Finite-Memory Strategy Synthesis for Robust Multidimensional Mean-Payoff Objectives

Yaron Velner
The Blavatnik School of Computer Science, Tel Aviv University, Israel

***Abstract*—Two-player games on graphs provide the mathematical foundation for the study of reactive systems. In the quantitative framework, an objective assigns a value to every play, and the goal of player 1 is to minimize the value of the objective. In this framework, there are two relevant synthesis problems to consider: the quantitative analysis problem is to compute the minimal (or infimum) value that player 1 can assure, and the boolean analysis problem asks whether player 1 can assure that the value of the objective is at most ν (for a given threshold ν). Mean-payoff expression games are played on a multidimensional weighted graph. An atomic mean-payoff expression objective is the mean-payoff value (the long-run average weight) of a certain dimension, and the class of mean-payoff expressions is the closure of atomic mean-payoff expressions under the algebraic operations of $\max, \min$, numerical complement and sum. In this work, we study for the first time the strategy synthesis problems for games with robust quantitative objectives, namely, games with mean-payoff expression objectives. While in general, optimal strategies for these games require infinite-memory, in synthesis we are typically interested in the construction of a finite-state system. Hence, we consider games in which player 1 is restricted to finite-memory strategies, and our main contribution is as follows. We prove that for mean-payoff expressions, the quantitative analysis problem is computable, and the boolean analysis problem is inter-reducible with Hilbert's tenth problem over rationals — a fundamental long-standing open problem in computer science and mathematics.**

I. Introduction

In the classical framework of boolean formal verification, program may only violate or satisfy a given specification, and in the framework of synthesis, the task is to automatically construct **a** program that satisfies the specification. The boolean framework does not discriminate between programs that satisfy a given specification, and consequently, it may produce (or verify) unreasonable implementations.

In the recent years, there is an emerging line of research that aims to measure the quality of a program with quantitative metrics, e.g., [1], [3], [5], [7], [8], [11]. The quantitative verification problem asks how well a program satisfies a given specification, and the synthesis task is to construct the optimal program with respect to a specification.

Quantitative verification and synthesis problems are modelled by infinite-duration games over weighted graphs. In these games, the set of vertices is partitioned into player-1 and player-2 vertices; initially, a pebble is placed on an initial vertex, and in every round, the player who owns the vertex that the pebble resides in, advances the pebble to an adjacent vertex. This process is repeated forever and give rise to a *play* that induces an infinite sequence of weights (or weight vectors), and a quantitative objective assigns a value to every play (or equivalently to every infinite sequence of weights).

The classical work on these games only considered games with single objectives, such as minimizing the long run average weight, or minimize the sum of weights. In order to have robust quantitative specifications, it is necessary to investigate games on graphs with multiple (and possibly conflicting) objectives. Typically, multiple objectives are modeled by multidimensional weight functions (e.g., [1], [4], [5], [10]), and the outcome of a *play* is a vector of values. In the boolean setting, the goal of player 1 is to satisfy a boolean condition on the values (with respect to a threshold vector). For example, player 1 needs to assure that the average response time (rt) of an arbiter is at most 2.4 and that the average energy consumption (ec) is below 7. In the quantitative setting, the outcome of a play is a unique (real) value, and the goal of player 1 is to minimize the value of the play. A multiple objective specification is modelled by algebraic operations on single objectives. In the example above, we define the quantitative objective $\max(rt - 2.4, ec - 7)$, and a non-positive value to the quantitative objective implies that the boolean objective is satisfied. In the general case, an objective is determined either by the projection of the weight function to one dimension, or it is formed by algebraic operation on two (or more) objectives. In the literature, the common and natural algebraic operations are $\min$,$\max$, *numerical complement* (multiplication by -1) and sum. We note that when the goal is to minimize the value of the objective, then the first three operations generalize the boolean disjunction, conjunction and negation. A quantitative objective is *robust* if it is closed under the four algebraic operations. So far, the only known class of robust quantitative objectives that has an effective algorithm for the model checking problem (that is, for solving one-player games) is the class of *mean-payoff expressions* [6], which is the closure of one-dimensional mean-payoff (long-run average of the weights) objectives to the four algebraic operation. For example, for an infinite sequence of vectors $a = a_1, a_2, \dots \in (\mathbb{R}^3)^\omega$ the objective

$$E(a) = LimAvg_1(a) + \min(LimAvg_2(a), -LimAvg_3(a))$$

is a mean-payoff expression (where $LimAvg_i$ is the long-run average of dimension i) and $E((1, 2, 6)^\omega) = 1+\min(2, -6) = -5$.

In the quantitative setting, there are two relevant synthesis problems: (i) the *quantitative analysis problem* is to compute the optimal (infimum) value that a player-1 strategy can assure;

and (ii) the *boolean analysis problem* is to determine whether player 1 can assure a value of at most ν to the objective (for a given ν). From the perspective of synthesis, these problems are most important when player 1 is restricted to finite-memory strategies.

For mean-payoff expressions, optimal finite-memory strategies may not always exist. Hence, the quantitative analysis problem is to compute the greatest lower bound on the minimal value that player 1 can assure. We note that since all model checking problems are decidable for mean-payoff expression, then the computability of the quantitative analysis will give us an effective algorithm to synthesize ϵ-optimal finite-memory strategies, and if the boolean analysis problem were decidable, then we would have an algorithm that construct the corresponding player-1 strategy.

Our contribution. In this paper, we consider for the first time the synthesis problem for a robust class of quantitative objectives. We prove computability for the quantitative synthesis problem, and we show that the boolean analysis problem is inter-reducible with Hilbert's tenth problem over rationals ($H10(\mathbb{Q})$), which is a fundamental long-standing open question in computer science and mathematics. We show that the problem is inter-reducible with $H10(\mathbb{Q})$ even when both players are restricted to finite-memory strategies, and we show that there is a fragment of mean-payoff expressions that is $H10(\mathbb{Q})$-hard when one or both players are restricted to finite-memory strategies, but decidable when both players may use infinite-memory strategies.

Our main technical contribution is the introduction of a general scheme that lifts a one-player game solution (equivalently, a model checking algorithm) to a solution for a two-player game (when player 1 is restricted to finite-memory strategies). The scheme works for a large class of quantitative objectives that have certain properties (which we define in Subsection II-B).

Related work. The class of mean-payoff expressions was introduced in [6], and the decidability of the model checking problems (which correspond to one-player games) was established. A simpler and more efficient algorithm for mean-payoff expression games was given in [15]. Mean-payoff games on multidimensional graphs were first studied in [9]. In these games the objective of player 1 was to satisfies a conjunctive condition (in the terms of this paper, the objective was a maximum of multiple one-dimensional objectives). In [16], a decidability for an objective that is formed by the $\min$ and $\max$ operators was established. But the proof can not be extended to include the numerical complement operator, and it does not scale for the case that player 1 is restricted to finite-memory strategies.

Structure of the paper. In the next section we bring the basic definitions for quantitative games and we define a class of quantitative objectives that have special properties. In Sections III and IV we give a generic solution for the synthesis problem of quantitative objectives that satisfies the special properties (and an overview of the solution is given in Subsection II-C). In Section V we show that mean-payoff expressions satisfy the special properties and the main results of the paper follow. Due to lack of space, some of the proofs were omitted, and the full proofs are in the appendix.

II. Games with Quantitative Objectives

In this section we bring the formal definitions for quantitative objectives and games on graphs with quantitative objectives (Subsection II-A). We define four special properties of quantitative objectives (Subsection II-B), and we give an informal overview for the two-player game solution of games with quantitative objectives that satisfy the special properties (Subsection II-C).

A. Quantitative games on graphs

Quantitative objectives. In this paper we consider directed finite graphs with a k-dimensional weight function that assigns a vector of rationals to each edge. A *quantitative objective* is a function that assigns a value to every infinite sequence of weight vectors. Formally an objective is a function $obj : (\mathbb{R}^k)^\omega \to \mathbb{R}$. A simple example for quantitative objective is to consider a one-dimensional weight function and an objective that assigns to each infinite path the maximal weight that occurs infinitely often in the path. An objective $obj : (\mathbb{R}^k)^\omega \to \mathbb{R}$ is called *prefix-independent* if for every $a_1 \in (\mathbb{R}^k)^*$ and $a_2 \in (\mathbb{R}^k)^\omega$ it holds that $obj(a_1 a_2) = obj(a_2)$.

Algebraic operations over quantitative objectives. The quantitative counterpart of the boolean operations of disjunction, conjunction and negation are the $\max$,$\min$ and numerical complement operators (numerical complement is multiplication by -1). The sum operator, which does not have a boolean counterpart, is also very natural operator in the framework of quantitative objectives. For two quantitative objectives obj_1 and obj_2, the quantitative objective $\mathrm{op}(obj_1, obj_2)$ (for $\mathrm{op} \in \{\min, \max, \mathrm{sum}\}$) assigns to every infinite sequence of weights $\ell \in (\mathbb{R}^k)^\omega$ the value $\mathrm{op}(obj_1(\ell), obj_2(\ell))$, and the numerical complement of obj_1 assigns the value of $-obj_1(\ell)$.

Robust quantitative objectives A class of quantitative objective $\mathcal{O}$ is *robust* if it is closed under the algebraic operations of $\min, \max, \mathrm{sum}$ and numerical complement. Formally, a class of objectives $\mathcal{O}$ is robust, if for every two objectives $obj_1, obj_2 \in \mathcal{O}$ the four quantitative objectives $obj_{\min}, obj_{\max}, obj_{\mathrm{sum}}$ and obj^- are in $\mathcal{O}$ (such that for every $\ell \in (\mathbb{R}^k)^\omega$ and $\mathrm{op} \in \{\min, \max, \mathrm{sum}\}$: $obj_{\mathrm{op}}(\ell) = \mathrm{op}(obj_1(\ell), obj_2(\ell))$ and $obj^-(\ell) = -obj_1(\ell)$). We note that in [6], [8], Chatterjee et al. gave a broader definition for robustness of quantitative objectives, but since the concrete objectives that we consider in this paper are robust according to both definitions, we prefer to use the narrower (and simpler) notion of robustness.

Games on graph. A game graph is a directed graph $G = (V = V_1 \cup V_2, v_0, E, w : E \to \mathbb{Q}^k)$, where V is the set of vertices; V_i is the set of *player i* vertices; v_0 is the *initial vertex*; $E \subseteq V \times V$ is the set of edges; and $w : E \to \mathbb{Q}^k$ is a multidimensional weight function (e.g., see Figure 1). A *play*

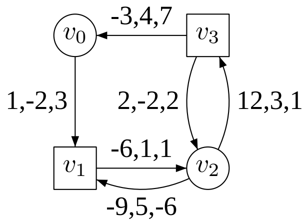

Fig. 1. Game graph G. Player 1 owns the round vertices.

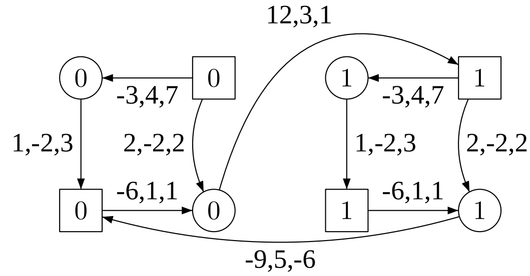

Fig. 2. Game graph G according to strategy σ.

is an infinite sequence of rounds. In the first round a pebble is placed on the initial vertex and in every round the player who owns the vertex of the pebble advance the pebble to an adjacent vertex. Hence, a play corresponds to an infinite path in the graph that begins in v_0 and the labeling of the play is the corresponding infinite sequence of weight vectors. A game graph is a *one-player game* if only one of the players has a vertex with out-degree more than one.

Strategies. A *strategy* is a recipe for determining the next move based on the *history* of the play. A *player-i strategy* is a function $\sigma : V^*V_i \to V$, such that for every finite path π that ends in vertex v we have $(v, \sigma(\pi)) \in E$. A strategy has *finite memory* if it can be implemented by a Moore machine $(M, m_0, \alpha_n, \alpha_u)$, where M is a finite set of memory states, m_0 is the initial memory state, $\alpha_u : M \times V \to M$ is the update function, and $\alpha_n : M \times V_i \to V$ is the next vertex function. If a play prefix is in state v_i and memory state M, then the strategy choice for the next vertex is $v = \alpha_n(M, v_i)$ and the memory is updated to $\alpha_u(M, v_i)$. A strategy is *memoryless* if it depends only in the current location of the pebble. Formally a player-i memoryless strategy is a function $\sigma : V_i \to V$. (We note that a memoryless strategy is also a finite-memory strategy.)

