3-2009

On Omega-Languages Defined by Mean-Payoff Conditions

Rajeev Alur
University of Pennsylvania, alur@cis.upenn.edu

Aldric Degorre
University of Grenoble

Oded Maler
University of Grenoble

Gera Weiss
University of Pennsylvania, gera@seas.upenn.edu

Recommended Citation

Rajeev Alur, Aldric Degorre, Oded Maler, and Gera Weiss, "On Omega-Languages Defined by Mean-Payoff Conditions", *Lecture Notes in Computer Science: Foundations of Software Science and Computational Structures* 5504, 333-347. March 2009. http://dx.doi.org/10.1007/978-3-642-00596-1_24

From the 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009.

On Omega-Languages Defined by Mean-Payoff Conditions

Abstract

In quantitative verification, system states/transitions have associated payoffs, and these are used to associate mean-payoffs with infinite behaviors. In this paper, we propose to define ω-languages via Boolean queries over mean-payoffs. Requirements concerning averages such as "the number of messages lost is negligible" are not ω-regular, but specifiable in our framework. We show that, for closure under intersection, one needs to consider multi-dimensional payoffs. We argue that the acceptance condition needs to examine the set of accumulation points of sequences of mean-payoffs of prefixes, and give a precise characterization of such sets. We propose the class of multi-threshold mean-payoff languages using acceptance conditions that are Boolean combinations of inequalities comparing the minimal or maximal accumulation point along some coordinate with a constant threshold. For this class of languages, we study expressiveness, closure properties, analyzability, and Borel complexity.

Disciplines

Computer Sciences

Comments

From the 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009.

On Omega-Languages Defined by Mean-Payoff Conditions

Rajeev Alur[1], Aldric Degorre[2], Oded Maler[2], Gera Weiss[1]

[1] Dept. of Computer and Information Science, University of Pennsylvania, USA
{alur, gera}@cis.upenn.edu
[2] CNRS - Verimag, University of Grenoble, France
{aldric.degorre, oded.maler}@imag.fr

Abstract. In *quantitative* verification, system states/transitions have associated payoffs, and these are used to associate *mean-payoffs* with infinite behaviors. In this paper, we propose to define ω-languages via Boolean queries over mean-payoffs. Requirements concerning averages such as "the number of messages lost is negligible" are not ω-regular, but specifiable in our framework. We show that, for closure under intersection, one needs to consider multi-dimensional payoffs. We argue that the acceptance condition needs to examine the set of *accumulation points* of sequences of mean-payoffs of prefixes, and give a precise characterization of such sets. We propose the class of *multi-threshold mean-payoff languages* using acceptance conditions that are Boolean combinations of inequalities comparing the minimal or maximal accumulation point along some coordinate with a constant threshold. For this class of languages, we study expressiveness, closure properties, analyzability, and Borel complexity.

1 Introduction

In algorithmic verification of reactive systems, the system is modeled as a finite-state transition system (possibly with fairness constraints), and requirements are captured as languages of infinite words over system observations [8, 9]. The most commonly used framework for requirements is the class of ω-regular languages. This class is expressive enough to capture many natural requirements, and has well-understood and appealing theoretical properties: it is closed under Boolean operations, it is definable by finite automata (such as deterministic parity automata or nondeterministic Büchi automata), it contains Linear Temporal Logic LTL, and decision problems such as emptiness, language inclusion are decidable [10, 11].

The classical verification framework only captures *qualitative* aspects of system behavior, and in order to describe *quantitative* aspects, for example, consumption of resources such as CPU and energy, a variety of extensions of system models, logics, and automata have been proposed and studied in recent years [1,3,5,7]. The best known, and the most relevant to our work, approach is as follows: a *payoff* (or a cost) is associated with each state (or transition) of the model, the *mean-payoff* of a finite run is simply the average of the payoffs of all the states in the run, and the mean-payoff of an infinite run is the limit, as n goes to infinity, of the mean-payoff of the prefix of length n. The notion of mean-payoff objectives was first studied in classical game theory, and more

recently in verification literature [5, 6, 12]. Most of this work is focused on computing the optimal mean-payoff value, typically in the setting of two-player games, and the fascinating connections between the mean-payoff and parity games.

In this paper, we propose and study ways of defining languages of infinite words based on the mean-payoffs. As a motivating example, suppose 1 denotes the condition that "message is delivered" and 0 denotes the condition that "message is lost." A behavior of the network is an infinite sequence over $\{0, 1\}$. Requirements such as "no message is ever lost" (always 1), "only finitely many messages are lost" (eventually-always 1), and "infinitely many messages are delivered" (infinitely-often 1), are all ω-regular languages. However, the natural requirement that "the number of lost messages is negligible" is not ω-regular. Such a requirement can be formally captured if we can refer to averages. For this purpose, we can associate a payoff with each symbol, payoff 0 with message lost and payoff 1 with message delivered, and require that mean-payoff of every infinite behavior is 1. As this example indicates, using mean-payoffs to define acceptance conditions can express meaningful, non-regular, and yet analyzable, requirements.

The central technical question for this paper is to define a precise query language for mapping mean-payoffs of infinite runs into Boolean answers so that the resulting class of ω-languages has desirable properties concerning closure, expressiveness, and analyzability. The obvious candidate for turning mean-payoffs into acceptance criteria is threshold queries of the form "is mean-payoff above (or below) a given threshold t". Indeed, this is implicitly the choice in the existing literature on decision problems related to mean-payoff models [5, 6, 12]. A closer investigation indicates that this is not a satisfactory choice for queries.

First, closure under intersection requires that we should be able to model multiple payoff functions. For this purpose, we define *d-payoff automata*, where d is the dimension of the payoffs, and each edge is annotated with a d-dimensional vector of payoffs. We prove that expressiveness strictly increases with the dimension. From the applications point of view, multi-payoffs allow to model requirements involving multiple quantities. Because we allow unbounded dimensions, one can also add coordinates that model weighted sums of the quantities, and put bounds on these coordinates too.

Second, the limit of the mean-payoffs of prefixes of an infinite run may not exist. This leads us to consider the set of *accumulation points* corresponding to a run. For single-dimensional payoffs, the set of these points is an interval. For multi-dimensional payoffs, we are not aware of existing work on understanding the structure of accumulation points. We establish a precise characterization of the structure of accumulation points: a set can be a set of accumulation points of a run of a payoff automaton if and only if it is closed, bounded, and connected.

Third, if we use mp to refer to the mean-payoff of a run, and consider four types of queries of the form $mp < t$, $mp \leq t$, $mp > t$, and $mp \geq t$, where t is a constant, we prove that the resulting four classes of ω-languages have incomparable expressiveness. Consequently acceptance condition needs to support combination of all such queries.

After establishing a number of properties of the accumulation points of multi-dimensional payoff automata, we propose the class of *multi-threshold mean-payoff languages*. For this class, the acceptance condition is a Boolean combination of constraints

of the form "is there an accumulation point whose ith projection is less than a given threshold t". We show that the expressive power of this class is incomparable to that of the class of ω-regular languages, that this class is closed under Boolean operations and has decidable emptiness problem. We also study its Borel complexity.

