycles with respective vector values $(1, 0)$ (on letter '$a$') and $(0, 1)$ (on letter '$b$'). The set $H = \mathsf{conv}(S_E)$ is the solid segment on Fig. 1 and contains the vector values of all lasso-words. However, other vector values can be obtained: consider the word $w = a^{n_1} b^{n_2} a^{n_3} b^{n_4} \dots$ where $n_1 = 1$ and $n_{i+1} = (n_1 + \dots + n_i)^2$ for all $i \geq 1$. It is easy to see that the value of $w$ according to $A_1$ is $0$ because the average number of $a$'s in the prefixes $a^{n_1} b^{n_2} \dots a^{n_i} b^{n_{i+1}}$ for $i$ odd is smaller than $\frac{n_1 + \dots + n_i}{n_1 + \dots + n_i + n_{i+1}} = \frac{1}{1 + n_1 + \dots + n_i}$ which tends to $0$ when $i \to \infty$. Since $A_1$ is a $\mathsf{LimInfAvg}$-automaton, the value of $w$ is $0$ in $A_1$, and by a symmetric argument the value of $w$ is also $0$ in $A_2$. Therefore the vector $(0, 0)$ is in the vector set of $E$. Note that $z = (z_1, z_2) = (0, 0)$ is the pointwise minimum of $x = (x_1, x_2) = (1, 0)$ and $y = (y_1, y_2) = (0, 1)$, i.e. $z = f_{\min}(x, y)$ where $z_1 = \min(x_1, y_1)$ and $z_2 = \min(y_1, y_2)$. In fact, the vector set is the whole triangular region in Fig. 1, i.e. $V_E = \{f_{\min}(x, y) \mid x, y \in \mathsf{conv}(S_E)\}$. □

We generalize $f_{\min}$ to finite sets of points $P \subseteq \mathbb{R}^n$ in $n$ dimensions as follows: $f_{\min}(P) \in \mathbb{R}^n$ is the point $p = (p_1, p_2, \dots, p_n)$ such that $p_i$ is the minimum $i^{\text{th}}$ coordinate of the points in $P$, for $1 \leq i \leq n$. For arbitrary $S \subseteq \mathbb{R}^n$, define $F_{\min}(S) = \{f_{\min}(P) \mid P$ is a finite subset of $S\}$. As illustrated in Example 1, the next lemma shows that the vector set $V_E$ is equal to $F_{\min}(\mathsf{conv}(S_E))$.

**Lemma 2.** *Let $E$ be a mean-payoff automaton expression built from deterministic* $\mathsf{LimInfAvg}$-*automata, and such that $A_E$ has only one strongly connected component. Then, the vector set of $E$ is $V_E = F_{\min}(\mathsf{conv}(S_E))$.*

For a general mean-payoff automaton expression $E$ (with both deterministic LimInfAvg- and LimSupAvg automata, and with multi-scc underlying graph), we can use the result of Lemma 2 as follows. We replace each LimSupAvg automaton $A_i$ occurring in $E$ by the LimInfAvg automaton $A'_i$ obtained from $A_i$ by replacing every weight wt by $-$wt. The duality of $\liminf$ and $\limsup$ yields $L_{A'_i} = -L_{A_i}$. In each strongly connected component $\mathcal{C}$ of the underlying graph of $A_E$, we compute $V_{\mathcal{C}} = F_{\min}(\mathsf{conv}(S_{\mathcal{C}}))$ (where $S_{\mathcal{C}}$ is the set of vector values of the simple cycles in $\mathcal{C}$) and apply the transformation $x_i \to -x_i$ on every coordinate $i$ where the automaton $A_i$ was originally a LimSupAvg automaton. The union of the sets $\bigcup_{\mathcal{C}} V_{\mathcal{C}}$ where $\mathcal{C}$ ranges over the strongly connected components of $A_E$ gives the vector set of $E$.