We denote the set of all player-i strategies by $\mathcal{S}_i$ and we denote the set of all player-i finite memory strategies by $\mathcal{FM}_i$.

Game graph according to a finite-memory strategy. For a game graph $G = (V = V_1 \cup V_2, E, w)$ and a player-1 finite-memory strategy $\sigma = (M, m_0, \alpha_u, \alpha_n)$, we denote the *game graph according to strategy* σ by G^σ, and we define it as follows:

- The vertices of G^σ are the Cartesian product $V \times M$; player-i vertices are $V_i \times M$; and the initial vertex of G^σ is (v_0, m_0).
- For a player-1 vertex (v, m), the only successor vertex is $(\alpha_n(v, m), \alpha_u(v, m))$. For a player-2 vertex (v, m) the set of successor vertices is $\{(u, n) \mid (v, u) \in E \text{ and } \alpha_u(v, m) = n\}$.

We note that the out-degree of all player-1 vertices is one, and thus G^σ is a *one-player game graph*. The main property of graphs according to a finite-memory strategy is that every infinite path in G^σ corresponds to a play that is consistent with σ in G. A game graph according to a memoryless strategy is a special case of games according to finite-memory strategies. In this case, the game graph is obtained from G by removing all the player's out-edges that are not chosen by the memoryless strategy.

Example 1. *Consider the game graph from Figure 1 and consider a player-1 strategy σ that in vertex v_2 moves the pebble to v_3 if v_2 was visited an odd number of times and otherwise it moves the pebble to v_1. For example, in the first time that v_2 is visited, player 1 moves the pebble to v_3, in the second time he will move the pebble to v_1, in the third time to v_3 and so on. The strategy σ requires one bit of memory (i.e., $M = \{0, 1\}$), and G^σ is illustrated in Figure 2 (the labeling of the nodes represents the memory state). In G^σ all the choices are done by player 2.*

Values of strategies and games. A tuple (σ, τ) of player-1 and player-2 strategies (respectively) uniquely defines a play $\pi_{\sigma,\tau}$ in a given graph. For a game graph G, a quantitative objective obj and a tuple of strategies (σ, τ) we denote $Val_{\sigma,\tau} = obj(\pi_{\sigma,\tau})$. In this paper, we assume that player 1 wishes to minimize[1] the value of the quantitative objective, and we define the *value* of a player-1 strategy σ to be $Val_\sigma = \sup_{\tau \in \mathcal{S}_2} Val_{\sigma,\tau}$. (Intuitively, this is the maximal value that player 2 can achieve against strategy σ.) The *minimal value of a game* is defined as $\inf_{\sigma \in \mathcal{FM}_1} Val_\sigma$. Intuitively, the minimal value of a game is the minimal value that player 1 can ensure by a finite-memory strategy.

Quantitative and boolean analysis For a given game graph, objective, and a rational threshold $r \in \mathbb{Q}$: The *quantitative analysis* task is to compute the minimal value of the game that can be enforced by a finite-memory strategy. The *boolean analysis* task is to decide whether there is a player-1 finite-memory strategy σ for which $Val_\sigma \leq r$. That is, whether player 1 can assure that a value of at most r for the objective.

Boolean games and winning strategies. A *boolean game* is a game on graph equipped with a winning condition $W \subseteq E^\omega$ (that is, a winning condition is a set of infinite paths). A play π is *winning* for player 1 if $\pi \in W$, and a strategy σ is a player-1 *winning strategy* if for every player-2 strategy τ we have $\pi_{\sigma,\tau} \in W$. For a quantitative objective obj and a threshold $\nu \in \mathbb{R}$ we denote by (obj, ν) the boolean winning condition $\{\pi \in E^\omega \mid obj(\pi) \leq r\}$.

B. One-player game solution

In this paper, we consider objectives that have special properties for their one-player game solution and we present a general scheme that lifts a one-player game solution into a two-player game solution. To formally define the special properties of the solutions, we bring the next definitions.

[1]Since we consider robust objectives, then the same results hold when player-1 goal is to maximize the value of the objective.

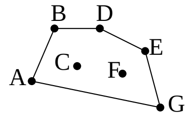

Fig. 3. $CONVEX(A, B, C, D, E, F, G)$ is the polygon $ABDEG$.

Definitions and notions for weighted graphs. Let $G = (V, E, w : E \to \mathbb{Q}^k)$ be a k-dimensional weighted graph. The *weight vector* of a finite path $\pi = e_1 \dots e_n$ is $w(\pi) = \sum_{i=1}^{n} w(e_i)$ and the *average weight* of a path is $Avg(\pi) = \frac{w(\pi)}{|\pi|}$. For a set of finite paths $\Pi = \{\pi_1, \dots, \pi_n\}$ we denote $Avg(\Pi) = \{Avg(\pi_1), \dots, Avg(\pi_n)\}$. We denote the set of simple cycles in G by $C(G)$, and we abbreviate $Avg(G) = Avg(C(G))$. For a finite set of vectors $V = \{v_1, v_2, \dots, v_n\} \in \mathbb{R}^k$, we denote $CONVEX(V) = \{\sum_{i=1}^{n} \alpha_i v_i \mid \sum_{i=1}^{n} \alpha_i = 1 \text{ and } \alpha_1, \dots, \alpha_n \geq 0\}$ (see Figure 3). We abbreviate $CONVEX(G) = CONVEX(Avg(G))$. An m-dimensional *simplex* is the set $\mathcal{S}(m) = \{(x_1, \dots, x_m) \in \mathbb{R}^m \mid x_i \geq 0 \wedge \sum_{i=1}^{m} x_i = 1\}$. The *simplex interior* is $\mathcal{SI}(m) = \{(x_1, \dots, x_m) \in \mathbb{R}^m \mid x_i > 0 \wedge \sum_{i=1}^{m} x_i = 1\}$, and the *rational interior of a simplex* is $\mathcal{QSI}(m) = \mathcal{SI}(m) \cap \mathbb{Q}^m$. When m is clear from the context we abbreviate $\mathcal{S}(m), \mathcal{SI}(m)$ and $\mathcal{QSI}(m)$ with $\mathcal{S}, \mathcal{SI}$ and $\mathcal{QSI}$ (respectively).

Solution for one-player game with special properties. A solution for a one-player quantitative game is a function f that assigns to every one-player game graph G the maximal value that the player can achieve in graph G. We note that for prefix-independent objectives, a function f' that assigns to each *strongly connected* graph its maximal value uniquely defines the solution function f (since the value of f is the maximal value of f' over all the strongly connected component of the graph). In this paper, we will consider only prefix-independent objective, hence, we define the special properties of a solution for strongly connected graphs. The special properties that we consider are:

1) *First-order definable.* For every $n \in \mathbb{N}$ there is a first-order formula $\zeta_n(\overline{x_1}, \dots, \overline{x_n}, y)$ over $\langle \mathbb{R}, =, <, +, \times \rangle$ such that for every graph G with $Avg(G) = \{\overline{x_1}, \dots, \overline{x_n}\}$ we have $f(G) = y$ if and only if $\zeta_n(\overline{x_1}, \dots, \overline{x_n}, y)$ holds. In addition we require ζ_n to be computable from n. In the sequel, we write $y = \zeta_n(\overline{x_1}, \dots, \overline{x_n})$ instead of $\zeta_n(\overline{x_1}, \dots, \overline{x_n}, y)$.
2) *Monotone in $CONVEX(G)$.* If for two (strongly connected) graphs H and G we have $CONVEX(G) \subseteq CONVEX(H)$, then $f(G) \leq f(H)$. As a consequence, we get that for a k-dimensional objective, f is a function from $(\mathbb{R}^k)^*$ to $\mathbb{R}$, and that $f(G) \equiv g(CONVEX(G))$ for some function $g : (\mathbb{R}^k)^* \to \mathbb{R}$. Hence, by abusing the notation, we sometime write $f(CONVEX(G))$ instead of $f(G)$.
3) *Continuous function.* f is a continues function. Formally, if f is the solution for a k-dimensional objective, then for every $n \in \mathbb{N}$ the function $\zeta_n : (\mathbb{R}^k)^n \to \mathbb{R}$ is a continues function.

We will show computability for the quantitative analysis problem for objectives that have a solution that satisfies the above three properties. We also consider a fourth special property, and we will show decidability for the boolean analysis problem for objectives that have a solution that satisfies all four properties.

4) *Fourth property.* A solution $f = \{\zeta_1, \dots, \zeta_n, \zeta_{n+1}, \dots\}$ satisfies the fourth property if the next problem is decidable (for the set $\{\zeta_1, \dots, \zeta_n, \zeta_{n+1}, \dots\}$):
   - Input: a threshold $\nu \in \mathbb{Q}$ and a set of n matrices $A_1, \dots, A_n$, where A_i is a $k \times m_i$ matrix for some $m_i \in \mathbb{N}$.
   - Task: determine if the inequality $\zeta_n(A_1 \cdot x_1, \dots, A_n \cdot x_m) \leq \nu$ subject to $x_i \in \mathcal{QSI}(m_i)$ is feasible (note that the result of the multiplication $A_i \cdot x_i$ is a vector of size k).

In the next example we demonstrate the above properties.

Example 2. *Consider the two-dimensional one-player solution function* $f(G) = \max_{(x,y) \in CONVEX(G)}[\max(x+y+10, -x+y+10, \min(-x+y-10, x+y-10))]$. *We demonstrate that f is first-order definable by giving the explicit formula for ζ_2, that is, the formula for a (strongly connected) graph with only two simple cycles with average weights (x_1, y_1) and (x_2, y_2).*

$$\begin{aligned}
&\zeta_2(x_1, y_1, x_2, y_2, r) \equiv \\
&\forall \alpha_1, \alpha_2, x, y(\alpha_1 \geq 0 \wedge \alpha_2 \geq 0 \wedge (\alpha_1 + \alpha_2 = 1) \wedge (x = \alpha_1 x_1 + \alpha_2 x_2) \wedge (y = \alpha_1 y_1 + \alpha_2 y_2)) \to \\
&r \geq \max(x+y+10, -x+y+10, \min(-x+y-10, x+y-10)) \\
&\wedge \\
&\exists \alpha_1, \alpha_2, x, y(\alpha_1 \geq 0 \wedge \alpha_2 \geq 0 \wedge (\alpha_1 + \alpha_2 = 1) \wedge (x = \alpha_1 x_1 + \alpha_2 x_2) \wedge (y = \alpha_1 y_1 + \alpha_2 y_2)) \wedge \\
&r = \max(x+y+10, -x+y+10, \min(-x+y-10, x+y-10))
\end{aligned}$$

(Technically max *and* min *are not in $\langle \mathbb{R}, <, +, \times \rangle$, but they are trivially definable in this dictionary.) Clearly if for two graphs we have $CONVEX(G_1) \subseteq CONVEX(G_2)$, then $f(G_1) \leq f(G_2)$ (hence, f is monotone), and ζ_2 is obviously a continuous function (and in general ζ_n is also continuous). Hence, f satisfies Properties 1-3. In Figure 4 we illustrate the geometrical interpretation of Property 2, namely, the fact that the value of f depends only in $CONVEX(G)$. The equality* $\max(x+y+10, -x+y+10, \min(-x+y-10, x+y-10)) = 0$ *is presented by the thick line. The points that are connected by the dotted line present the weights of the simple cycles of a strongly connected graph G_1 and the points that are connected by the dashed line present the weights of the simple cycles of a strongly connected graph G_2. The reader can see that $CONVEX(G_1)$ is below the thick line and $CONVEX(G_2)$ intersects with it. Hence, $f(G_1) < 0$ and $f(G_2) > 0$.*

C. Informal overview of the solution for two-player games

The key notion for our solution is games according to strategies. When a one-player solution f is given, the boolean analysis problem amounts to determining whether there is a finite-memory strategy σ such that for every strongly-connected component (SCC) $S \in G^\sigma$ it holds that $f(CONVEX(S)) \leq \nu$. In Lemma 4 we show that w.l.o.g we may assume that for

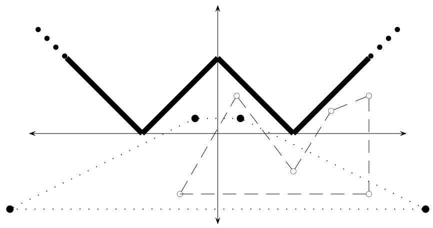

Fig. 4.

any σ the graph G^σ is strongly connected. Hence, in Section III we investigate the set $\{CONVEX(G^\sigma) \mid \sigma \in \mathcal{FM}_1\}$ and obtain a computable characterization for it. In Section IV we exploit the properties of the one-player solution and the results of Section III and we obtain a first-order formula over **rationals** that computes the values that player 1 can enforce. We use the fact that f is continuous to show that the formula has the same infimum over rationals and reals, and hence, due to Tarski's Theorem the infimum value is computable. We also show that if Property 4 holds, then one can effectively determine whether the formula has an assignment that gives a value of at most ν. In Section V we apply these results for mean-payoff expressions. We show that their one-player solution satisfies Properties 1-3, and that it satisfies property 4 if and only if $H10(\mathbb{Q})$ is decidable.