2 Definitions

2.1 Multi-payoff automata

Multi-payoff automata are defined as automata with labels, called payoffs, attached to transitions. In this paper, payoffs are vectors in a finite dimensional Euclidean space.

Definition 1 (d-Payoff automaton) *A d-payoff automaton, with $d \in \mathbb{N}$, is a tuple $\langle A, Q, q_0, \delta, w\rangle$ where A and Q are finite sets, representing the alphabet and states of the automaton, respectively; $q_0 \in Q$ is an initial state; $\delta \in Q \times A \to Q$ is a total transition function (also considered as a set of transitions $(q, a, \delta(q, a))$) and $w\colon \delta \to \mathbb{R}^d$ is a function that maps each transition to a d-dimensional vector, called payoff.*

Note that we consider only deterministic complete automata.

Definition 2 *The following notions are defined for payoff automata:*

- *A finite* run *of an automaton is a sequence of transitions of the following type: $(q_1, a_1, q_2)(q_2, a_2, q_3)\dots(q_i, a_i, q_{i+1})$. An infinite run is an infinite sequence of transitions such that any prefix is a finite run.*
- *We denote by $\lambda(r)$ the word of the symbols labelling the successive transitions of the run r, i.e. $\lambda((q_1, a_1, q_2)\cdots(q_n, a_n, q_{n+1})) = a_1 \cdots a_n$.*
- *A run is* initial *if $q_1 = q_0$.*
- *By $\mathrm{run}_{\mathcal{A}}(u)$ we denote the initial run r in $\mathcal{A}$ such that $u = \lambda(r)$*
- *A* cycle *is a run $(q_1, a_1, q_2)(q_2, a_2, q_3)\dots(q_i, a_i, q_{i+1})$ such that $q_1 = q_{i+1}$. A cycle is* simple *if no proper subsequence is a cycle.*
- *For a word or run u, $u\restriction n$ denotes the prefix of length n of u, and $u[n]$ the nth element of u.*
- *The payoff of a finite run r is* $\mathrm{payoff}(r) = \sum_{i=1}^{|r|} w(r[i])$.
- *The mean-payoff of a run r is* $\mathrm{mp}(r) = \mathrm{payoff}(r)/|r|$.
- *A subset of the states of an automaton is* strongly connected *if, for any two elements of that subset, there is a path from one to the other.*
- *A* strongly connected component *(SCC) is a strongly connected subset that is not contained in any other strongly connected subset.*
- *A SCC is* terminal *if it is reachable and there is no path from the SCC to any other SCC.*

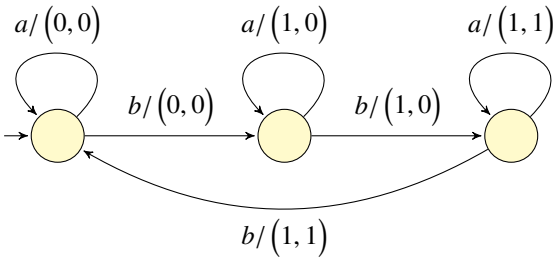
2.2 Acceptance

In the literature, the mean-payoff value of a run is generally associated to the "limit" of the averages of the prefixes of the run. As that limit does not always exist, standard definitions only consider the lim inf of that sequence (or sometimes lim sup) and, more specifically, threshold conditions comparing those quantities with fixed constants [2, 4, 5, 12]. As that choice is arbitrary, and more can be said about the properties of that sequence than the properties of just its lim inf or even lim sup, in particular when $d > 1$, we choose to consider the entire set of accumulation points of that sequence.

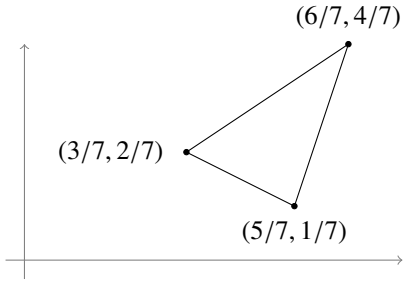
A point x is an accumulation point of the sequence $x_0, x_1, \ldots$ if, for every open set containing x, there are infinitely many indices such that the corresponding elements of the sequence belong to the open set.

Definition 3 *We denote by* $\mathrm{Acc}(x_n)_{n=1}^{\infty}$ *the set of accumulation points of the sequence* $(x_n)_{n=1}^{\infty}$. *If r is a run of a d-payoff automaton $\mathcal{A}$,* $\mathrm{Acc}_{\mathcal{A}}(r) = \mathrm{Acc}(\mathrm{mp}(r\restriction n))_{n=1}^{\infty}$, *and for a word w,* $\mathrm{Acc}_{\mathcal{A}}(w) = \mathrm{Acc}_{\mathcal{A}}(\mathrm{run}(w))$.

Example 1. Consider the 2-payoff automaton



where edges are annotated with expression of the form σ/v meaning that the symbol σ triggers a transition whose payoff is v. Let $w = \prod_{i=0}^{\infty} a^{2^i-1}b$ be an infinite word where b's are isolated by sequences of a's with exponentially increasing lengths. The set $\mathrm{Acc}_{\mathcal{A}}(w)$ is the triangle



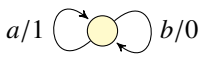
as we show next. By direct calculation we get that $\lim_{n\to\infty} \mathrm{mp}(w\restriction \sum_{i=0}^{3n} 2^i) = (6/7, 4/7)$, $\lim_{n\to\infty} \mathrm{mp}(w\restriction \sum_{i=0}^{3n+1} 2^i) = (3/7, 2/7)$, and $\lim_{n\to\infty} \mathrm{mp}(w\restriction \sum_{i=0}^{3n+2} 2^i) = (5/7, 1/7)$. Furthermore, for every $n \in \mathbb{N}$, $j \in \{0, 1, 2\}$ and $k \in \{0, \ldots, 2^{3n+j+1}\}$, the vector $\mathrm{mp}(w\restriction k + \sum_{i=0}^{3n+j} 2^i)$ is in the convex hull of $\mathrm{mp}(w\restriction \sum_{i=0}^{3n+j} 2^i)$ and $\mathrm{mp}(w\restriction \sum_{i=0}^{3n+j+1} 2^i)$ and the

maximal distance between points visited on this line goes to zero as n goes to infinity. Together, we get that the points to which the mean-payoff gets arbitrarily close are exactly the points on the boundary of the above triangle. Similarly, if we choose the word $w' = \prod_{i=0}^{\infty} a^{3^i-1}b$, we get that $\mathrm{Acc}_{\mathcal{A}}(w')$ is the boundary of the triangle $(4/13, 3/13), (10/13, 1/13), (12/13, 9/13)$. □

We say that a word or run is *convergent*, whenever its set of accumulation points is a singleton, i.e. when its sequence of mean payoffs converges. For instance, periodic runs are convergent because the mean-payoffs of the prefixes $r{\restriction}n$ of an infinite run $r = r_1 r_2^{\omega}$ converge to the mean-payoff of the finite run r_2, when n goes to infinity.