**Theorem 1.** *Let $E$ be a mean-payoff automaton expression built from deterministic* LimInfAvg-*automata, and let $\mathcal{Z}$ be the set of strongly connected components in $A_E$. For a strongly connected component $\mathcal{C}$ let $S_{\mathcal{C}}$ denote the set of vector values of the simple cycles in $\mathcal{C}$. The vector set of $E$ is $V_E = \bigcup_{\mathcal{C} \in \mathcal{Z}} F_{\min}(\mathsf{conv}(S_{\mathcal{C}}))$.*

## 4 Computation of $F_{\min}(\mathsf{conv}(S))$ for a Finite Set $S$

It follows from Theorem 1 that the vector set $V_E$ of a mean-payoff automaton expression $E$ can be obtained as a union of sets $F_{\min}(\mathsf{conv}(S))$, where $S \subseteq \mathbb{R}^n$ is a finite set. However, the set $\mathsf{conv}(S)$ being in general infinite, it is not immediate that $F_{\min}(\mathsf{conv}(S))$ is computable. In this section we consider the problem of computing $F_{\min}(\mathsf{conv}(S))$ for a finite set $S$. In subsection 4.1 we present an explicit construction and in subsection 4.2 we give a geometric construction of the set as a set of linear constraints. We first present some properties of the set $F_{\min}(\mathsf{conv}(S))$.

**Lemma 3.** *If $X$ is a convex set, then $F_{\min}(X)$ is convex.*

By Lemma 3, the set $F_{\min}(\mathsf{conv}(S))$ is convex, and since $F_{\min}$ is a monotone operator and $S \subseteq \mathsf{conv}(S)$, we have $F_{\min}(S) \subseteq F_{\min}(\mathsf{conv}(S))$ and thus $\mathsf{conv}(F_{\min}(S)) \subseteq F_{\min}(\mathsf{conv}(S))$. The following proposition states that in two dimensions the above sets coincide.

**Proposition 1.** *Let $S \subseteq \mathbb{R}^2$ be a finite set. Then,* $\mathsf{conv}(F_{\min}(S)) = F_{\min}(\mathsf{conv}(S))$.

We show in the following example that in three dimensions the above proposition does not hold, i.e., we show that $F_{\min}(\mathsf{conv}(S_E)) \neq \mathsf{conv}(F_{\min}(S_E))$ in $\mathbb{R}^3$.

*Example 2.* We show that in three dimension there is a finite set $S$ such that $F_{\min}(\mathsf{conv}(S)) \not\subseteq \mathsf{conv}(F_{\min}(S))$. Let $S = \{q, r, s\}$ with $q = (0, 1, 0)$, $r = (-1, -1, 1)$, and $s = (1, 1, 1)$. Then $f_{\min}(r, s) = r$, $f_{\min}(q, r, s) = f_{\min}(q, r) = t = (-1, -1, 0)$, and $f_{\min}(q, s) = q$. Therefore $F_{\min}(S) = \{q, r, s, t\}$. Consider $p = (r + s)/2 = (0, 0, 1)$. We have $p \in \mathsf{conv}(S)$ and $f_{\min}(p, q) = (0, 0, 0)$. Hence $(0, 0, 0) \in F_{\min}(\mathsf{conv}(S))$. We now show that $(0, 0, 0)$ does not belong to $\mathsf{conv}(F_{\min}(S))$. Consider $u = \alpha_q \cdot q + \alpha_r \cdot r + \alpha_s \cdot s + \alpha_t \cdot t$ such that $u$ in $\mathsf{conv}(F_{\min}(S))$. Since the third coordinate is non-negative for $q, r, s$, and $t$, it follows that if $\alpha_r > 0$ or $\alpha_s > 0$, then the third coordinate of $u$ is positive. If $\alpha_s = 0$ and $\alpha_r = 0$, then we have two cases: (a) if $\alpha_t > 0$, then the first coordinate of $u$ is negative; and (b) if $\alpha_t = 0$, then the second coordinate of $u$ is 1. It follows $(0, 0, 0)$ is not in $\mathsf{conv}(F_{\min}(S))$. □

### 4.1 Explicit construction

Example 2 shows that in general $F_{\min}(\mathsf{conv}(S)) \not\subseteq \mathsf{conv}(F_{\min}(S))$. In this section we present an explicit construction that given a finite set $S$ constructs a finite set $S'$ such that (a) $S \subseteq S' \subseteq \mathsf{conv}(S)$ and (b) $F_{\min}(\mathsf{conv}(S)) \subseteq \mathsf{conv}(F_{\min}(S'))$. It would follow that $F_{\min}(\mathsf{conv}(S)) = \mathsf{conv}(F_{\min}(S'))$. Since convex hull of a finite set is computable and $F_{\min}(S')$ is finite, this would give us an algorithm to compute $F_{\min}(\mathsf{conv}(S))$. For simplicity, for the rest of the section we write $F$ for $F_{\min}$ and $f$ for $f_{\min}$ (i.e., we drop the $\min$ from subscript). Recall that $F(S) = \{f(P) \mid P \text{ finite subset of } S\}$ and let $F_i(S) = \{f(P) \mid P \text{ finite subset of } S \text{ and } |P| \leq i\}$. We consider $S \subseteq \mathbb{R}^n$.