III. $CONVEX$ Cycles Problem

In this section, we consider the next problem:

Problem 1 ($CONVEX$ cycles problem)**.**
- *Input: a k-dimensional game graph G and a set of k-dimensional vectors $\mathcal{V}$.*
- *Task: determine whether there is a player-1 finite-memory strategy σ such that $CONVEX(G^\sigma) \subseteq CONVEX(\mathcal{V})$. (We call such strategy a* realizing *strategy.)*

We first present the solution for the above problem, and then we show how to find all the sets of vectors for which there is a realizing player-1 finite-memory strategy.

The solution for Problem 1 relies on the next lemma.

Lemma 1. *For a game graph G and a set of vectors $\mathcal{V}$, there exists a player-1 finite-memory strategy σ for which $CONVEX(G^\sigma) \subseteq CONVEX(\mathcal{V})$ iff for every player-2 memoryless strategy τ_i there exists a player-1 finite-memory strategy σ_i such that $CONVEX((G^{\tau_i})^{\sigma_i}) \subseteq CONVEX(\mathcal{V})$.*

Proof: The proof for the direction from right to left is trivial (since any cycle in $(G^{\tau_i})^\sigma$ is also a cycle in G^σ).

Our proof for the converse direction is inspired by [12], and the key intuition of the proof, is the following. Let v be a player-2 vertex, with two out-edges e_1 and e_2, and let $G_1 = G - \{e_1\}$ and $G_2 = G - \{e_2\}$. Suppose that player 1 has two finite-memory strategies σ_1 and σ_2 such that $CONVEX(G_i^{\sigma_i}) \subseteq CONVEX(\mathcal{V})$ (for $i = 1, 2$). Then player 1 can combine the two strategies over $G - \{e_1\}$ and $G - \{e_2\}$, and he can obtain a finite-memory strategy σ, such that each simple cycles in G^σ is a convex combination of cycles from $G_1^{\sigma_1}$ and $G_2^{\sigma_2}$, and hence, $CONVEX(G^\sigma) \subseteq CONVEX(\mathcal{V})$. Hence, either player 1 has a realizable strategy for G or he does not have realizable strategy for $G_1 = G - \{e_1\}$ or for $G_2 = G - \{e_2\}$. Since this holds for every player-2 state, the proof follows.

In order to formally prove the key intuition we claim that if player 1 has a realizable strategy σ_i against any player-2 memoryless strategy, then he has a realizable strategy σ that satisfies $CONVEX(G^\sigma) \subseteq CONVEX(\mathcal{V})$, and we prove the claim by induction on the number of player-2 vertices with out-degree greater then one. The base case, where all of player-2 vertices have out-degree one, is trivial. For the inductive step, let us assume that there is a player-2 vertex v with out-edges e_1 and e_2 (if there is no such vertex, then we are in the base case). For $i = 1, 2$, let G_i be $G - \{e_i\}$. If player-2 has a violating memoryless strategy in either G_1 or G_2, then surely this is also a violating memoryless strategy for G, and the claim follows. Otherwise, we construct a realizable player-1 strategy in G in the following way. For $i = 1, 2$, let σ_i be a finite-memory player-1 realizable strategy in G_i, If in σ_1 (resp., σ_2), the vertex v is unreachable then it is surely a winning strategy also for G. Otherwise, there exists a memory state m such that (m, v) is a vertex in $G_1^{\sigma_1}$, and we denote by σ_1' the strategy that is obtained by changing σ_1 initial memory state to m. We construct σ in the following way. The memory structure of σ is a tuple $(M_1, M_2, \{1, 2\})$ where M_1 is the memory structure of σ_1', M_2 is the memory structure of σ_2, and the third value in the tuple indicates if we are playing according to σ_1' or σ_2. At the beginning of a play, σ decides according to σ_2 (and updates M_2 accordingly). If σ decides according to σ_2 and edge e_1 is visited, then σ decides according to σ_1' (and updates M_1 accordingly), until edge e_2 is visited, and then σ again decides according to σ_2, and so on. We note that σ is a finite-memory strategy, and that any simple cycle in G^σ is a composition of simple cycles from $G_1^{\sigma_1'}$ and $G_2^{\sigma_2}$. Hence, the average weight of any simple cycle in G^σ is $\lambda x_1 + (1-\lambda) x_2$ for some $\lambda \in [0, 1]$ and $x_i \in Avg(G_i) \subseteq CONVEX(\mathcal{V})$. And thus, a convex combination of x_1 and x_2 is also in $CONVEX(\mathcal{V})$, and we get that $CONVEX(G^\sigma) \subseteq CONVEX(\mathcal{V})$. Therefore, σ is a realizing strategy and the proof is completed. ■

We now wish to characterize all the sets of vectors that have a realizing strategy. For this purpose we bring the next definition. For a player-2 memoryless strategy τ, let Π_e^τ be the (finite) set of Eulerian cyclic paths in G^τ, that is Π_e^τ contains only cyclic paths that visit every edge at most once. For every path $\pi \in \Pi_e^\tau$, let $c_1, \ldots, c_t$ be the simple cycles that occur in π and we associate a $t \times k$ matrix A_π to every path π such that the i-th column of the matrix is $Avg(c_i)$. We observe that

$$\{Avg(\pi) \mid \ \pi \text{ is a cyclic path in } G^\tau\} = \bigcup_{\pi \in \Pi_e^\tau} \{A_\pi \cdot x \mid x \in \mathcal{QSI}\}$$

The next lemma shows how to compute the realizable sets of vectors.

Lemma 2. *Let G be a k-dimensional graph, and let $\tau_1, \ldots, \tau_\ell$ be the (finitely many) player-2 memoryless strategies in G. A set of vectors $\mathcal{V} \subseteq \mathbb{R}^k$ is realizable if and only if there exist $x_1, \ldots, x_\ell \in \mathbb{R}^k$ such that $x_i \in \bigcup_{\pi \in \Pi_e^{\tau_i}} \{A_\pi \cdot x \mid x \in \mathcal{QSI}\}$*

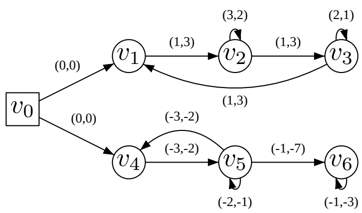

Fig. 5. Game graph G.

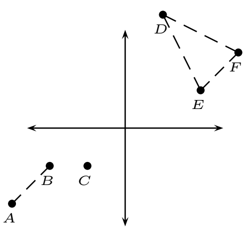

Fig. 6.

(for every $i \in \{1, \ldots, \ell\}$) and $CONVEX(x_1, \ldots, x_\ell) \subseteq CONVEX(\mathcal{V})$.

Proof: First we characterize the realizable vector sets when a player-2 memoryless strategy τ is given, that is, we characterize the realizable vectors in a one-player game. A finite-memory strategy σ in a one-player graph G^τ is an ultimately periodic infinite path, and $(G^\tau)^\sigma$ is a *lasso shaped graph* with exactly one cycle. The cycle of $(G^\tau)^\sigma$ is obviously a cyclic path in G^τ, and thus $\mathcal{V}$ is realizable in G^τ iff there is a cyclic path π in G^τ with $Avg(\pi) \in CONVEX(\mathcal{V})$.

Hence, by Lemma 1, we get that $\mathcal{V}$ is realizable iff for every player-2 memoryless strategy τ_i there is a cyclic path π_i in G^{τ_i} with $Avg(\pi_i) \in CONVEX(\mathcal{V})$. Since such witness π_i exists iff there exists $x_i \in \bigcup_{\pi \in \Pi_e^{\tau_i}} \{A_\pi \cdot x \mid x \in \mathcal{QSI}\}$ with $Avg(\pi_i) = x_i$, then the proof is completed. ■

In the next example we illustrate the geometrical interpretation of Lemma 2.

Example 3. *Consider the game graph G in Figure 5, where the box vertices are controlled by player 2. Player 2 has two possible memoryless strategies, namely, τ_1 that always follow the edge $v_0 \to v_1$ and τ_2 that always follow $v_0 \to v_4$. In G^{τ_1} the set of Eulerian cyclic paths $\Pi_e^{\tau_1}$ contains all cyclic sub-paths of the Eulerian cyclic path $v_1 \to v_2 \to v_2 \to v_3 \to v_3 \to v_1$. Hence, the average weight of any infinite lasso path in G^{τ_1} is a convex combination of $Avg(v_1 \to v_2 \to v_3 \to v_1)$, $Avg(v_2 \to v_2)$ and $Avg(v_3 \to v_3)$ (points D, F and E in Figure 6). In G^{τ_2}, an Eulerian cyclic path is either a sub-path of $v_4 \to v_5 \to v_5 \to v_4$ or the path $v_6 \to v_6$. Hence, the average weight of any infinite lasso path is either a convex combination of $Avg(v_4 \to v_5 \to v_4)$ and $Avg(v_5 \to v_5)$ (points A and B in Figure 6), or it is $Avg(v_6 \to v_6)$ (point C in Figure 6). By Lemma 2, we get that a set of vectors $\mathcal{V}$ is realizable if and only if $CONVEX(\mathcal{V})$ intersects with the polygon DEF and with either the line AB or with the point C (or with both).*

IV. Generic Solution for Games with Quantitative Objectives

In this section we solve the quantitative analysis problem for games with quantitative objectives that satisfy Properties 1-3 and we solve the boolean analysis problem for objectives that satisfy Properties 1-4. We first give a conceptual (i.e., not always computable) solution for the boolean analysis problem, and then extend the solution for the quantitative analysis problem.

An equivalent formulation for the boolean analysis problem is to ask whether for a game graph G and a threshold ν there is a player-1 (finite-memory) strategy σ such that the one-player solution over G^σ is at most ν. By the third property (convex monotonicity), it is enough to determine whether there is σ such that for every SCC S of G^σ it holds that $f(CONVEX(S)) \leq \nu$ (where f is the solution for the one-player game). However, we first show how to determine whether there is σ such that $f(CONVEX(G^\sigma)) \leq \nu$ and only then solve the original problem.

Lemma 3. *Let f be a one-player solution that satisfies Properties 1-3. Then $\inf_{\sigma \in \mathcal{FM}_1} f(CONVEX(G^\sigma))$ is computable (when the input is a game graph G). If f also satisfies Property 4, then the problem of determining whether there is a player-1 finite-memory strategy σ such that $f(CONVEX(G^\sigma)) \leq \nu$ is decidable (when the input is G and ν).*

Proof: Let $\tau_1, \ldots, \tau_m$ be all player-2 memoryless strategies in G (note that m is at most exponential in $|G|$). By Lemma 2, and by the monotonicity of f, there is a player-1 strategy σ that satisfies $f(CONVEX(G^\sigma)) \leq \nu$ if and only if there are matrices $A_{\pi_1}, \ldots, A_{\pi_m}$ and vectors $x_1, \ldots, x_m$ such that $\pi_i \in \Pi_e^{\tau_i}$ and $x_i \in \mathcal{QSI}$ and $\zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m) \leq \nu$. For every τ_i, the set $\Pi_e^{\tau_i}$ is finite (and at most of exponential size). Hence, we can enumerate all m-tuples of $\Pi_e^{\tau_1} \times \cdots \times \Pi_e^{\tau_m}$ and check if for at least one tuple there is a solution to the inequality $\zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m) \leq \nu$. If f satisfies Property 4, then for a given $\pi_1, \ldots, \pi_m$ we can effectively check if the inequality is satisfiable. Hence, we can effectively determine whether there is σ such that $f(CONVEX(G^\sigma)) \leq \nu$. Moreover, for a given $\pi_1, \ldots, \pi_m$ the expression $\inf_{x_1, \ldots, x_m \in \mathcal{SI}} \zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m)$ is first-order definable (recall that ζ_m is first-order definable) over $\langle \mathbb{R}, <, +, \times \rangle$ (note that x_i ranges over $\mathcal{SI}$ and not over $\mathcal{QSI}$) and therefore, by Taski's Theorem [13] its value is computable. Since ζ_m is continues we get that $\inf_{x_1, \ldots, x_m \in \mathcal{SI}} \zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m) = \inf_{x_1, \ldots, x_m \in \mathcal{QSI}} \zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m)$. Finally, by Lemma 2, we have that $\inf_{\sigma \in \mathcal{FM}_1} f(CONVEX(G^\sigma)) = \inf_{\pi_1 \in \Pi_e^{\tau_1}, \ldots, \pi_m \in \Pi_e^{\tau_m}} \inf_{x_1, \ldots, x_m \in \mathcal{QSI}} \zeta_m(A_{\pi_1} \cdot x_1, \ldots, A_{\pi_m} \cdot x_m)$, and since $\Pi_e^{\tau_i}$ is finite we get that $\inf_{\sigma \in \mathcal{FM}_1} f(CONVEX(G^\sigma))$ is computable. ■

Before presenting the algorithm for the boolean analysis problem we recall the (standard) definitions of *winning regions* and *attractors*. Let G be a game graph G with an initial vertex v_0, and let v be an arbitrary vertex in G. We denote by (G, v) the game graph that is formed from G by changing the initial vertex to v. We say that a vertex v is in *player-1 winning region* (denoted by Win_1) if player 1 wins in (G, v) (that is, player 1 has a finite-memory strategy that assures a value at most ν to the objective). The *player-1 attractor set* of a vertex v (denoted by $Attr_1(v)$) contains all the vertices from which player 1 can force reachability to v (after finite number of rounds). It is well known that the attractor set of a vertex is computable (even in linear time) and that player 1 can force reachability by a finite-memory strategy (in fact, even by a memoryless strategy). The next remark shows another important property of attractors and winning regions.