Definition 4 *An infinite run r is accepted by a d-payoff automaton $\mathcal{A}$ with condition F, where F is a predicate on $2^{\mathbb{R}^d}$, if and only if $F(\mathrm{Acc}_{\mathcal{A}}(r))$. An infinite word u is accepted if and only if* run(u) *is accepted. We denote by $L(\mathcal{A}, F)$ the language of words accepted by $\mathcal{A}$ with condition F. In the following, we call* mean-payoff language*, any language accepted by a d-payoff automaton with such a condition. If d is one and $F(S)$ is of the form* $\mathrm{extr}\, S \bowtie C$ *where* $\mathrm{extr} \in \{\inf, \sup\}$*, $\bowtie \in \{<, \leq, >, \geq\}$, and C is a real constant; we say that F is a threshold condition.*

Example 2. For the 1-payoff automaton



let the acceptance condition $F(S)$ be true iff $S = \{0\}$. This defines the language of words having negligibly many a's. □

3 Expressiveness

3.1 Comparison with ω-regular languages

Before proving specific results on the class of mean-payoff languages, we show that it is incomparable with the class of ω-regular languages. In this context, we call specification types incomparable if each type of specification can express properties that are not expressible in the other type. Incomparability of mean-payoff and ω-regular specifications is, of course, a motivation for studying mean-payoff languages.

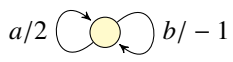
We will need the following ad-hoc pumping lemma for ω-regular languages.

Lemma 1 (Pumping lemma) *Let L be an ω-regular language. There exists $p \in \mathbb{N}$ such that, for each $w = u_1 w_1 u_2 w_2 \dots u_i w_i \dots \in L$ such that $|w_i| \geq p$ for all i, there are sequences of finite words $(x_i)_{i\in\mathbb{N}}, (y_i)_{i\in\mathbb{N}}, (z_i)_{i\in\mathbb{N}}$ such that, for all i, $w_i = x_i y_i z_i$, $|x_i y_i| \leq p$ and $|y_i| > 0$ and for every sequence of pumping factors $(j_i)_{i\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$, the pumped word $u_1 x_1 y_1^{j_1} z_1 u_2 x_2 y_2^{j_2} z_2 \dots u_i x_i y_i^{j_i} z_i \dots$ is in L.*

Proof. Similar to the proof of the pumping lemma for finite words. □

Proposition 1 *There exists a mean-payoff language, defined by a* 1*-payoff automaton and a threshold acceptance condition, that is not* ω*-regular.*

Proof. Consider the 1-payoff automaton



We show that $L = \{w | \inf \mathrm{mp}_{\mathcal{A}}(w) \leq 0\}$ is not regular. For any p, the word $w = (a^p b^{2p})^\omega$ is in that language. Assuming, towards contradiction, that the language is regular and using the pumping Lemma 1 on w, we can select as factors w_i the sequences of a an choose $j_i = 2$ to obtain a word w' that should be in L. But since $\mathrm{mp}_A(w')$ is a singleton bigger than zero, w' does not satisfy the acceptance condition and therefore is not in L, a contradiction. □

Proposition 2 *There exists an* ω*-regular language that is not a mean-payoff language.*

Proof. Let $L = (a^* b)^\omega$. We will show that, in any payoff automaton, we can find two words u_1 and u_2, u_1 having infinitely often b and u_2 having eventually only a, and such that $Acc(u_1) = Acc(u_2)$. Then obviously no general mean-payoff acceptance condition can distinguish those two words although $u_1 \in L$ and $u_2 \notin L$.

Let us construct the counter-example. Suppose $\mathcal{A}$ is a payoff automaton recognizing L with some predicate F. Let c_a be a cycle such that $\lambda(c_a)$ contains only a's and c_b a cycle such that $\lambda(c_b)$ contains at least one b, both starting in some state q in a terminal strongly connected component of $\mathcal{A}$, and let p be an initial run leading to q.

The mean-payoffs of the run $r = p \prod_{i=1}^{\infty} c_a^i c_b$, which should be accepted, converge to $\mathrm{mp}_{\mathcal{A}}(c_a)$, which is also the mean-payoff of $p c_a^\omega$, which should be rejected but has to be accepted by $\mathcal{A}$, since it has the same mean-payoff as r. □

3.2 Topology of mean-payoff accumulation points

In this section we discuss the structure of the set of accumulation points. In particular we characterize the sets that are the accumulation points of some run of a payoff automaton.

If S is a strongly connected component of an automaton, and C is the set of simple cycles in S, then we denote by ConvHull(S) the convex hull of $\{\mathrm{mp}(c) | c \in C\}$.

Theorem 1 *Let* r *be an infinite run of a* d*-payoff automaton, then* Acc(r) *is a closed, connected and bounded subset of* $\mathbb{R}^d$.

Proof.
Closed: True for any set of accumulation points: let (a_n) be a sequence in a topological space, and $(x_n) \in \mathrm{Acc}(a_n)_{n=1}^{\infty}$ be a sequence of accumulation points converging to a point x. For any x_i, we can choose a sub-sequence (a_{i_n}) converging to x_i. Now we can construct a sub-sequence of elements that converges to x: for every i, take the first element $a_{i_{f(i)}}$ of a_{i_n} which is at a distance smaller than 2^{-i} from x_i such that $f(i) > f(i-1)$. Then the sequence $(a_{i_{f(i)}})_{i \in \mathbb{N}}$ converges to x.

Bounded: As we are speaking of a sequence of averages of the (finite) set of payoffs, it is clear that the sequence of mean-payoffs remains in the convex hull of that set, which is bounded.
Connected: Proof by contradiction. Suppose there exists two disjoint open sets O_1 and O_2 such that $Acc(r) \subseteq O_1 \cup O_2$. Let d be the distance between O_1 and O_2. As those sets are open and disjoint, $d > 0$. But the vector between two successive averages is $\text{payoff}(r\restriction n)/n - \text{payoff}(r\restriction n-1)/n-1 = (1/n)(\text{payoff}(r\restriction n-1)) + w(r[n]) - n/(n-1)\,\text{payoff}(r\restriction n-1)) = (1/n)(w(r[n]) - \text{mp}(r\restriction n))$, whose norm is smaller than Δ/n, where $\Delta = \max\{|||w(t) - w(t')|||t, t' \in \delta\}$. If a run has accumulations points in both O_1 and O_2, then there exist $n > \Delta/d$ such that the nth step is in O_1 and the $(n+1)$th in O_2. The distance between those two points has to be both greater than d and smaller than Δ/n, which is not possible. □

As a remark, we can say more than boundedness: indeed a run eventually comes into a SCC it never leaves. The contribution of the payoffs of the SCC becomes then dominant as n goes to the infinity. Even better, actually, the contribution of the simple cycles of that SCC is dominant. Thus the set of accumulation points is included in the convex hull of the simple cycles of the SCC.