**Lemma 4.** *Let $S \subseteq \mathbb{R}^n$. Then, $F(S) = F_n(S)$ and $F_n(S) \subseteq F_2^{n-1}(S)$.*

**Iteration of a construction $\gamma$.** We will present a construction $\gamma$ with the following properties: input to the construction is a finite set $Y$ of points, and the output $\gamma(Y)$ satisfies the following properties

1. **(Condition C1).** $\gamma(Y)$ is finite and subset of $\mathsf{conv}(Y)$.
2. **(Condition C2).** $F_2(\mathsf{conv}(Y)) \subseteq \mathsf{conv}(F(\gamma(Y)))$.

Before presenting the construction $\gamma$ we first show how to iterate the construction to obtain the following result: given a finite set of points $X$ we construct a finite set of points $X'$ such that $F(\mathsf{conv}(X)) = \mathsf{conv}(F(X'))$.

*Iterating* $\gamma$. Consider a finite set of points $X$, and let $X_0 = X$ and $X_1 = \gamma(X_0)$. Then

$$\mathsf{conv}(X_1) \subseteq \mathsf{conv}(\mathsf{conv}(X_0)) \quad \text{(since by Condition \textbf{C1} we have } X_1 \subseteq \mathsf{conv}(X_0))$$

and hence $\mathsf{conv}(X_1) \subseteq \mathsf{conv}(X_0)$; and

$$F_2(\mathsf{conv}(X_0)) \subseteq \mathsf{conv}(F(X_1)) \qquad \text{(by Condition \textbf{C2})}$$

For $i \geq 2$, let $X_i = \gamma(X_{i-1})$, and then by iteration we obtain that for $X_{n-1}$ we have

$$(1)\ \mathsf{conv}(X_{n-1}) \subseteq \mathsf{conv}(X_0) \qquad (2)\ F_2^{n-1}(\mathsf{conv}(X_0)) \subseteq \mathsf{conv}(F(X_{n-1}))$$

From (1) and (2) above, along with the aid of Lemma 4 and Lemma 3, we show the following properties:

$$(A)\ F(\mathsf{conv}(X_0)) = F_n(\mathsf{conv}(X_0)) \subseteq F_2^{n-1}(\mathsf{conv}(X_0)) \subseteq \mathsf{conv}(F(X_{n-1}))$$

$$(B)\ \mathsf{conv}(F(X_{n-1})) \subseteq \mathsf{conv}(F(\mathsf{conv}(X_{n-1}))) \subseteq F(\mathsf{conv}(X_0))$$

By (A) and (B) above we have $F(\mathsf{conv}(X_0)) = \mathsf{conv}(F(X_{n-1}))$. Thus given the finite set $X$, we have the finite set $X_{n-1}$ such that (a) $X \subseteq X_{n-1} \subseteq \mathsf{conv}(X)$ and (b) $F(\mathsf{conv}(X)) = \mathsf{conv}(F(X_{n-1}))$. We now present the construction $\gamma$ to complete the result.

**The construction $\gamma$.** Given a finite set $Y$ of points $Y' = \gamma(Y)$ is obtained by adding points to $Y$ in the following way:

– For all $1 \leq k \leq n$, we consider all $k$-dimensional coordinate planes $\Pi$ supported by a point in $Y$;
– Intersect each coordinate plane $\Pi$ with $\mathsf{conv}(Y)$ and the result is a convex polytope $Y_\Pi$;
– We add the corners (or extreme points) of each polytope $Y_\Pi$ to $Y$.

The proof that the above construction satisfies condition **C1** and **C2** is given in the fuller version [3], and thus we have the following result.