Remark 1. *Let G be a game graph over a boolean objective that is formed by a quantitative objective with a solution function f and a threshold ν, and let v be a vertex in G. Then for every vertex $u \notin Attr_1(v)$, if σ is a finite-memory player-1 strategy for (G, u), then σ is a winning strategy in $(G - Attr_1(v), u)$.*

Proof: We denote $H = G - Attr_1(v)$ and we observe that $(H, u)^\sigma$ is a subgraph of $(G, u)^\sigma$. Hence, for every SCC $S \in H$ there is a corresponding SCC $S' \in G$ such that $f(CONVEX(S')) \leq \nu$. Since $CONVEX(S) \subseteq CONVEX(S')$ and by the monotonicity of f we get that $f(CONVEX(S)) \leq \nu$ and therefore σ is a winning strategy in (H, u). ∎

Algorithm 1 computes player-1 winning region, and we prove its correctness in Lemma 4

Algorithm 1 Player-1 winning region computation for quantitative objectives

```
WINNINGREGION(G, f, ν)
  for v ∈ G do
    if ∃σ s.t f(CONVEX((G, v)^σ)) ≤ ν then
      W ← Attr_1(v)
      W ← W ∪ WINNINGREGION(G − Attr_1(v), f, ν)
      return W
    end if
  end for
  return ∅
```

Lemma 4. *Algorithm 1 computes player-1 winning region.*

Proof: We first prove that in every step of the algorithm, if a vertex $u \in W$, then $u \in Win_1$. We prove the assertion by considering the next three cases: (i) There is a strategy σ for which $f(CONVEX((G, u)^\sigma)) \leq \nu$. In this case, for every SCC $S \in (G, u)^\sigma$ we have that $CONVEX(S) \subseteq CONVEX((G, u)^\sigma)$ and by the monotonicity of f we get that $f(CONVEX(S)) \leq \nu$. Hence, $u \in Win_1$. (ii) There is a vertex v and a strategy σ s.t $f(CONVEX((G, v)^\sigma)) \leq \nu$ and $u \in Attr_1(v)$. In this case, v is in player-1 winning region and therefore the attractor of v is also in Win_1. (iii) For some vertex v we have $u \in$ WINNINGREGION$(G - Attr_1(v), f, \nu)$ and $f(CONVEX((G, v)^\sigma)) \leq \nu$ for some strategy σ. By a simple induction on the size of the graph we get that u is in player-1 winning region for the game graph $G - Attr_1(v)$. The following strategy is a winning strategy for (G, u): (a) play according to the winning strategy over $G - Attr_1(v)$; (b) if the pebble is in vertex v, then play according to σ. Hence, if $u \in W$, then $u \in Win_1$ and we get that $Win_1 \supseteq W$.

In order prove the converse direction, we first prove that if $Win_1 \neq \emptyset$, then $W \neq \emptyset$. Indeed, if $v \in Win_1$, then for some strategy σ we have that for every SCC $S \in (G, v)^\sigma$ it holds that $f(CONVEX(S)) \leq \nu$. Let S' be a terminal SCC in $(G, v)^\sigma$ and let (u, m) be a vertex in S' (where u is a vertex in G and m is a memory state of σ). Let σ' be the strategy that is formed by changing σ initial memory state to m. Then $(G, u)^{\sigma'} = S'$, and therefore $f(CONVEX((G, u)^{\sigma'})) \leq \nu$. Hence, the **if** condition in the **for** loop is satisfied at least once, and $W \neq \emptyset$. We are now ready to prove that $Win_1 \subseteq W$. Towards a contradiction we assume the existence of $u \in (Win_1 - W)$. By the definition of Algorithm 1 it follows that there is a subgraph $H \subseteq G$ such that $u \in H$ and the algorithm returns $\emptyset$ when it runs over H. Hence, player-1 winning region in H is empty (namely, $u \notin Win_1$ over game graph H) and by Remark 1 we get that $u \notin Win_1$ in game graph G and the contradiction follows. Thus $Win_1 \subseteq W$. ∎

We present a similar algorithm for the computation of quantitative analysis of quantitative objectives. For this purpose we extend the notion of winning regions to quantitative objectives by defining *value regions*. For a threshold ν we say that a vertex v is in *ν value region* (denoted by $VR(\nu)$) if $\inf_{\sigma \in \mathcal{FM}_1} \sup_{\tau \in \mathcal{S}_2} Val_{\sigma,\tau} = \nu$. Algorithm 2 computes value regions by a call to VALUEREGION$(G, f, -\infty)$, and its correctness follows by the same arguments as in the proof of Lemma 4. We note that if f satisfies Properties 1-3, then by Lemma 3, there is an effective procedure to compute $\inf_{\sigma \in \mathcal{FM}_1} f(CONVEX((G, v)^\sigma))$ (hence, Algorithm 2 can be effectively executed) and if f satisfies Properties 1-4, then by the same lemma we get that there is a procedure to determine whether $f(CONVEX((G, v)^\sigma)) \leq \nu$ (hence, Algorithm 1 can be effectively executed). Hence, we get the main result of this section.

Theorem 1. *Let f be the one-player solution of a quantitative objective.*

- *If f satisfies Properties 1-3, then the corresponding quantitative analysis problem is computable.*
- *If f satisfies Properties 1-4, then the corresponding boolean analysis problem is decidable.*

We note that Theorem 1 provides a recipe for the construction of ϵ-optimal strategies. If the infimum value that player 1 can achieve is ν, then the process that enumerates all $\sigma \in \mathcal{FM}_1$ and halts if the one-player solution of G^σ is at most $\nu + \epsilon$ will always terminate. Similarly, if the boolean analysis problem is decidable, then it is possible to effectively construct a finite-memory strategy that assures the corresponding threshold.

Algorithm 2 Value region computation for quantitative objectives. The algorithm invokes $\textsc{ValueRegion}(G, f, -\infty)$.

```
ValueRegion(G, f, ValLowerBound)
  if G ≠ ∅ then
    for v ∈ G do
      I[v] ← max(inf_{σ∈FM_1} f(CONVEX((G,v)^σ)), ValLowerBound)
    end for
    u ← argmin_{v∈G} I[v]                      /* Choose u s.t I[u] = min_{v∈G} I[v] */
    VR(I[u]) ← VR(I[u]) ∪ Attr_1(u)            /* Add Attr_1(u) to the value region of I[u] */
    return ValueRegion(G−Attr_1(u), f, I[u])   /* Continue the computation recursively. The new lower bound is I[u]. */
  end if
```

V. Games with Mean-Payoff Expression Objectives

In this section, we bring the formal definition of mean-payoff expressions and we use the results of Section IV to analyze games with mean-payoff expressions. We define mean-payoff expressions in Subsection V-A and we analyze mean-payoff expression games in Subsection V-B.

A. Mean-payoff expression objectives

The class of mean-payoff expressions is the closure of single dimension mean-payoff objectives to the algebraic operations of $\min, \max, \mathrm{sum}$ and numerical complement. Formally, for an infinite sequence of reals $\rho = a_1, a_2, \dots \in \mathbb{R}^\omega$, we denote $\mathit{LimInfAvg}(\rho) = \liminf_{n\to\infty} \frac{a_1+\dots+a_n}{n}$ and $\mathit{LimSupAvg}(a_1, a_2, \dots) = \limsup_{n\to\infty} \frac{a_1+\dots+a_n}{n}$. For an infinite sequence of vectors $\rho = v_1, v_2 \dots \in (\mathbb{R}^k)^\omega$ we denote be the projection of ρ to the i-th dimension by ρ_i, and we denote $\mathit{LimInfAvg}_i(\rho) = \mathit{LimInfAvg}(\rho_i)$ and $\mathit{LimSupAvg}_i(\rho) = \mathit{LimSupAvg}(\rho_i)$. An *atomic expression* over $\mathbb{R}^k$ is either $\mathit{LimInfAvg}_i$ or $\mathit{LimSupAvg}_i$. If E_1 and E_2 are expressions, then $-E_1$, $\max(E_1, E_2)$, $\min(E_1, E_2)$ and $\mathrm{sum}(E_1, E_2)$ are also expressions. For a sequence $\rho \in (\mathbb{R}^k)^\omega$ and an expression E, the value of $E(\rho)$ is $\mathit{LimInfAvg}_i(\rho)$ if $E = \mathit{LimInfAvg}_i$, $\mathit{LimSupAvg}_i(\rho)$ if $E = \mathit{LimSupAvg}_i$, $\mathrm{op}(E_1(\rho), E_2(\rho))$ if $E = \mathrm{op}(E_1, E_2)$ (for $\mathrm{op} \in \{\min, \max, \mathrm{sum}\}$) and $-E_1(\rho)$ if $E = -E_1$. Over $\mathbb{R}^2$, a possible expression is $E = \min(\mathit{LimInfAvg}_1, \mathit{LimSupAvg}_1 + \mathit{LimInfAvg}_2) + \max(\mathit{LimInfAvg}_1, \mathit{LimSupAvg}_2)$, and the value of E for the sequence $(-1, 1)^\omega$ is $\min(-1, -1+1) + \max(-1, 1) = 0$.

We say that an expression E is of *normal form* if (i) the numerical complement does not occur in E; and (ii) for every dimension i, there is at most one occurrence of an atomic expression $A_i \in \{\mathit{LimInfAvg}_i, \mathit{LimSupAvg}_i\}$; and (iii) $E = \max(E_1, \dots, E_\ell)$, where E_i is a *max-free* expression (that is, the $\max$ operator does not occur in E_i). The next simple lemma shows that w.l.o.g we may consider only games over normal form expressions.

Lemma 5. *For every k-dimensional weighted graph G with a weight function w and an expression E, we can effectively construct an m-dimensional weight function w' and a normal form expression F such that every infinite path in G gets the same value according to (E, w) and according to (F, w').*

Proof: We can easily overcome the restriction on the number of atomic expressions per dimension by creating several *copies* of the same dimension (that is, additional dimensions with weights that are identical to the original dimension). We can create an equivalent numerical complement free expression by the following recursive process. If $E = -\mathit{LimInfAvg}_i$ (respectively, $E = -\mathit{LimSupAvg}_i$), then we multiply all the weights in dimension i by -1 and define $F = \mathit{LimSupAvg}_i$ (resp. $F = \mathit{LimInfAvg}_i$). F is equivalent to E since $\mathit{LimInfAvg}(a_1, a_2, \dots) = -\mathit{LimSupAvg}(-a_1, -a_2, \dots)$. If $E = -\mathrm{op}(E_1, E_2)$, then we recursively change the weights and construct normal form expressions F_1 and F_2 that are equivalent to $-E_1$ and $-E_2$, and return the normal form expression $F = \mathrm{op}(F_1, F_2)$. And we similarly handle the expression $E = \mathrm{op}(E_1, E_2)$. Finally, if we have a numerical complement free expression E, then we construct an equivalent expression $F = \max(F_1, \dots, F_\ell)$, where F_i is a max free expression, by the following recursive procedure: If E is an atomic expression, then we return $F = \max(E, E)$. If $E = \mathrm{op}(E_1, E_2)$, then we recursively construct two expressions F_1 and F_2, such that F_i is equivalent to E_i and $F_1 = \max(G_1, \dots, G_r)$, $F_2 = \max(H_1, \dots, H_q)$ (where H_i and G_i are max-free expressions), and we return $F = \max_{i\in\{1,\dots,r\}, j\in\{1,\dots,q\}}\{\mathrm{op}(G_i, H_j)\}$. ■

Hence, in the rest of the paper we will assume w.l.o.g that all the expressions are of normal form.