The following theorem is a converse to Theorem 1.

Theorem 2 *For every non-empty, closed, bounded and connected set $D \subset \mathbb{R}^d$, there is a d-payoff automaton and a run r of that automaton such that* $\text{Acc}(r) = D$.

Proof. Take any automaton with a reachable SCC such that D is contained in the convex hull of the cycles of the SCC.

For every integer $n > 0$, let $\{O_{i,n} : i = 1, \dots, l_n\}$ be a finite coverage of D by open sets of diameter smaller than $1/n$. Such a coverage exists, for example, by covering D by spheres of diameter $1/n$.

Suppose p is a finite initial run going into the SCC. For every n and every i, we can prolong p with a suffix c such that $\text{mp}(pc) \in O_{n,i}$ and pc is in the SCC (form the end of p onwards). For that, we need c to be long enough and have the right proportions of simple cycles. Furthermore, as $\text{mp}(pc\restriction l+1) - \text{mp}(pc\restriction l)$ becomes smaller when l goes to infinity, we can make the distance of $\text{mp}(pc\restriction l)$ from D converge to zero when l goes to infinity.

As the set $(O_{n,i})_{n,i\in\mathbb{N}\times\mathbb{N}}$ is countable, we can construct recursively the successive suffixes $c_{1,1}, c_{1,2}, \dots, c_{2,1}, c_{2,2}, \dots$ such that $\text{mp}(pc_{1,1}c_{1,2}\dots_{2,1}\, c_{2,2}\dots c_{n,i})$ is in $O_{n,i}$, and such that for every l, $\text{mp}(p\prod_{ji\in\mathbb{N}\times\mathbb{N}} c_{ji}\restriction l)$ is at a distance smaller than K/l from D.

Let $x \in D$. Then for every n, $x \in O_{n,i}$ for some i, thus for every n, the sequence of mean-payoffs comes within a radius $1/n$ from x, which means x is an accumulation point. Conversely, if $y \notin D$, as D is closed, it is at a distance $\delta > 0$ from D, moreover there exist a l such that $\text{mp}(pc\restriction l)$ never exits a radius $\epsilon < \delta$ around D and therefore the sequence of mean-payoff will never come in a radius $\delta - \epsilon$ from y. So y is not an accumulation point. We conclude that $\text{Acc}_{\mathcal{A}}(r)$ is exactly D. □

Actually, like for Theorem 1, a careful examination of the proof reveals that a stronger statement is true. Specifically, it is easy to verify that any closed, bounded and connected set contained in any of the SCC of an automaton is the set of accumulation points of some run of that automaton.

3.3 Comparison of threshold mean-payoff languages

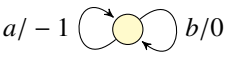
We study mean-payoff languages where the minimum and maximum of the set of accumulation points are compared with a given threshold. We assume, without loss of generality, that the threshold is zero because changing the threshold is equivalent to an affine transformation of the payoffs. We show that the different threshold languages are incomparable in expressive power.

Definition 5 *We denote by $\mathcal{L}_{\bowtie}$ the class of mean-payoff languages accepted by a* 1*-payoff automaton with the condition* $\min \mathrm{Acc}(w) \bowtie 0$*, where* $\bowtie$ *is* $<, >, \leq$ *or* $\geq$*.*

Note that these languages are the winning conditions used to define mean-payoff games, e.g. in [12], because $\min \mathrm{Acc}(w) = \liminf_{n\to\infty} \mathrm{mp}_{\mathcal{A}}(w \upharpoonright n)$. We do not need to discuss the class of languages defined as complements of these conditions because $\mathcal{L}_{>}$ is $\mathrm{co}\,\mathcal{L}_{\leq}$ and $\mathcal{L}_{\geq}$ is $\mathrm{co}\,\mathcal{L}_{<}$, where $\mathrm{co}\,\mathcal{L}_{\bowtie}$ is the set of languages defined as sets of words that do not satisfy $\min \mathrm{Acc}(w) \bowtie 0$ for some automaton.

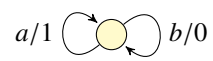
Theorem 3 *The classes $\mathcal{L}_{<}$, $\mathcal{L}_{\leq}$,$\mathcal{L}_{\geq}$ and $\mathcal{L}_{>}$ are incomparable.*

Proof. We begin by showing that $\mathcal{L}_{<}$ and $\mathcal{L}_{\leq}$ are incomparable. Consider the automaton



and the language $L = \{w | \min \mathrm{Acc}(w) < 0\}$. Suppose, towards contradiction, that there exists an automaton $\mathcal{A}'$ accepting L with a $\mathcal{L}_{\leq}$ condition. Consider c_a and c_b two cycles in $\mathcal{A}'$, labelled respectively with a word in $a(a+b)^i$ and with b^j for some integers i and j, and start from a same reachable state q (two such cycles exist at least in any terminal strongly connected component). Let p be a finite initial run ending at q. As pc_a^ω is a run of $\mathcal{A}'$ which should be accepted, it is necessary that $\mathrm{payoff}_{\mathcal{A}'}(c_a) \leq 0$, and as pc_b^ω should not be accepted, it is necessary that $\mathrm{payoff}_{\mathcal{A}'}(c_b) > 0$. For all k, the run $p(c_a c_b^k)^\omega$ should be accepted. So it is necessary that for all k, $\mathrm{payoff}_{\mathcal{A}'}(c_a c_b^k) \leq 0$. Thus $\mathrm{payoff}_{\mathcal{A}'}(c_a) + k\,\mathrm{payoff}_{\mathcal{A}'}(c_b) \leq 0$, which is possible if and only if $\mathrm{payoff}_{\mathcal{A}'}(c_b) \leq 0$ and contradicts $\mathrm{payoff}_{\mathcal{A}'}(c_b) > 0$. Thus L cannot be accepted by a $\mathcal{L}_{\leq}$ condition.

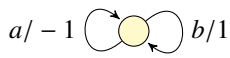
Conversely, consider the automaton



with the language L defined by the $\mathcal{L}_{\leq}$ acceptance condition. Towards contradiction, assume that $\mathcal{A}'$ is an automaton accepting L with the $\mathcal{L}_{<}$ acceptance contradiction. If c_a, c_b and p are defined in $\mathcal{A}'$, the same way as before, we should have $\mathrm{payoff}_{\mathcal{A}'}(c_a) \geq 0$ and $\mathrm{payoff}_{\mathcal{A}'}(c_b) < 0$. For all k, the run $p(c_a c_b^k)$ should be rejected, so it is necessary that $\mathrm{payoff}_{\mathcal{A}'}(c_a c_b^k) \geq 0$. Thus for all k, $\mathrm{payoff}_{\mathcal{A}'}(c_a) + k\,\mathrm{payoff}_{\mathcal{A}'}(c_b) \geq 0$, which is possible if and only if $\mathrm{payoff}_{\mathcal{A}'}(c_b) \geq 0$ and contradicts $\mathrm{payoff}_{\mathcal{A}'}(c_b) < 0$. Therefore L cannot be expressed by a $\mathcal{L}_{<}$ condition.