**Theorem 2.** *Given a finite set $S \subseteq \mathbb{R}^n$ such that $|S| = m$, the following assertion holds: a finite set $S'$ with $|S'| \leq m^{2^n} \cdot 2^{n^2+n}$ can be computed in $m^{O(n \cdot 2^n)} \cdot 2^{O(n^3)}$ time such that (a) $S \subseteq S' \subseteq \mathsf{conv}(S)$ and (b) $F_{\min}(\mathsf{conv}(S)) = \mathsf{conv}(F_{\min}(S'))$.*

### 4.2 Linear constraint construction

In the previous section we presented an explicit construction of a finite set of points whose convex hull gives us $F_{\min}(\mathsf{conv}(S))$. The explicit construction shows interesting properties of the set $F_{\min}(\mathsf{conv}(S))$, however, the construction is inefficient computationally. In this subsection we present an efficient geometric construction for the computation of $F_{\min}(\mathsf{conv}(S))$ for a finite set $S$. Instead of constructing a finite set $S' \subseteq \mathsf{conv}(S)$ such that $\mathsf{conv}(S') = F_{\min}(\mathsf{conv}(S))$, we represent $F_{\min}(\mathsf{conv}(S))$ as a finite set of linear constraints.

Consider the *positive orthant* anchored at the origin in $\mathbb{R}^n$, that is, the set of points with non-negative coordinates: $\mathbb{R}^n_+ = \{(z_1, z_2, \ldots, z_n) \mid z_i \geq 0, \forall i\}$. Similarly, the *negative orthant* is the set of points with non-positive coordinates, denoted as $\mathbb{R}^n_- = -\mathbb{R}^n_+$. Using vector addition, we write $y + \mathbb{R}^n_+$ for the positive orthant anchored at $y$. Similarly, we write $x + \mathbb{R}^n_- = x - \mathbb{R}^n_+$ for the negative orthant anchored at $x$. The positive and negative orthants satisfy the following simple *duality relation*: $x \in y + \mathbb{R}^n_+$ iff $y \in x - \mathbb{R}^n_+$.

Note that $\mathbb{R}^n_+$ is an $n$-dimensional convex polyhedron. For each $1 \leq j \leq n$, we consider the $(n-1)$-dimensional face $\mathbb{L}_j$ spanned by the coordinate axes except the $j^{\text{th}}$ one, that is, $\mathbb{L}_j = \{(z_1, z_2, \ldots, z_n) \in \mathbb{R}^n_+ \mid z_j = 0\}$.

We say that $y + \mathbb{R}^n_+$ is *supported* by $X$ if $(y + \mathbb{L}_j) \cap X \neq \varnothing$ for every $1 \leq j \leq n$. Assuming $y + \mathbb{R}^n_+$ is supported by $X$, we can construct a set $Y \subseteq X$ by collecting one point per $(n-1)$-dimensional face of the orthant and get $y = f(Y)$. It is also allowed that two faces contribute the same point to $Y$. Similarly, if $y = f(Y)$ for a subset $Y \subseteq X$, then the positive orthant anchored at $y$ is supported by $X$. Hence, we get the following lemma.

**Lemma 5 (Orthant Lemma).** *$y \in F_{\min}(X)$ iff $y + \mathbb{R}^n_+$ is supported by $X$.*

*Construction.* We use the Orthant Lemma to construct $F_{\min}(X)$. We begin by describing the set of points $y$ for which the $j^{\text{th}}$ face of the positive orthant anchored at $y$ has a non-empty intersection with $X$. Define $F_j = X - \mathbb{L}_j$, the set of points of the form $x - z$, where $x \in X$ and $z \in \mathbb{L}_j$.

**Lemma 6 (Face Lemma).** *$(y + \mathbb{L}_j) \cap X \neq \varnothing$ iff $y \in F_j$.*

*Proof.* Let $x \in X$ be a point in the intersection, that is, $x \in y + \mathbb{L}_j$. Using the duality relation for the $(n-1)$-dimensional orthant, we get $y \in x - \mathbb{L}_j$. By definition, $x - \mathbb{L}_j$ is a subset of $X - \mathbb{L}_j$, and hence $y \in F_j$. □

It is now easy to describe the set defined in our problem statement.