B. Synthesis of a finite-memory controller for mean-payoff expression objectives

In this subsection we apply Theorem 1 to mean-payoff expression objectives. We first prove that the solution for mean-payoff expressions satisfies Properties 1-3 , and thus the quantitative analysis problem is computable for mean-payoff expression games. We then show that the boolean analysis problem is inter-reducible with Hilbert's tenth problem over rationals ($\mathit{H10}(\mathbb{Q})$) by showing that an effective algorithm for $\mathit{H10}(\mathbb{Q})$ implies that mean-payoff expressions satisfy Property 4, and by a reduction from $\mathit{H10}(\mathbb{Q})$ to mean-payoff expression games.

One-player games were solved in [6] and in [15]. We present our solution from [15] to establish properties of the one-player solution. For an expression E and a one-player game (G, v_0), that is, a game over graph G with initial vertex v_0, we say that a threshold ν is *feasible* if the player has a strategy that achieves a value at least ν (we recall that in the one-player setting, the player aim to maximize the value of the objective). The *max-free* constraints were presented in [14] (Section A.4), and they describe the feasible thresholds of a max-free expression.

Definition 1 (Max-free constraints)**.** *Let G be a strongly-connected k-dimensional game graph, and we recall that $C(G)$ is the set of simple cycles of G. Let E be a max-free expression such that the first j dimensions of G occur in E as lim-inf (and the others as lim-sup). We define a variable X^i_c for every simple cycle c and index $i \in \{j+1,\dots,k\}$, and we define a vector of variables $\overline{r} = (r_1,\dots,r_{2k})$. The the max-free constraints for threshold $\nu \in \mathbb{Q}$ are*

1) $\sum_{c\in C(G)} X^i_c Avg_m(c) \geq r_m$ *for every* $i \in \{j+1,\dots,k\}$ *and* $m \in \{1,\dots,j,i\}$
2) $\sum_{c\in C(G)} X^i_c = 1$ *for every* $i \in \{j+1,\dots,k\}$
3) $X^i_c \geq 0$ *for every* $i \in \{j+1,\dots,k\}$ *and* $c \in C(G)$
4) $M_E \times \overline{r} \geq (0,\dots,0,\nu)^T$

where M_E is a matrix that is independent of the graph, and computable from E. (We note that in [14], the first type of constraints was $\sum_{c\in C(G)} X^i_c w_m(c) \geq r_m$ and the second type of constraints was $\sum_{c\in C(G)} |c| X^i_c = 1$. It is straight forward to verify that the constraints are equivalent — in terms of feasibility. In addition, the fourth constraint was presented as $M_E \times \overline{r} \geq \overline{b}_\nu$; but the proof of Lemma 7 in [14] implies that $\overline{b}_\nu = (0,\dots,0,\nu)^T$.) We proved in [15] that a threshold ν is feasible if and only if the corresponding max-free constraints are feasible. For a max-free expression E, a strongly-connected graph G and a threshold ν, we denote the max-free constraints by $\mathrm{MFC}(E,G,\nu)$ and we observe that for a (normal-form) mean-payoff expression $E = \max(E_1,\dots,E_\ell)$ and a strongly-connected graph G, the solution function for the one-player game is $f(G) = \max\{\nu \in \mathbb{R} \mid \exists i \in \{1,\dots,\ell\} \text{ s.t } \mathrm{MFC}(E_i,G,\nu) \text{ is feasible}\}$. By the definition of the max-free constraints, it easily follows that the solution is function first-order definable and continuous (i.e., it satisfies Properties 1 and 3). In the next Lemma we prove that the solution also satisfies the second property.

Lemma 6. *Let E be a mean-payoff expression over k dimensions, and let f be its one-player solution function. Then for every two strongly-connected graphs G and H: if $CONVEX(H) \subseteq CONVEX(G)$, then $f(H) \leq f(G)$.*

Proof: Since we assume that $E = \max(E_1,\dots,E_n)$, where E_i is a max-free expression, it is enough to prove that if a threshold ν is feasible in H for the max-free expression E_i, then it is also feasible in G. Let $C^G_1,\dots,C^G_n$ and $C^H_1,\dots,C^H_m$ be the simple cycles of G and H respectively. We note that since $CONVEX(H) \subseteq CONVEX(G)$, then for every convex combination $x_1,\dots,x_m$, there is a convex combination $y_1,\dots,y_n$ such that $\sum_{i=1}^m x_i Avg(C^H_i) = \sum_{i=1}^n y_i Avg(C^G_i)$. Hence, a solution for the max-free constraints over graph H induces a solution for the max-free constraints over G (by replacing, in the inequalities of constraints 1 over graph H, every convex combination of cycles of H by the corresponding convex combination of cycles of G).

Thus, every threshold that is feasible for H is also feasible for G, and the proof follows. ■

Hence, the one-player solution function of mean-payoff expressions satisfies Properties 1-3 and the next theorem follows.

Theorem 2. *The quantitative analysis problem for mean-payoff expression games (where player 1 is restricted to finite-memory strategies) is computable.*

We now show that the solution for one-player mean-payoff expression games satisfies Property 4 if and only if $H10(\mathbb{Q})$ is decidable. We first prove the direction from right to left.

Lemma 7. *If $H10(\mathbb{Q})$ is decidable, then mean-payoff expressions satisfy the fourth property.*

Proof: Let G be an arbitrary strongly connected graph with n simple cycles, let $C(G) = \{C_1,\dots,C_n\}$ be its set of simple cycles, let $E = \max(E_1,\dots,E_m)$ be a mean-payoff expression (where E_i is a max-free expression), and let ν be a rational threshold. We recall that the sentence $(\zeta_n(Avg(C_1),\dots,Avg(C_n) \leq \nu)$ is equivalent to the statement: "For every $y > \nu$ and $i \in \{1,\dots,m\}$, the constraints $\mathrm{MFC}(E_i,G,y)$ are infeasible." By the definition of the max-free constraints, when the set $Avg(G)$ is fixed the above statement is easily reduced to the infeasibility of m linear systems. Motzkins Transposition Theorem (e.g., Theorem 1 in [2]) gives a witness to the infeasibility of a set of linear inequalities. We use Lagrange four-square Theorem to construct a Diophantine equation that has a rational root if and only if the witness exists. We show that the construction works also when $Avg(G)$ is not fixed, i.e., when $Avg(G) = \{A_1\overline{x_1},\dots,A_1\overline{x_n}\}$ for some n matrices $A_1,\dots,A_n$ and n vectors of rational variables $\overline{x_1},\dots,\overline{x_n} \in \mathcal{QSI}$. (The details of the construction are given in the appendix.) Hence, if $H10(\mathbb{Q})$ is decidable, then mean-payoff expressions satisfy the fourth property. ■

We now prove the reduction from $H10(\mathbb{Q})$ to the boolean analysis of mean-payoff expression games, and we show that there is a reduction even for a simpler subclass of mean-payoff expressions. An expression E is sum-free and *LimInfAvg*-only if only the min and max operators occur in E and all the atomic expressions in E are of the form $LimInfAvg_i$. (In addition, the numerical complement operator also does not occur). The next lemma shows that the boolean analysis problem for sum-free *LimInfAvg*-only expressions is $H10(\mathbb{Q})$-hard.

Lemma 8. *If the boolean analysis problem is decidable for* sum-free *LimInfAvg-only expressions, then $H10(\mathbb{Q})$ is decidable.*

Proof: We only present a rough and informal sketch of the proof. The full proof is given in the appendix. We first show a reduction from $H10(\mathbb{Q})$ to the problem of finding a rational solution for two set of variables $Q = \{q_1,\dots,q_n\}$ and $P = \{p_1,\dots,p_n\}$ and a set of constraints, each of them is with the form: (i) $\sum_{i\in I} \alpha_i q_i \leq 0$, for some $I \subseteq \{1,\dots,n\}$; or (ii) $\sum_{i\in I} \alpha_i p_i \leq 0$, for some $I \subseteq \{1,\dots,n\}$; or (iii) $q_i p_j = q_k p_\ell$ for some $i,j,k,\ell \in \{1,\dots,n\}$; subject to $q_i, p_i > 0$. We then show a reduction from the boolean

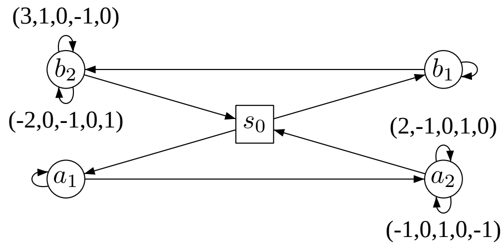

Fig. 7.

analysis problem to the above problem. We illustrate the reduction by showing the construction for the set of constraints $\{q_1 - 2q_2 \leq 0, 2p_1 - 3p_2 \leq 0, p_1q_1 = p_2q_2\}$. For the these constraints we build a game graph G that is illustrated in Figure 7. In the figure we explicitly show only part of the weight vectors and only part of the dimensions. The initial vertex of G is s_0 and this vertex is a player-2 vertex (and the rest are player-1 vertices). The objective of the game is the expression $E = \max(1, \min(2, 3), \min(4, 5))$, where i stands for $\mathit{LimInfAvg}_i$ (for $i = 1, 2, 3, 4, 5$), and the threshold is 0. In G, player 2 has only two memoryless strategies, namely $\tau_1 = s_0 \to b_1$ and $\tau_1 = s_0 \to a_1$. We rely on Lemma 4 and show that player 1 has a winning strategy if and only if there is a finite path π_1 that visits only the self-loops of b_2 and a path π_2 that visits only the self-loops of a_2 such that every vector in $v \in \mathit{CONVEX}(\mathit{Avg}(\pi_1), \mathit{Avg}(\pi_2))$ satisfies the winning condition, i.e., if $v = (v_1, v_2, v_3, v_4, v_5)$, then $\max(v_1, \min(v_2, v_3), \min(v_4, v_5)) \leq 0$. We observe that for any such path π_1 it holds that $\mathit{Avg}(\pi_1) = q_1(3, 1, 0, -1, 0) + q_2(-2, 0, -1, 0, 1)$ and similarly $\mathit{Avg}(\pi_2) = p_1(2, -1, 0, 1, 0) + p_2(-1, 0, 1, 0, -1)$ for some positive rational q_1, q_2, p_1, p_2. We further observe that if $q_1 - 2q_2 > 0$, then $\mathit{Avg}(\pi_1)$ is positive in the first dimension, and thus there is a vector $v \in \mathit{CONVEX}(\mathit{Avg}(\pi_1), \mathit{Avg}(\pi_2))$ that gives a positive value to the expression. Hence, it must hold that $q_1 - 2q_2 \leq 0$ and similarly $2p_1 - 3p_2 \leq 0$. Moreover, we prove that if $p_1q_1 \neq p_2q_2$, then there exists $v \in \mathit{CONVEX}(\mathit{Avg}(\pi_1), \mathit{Avg}(\pi_2))$ that is positive either in dimensions 2 and 3 or in dimensions 4 and 5. Hence, it must hold that $p_1q_1 = p_2q_2$ and the proof follows. ■

The next theorem summarize the results of Lemmas 8 and 7

Theorem 3. *The boolean analysis problem for mean-payoff expression games (when player 1 is restricted to finite-memory strategies) is inter-reducible with $H10(\mathbb{Q})$, and it is $H10(\mathbb{Q})$-hard even for* sum*-free $\mathit{LimInfAvg}$-only expressions.*

We also consider the case where both players are restricted to finite-memory strategies. In this setting, the quantitative analysis problem is to compute $\inf_{\sigma \in \mathcal{FM}_1} \sup_{\tau \in \mathcal{FM}_2} \mathit{Val}_{\sigma,\tau}$. The boolean analysis problem is to determine whether player 1 has a finite-memory strategy that assures a value of at most ν against any player-2 finite-memory strategy.