These counter examples can be adapted for proving that any class with strict inequality symbol is incomparable to any class with a weak inequality symbol. It remains to prove the incompatibility of $\mathcal{L}_<$ and $\mathcal{L}_>$ and that of $\mathcal{L}_\leq$ and $\mathcal{L}_\geq$.

Consider the language L defined by the automaton



(denoted by $\mathcal{A}$) with the $\mathcal{L}_\leq$ acceptance condition. Suppose, towards contradiction, that $\mathcal{A}'$ is an automaton defining the same language with a $\mathcal{L}_\geq$ condition. Choose two cycles c_a and c_b starting from two states q_a and q_b in a same terminal strongly connected component of $\mathcal{A}'$, such that c_a is labelled by a^l and c_b is labelled by b^m for some l and m, an initial run p going to q_b and two runs u_{ab} and u_{ba} going from q_a to q_b and from q_b to q_a, respectively. Because $pu_{ba}c_a^\omega$ should be accepted and pc_b^ω should be rejected, we must have $\text{payoff}_{\mathcal{A}'}(c_b) < 0$ and $\text{payoff}_{\mathcal{A}'}(c_a) \geq 0$. Consider $r = p\prod_{i\in\mathbb{N}} c_b^{2^i l}u_{ba}c_a^{2^i m}u_{ab}$. Then $\text{mp}_{\mathcal{A}}(\lambda(p\prod_{i=0}^n c_b^{2^i l}u_{ba}c_a^{2^i m}u_{ab}))$ converges to 0 as n goes to infinity, thus $\lambda(r) \in L$, and therefore r should be accepted by $\mathcal{A}'$ with the $\mathcal{L}_\geq$ condition. But $\text{mp}_{\mathcal{A}'_3}(word(p\prod_{i=0}^n c_b^{2^i l}u_{ba}c_a^{2^i m}u_{ab}c_b^{2^{n+1}l}))$ converges to a negative limit, thus r cannot be accepted by $\mathcal{A}'_3$ with the $\mathcal{L}_\geq$ condition, leading to contradiction.

Conversely, consider the language L, defined with the same automaton, but with the $\mathcal{L}_\geq$ condition. Suppose $\mathcal{A}'$ is an automaton defining the same L with the $\mathcal{L}_\leq$ condition. We can choose two cycles c_a and c_b starting from two states q_a and q_b in a same terminal strongly connected component of $\mathcal{A}'$ such that c_a is labelled by a^l and c_b is labelled by b^m for some l and m, an initial run p going to q_b and two runs u_{ab} and u_{ba} going from q_a to q_b and from q_b to q_a. Then we should have $\text{payoff}_{\mathcal{A}'}(c_b) \leq 0$ and $\text{payoff}_{\mathcal{A}'}(c_a) > 0$ (because $pu_{ba}c_a^\omega$ should be rejected and pc_b^ω should be accepted). Consider $r = p\prod_{i\in\mathbb{N}} u_{ba}c_a^{2^i m}u_{ab}c_b^{2^i l}$. Then $\text{mp}_{\mathcal{A}_3}(\lambda(p(\prod_{i=0}^n u_{ba}c_a^{2^i m}u_{ab}c_b^{2^i l})u_{ba}c_a^{2^{n+1}m}))$ converges to a negative limit as n goes to infinity, thus $\lambda(r) \notin L$, and therefore r should be rejected by $\mathcal{A}'$ with the $\mathcal{L}_\leq$ condition. But $\text{mp}_{\mathcal{A}'}(\lambda(p\prod_{i=0}^n u_{ba}c_a^{2^i m}u_{ab}c_b^{2^i l}))$ converges to 0, thus r is accepted by $\mathcal{A}'$ with the $\mathcal{L}_\leq$ condition, which contradicts the fact that it recognizes L.

We have thus established the incomparability of $\mathcal{L}_\geq$ and $\mathcal{L}_\leq$. As $\mathcal{L}_<$ and $\mathcal{L}_>$ are the classes of the complements of languages in respectively $\mathcal{L}_\geq$ and $\mathcal{L}_\leq$, it also implies the incomparability of the latter pair. □

The incompatibility of threshold classes shows how arbitrary the choice of only one of them as a standard definition would be. This suggests definition of a larger class including all of them.

3.4 Mean-payoff languages in the Borel hierarchy

For a topological space X, we denote by Σ_1^0 the set of open subsets by and Π_1^0 the set of closed subsets. The Borel hierarchy is defined inductively as the two sequences (Σ_α^0) and (Π_α^0), where $\Sigma_\alpha^0 = (\bigcup_{\beta<\alpha}\Pi_\beta^0)_\sigma$, and $\Pi_\alpha^0 = (\bigcup_{\beta<\alpha}\Sigma_\beta^0)_\delta$, where α and β are ordinals and $(\bullet)_\sigma$, $(\bullet)_\delta$ denote closures under countable intersections and unions, respectively.

We consider the standard topology over A^ω with the base $\{wA^\omega \colon w \in A^*\}$, i.e. a subset of A^ω is open if and only if it is a union of sets, each set consists of all possible continuations of a finite word.

Theorem 4 *The following facts hold:* $\mathfrak{L}_{\leq} \subset \Pi_2^0$, $\mathfrak{L}_{\leq} \not\subseteq \Sigma_2^0$, $\mathfrak{L}_{<} \subset \Sigma_3^0$ *and* $\mathfrak{L}_{<} \not\subseteq \Pi_3^0$.

Proof. – Let $L \in \mathfrak{L}_{\leq}$, then there exists $d \in \mathbb{N}$, a 1-payoff automaton $\mathcal{A}$ such that $L = \{w \in A^\omega | \min(Acc_{\mathcal{A}}(w)) \leq 0\}$. Therefore we can write L as

$$\begin{aligned} L &= \bigcap_{N \in \mathbb{N}} \{w \in A^\omega | \forall m \in \mathbb{N} \exists n > m \, \mathrm{mp}_{\mathcal{A}}(w\restriction n) < 1/N)\} \\ &= \bigcap_{N \in \mathbb{N}, m \in \mathbb{N}} \bigcup_{n > m} \{w \in A^\omega | \, \mathrm{mp}_{\mathcal{A}}(w\restriction n) < 1/N)\}. \end{aligned}$$

For any N and m the condition $\mathrm{mp}_{\mathcal{A}}(w\restriction n) < 1/N)$ is independent of the suffix past the nth symbol of w and therefore the set $\{w \in A^\omega | \, \mathrm{mp}_{\mathcal{A}}(w\restriction n) < 1/N)\}$ is clopen. We get that $\mathfrak{L}_{\leq} \in \Pi_2^0$.