**Lemma 7 (Characterization).** $F_{\min}(X) = \bigcap_{j=1}^{n} F_j$.

*Proof.* By the Orthant Lemma, $y \in F_{\min}(X)$ iff $y + \mathbb{R}_+^n$ is supported by $X$. Equivalently, $(y + \mathbb{L}_j) \cap X \neq \varnothing$ for all $1 \leq j \leq n$. By the Face Lemma, this is equivalent to $y$ belonging to the common intersection of the sets $F_j = X - \mathbb{L}_j$. □

**Algorithm for computation of $F_{\min}(\mathsf{conv}(S))$.** Following the construction, we get an algorithm that computes $F_{\min}(\mathsf{conv}(S))$ for a finite set $S$ of points in $\mathbb{R}^n$. Let $|S| = m$. We first represent $X = \mathsf{conv}(S)$ as intersection of half-spaces: we require at most $m^n$ half-spaces (linear constraints). It follows that $F_j = X - \mathbb{L}_j$ can be expressed as $m^n$ linear constraints, and hence $F_{\min}(X) = \bigcap_{j=1}^{n} F_j$ can be expressed as $n \cdot m^n$ linear constraints. This gives us the following result.

**Theorem 3.** *Given a finite set $S$ of $m$ points in $\mathbb{R}^n$, we can construct in $O(n \cdot m^n)$ time $n \cdot m^n$ linear constraints that represent $F_{\min}(\mathsf{conv}(S))$.*

## 5 Mean-Payoff Automaton Expressions are Decidable

Several problems on quantitative languages can be solved for the class of mean-payoff automaton expressions using the vector set. The decision problems of quantitative emptiness and universality, and quantitative language inclusion and equivalence are all decidable, as well as questions related to cut-point languages, and computing distance between mean-payoff languages.

*Decision problems and distance.* From the vector set $V_E = \{\langle L_{A_1}(w), \dots, L_{A_n}(w)\rangle \in \mathbb{R}^n \mid w \in \Sigma^\omega\}$, we can compute the *value set* $L_E(\Sigma^\omega) = \{L_E(w) \mid w \in \Sigma^\omega\}$ of values of words according to the quantitative language of $E$ as follows. The set $L_E(\Sigma^\omega)$ is obtained by successive application of *min*-, *max*- and *sum-projections* $p_{ij}^{\min}, p_{ij}^{\max}, p_{ij}^{\mathrm{sum}} : \mathbb{R}^k \to \mathbb{R}^{k-1}$ where $i < j \leq k$, defined by

$$p_{ij}^{\min}((x_1, \dots, x_k)) = (x_1, \dots, x_{i-1}, \min(x_i, x_j), x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots x_k),$$
$$p_{ij}^{\mathrm{sum}}((x_1, \dots, x_k)) = (x_1, \dots, x_{i-1}, \quad x_i + x_j \quad, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots x_k),$$

and analogously for $p_{ij}^{\max}$. For example, $p_{12}^{\max}(p_{23}^{\min}(V_E))$ gives the set $L_E(\Sigma^\omega)$ of word values of the mean-payoff automaton expression $E = \max(A_1, \min(A_2, A_3))$.

Assuming a representation of the polytopes of $V_E$ as a boolean combination $\varphi_E$ of linear constraints, the projection $p_{ij}^{\min}(V_E)$ is represented by the formula

$$\psi = (\exists x_j : \varphi_E \wedge x_i \leq x_j) \vee (\exists x_i : \varphi_E \wedge x_j \leq x_i)[x_j \leftarrow x_i]$$

where $[x \leftarrow e]$ is a substitution that replaces every occurrence of $x$ by the expression $e$. Since linear constraints over the reals admit effective elimination of existential quantification, the formula $\psi$ can be transformed into an equivalent boolean combination of linear constraints without existential quantification. The same applies to max- and sum-projections.

Successive applications of min-, max- and sum-projections (following the structure of the mean-payoff automaton expression $E$) gives the value set $L_E(\Sigma^\omega) \subseteq \mathbb{R}$ as a boolean combination of linear constraints, hence it is a union of intervals. From this set, it is easy to decide the quantitative emptiness problem and the quantitative universality problem: there exists a word $w \in \Sigma^\omega$ such that $L_E(w) \geq \nu$ if and only if $L_E(\Sigma^\omega) \cap [\nu, +\infty[ \neq \varnothing$, and $L_E(w) \geq \nu$ for all words $w \in \Sigma^\omega$ if and only if $L_E(\Sigma^\omega) \cap ]-\infty, \nu[ = \varnothing$.