Theorem 4. *When both players are restricted to finite-memory strategies: (i) the quantitative analysis problem for mean-payoff expression games is computable; (ii) the boolean analysis problem for mean-payoff expression games is inter-reducible with $H10(\mathbb{Q})$, and it is $H10(\mathbb{Q})$-hard even for* sum*-free $\mathit{LimInfAvg}$-only expressions.*

Proof: We present a sketch of the proof. The full proof is given in the appendix. Informally, when both players are restricted to finite-memory strategies, the outcome of a play is an ultimately periodic path, and thus we may assume that all the atomic expressions are of the form of $\mathit{LimInfAvg}$ (because for periodic paths we have $\mathit{LimInfAvg}(\pi) = \mathit{LimSupAvg}(\pi)$). We also show that if all the atomic expressions are $\mathit{LimInfAvg}$ and player 1 is restricted to finite-memory strategies, then player 2 can achieve a value greater than ν if and only if he can do it with a finite-memory strategy, and the proof follows. ■

As a final remark, we note that while the boolean analysis for sum-free $\mathit{LimInfAvg}$-only expressions is $H10(\mathbb{Q})$-hard when player 1 is restricted to a finite-memory strategy (and also when both players are restricted to finite-memory strategies), the next lemma shows that the problem is decidable when both players may use arbitrary strategies.

Lemma 9 (Theorem 5 in [16])**.** *When both players may use arbitrary strategies, the boolean analysis of* sum*-free $\mathit{LimInfAvg}$-only expression games is decidable.*

Proof: The proof follows from Theorem 5 in [16] due to the fact that there is an immediate translation from sum-free $\mathit{LimInfAvg}$-only expressions to the $\bigvee\bigwedge \mathrm{MeanPayoffInf}^{\leq}(\nu)$ objectives that were defined in [16]. ■

References

[1] R. Alur, A. Degorre, O. Maler, and G. Weiss. On omega-languages defined by mean-payoff conditions. FOSSACS, 2009.

[2] A. Ben-Israel. Motzkins transposition theorem, and the related theorems of farkas, gordan and stiemke. *Encyclopaedia of Mathematics*, 2001.

[3] U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *LICS*, 2011.

[4] T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Two views on multiple mean-payoff objectives in markov decision processes. In *LICS*, 2011.

[5] T. Brázdil, K. Chatterjee, A. Kucera, and P. Novotný. Efficient controller synthesis for consumption games with multiple resource types. In *CAV*, 2012.

[6] K. Chatterjee, L. Doyen, H. Edelsbrunner, T. A. Henzinger, and P. Rannou. Mean-payoff automaton expressions. In *CONCUR*, 2010.

[7] K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages. *LICS*, 2009.

[8] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 2010.

[9] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *FSTTCS*, 2010.

[10] K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *CONCUR*, 2012.

[11] M. Droste and G. Rahonis. Weighted automata and weighted logics on infinite words. In *Developments in Language Theory*, 2006.

[12] E. Kopczyński. Half-positional determinacy of infinite games. *ICALP*, 2006.

[13] A. Tarski. A decision method for elementary algebra and geometry. 1951.

[14] Y. Velner. The complexity of mean-payoff automaton expression. *CoRR*, 2011.

[15] Y. Velner. The complexity of mean-payoff automaton expression. *ICALP*, 2012.

[16] Y. Velner and A. Rabinovich. Church synthesis problem for noisy input. In *FOSSACS*, 2011.

Appendix A
Proof of Lemma 7

Let G be an arbitrary strongly connected graph with n simple cycles, let $C(G) = \{C_1, \dots, C_n\}$ be its set of simple cycles, let $E = \max(E_1, \dots, E_m)$ be a mean-payoff expression (where E_i is a max-free expression), and let ν be a rational threshold. We recall that the sentence $(\zeta_n(Avg(C_1), \dots, Avg(C_n) \leq \nu))$ is equivalent to the statement:

For every $y > \nu$ and $i \in \{1, \dots, m\}$, the constraints $\mathrm{MFC}(E_i, G, y)$ are infeasible.

By the definition of the max-free constraints, when the set $Avg(G)$ is fixed the above statement is easily reduced to the infeasibility of m linear systems, each of them is of the form:

$$A^i_{Avg(G)}\overline{x} \leq \overline{b^i} \text{ and } B^i_{Avg(G)}\overline{x} < \overline{c^i}$$

By Motzkins Transposition Theorem (e.g., Theorem 1 in [2]) the infeasibility of a linear system $A^i_{Avg(G)}\overline{x} \leq \overline{b^i}$ and $B^i_{Avg(G)}\overline{x} < \overline{c^i}$ is equivalent to the existence of two non-negative vectors $\overline{y}, \overline{z} \geq \overline{0}$ such that either

- $\overline{z} = 0$ and $(A^i_{Avg(G)})^T\overline{y} = 0$ and $\overline{b^i}^T\overline{y} < 0$; or
- $\overline{z} \neq 0$ and $(A^i_{Avg(G)})^T\overline{y} + (B^i_{Avg(G)})^T\overline{z} = 0$ and $\overline{b^i}^T\overline{y} + \overline{c^i}^T\overline{z} \leq 0$

Since every linear inequality has a rational solution (when the coefficients are rational) we get that if such $\overline{y}$ and $\overline{z}$ exist, then there also exist rational $\overline{y}$ and $\overline{z}$ that satisfy the above. Hence the above statement is equivalent to the rational feasibility of the following constraints (for variables $\overline{y} = (y_1, \dots, y_r), \overline{z} = (z_1, \dots, z_r), p_1, p_2$ and q):

- $p_1 > 0, p_2 > 0, q \geq 0$
- $\overline{y} \geq \overline{0}, \overline{z} \geq \overline{0}$
- $(A^i_{Avg(G)})^T\overline{y} + (B^i_{Avg(G)})^T\overline{z} = 0$
- $(\sum_{j=1}^r z_i - p_1)(\overline{b^i}^T\overline{y} + p_2) = 0$
- $(\sum_{j=1}^r z_i)(\overline{b^i}^T\overline{y} + \overline{c^i}^T\overline{z} + q) = 0$

By Lagrange's four-square Theorem, every natural number is the sum of four integer squares. Therefore, every inequality of the form $x \geq 0$ is equivalent to the rational feasibility of the equation

$$x = \frac{x_1^2 + x_2^2 + x_3^2 + x_4^2}{1 + x_5^2 + x_6^2 + x_7^2 + x_8^2}$$

and every inequality of the form $x > 0$ is equivalent to the rational feasibility of the equation

$$x = \frac{1 + x_1^2 + x_2^2 + x_3^2 + x_4^2}{1 + x_5^2 + x_6^2 + x_7^2 + x_8^2}$$

and the equations of the above form can be easily transformed into Diophantine equations. Hence, we get that the infeasibility of a linear system

$$A^i_{Avg(G)}\overline{x} \leq \overline{b^i} \text{ and } B^i_{Avg(G)}\overline{x} < \overline{c^i}$$

is equivalent to the rational feasibility of several Diophantine equations $D_1 = 0, D_2 = 0, \dots, D_r = 0$, and therefore it is equivalent to the rational feasibility of $D^i = D_1^2 + \cdots + D_r^2 = 0$. Therefore, when the set of simple cycles is fixed, the simultaneous infeasibility of all the max-free constraints is equivalent to the rational feasibility of the Diophantine equation $D_{Avg(G)} = \sum_{i=1}^m (D^i)^2 = 0$. We also note that if $Avg(G)$ is not fixed, that is $Avg(C_i)$ is a vector of variables (for $i = 1, \dots, n$), then $D_{Avg(C)} = 0$ remains a Diophantine equation.

We are now ready to prove that the solution for one-player mean-payoff games is satisfies Property 4 (if $H10(\mathbb{Q})$ is decidable). For n matrices $A_1, \dots, A_n$ the rational satisfiability of $\zeta_n(A_1\overline{x_1}, \dots, A_n\overline{x_n}) \leq \nu$ is equivalent to the existence of a rational solution to $D_{Avg(G)} = 0$ (for $Avg(G) = \{A_1\overline{x_1}, \dots, A_n\overline{x_n}\}$). We can encode the requirement that $x_1, \dots, x_n \in \mathcal{QSI}$ by a Diophantine equations $K(\overline{x_1}, \dots, \overline{x_n}) = 0$ by the same techniques we used for the construction of $D_{Avg(G)} = 0$. Hence, the satisfiability of $\zeta_n(A_1\overline{x_1}, \dots, A_n\overline{x_n}) \leq \nu$ is equivalent to the existence of a rational solution to the Diophantine equation $K^2 + D^2 = 0$, and if $H10(\mathbb{Q})$ is decidable, then we can effectively determine whether $K^2 + D^2 = 0$ has a rational solution and the proof follows.

Appendix B
Proof of Lemma 8

We prove Lemma 8 in the next three subsections. In the first subsection we present an alternative formulation for $H10(\mathbb{Q})$. In the second subsection we prove a simple technical lemma on vectors. In the third subsection we present a reduction from the problem we presented in the first subsection to sum-free *LimInfAvg*-only games, and the reduction relies on the lemma that we prove in the second subsection.

We note that the first two subsection are technical and tedious, but they relay only on basic algebra.

A. Alternative formulations of $H10(\mathbb{Q})$

In this subsection, we present five problems; the first problem is $H10(\mathbb{Q})$, and we show a reduction from the i-th problem to the $i+1$-th problem, for $i = 1, 2, 3, 4$. Thus, we get that there is a reduction from $H10(\mathbb{Q})$ to the fifth problem (that is, Problem 6), and in the third subsection we will show a reduction from that problem to mean-payoff expression games.

Problem 2 ($H10(\mathbb{Q})$)**.** *For a polynomial P, find a rational solution to*

$$P(q_1, \dots, q_n) = 0$$

Problem 3. *Find a rational solution to*

$$q_0 \cdot P(\frac{q_1}{q_0}, \dots, \frac{q_n}{q_0}) = 0$$

(for a polynomial P) subject to

- *$q_0 \leq q_i$ for every $i = 1, \dots, n$; and*
- *$q_i \geq 1$ for every $i = 0, \dots, n$.*

Lemma 10. *There is a reduction from $H10(\mathbb{Q})$ to Problem 3.*

Proof: We first note that we can easily reduce $H10(\mathbb{Q})$ to the problem of finding a rational solution for the polynomial equation $D(q_1, q_2, \dots, q_n) = 0$ subject to

$q_1, q_2, \ldots, q_n \geq 1$. (The reduction is trivial, a polynomial equation $P(q_1, \ldots, q_n) = 0$ has a solution if and only if the polynomial equation $D(p_1, \ldots, p_{2n}) = P(p_1 - p_2, p_3 - p_4, \ldots, p_{2n-1} - p_{2n}) = 0$ has a solution that satisfies $p_i \geq 1$.) We define $P = D$ and we note that $q_0 \cdot P = 0$ has a rational solution (subject to $q_0 \geq 1$) if and only if $P = 0$ has a rational solution, and it is trivial to observe that $P = 0$ has a rational solution (subject to $q_0 \geq q_i$ and $q_i \geq 1$) if and only if $D = 0$ has a rational solution (subject to $q_i \geq 1$). ■

Problem 4. *For a given a set of variables $Q = \{q_1, \ldots, q_n\}$, and a set of equations such that at most one equation is of the form*

$\sum_{i \in I} \alpha_i q_i = 0$, for some $I \subseteq \{0, \ldots, n\}$ and $\alpha_i \in \mathbb{Q}$ for every $i \in I$.

and all the other equations are of the form

$q_i q_j = q_k q_\ell$ for some $i, j, k, \ell \in \{0, \ldots, n\}$.

find a rational solution that satisfies $1 \leq q_0 \leq q_i$ for every $i = 1, \ldots, n$.