– We prove $\mathfrak{L}_{>} \not\subseteq \Pi_2^0$, which is the same as $\mathfrak{L}_{\leq} \not\subseteq \Sigma_2^0$ because $\mathfrak{L}_{>} = \mathrm{co}\, \mathfrak{L}_{\leq}$. Let L be the set of words on alphabet $A = \{a, b\}$ having more than negligibly many b. We already demonstrated that $L \in \mathfrak{L}_{>}$. Suppose $L \in \Pi_2^0$. Then $L = \bigcap_{i \in \mathbb{N}} L_i A^\omega$ for some family of languages of finite words L_i. We can assume without loss of generality that the words of L_i have all length i. For all m, the word $w_m = (\prod_{j=1}^{m-1} a^{2^j} b)(a^{2^m} b)^\omega \in L$ (as it is ultimately periodic with a period where the proportion of b is not 0). For the word $w = \prod_{j=1}^{\infty} a^{2^j} b$, it means that any prefix $w\restriction i$ of length i is in L_i. This is a contradiction, because $w \notin L$.

– For the two last items of the theorem: Chatterjee exhibited in [2] a Π_3^0-hard language in $\mathfrak{L}_{\geq}$. He also established that this class is included in Π_3^0. As $\mathfrak{L}_{\geq} = \mathrm{co}\, \mathfrak{L}_{<}$, that proves what we need.

□

3.5 Dimensionality

In this section we analyze closure properties of mean-payoff languages defined by automata with a fixed dimension.

The following lemma shows that, for any d, the class of mean-payoff languages definable by d-payoff automata is not closed under intersection.

Lemma 2 *If d_1 and d_2 are two integers, then there exists L_1 and L_2, two mean-payoff languages of dimensions d_1 and d_2 such that L_1 and L_2 contain only convergent words and $L_1 \cap L_2$ is not definable as a dimension d mean-payoff language with $d < d_1 + d_2$.*

Proof. Let $A = \{a_1, \ldots, a_{d_1}\}$ and $B = \{b_1, \ldots, b_{d_2}\}$ be two disjoint alphabets. Let $\mathcal{A}_1$ be the one-state d_1-payoff automaton on alphabet $A \cup B$, such that the payoff of the transition (q_0, a_i, q_0) is 1 on the ith coordinate and 0 in the other coordinates and the payoff of the transition (q_0, b_i, q_0) is 0. And let $\mathcal{A}_2$ be the d_2-payoff one-state automaton defined similarly by swapping a and b.

Let L_i be the language defined on $\mathcal{A}_i$ by predicate F_i, testing equality with the singleton $\{l_i\}$, where l_i is in the simplex defined by the $d_i + 1$ different payoffs of the transitions of $\mathcal{A}_i$. In the proof of Theorem 2 we establish that the L_i are not empty.

Let $w \in (A + B)^\omega$, then w is in L_1 if and only if the proportion of a_i in every prefix tends to the ith coordinate of l_1, and it is in L_2 if and only if the proportion of b_i in every prefix tends to the ith coordinate of l_2.

Then for w to be in $L_1 \cap L_2$, it is necessary that the proportion of every symbols tends to either a coordinate of l_1, if that symbol is a a_i, or a coordinate of l_2, if it is a b_i.

Now suppose $L_1 \cap L_2$ is recognized by a d-payoff automaton with $d < d_1 + d_2$. Choose one terminal strongly connected component of $\mathcal{A}$ and consider for every symbol a of the alphabet a cycle labeled by a word in a^*, starting at some state q_a. Let also be p an initial run going to q_a and for every pair of symbols a, b a path u_{ab} going from q_a to q_b.

Only looking at the runs in the language $p\{u_{a_1 a} c_a^* u_{a a_1} | a \in A \cup B\}^\omega$, it is possible to converge to any proportion of the symbols of $A \cup B$, and thus have runs whose labeling word is in L. But as the payoffs are in dimension d, and the number of symbols is $d_1 + d_2 > d$, that language also contains runs converging to different symbol proportions but still having the same mean-payoff limit. Those runs are accepted by $\mathcal{A}$ but are not labeled by a word in L. □

Next, we prove that the intersection of two languages of dimensions d_1 and d_2 is a language of dimension $d_1 + d_2$. This will be proved constructively, by showing that the intersection language is the language defined on the product automaton with the "product" condition. Before going to the statement of the lemma, we need to define what those products are.

Definition 6 *If F_1 and F_2 are predicates on $2^{\mathbb{R}^{d_1}}$ and $2^{\mathbb{R}^{d_2}}$, we denote by $F_1 \Cap F_2$ the predicate on $2^{\mathbb{R}^{d_1+d_2}}$ which is true for $X \subseteq \mathbb{R}^{d_1+d_2}$ if and only if $F_1(p_1(X))$ and $F_2(p_2(X))$, where p_1 is the projection on the d_1 first coordinates and p_2 on the d_2 last.*

Definition 7 (Weighted automata product) *If $\mathcal{A}_1 = \langle A, Q_1, q_0^1, \delta_1, w_1 \rangle$ is a d_1-payoff automaton and $\mathcal{A}_2 = \langle A, Q_2, q_0^2, \delta_2, w_2 \rangle$, a d_2-payoff automaton, then we define $\mathcal{A}_1 \otimes \mathcal{A}_2 = \langle A, Q_1 \times Q_2, (q_0^1, q_0^2), \delta_{1\otimes 2}, w_{1\otimes 2} \rangle$, the product of $\mathcal{A}_1$ and $\mathcal{A}_2$, a $(d_1 + d_2)$-payoff automaton such that*

- $\delta_{1\otimes 2} = \{((q_1, q_2), a, (q_1', q_2')) | (q_1, a, q_1') \in \delta_1 \wedge (q_2, a, q_2') \in \delta_2 \wedge a \in A\}$,
- $w_{1\otimes 2} \colon \delta_{1\otimes 2} \to \mathbb{R}^{d_1+d_2}$ *is such that if* $w_1(q_1, a, q_1') = (x_1, \dots x_{d_1})$ *and* $w_2(q_2, a, q_2') = (y_1, \dots y_{d_2})$, *then* $w((q_1, q_2), a, (q_1', q_2')) = (x_1, \dots x_{d_1}, y_1, \dots y_{d_2})$.

But before we state the theorem, we need the following lemma:

Lemma 3 *If r is a run of a d-payoff automaton $\mathcal{A}$ and p is a projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$, with $d' < d$, then* $\mathrm{Acc}(p(\mathrm{mp}_{\mathcal{A}}(r\restriction n))) = p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$

Proof. Let $x' \in \mathrm{Acc}(p(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$. For any $i \in \mathbb{N}$, $p(\mathrm{mp}_{\mathcal{A}}(r\restriction n))$ eventually comes into a distance $1/i$ from x', for some index n_i. For $j > i$ $\mathrm{mp}_{\mathcal{A}}(r\restriction n_j)$ remains in $B(x', 1/i) \times K$ (where K is a compact of $\mathbb{R}^{d-d'}$), as this product is compact, it has at least one accumulation point. Thus the distance from x' to $p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$ is 0. But $\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n))$ is

a closed set and p, being a projection, is continuous, so $p(\mathrm{Acc}(\mathrm{mp}(r\restriction n)))$ is closed too, which means $x' \in p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$, and so $\mathrm{Acc}(p(\mathrm{mp}(r\restriction n))) \subseteq p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$.