In the same way, we can decide the quantitative language inclusion problem "is $L_E(w) \leq L_F(w)$ for all words $w \in \Sigma^\omega$ ?" by a reduction to the universality problem for the expression $F - E$ and threshold $0$ since mean-payoff automaton expressions are closed under sum and complement. The quantitative language equivalence problem is then obviously also decidable.

Finally, the distance between the quantitative languages of $E$ and $F$ can be computed as the largest number (in absolute value) in the value set of $F - E$. As a corollary, this distance is always a rational number.

*Comparison with [1].* The work in [1] considers deterministic mean-payoff automata with multiple payoffs. The weight function in such an automaton is of the form $\mathsf{wt} : \delta \to \mathbb{Q}^d$. The value of a finite sequence $(v_i)_{1 \leq i \leq n}$ (where $v_i \in \mathbb{Q}^d$) is the mean of the tuples $v_i$, that is a $d$-dimensional vector $\mathsf{Avg}_n = \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i$. The "value" associated to an infinite run (and thus also to the corresponding word, since the automaton is deterministic) is the set $Acc \subseteq \mathbb{R}^d$ of accumulation points of the sequence $(\mathsf{Avg}_n)_{n \geq 1}$.

In [1], a query language on the set of accumulation points is used to define *multi-threshold mean-payoff languages*. For $1 \leq i \leq n$, let $p_i : \mathbb{R}^n \to \mathbb{R}$ be the usual projection along the $i^{\text{th}}$ coordinate. A query is a boolean combination of atomic threshold conditions of the form $\min(p_i(Acc)) \sim \nu$ or $\max(p_i(Acc)) \sim \nu$ where $\sim \in \{<, \leq, \geq, >\}$ and $\nu \in \mathbb{Q}$. A word is accepted if the set of accumulation points of its (unique) run satisfies the query. Emptiness is decidable for such multi-threshold mean-payoff languages, by an argument based on the computation of the convex hull of the vector values of the simple cycles in the automaton [1] (see also Lemma 1). We have shown that this convex hull $\mathsf{conv}(S_E)$ is not sufficient to analyze quantitative languages of mean-payoff automaton expressions. It turns out that a richer query language can also be defined using our construction of $F_{\min}(\mathsf{conv}(S_E))$.

In our setting, we can view a $d$-dimensional mean-payoff automaton $A$ as a product $P_A$ of 2d copies $A_t^i$ of $A$ (where $1 \leq i \leq d$ and $t \in \{\mathsf{LimInfAvg}, \mathsf{LimSupAvg}\}$), where $A_t^i$ assigns to each transition the $i^{\text{th}}$ coordinate of the payoff vector in $A$, and the automaton is interpreted as a $t$-automaton. Intuitively, the set $Acc$ of accumulation points of a word $w$ satisfies $\min(p_i(Acc)) \sim \nu$ (resp. $\max(p_i(Acc) \sim \nu)$ if and only if the value of $w$ according to the automaton $A_t^i$ for $t = \mathsf{LimInfAvg}$ (resp. $t = \mathsf{LimSupAvg}$) is $\sim \nu$. Therefore, atomic threshold conditions can be encoded as threshold conditions on single variables of the vector set for $P_A$. Therefore, the vector set computed in Section 4 allows to decide the emptiness problem for multi-threshold mean-payoff languages, by checking emptiness of the intersection of the vector set with the constraint corresponding to the query.

Furthermore, we can solve more expressive queries in our framework, namely where atomic conditions are linear constraints on $\mathsf{LimInfAvg}$- and $\mathsf{LimSupAvg}$-values. For example, the constraint $\mathsf{LimInfAvg}(\mathsf{wt}_1) + \mathsf{LimSupAvg}(\mathsf{wt}_2) \sim \nu$ is simply encoded as $x_k + x_l \sim \nu$ where $k, l$ are the indices corresponding to $A^1_{\mathsf{LimInfAvg}}$ and $A^2_{\mathsf{LimSupAvg}}$ respectively. Note that the trick of extending the dimension of the $d$-payoff vector with, say $\mathsf{wt}_{d+1} = \mathsf{wt}_1 + \mathsf{wt}_2$, is not equivalent because $\mathsf{Lim}\{^{\mathsf{Sup}}_{\mathsf{Inf}}\}\mathsf{Avg}(\mathsf{wt}_1) \pm \mathsf{Lim}\{^{\mathsf{Sup}}_{\mathsf{Inf}}\}\mathsf{Avg}(\mathsf{wt}_2)$ is not equal to $\mathsf{Lim}\{^{\mathsf{Sup}}_{\mathsf{Inf}}\}\mathsf{Avg}(\mathsf{wt}_1 \pm \mathsf{wt}_2)$ in general (no matter the choice of $\{^{\mathsf{Sup}}_{\mathsf{Inf}}\}$ and $\pm$). Hence, in the context of non-quantitative languages our results also provide a richer query language for the deterministic mean-payoff automata with multiple payoffs.