Lemma 11. *There is a reduction from Problem 3 to Problem 4.*

Proof: We prove the lemma by giving a generic example that demonstrates the reduction. Suppose that the equation with the form of Problem 3 is $q_0 \cdot P(\frac{q_1}{q_0}, \ldots, \frac{q_n}{q_0}) = 5\frac{q_1^2 q_2 q_3^3}{q_0^5} + \frac{q_1^2}{q_0} + 7q_0$, then we reduce it to a problem with the form of Problem 4 by defining the following equations:

- $p_0 \cdot q_0 = q_0 \cdot q_0$ (equivalent to $p_0 = q_0$)
- $p_1 \cdot q_0 = q_1 \cdot q_1$ (equivalent to $p_1 = \frac{q_1^2}{q_0}$)
- $p_2 \cdot q_0 = q_3 \cdot q_3$ and $p_3 \cdot q_0 = p_2 \cdot q_3$ (equivalent to $p_3 = \frac{q_3^3}{q_0^2}$)
- $p_4 \cdot q_0 = p_1 \cdot p_3$ (equivalent to $p_4 = \frac{q_1^2 q_3^3}{q_0^4}$)
- $p_5 \cdot q_0 = p_4 q_2$ (equivalent to $p_5 = \frac{q_1^2 q_2 q_3^3}{q_0^5}$)
- $5p_5 + p_1 + 7q_0 = 0$, subject to $1 \leq q_0 \leq q_1, q_2, q_3, p_0, p_1, p_2, p_3, p_4, p_5$ and $q_i, p_j \geq 1$ (equivalent to $q_0 \cdot P(\frac{q_1}{q_0}, \ldots, \frac{q_n}{q_0}) = 0$)

A solution to the above equations that satisfies $1 \leq q_0 \leq q_1, q_2, q_3, p_1, p_2, p_3, p_4, p_5$ is clearly a solution for $q_0 \cdot P = 0$ that satisfies Problem 3 conditions. Conversely, a solution to $q_0 \cdot P = 0$ that satisfies Problem 3 conditions is a solution for the above constraints, and since $1 \leq q_0 \leq q_1, q_2, q_3$ we also get that $q_0 \leq p_1, p_2, p_3, p_4, p_5$ and a solution to the above equitations follows. ■

Problem 5. *For a given sets of variables $Q = \{q_1, \ldots, q_n\}$, $P = \{p_1, \ldots, p_n\}$, and a given set of equations, each of the form of either:*

- *$\sum_{i \in I} \alpha_i q_i = 0$, for some $I \subseteq \{1, \ldots, n\}$; or*
- *$q_i p_j = q_k p_\ell$ for some $i, j, k, \ell \in \{1, \ldots, n\}$; or*
- *$q_i = \frac{1}{2}\sum_{j=1}^n q_j$, for some $i \in \{1, \ldots, n\}$; or*
- *$p_i = \frac{1}{2}\sum_{j=1}^n p_j$, for some $i \in \{1, \ldots, n\}$; or*

find a rational solution that satisfies

- *$q_1 \leq q_i$, for $i = 1, \ldots, n$; and*
- *$q_i, p_i \geq 1$ for $i = 1, \ldots, n$; and*
- *$\sum_{i=1}^n p_i = \sum_{i=1}^n q_i$.*

Lemma 12. *There is a reduction from Problem 4 to Problem 5.*

Proof: To show a reduction, we need to show how to encode an equation of the form of $q_1 q_2 = q_3 q_4$ with equations of the above form. For this purpose we define the equations:

- $q_{n+1} = \frac{1}{2}\sum_{j=1}^{n+1} q_j$ and $p_{n+1} = \frac{1}{2}\sum_{j=1}^{n+1} p_j$
- $q_2 p_{n+1} = q_{n+1} p_1$
- $q_4 p_{n+1} = q_{n+1} p_2$
- $q_1 p_1 = q_3 p_2$

It is straight forward to observe that if $\sum_{j=1}^{n+1} q_j = \sum_{j=1}^{n+1} p_j$ then the above set of equations are equivalent to $q_1 q_2 = q_3 q_4$. ■

Problem 6. *For a given sets of variables $Q = \{q_1, \ldots, q_n\}$, $P = \{p_1, \ldots, p_n\}$, and a given set of constraints, each of the form of either:*

- *$\sum_{i \in I} \alpha_i q_i \leq 0$, for some $I \subseteq \{1, \ldots, n\}$; or*
- *$\sum_{i \in I} \alpha_i p_i \leq 0$, for some $I \subseteq \{1, \ldots, n\}$; or*
- *$q_i p_j = q_k p_\ell$ for some $i, j, k, \ell \in \{1, \ldots, n\}$*

find a rational solution that satisfies

- *$q_i, p_i > 0$ for $i = 1, \ldots, n$*

Lemma 13. *There is a reduction from Problem 5 to Problem 6.*

Proof: The reduction is straight forward. We replace every equation of the form of $\sum_{i \in I} \alpha_i q_i = 0$ with two constraints $\sum_{i \in I} \alpha_i q_i \leq 0$ and $\sum_{i \in I} -\alpha_i q_i \leq 0$. We replace $q_i = \frac{1}{2}\sum_{j=1}^n q_j$ with $\sum_{j \in \{1,\ldots,n\}-\{i\}} \frac{1}{2}q_j - \frac{1}{2}q_i \leq 0$ and $\sum_{j \in \{1,\ldots,n\}-\{i\}} -\frac{1}{2}q_j + \frac{1}{2}q_i \leq 0$. We replace $p_i = \frac{1}{2}\sum_{j=1}^n p_j$ with $\sum_{j \in \{1,\ldots,n\}-\{i\}} \frac{1}{2}p_j - \frac{1}{2}p_i \leq 0$ and $\sum_{j \in \{1,\ldots,n\}-\{i\}} -\frac{1}{2}p_j + \frac{1}{2}p_i \leq 0$. In addition, we add n constraints $q_1 \leq q_i$ for $i = 1, \ldots, n$. It is straight forward to observe that if the above formed constraints have a rational solution $Q = \{q_1, \ldots, q_n\}, P = \{p_1, \ldots, p_n\}$ that satisfies $q_i, p_i > 0$, then for every rational positive m we get that $mQ = \{mq_1, \ldots, mq_n\}, P = \{p_1, \ldots, p_n\}$ and $Q = \{q_1, \ldots, q_n\}, mP = \{mp_1, \ldots, mp_n\}$ are also solutions. Hence, a solution to the formed constraints implies that there is a solution that satisfies $q_i, p_i \geq 1$ and $\sum_{i=1}^n p_i = \sum_{i=1}^n q_i$. And conversely, if the formed constraints are not satisfiable, then clearly the original equations are not solvable. ■

B. Auxiliary lemma

In this subsection, we prove the next lemma.

Lemma 14. *Let $\alpha_1, \alpha_2, \beta_1, \beta_2$ be strictly positive rationals, and let $v_1(\alpha_1) = \alpha_1 \cdot (-1, 0, 1, 0)$, $v_2(\alpha_2) = \alpha_2 \cdot (0, 1, 0, -1)$, $u_1(\beta_1) = \beta_1 \cdot (1, 0, -1, 0)$, and $u_2(\beta_2) = \beta_2 \cdot (0, -1, 0, 1)$. For every $m, n \in \mathbb{Q}$ we denote by the vector $x(m, n) = (x_1, x_2, x_3, x_4)$ the sum $m(v_1 + v_2) + n(u_1 + u_2)$. Then the following assertions are equivalent:*

1) *$\frac{\beta_1}{\alpha_1} = \frac{\beta_2}{\alpha_2}$.*
2) *For every non-negative rationals m, n: $\max(\min(x_1, x_2), \min(x_3, x_4)) \leq 0$.*

Proof: By definition $x_1 = -m\alpha_1 + n\beta_1$, $x_2 = m\alpha_2 - n\beta_2$, $x_3 = -x_1$ and $x_4 = -x_2$.

We first prove that assertion 1 implies assertion 2. Suppose that $\frac{\beta_1}{\alpha_1} = \frac{\beta_2}{\alpha_2}$, let m and n be arbitrary non-negative rationals, and we denote $k = \frac{m}{n}$. In order to prove that

$\max(\min(x_1,x_2),\min(x_3,x_4)) \le 0$, it is enough to show that if $x_1 > 0$, then $x_2 < 0$ (since in this case $x_3 = -x_1 < 0$). Suppose that $x_1 > 0$. Hence, $\beta_1 > k\alpha_1$, and we get that $k < \frac{\beta_1}{\alpha_1}$. By definition, $x_2 = n(k\alpha_2 - \beta_2)$, and since we assumed that $\frac{\beta_1}{\alpha_1} = \frac{\beta_2}{\alpha_2}$, and we proved that $k < \frac{\beta_1}{\alpha_1}$, we get that $x_2 < 0$, and the claim that assertion 1 implies assertion 2 follows.

In order to prove that assertion 2 implies assertion 1, we consider two distinct cases. In the first case we assume (towards a contradiction) that $\frac{\beta_1}{\alpha_1} > \frac{\beta_2}{\alpha_2}$, and we choose m and n that satisfy $\frac{\beta_1}{\alpha_1} > k = \frac{m}{n} > \frac{\beta_2}{\alpha_2}$. We claim that $x_1 > 0$ and $x_2 > 0$, and therefore a contradiction to the assumption that $\max(\min(x_1,x_2),\min(x_3,x_4)) \le 0$ follows. Indeed, since $\frac{\beta_1}{\alpha_1} > k$, then $x_1 = n(-k\alpha_1 + \beta) > 0$, and since $k > \frac{\beta_2}{\alpha_2}$, then $x_2 = n(k\alpha_2 - \beta_2) > 0$. In the second case, we assume that $\frac{\beta_1}{\alpha_1} < \frac{\beta_2}{\alpha_2}$, and by similar arguments, we get that $x_3, x_4 > 0$ and a contradiction follows. Hence, in both cases we get that assertion 2 implies assertion 1, and the proof of the lemma follows. ■

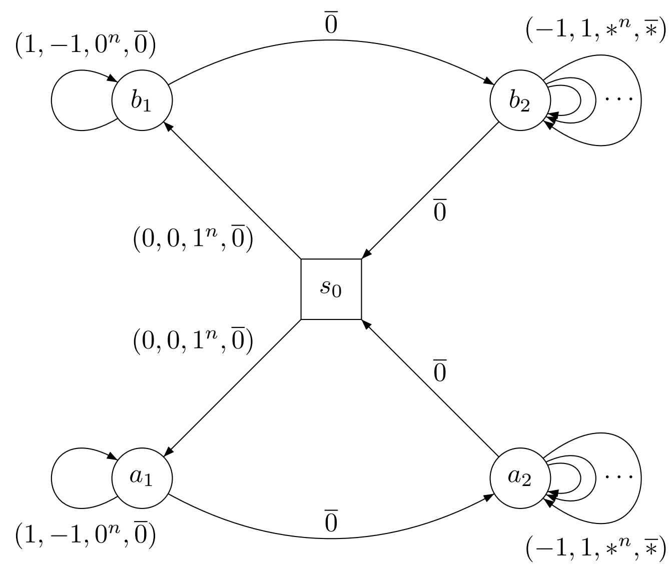

Fig. 8. The graph that is formed by the reduction. s_0 is player-2 state and a_1, a_2, b_1, b_2 are player-1 states. $\overline{0}$ denotes a vector of zeros; 1^n denotes a vector of n ones, 0^n denotes a vector of n zeros, and the weights of $\overline{*}$ and $*^n$ are given in the description of the reduction

C. *The reduction*

In this subsection, we present a reduction from Problem 6 to the boolean synthesis problem for mean-payoff expressions (when player 1 is restricted to finite-memory strategies). The reduction is as following: For a given sets of variables $Q = \{q_1,\dots,q_n\}$, $P = \{p_1,\dots,p_n\}$, and a given set of constraints, each of the form of either:

- $\sum_{i\in I}\alpha_i q_i \le 0$, for some $I \subseteq \{1,\dots,n\}$; or
- $\sum_{i\in I}\alpha_i p_i \le 0$, for some $I \subseteq \{1,\dots,n\}$; or
- $q_i p_j = q_k p_\ell$ for some $i,j,k,\ell \in \{1,\dots,n\}$

We denote by t_1 the number of constraints that are of the first form, and w.l.o.g we assume that the number of constraints that are of the second form is also t_1. We denote by t_2 the number of constraints that are of the third form. We construct a $k = 2+n+t_1+4t_2$ dimensional game graph with 5 states (see Figure 8), and an expression

$$E = \max(\mathit{LimInfAvg}_1,\dots,\mathit{LimInfAvg}_{2+n+t_1}, E_1,\dots,E_{2t_2})$$

where

$$E_i = \min(\mathit{LimInfAvg}_{2+n+t_1+2i}, \mathit{LimInfAvg}_{2+n+t_1+2i+1})$$

The transitions of the graph are described in Figure 8, and each of the states a_2 and b_2 has n self-loop edges.