Conversely, if $x' \in p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r \restriction n)))$ a sub-sequence $\mathrm{mp}_{\mathcal{A}}(r \restriction n_i)$ converges to a x such that $x' = p(x)$, and thus $p(\mathrm{mp}_{\mathcal{A}}(r \restriction n_i))$ converges to x', which means $x' \in \mathrm{Acc}(p(\mathrm{mp}(r\restriction n)))$. We conclude that $\mathrm{Acc}(p(\mathrm{mp}_{\mathcal{A}}(r\restriction n))) = p(\mathrm{Acc}(\mathrm{mp}_{\mathcal{A}}(r\restriction n)))$. □

Now we have all the needed tools, we can characterize the intersection of two mean-payoff languages as another mean-payoff language defined on an automaton whose dimension is known.

Proposition 3 *For any two d_1-payoff and d_2-payoff automata $\mathcal{A}_1$ and $\mathcal{A}_2$ and any two predicates F_1 and F_2 on respectively $2^{\mathbb{R}^{d_1}}$ and $2^{\mathbb{R}^{d_2}}$, the following equality holds: $L(\mathcal{A}_1, F_1) \cap L(\mathcal{A}_2, F_2) = L(\mathcal{A}_1 \otimes \mathcal{A}_2, F_1 \Cap F_2)$.*

Proof. Suppose $u \in A^\omega$, then the sequence of mean-payoffs of run r of u in $\mathcal{A}_1 \otimes \mathcal{A}_2$ are the projections by p_1 and p_2 of the sequence of mean-payoffs of some runs r_1 and r_2 in $\mathcal{A}_1$ and r_2 in $\mathcal{A}_2$ whose labeling is u. And conversely, if u has runs r_1 and r_2 in $\mathcal{A}_1$ and r_2 in $\mathcal{A}_2$, then it has a run r in $\mathcal{A}_1 \otimes \mathcal{A}_2$ whose sequence of mean-payoffs projects by p_1 and p_2 onto those of r_1 and r_2.

If r, r_1, and r_2 are such runs (the payoffs of r projecting on those of r_1 and r_2), then using lemma 3, we find that $\mathrm{Acc}_{\mathcal{A}}(r_1) = \mathrm{Acc}(p_1(\mathrm{mp}(r\restriction n))) = p_1(\mathrm{Acc}(\mathrm{mp}(r\restriction n)))$ and that $\mathrm{Acc}_{\mathcal{A}}(r_2) = \mathrm{Acc}(p_2(\mathrm{mp}(r\restriction n))) = p_2(\mathrm{Acc}(\mathrm{mp}(r\restriction n)))$.

But by definition $(F_1 \Cap F_2)(\mathrm{Acc}(\mathrm{mp}(r \restriction n)))$ holds iff $F_1(p_1(\mathrm{Acc}(\mathrm{mp}(r \restriction n))))$ and $F_2(p_2(\mathrm{Acc}(\mathrm{mp}(r\restriction n))))$ hold, thus it holds iff $F_1(\mathrm{Acc}_{\mathcal{A}_1}(r_1))$ and $F_2(\mathrm{Acc}_{\mathcal{A}_2}(r_2))$ hold.

From that we deduce that a word is in $L(\mathcal{A} \otimes \mathcal{A}, F_1 \Cap F_2)$ if and only if it is both in $L(\mathcal{A}_1, F_1)$ and $L(\mathcal{A}_2, F_2)$. □

4 An analyzable class of mean-payoff languages

4.1 The class of multi-threshold mean-payoff languages

As a candidate for a class of mean-payoff languages that is closed under complementation and includes all the expected standard mean-payoff language classes, we propose the following definition.

Definition 8 *A language L is a multi-threshold mean-payoff language (denoted by $L \in \mathfrak{L}_{mt}$) if it is the mean-payoff language defined on some d-payoff automaton $\mathcal{A}$, with a predicate F such that $F(S)$ is a Boolean combination of threshold conditions on $p_i(S)$ where p_i is the projection along the ith coordinate.*

Example 3. Consider the automaton given in Example 1 and the multi-threshold mean-payoff language $L = \{w | \min p_1(\mathrm{Acc}(w)) > .1 \wedge \max p_1(\mathrm{Acc}(w)) < .9 \wedge \min p_2(\mathrm{Acc}(w)) > .1 \wedge \max p_2(\mathrm{Acc}(w)) < .9\}$. For the word w, defined in Example 1, the set of accumulation points is shown to be a triangle that is contained in the box $\{x | .1 < p_1(x) < .9 \wedge .1 < p_2(x) < .9\}$ and therefore $w \in L$.

Geometrically, multi-threshold acceptance conditions can be visualized as specifying constraints on the maximal and minimal projection of $\mathrm{Acc}(w)$ on the axes. Since we can extend the payoff vectors by adding a coordinate whose values are a linear combination of the other coordinates, also threshold constraints involving minimal and maximal elements of the projection of $\mathrm{Acc}(w)$ on other lines are expressible, as shown in the following example.

Example 4. Assume that, with the automaton given in Example 1, we want to accept the words w such that $\mathrm{Acc}(w)$ is contained in the triangle $(.2, .2)-(.8, .2)-(.2, .8)$. We can do so by extending the dimension of the payoffs and renaming $(0, 0) \mapsto (0, 0, 0)$, $(1, 0) \mapsto (1, 0, 1)$, and $(1, 1) \mapsto (1, 1, 2)$. Namely, by adding a coordinate whose value is the sum of the other two coordinates. Then, $L = \{w | \min p_1(\mathrm{Acc}(w)) > .2 \wedge \min p_2(\mathrm{Acc}(w)) > .2 \wedge \max p_3(\mathrm{Acc}(w)) < 1\}$ is the wanted language.

4.2 Closure under Boolean operations

We prove here that $\mathfrak{L}_{mt}$ is in fact the Boolean closure of $\mathfrak{L}_{\lessgtr} \triangleq \mathfrak{L}_{<} \cup \mathfrak{L}_{\leq} \cup \mathfrak{L}_{>} \cup \mathfrak{L}_{\geq}$.

Theorem 5 *$\mathfrak{L}_{mt}$ is closed under Boolean operations and any language in $\mathfrak{L}_{mt}$ is a Boolean combination of languages in $\mathfrak{L}_{\lessgtr}$.*

Proof. Closure by complementation: let L be a $\mathfrak{L}_{mt}$ language, defined on some automaton $\mathcal{A}$ by a predicate P. $w \in L$ iff $P(\mathrm{Acc}_{\mathcal{A}}(w))$. So $w \in L^c$ iff $w \notin L$, that is iff $\neg P(\mathrm{Acc}_{\mathcal{A}}(w))$. But $\neg P$ is also a Boolean combination of threshold conditions, thus L^c is a $\mathfrak{L}_{mt}$ language.