*Complexity.* All problems studied in this section can be solved easily (in polynomial time) once the value set is constructed, which can be done in quadruple exponential time. The quadruple exponential blow-up is caused by $(a)$ the synchronized product construction for $E$, $(b)$ the computation of the vector values of all simple cycles in $A_E$, $(c)$ the construction of the vector set $F_{\min}(\mathsf{conv}(S_E))$, and $(d)$ the successive projections of the vector set to obtain the value set. Therefore, all the above problems can be solved in 4EXPTIME.

**Theorem 4.** *For the class of mean-payoff automaton expressions, the quantitative emptiness, universality, language inclusion, and equivalence problems, as well as distance computation can be solved in 4EXPTIME.*

Theorem 4 is in sharp contrast with the nondeterministic and alternating mean-payoff automata for which language inclusion is undecidable (see also Table 1). The following theorem presents the undecidability result that is derived from the results of [10].

**Theorem 5.** *The quantitative universality, language inclusion, and language equivalence problems are undecidable for nondeterministic mean-payoff automata; and the quantitative emptiness, universality, language inclusion, and language equivalence problems are undecidable for alternating mean-payoff automata.*

## 6 Expressive Power and Cut-point Languages

We study the expressive power of mean-payoff automaton expressions $(i)$ according to the class of quantitative languages that they defi ne, and $(ii)$ according to their cut-point languages.

*Expressive power comparison.* We compare the expressive power of mean-payoff automaton expressions with nondeterministic and alternating mean-payoff automata. The results of [6] show that there exist deterministic mean-payoff automata $A_1$ and $A_2$ such that $\min(A_1, A_2)$ cannot be expressed by nondeterministic mean-payoff automata. The results of [5] shows that there exists deterministic mean-payoff automata $A_1$ and $A_2$

such that $\mathrm{sum}(A_1, A_2)$ cannot be expressed by alternating mean-payoff automata. It follows that there exist languages expressible by mean-payoff automaton expression that cannot be expressed by nondeterministic and alternating mean-payoff automata. In Theorem 6 we show the converse, that is, we show that there exist languages expressible by nondeterministic mean-payoff automata that cannot be expressed by mean-payoff automaton expression. It may be noted that the subclass of mean-payoff automaton expressions that only uses min and max operators (and no sum operator) is a strict subclass of alternating mean-payoff automata, and when only the max operator is used we get a strict subclass of the nondeterministic mean-payoff automata.

**Theorem 6.** *Mean-payoff automaton expressions are incomparable in expressive power with nondeterministic and alternating mean-payoff automata: (a) there exists a quantitative language that is expressible by mean-payoff automaton expressions, but cannot be expressed by alternating mean-payoff automata; and (b) there exists a quantitative language that is expressible by a nondeterministic mean-payoff automaton, but cannot be expressed by a mean-payoff automaton expression.*

*Cut-point languages.* Let $L$ be a quantitative language over $\Sigma$. Given a threshold $\eta \in \mathbb{R}$, the *cut-point language* defined by $(L, \eta)$ is the language (i.e., the set of words) $L^{\geq\eta} = \{w \in \Sigma^\omega \mid L(w) \geq \eta\}$. It is known for deterministic mean-payoff automata that the cut-point language may not be $\omega$-regular, while it is $\omega$-regular if the threshold $\eta$ is *isolated*, i.e. if there exists $\epsilon > 0$ such that $|L(w) - \eta| > \epsilon$ for all words $w \in \Sigma^\omega$ [6].

We present the following results about cut-point languages of mean-payoff automaton expressions. First, we note that it is decidable whether a rational threshold $\eta$ is an isolated cut-point of a mean-payoff automaton expression, using the value set (it suffices to check that $\eta$ is not in the value set since this set is closed). Second, isolated cut-point languages of mean-payoff automaton expressions are *robust* as they remain unchanged under sufficiently small perturbations of the transition weights. This result follows from a more general robustness property of weighted automata [6] that extends to mean-payoff automaton expressions: if the weights in the automata occurring in $E$ are changed by at most $\epsilon$, then the value of every word changes by at most $\max(k, 1) \cdot \epsilon$ where $k$ is the number of occurrences of the $\mathrm{sum}$ operator in $E$. Therefore $D_{\sup}(L_E, L_{F^\epsilon}) \to 0$ when $\epsilon \to 0$ where $F^\epsilon$ is any mean-payoff automaton expression obtained from $E$ by changing the weights by at most $\epsilon$. As a consequence, isolated cut-point languages of mean-payoff automaton expressions are robust. Third, the isolated cut-point language of mean-payoff automaton expressions is $\omega$-regular. To see this, note that every strongly connected component of the product automaton $A_E$ contributes with a closed convex set to the value set of $E$. Since the $\max$-, $\min$- and $\mathrm{sum}$-projections are continuous functions, they preserve connectedness of sets and therefore each scc $C$ contributes with an interval $[m_C, M_C]$ to the value set of $E$. An isolated cut-point $\eta$ cannot belong to any of these intervals, and therefore we obtain a Büchi-automaton for the cut-point language by declaring to be accepting the states of the product automaton $A_E$ that belong to an scc $C$ such that $m_C > \eta$. Hence, we get the following result.

**Theorem 7.** *Let $L$ be the quantitative language of a mean-payoff automaton expression. If $\eta$ is an isolated cut-point of $L$, then the cut-point language $L^{\geq\eta}$ is $\omega$-regular.*

## 7 Conclusion and Future Works

We have presented a new class of quantitative languages, the *mean-payoff automaton expressions* which are both robust and decidable (see Table 1), and for which the distance between quantitative languages can be computed. The decidability results come with a high worst-case complexity, and it is a natural question for future works to either improve the algorithmic solution, or present a matching lower bound. Another question of interest is to find a robust and decidable class of quantitative languages based on the discounted sum measure [4].

## References


1. R. Alur, A. Degorre, O. Maler, and G. Weiss. On omega-languages defined by mean-payoff conditions. In *Proc. of FOSSACS*, LNCS 5504, pages 333–347. Springer, 2009.
2. M. Bojanczyk. Beyond omega-regular languages. In *Proc. of STACS*, LIPIcs 3. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2010.
3. K. Chatterjee, L. Doyen, H. Edelsbrunner, T.A. Henzinger, and P. Rannou. Mean-payoff automaton expressions. *CoRR*, abs/1006.1492, 2010.
4. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proc. of CSL*, LNCS 5213, pages 385–400. Springer, 2008.
5. K. Chatterjee, L. Doyen, and T. A. Henzinger. Alternating weighted automata. In *Proc. of FCT*, LNCS 5699, pages 3–13. Springer, 2009.
6. K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages. In *Proc. of LICS*, pages 199–208. IEEE, 2009.
7. K. Chatterjee, A. Ghosal, T. A. Henzinger, D. Iercan, C. Kirsch, C. Pinello, and A. Sangiovanni-Vincentelli. Logical reliability of interacting real-time tasks. In *Proc. of DATE*, pages 909–914. ACM, 2008.
8. L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proc. of LICS*, pages 454–465. IEEE, 1998.
9. L. de Alfaro, R. Majumdar, V. Raman, and M. Stoelinga. Game relations and metrics. In *Proc. of LICS*, pages 99–108. IEEE, 2007.
10. A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruńczyk. Energy and mean-payoff games with imperfect information. In *Proc. of CSL*, LNCS. Springer, 2010. To appear.
11. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled markov systems. In *Proc. of CONCUR*, LNCS 1664, pages 258–273. Springer, 1999.
12. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007.
13. M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. Springer, 2009.
14. M. Droste and D. Kuske. Skew and infinitary formal power series. In *Proc. of ICALP*, LNCS 2719, pages 426–438. Springer, 2003.
15. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
16. O. Kupferman and Y. Lustig. Lattice automata. In *Proc. of VMCAI*, LNCS 4349, pages 199–213. Springer, 2007.
17. M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.
18. M. P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.
19. E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco. Probabilistic finite-state machines-part I. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1013–1025, 2005.
20. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.