Closure by intersection: let L_1 and L_2 be two $\mathfrak{L}_{mt}$ languages defined respectively on the automata $\mathcal{A}$ and $\mathcal{A}$ by the predicates P_1 and P_2. Then $L_1 \cap L_2 = L(\mathcal{A} \otimes \mathcal{A}, P_1 \Cap P_2)$ (Theorem 3). It is easy to see that $P_1 \Cap P_2$ is still a Boolean combination of thresholds, and thus $L(\mathcal{A} \otimes \mathcal{A}, P_1 \Cap P_2)$ is in $\mathfrak{L}_{mt}$.

The other Boolean operations can be written as a combination of complementation and intersection.

Now we show, by induction on height of the formula of a predicate, that any $\mathfrak{L}_{mt}$ language is a Boolean combination of $\mathfrak{L}_{\lessgtr}$ languages.

We can, without loss of generality, suppose that every threshold concerns a different coordinate (if a coordinate has several thresholds, we can duplicate that coordinate, keeping the same values, and the language will remain the same). We can also assume that the predicate is written only with combinations of conjunctions and negations of thresholds.

- If the height is 0, that means that the condition is only one threshold on a multi-payoff automaton. The recognized language is the same as that of the automaton projected on the tested coordinate, so it is in $\mathfrak{L}_{\lessgtr}$.
- If the predicate is $\neg P$, then the recognized language is the complement of $L(\mathcal{A}, P)$, which is a Boolean combination of $\mathfrak{L}_{mt}$ languages of lesser height.
- If the predicate is $P = P_1 \wedge P_2$, let us call $\mathcal{A}_i$, a copy of $\mathcal{A}$ whose payoffs are projected on the subspace tested in P_i. Then $\mathcal{A}$ is isomorphic to $\mathcal{A}_1 \otimes \mathcal{A}_2$. Furthermore, as the set of coordinates that are tested in P_1 and P_2 are disjoint, their exists

some P'_1 and P'_2 with the same heights as P_1 and P_2, such that $P = P'_1 \Cap P'_2$. Thus $L = L(\mathcal{A}_1, P'_1) \cap L(\mathcal{A}_2, P'_2)$ (Theorem 3), which is a Boolean combination of $\mathfrak{L}_{mt}$ languages of lesser height.

□

4.3 Decidability

Theorem 6 *The emptiness of a language of $\mathfrak{L}_{mt}$ is decidable.*

Proof. We can assume the predicate of acceptance is written in disjunctive normal form (if not, we can find an equivalent DNF formula). Then we can see that a run is accepted whenever its set of accumulation points satisfies at least one of the disjuncts, and in a disjunct, every literal has to be satisfied. If we know how to check if a literal is satisfied, then this provides an obvious decision algorithm for one run.

Then it is easy to see that there are two types of literal. Some say that there must exist an accumulation point whose tested coordinate is greater or smaller than the threshold, we call those existential literals. The other literals say that every accumulation point should have the tested coordinate above or below the threshold, those we call universal literals.

For checking the emptiness of $L(\mathcal{A}, F)$, we propose the following algorithm: Try, for every a disjunct of F and every reachable SCC C of $\mathcal{A}$, to compute $P(C)$,the convex hull of the payoffs of its transitions, then compute C' the intersection of $P(C)$ with every universal literal, and finally check whether it intersects with every existential literal of the disjunct. If it does, then return true, else loop. If you exhausted the combinations, return false.

If this algorithm returns true, because C' is a convex polyhedron included in C and intersecting with every existential literal, we can construct D which is connected, closed, included in C', and intersects with every existential literal (take for instance the convex hull of a family consisting in one point in every intersection of C' with an existential literal). We can see that $F(D)$ holds. Then, Theorem 2 says there exist a run r such that $\mathrm{Acc}_{\mathcal{A}}(r) = D$, and thus there exist a word which that run and therefore is in $L(\mathcal{A}, F)$.

If that algorithm returns false, for every reachable SCC C, if you choose a closed connected subset D of $P(C)$ (as Theorem 1 says sets of accumulation points have to be), then for every disjunct, D either is not completely included in some universal literal, either does not intersect with some existential literal. In both case, D does not make the disjunct true. So F holds for no set of accumulation points of a run of $\mathcal{A}$, which implies that $L(\mathcal{A}, F)$ is empty. □

5 Summary and Future Directions

We proposed a definition of ω-languages using Boolean combination of threshold predicates over mean-payoffs. This type of specifications allows to express requirements concerning averages such as “no more than 10% of the messages are lost” or “the number of messages lost is negligible”. The later is not expressible by ω-regular requirements. We showed that if closure under intersection is needed, multi-dimensional

payoffs have to be considered. For runs of d-payoff automata, we studied acceptance conditions that examine the set of *accumulation points* and characterized those sets as all closed, bounded and connected subsets of $\mathbb{R}^d$.

The class of *multi-threshold mean-payoff languages* was proposed, using acceptance conditions that are Boolean combinations of inequalities comparing the minimal or maximal accumulation point along some coordinate with a constant threshold. We studied expressiveness, closure properties, analyzability, and Borel complexity.

Possible direction for future include extension to non-deterministic automata, and the study of multi-mean-payoff games.

Acknowledgments

This research was partially supported by NSF grants CNS 0524059 and CPA 0541149, and the French MINALOGIC project ATHOLE.

References

1. R. Alur, A. Kanade, and G. Weiss. Ranking automata and games for prioritized requirements. In A. Gupta and S. Malik, editors, *CAV*, volume 5123 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2008.
2. K. Chatterjee. Concurrent games with tail objectives. *Theor. Comput. Sci.*, 388(1-3):181–198, 2007.
3. K. Chatterjee, L. de Alfaro, and T. Henzinger. The complexity of quantitative concurrent parity games. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 678–687, 2006.
4. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proceedings of CSL 2008: Computer Science Logic*, Lecture Notes in Computer Science. Springer-Verlag, 2008.
5. K. Chatterjee, T. Henzinger, and M. Jurdziński. Mean-payoff parity games. In *Proceedings of the 20th Annual Symposium on Logic in Computer Science*, pages 178–187. IEEE Computer Society Press, 2005.
6. H. Gimbert and W. Zielonka. Deterministic priority mean-payoff games as limits of discounted games. In *ICALP*, pages 312–323, 2006.
7. O. Kupferman and Y. Lustig. Lattice automata. In *Proc. 8th Intl. Conf. Verification, Model Checking, and Abstract Interpretation*, LNCS 4349, pages 199–213. Springer, 2007.
8. Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems: Specification*. Springer-Verlag, 1991.
9. A. Pnueli. The temporal logic of programs. In *18th IEEE Symposium on the Foundations of Computer Science (FOCS'77)*, pages 46–57, 1977.
10. W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 133–191. Elsevier Science Publishers, 1990.
11. M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.
12. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.