The weight vector $\overline{w}$ of the i-th self-loop edge of state a_2 is determined according to the next rules:

1) The first two dimensions of $\overline{w}$ are -1 and $+1$ (respectively). Intuitively, this assures that player 1 will not stay forever in state a_1 or in state a_2.
2) The weight of dimension $2+i$ is -1 and for $j \in \{1,\dots,n\} - \{i\}$ the weight of dimensions j is 0. Intuitively, this assures that player 1 will visit edge i at least once.
3) If the j-th type-1 equation is $\sum_{m\in I}\alpha_m q_m \le 0$, then if $i \in I$, then the weight in dimension $2+n+j$ is $-\alpha_m$. Otherwise, we assign zero for this dimension. Intuitively, this enforce player 1 to visits edge i for q_i times in such way that $\sum_{m\in I}\alpha_m q_m \le 0$.
4) If the j-th type-3 equation is $q_m p_r = q_k p_\ell$, then the weights of the four dimensions $2+n+t_1+4j, 2+n+t_1+4j+1, 2+n+t_1+4j+2, 2+n+t_1+4j+3$ are:
   - If $i = m$, then the weights are $(-1,0,1,0)$
   - If $i = k$, then the weights are $(0,1,0,-1)$
   - Otherwise, the weights are $(0,0,0,0)$

The weight vector $\overline{w}$ of the i-th self-loop edge of state b_2 is determined according to the next rules:

1) The first $2+n$ dimensions are determined by the same rules that we presented to the self-loop edges of state a_2.
2) If the j-th type-2 equation is $\sum_{m\in I}\alpha_m p_m \le 0$, then if $i \in I$, then the weight in dimension $2+n+j$ is $-\alpha_m$. Otherwise, we assign zero for this dimension. Intuitively, this enforce player 1 to visits edge i for p_i times in such way that $\sum_{m\in I}\alpha_m p_m \le 0$.
3) If the j-th type-3 equation is $q_m p_r = q_k p_\ell$, then the weights of the four dimensions $2+n+t_1+4j, 2+n+t_1+4j+1, 2+n+t_1+4j+2, 2+n+t_1+4j+3$ are:

- If $i = m$, then the weights are $(1, 0, -1, 0)$
- If $i = k$, then the weights are $(0, -1, 0, 1)$
- Otherwise, the weights are $(0, 0, 0, 0)$

In the rest of this subsection, we will prove that player 1 has a finite-memory strategy that assures non-positive value for the expression E if and only if the given set of equations has a solution that satisfies Problem 6 limitations.

In the next lemmas we prove key properties of the game. The first lemma characterized the one-player game solution for the expression E.

Lemma 15. *Let G be an arbitrary strongly connected k-dimensional weighted one-player game graph, and let f be the one-player solution for the expression E. Then $f(G) > 0$ if and only if*

- *G has a simple cycle with positive average weight in a dimension $i \in \{1, \dots, 2 + n + t_1\}$; or*
- *G has two simple cycles C_1 and C_2, and there exist an index $i \in \{1, \dots, 2t_2\}$ and two positive rationals m, n for which*

  $mAvg(C_1) + nAvg(C_2)$ is positive is dimension $2 + n + t_1 + 2i$ and in dimension $2 + n + t_1 + 2i + 1$.

Proof: The proof follows directly by the definitions of the max-free constraints (Definition 1). ■

Lemma 16. *In the mean-payoff expression game over game graph G (that is constructed by the reduction) and threshold 0, player 1 wins from vertex s_0 if and only if he has a finite-memory strategy σ such that $f(CONVEX(G^\sigma)) \leq 0$ (where f is the one-player solution for the expression E).*

Proof: By the construction of G it follows that if player 1 strategy is to loop for ever in state a_1 or b_1, then the lim-inf of the average weight in dimension 1 will be 1 and E will get a positive value. Similarly, if player 1 strategy is to loop forever in state b_2 or a_2, then the average weight in dimension 2 is positive, and so does the value of E. Hence, every player-1 winning strategy will visit the initial state s_0 infinitely often. Therefore, if σ' is a player-1 winning strategy, then every SCC in $G^{\sigma'}$ contains a vertex (s_0, m) (for some memory state m). Let S be a terminal SCC in $G^{\sigma'}$ and let (s_0, m) be a vertex in S. We construct the witness strategy σ by changing the initial memory state o σ' to m. If σ' is a winning strategy, then by definition $f(S) \leq 0$ and since $G^\sigma = S$ we get that $f(CONVEX(G)) \leq 0$.

Hence, if player 1 wins in the game, then such σ exists, and the proof for the converse direction is trivial (since such a strategy σ is a winning strategy). ■

In the game graph G player 2 has only two possible memoryless strategies: the first strategy is to follow the edge (s_0, a_1), and we denote this strategy by τ_1, and the second strategy is to follow (s_0, b_1), and we denote it by τ_2.

Lemma 17. *There exists a player-1 strategy for which $f(G^\sigma) \leq 0$ if and only if there exist cyclic paths π_1 and π_2 such that π_i is a cyclic path in G^{τ_i} that visits all the edges of G^{τ_i} and $f(CONVEX(\pi_1, \pi_2)) \leq 0$.*

Proof: By Lemma 1 such σ exists if and only if there exist two ultimately periodic paths ρ_1 and ρ_2 such that ρ_i is an infinite path in the graph G^{τ_i} and $f(CONVEX(Avg(\rho_1), Avg(\rho_2))) \leq 0$. Hence, the proof for the direction from right to left follows. In order to prove the converse direction we assume that such ρ_1 and ρ_2 exist and show how to construct π_1 and π_2. Let $\rho_1 = \pi_0(\pi_1)^\omega$ (i.e., π_1 is the periodic finite path in ρ_1). We claim that if π_1 does not contain all the edges of G^{τ_1}, then $f(\{Avg(\pi_1)\}) > 0$. The proof of the claim is by considering the following distinct cases:

- Case 1: if π_1 contains only the cycles $s_0 \to a_1 \to a_2 \to s_0$, then the value of $Avg(\pi)$ is positive in the third dimension.
- Case 2: if π_1 contains only the self loop of a_1, then the value of the first dimension of $Avg(\pi)$ is positive
- Case 3: if π_1 does not contain the self loop of a_1, and contains some of the self loops of a_2, then the second dimension of $Avg(\pi)$ is positive.
- Case 4: if π_1 contains the cycle $s_0 \to a_1 \to a_2 \to s_0$, the self loop of a_1 and not the i-th self loop of a_2, then dimension $2 + i$ of $Avg(\pi)$ is positive.

Hence, if π_1 does not contain all the edges of G^{τ_1}, then we get that $f(Avg(\rho_1)) > 0$ (since $Avg(\rho_1) = Avg(\pi_1)$), and since f is monotone, we get that $f(CONVEX(Avg(\rho_1), Avg(\rho_2)) > f(Avg(\rho_1)) > 0$, which contradict the definition of ρ_1. We construct the witness path π_2 in a similar way (i.e., by defining $\rho_2 = \pi'_0(\pi_2)^\omega$, and the proof that π_2 contains all the edges of G^{τ_2} is similar. Since $Avg(\pi_i) = Avg(\rho_i)$, we get that $f(CONVEX(Avg(\pi_1), Avg(\pi_2)) \leq 0$ and the proof is complete. ■

We now give two additional definitions and then prove the correctness of the reduction. Let $C_1, \dots, C_n$ be the simple cycles of G^{τ_1}. We denote $\mathcal{QSI}(G^{\tau_1}) = \{\{v \in \mathbb{Q}^k \mid \exists (x_1, \dots, x_n) \in \mathcal{QSI}(n) \text{ s.t } v = \sum_{i=1}^n x_i Avg(C_i)\}$, and we similarly define $\mathcal{QSI}(G^{\tau_2})$. We say that two vectors v_1 and v_2 are *satisfactory* if $f(CONVEX(v_1, v_2)) \leq 0$. We are now ready to prove the correctness of the reduction, and by Lemma 16 and Lemma 17 it is enough to prove that there exists $v_i \in \mathcal{QSI}(G^{\tau_i})$ (for $i = 1, 2$) such that v_1, v_2 are satisfactory vectors if and only if the given set of equations has a rational solution.

We first prove the direction from right to left. Suppose that the given set of equations has a rational solution P, Q that satisfies $q_i, p_i > 0$. We construct the vector $v_1 \in \mathcal{QSI}(G^{\tau_1})$ by taking $\frac{1}{1+2\sum_{i=1}^n q_i}$ fraction of the average weight of the cycle $s_0 \to a_1 \to a_2 \to s_0$, $\frac{2\sum_{i=1}^n q_i}{1+2\sum_{i=1}^n q_i}$ fraction of the average weight of the self loop of a_1 and $\frac{q_i}{1+2\sum_{i=1}^n q_i}$ fraction of the average weight of the i-th self loop of a_2. Similarly, we construct the vector $v_2 \in \mathcal{QSI}(G^{\tau_2})$ by taking $\frac{1}{1+2\sum_{i=1}^n p_i}$ fraction of the average weight of the cycle $s_0 \to b_1 \to b_2 \to s_0$, $\frac{2\sum_{i=1}^n p_i}{1+2\sum_{i=1}^n p_i}$ fraction of the average weight of the self loop of b_1 and $\frac{p_i}{1+2\sum_{i=1}^n p_i}$ fraction of the average weight of the i-th self loop of b_2. By the construction of G, and since P and Q are solutions for the equations, it is straight forward to verify that the first $2+n+t_1$ dimensions of v_1 and v_2 are non-positive. In addition, by Lemma 14, and since P and Q satisfies all the equations of the form $q_i p_j = q_k p_\ell$, we get that for every positive $m, n \in \mathbb{Q}$ we have that $mv_1 + nv_2$ are non-positive in dimension $2+n+t_1+2i$ or in dimension $2+n+t_1+2i+1$ for every $i = 1, \ldots, 2t_4$. Hence, by Lemma 15, the vectors v_1, v_2 are satisfactory.

Conversely, suppose that there exist $v_i \in \mathcal{QSI}(G^{\tau_i})$ (for $i = 1, 2$) such that v_1, v_2 are satisfactory vectors. We denote by $w_{s_0,a}$ the average weight of the cycle $s_0 \to a_1 \to a_2 \to s_0$, by w_{a_1} the average weight of the self loop of a_1, and by $w^i_{a_2}$ the average weight of the i-th self loop of a_2. By definition, there exists $n+2$ positive rationals $x, y, q_1, \ldots, q_n$ for which $v_1 = xw_{s_0,a} + yw_{a_1} + \sum_{i=1}^{n} q_i w^i_{a_2}$. Similarly, we denote by $w_{s_0,b}$ the average weight of the cycle $s_0 \to b_1 \to b_2 \to s_0$, by w_{b_1} the average weight of the self loop of b_1, and by $w^i_{b_2}$ the average weight of the i-th self loop of b_2, and by definition, there exists $n+2$ positive rationals $x, y, p_1, \ldots, p_n$ for which $v_1 = xw_{s_0,b} + yw_{b_1} + \sum_{i=1}^{n} p_i w^i_{b_2}$. We claim the $Q = \{q_1, \ldots, q_n\}, P = \{p_1, \ldots, p_n\}$ are a solution to the given set of equations. By Lemma 15 and by the construction of the graph, it immediately follows that Q and P satisfy all the type-1 and type-2 constraints. In addition, by Lemma 14 (and by Lemma 15) we get that all the type-3 equations are also satisfied. Hence, we get that if there exist $v_i \in \mathcal{QSI}(G^{\tau_i})$ (for $i = 1, 2$) such that v_1, v_2 are satisfactory vectors, then the given set of constraints have a solution.

To conclude, we get that the boolean analysis problem for mean-payoff expressions is harder than $H10(\mathbb{Q})$, and the proof of Lemma 8 follows.

APPENDIX C
PROOF OF THEOREM 4

When both players are restricted to finite-memory strategies the outcome of the game is an ultimately periodic path $\pi = \pi_1(\pi_2)^\omega$. Thus, for every dimension i we have $\mathit{LimInfAvg}_i(\pi) = \mathit{LimSupAvg}_i(\pi)$. Hence, w.l.o.g we may assume that the game objective is a $\mathit{LimInfAvg}$-only expression. In this section, we will show a reduction from games in which both players are restricted to finite-memory strategies to games in which only player 1 is restricted to finite-memory strategies. The reduction is based on the next lemma.

Lemma 18. *Let E be a $\mathit{LimInfAvg}$-only expression and let G be a multidimensional weighted graph, and the goal of player 1 is to assure $E \le \nu$. Then a player-1 finite-memory strategy is winning if and only if it wins against every player-2 finite-memory strategy.*

Proof: The proof for the direction from left to right is trivial. To prove the converse direction we fix a player-1 finite-memory strategy σ and we show that if player 2 a strategy that wins against σ, then he also has a finite-memory winning strategy. We note that when σ is fixed, a player-2 strategy is an infinite path in G^σ and a player-2 finite-memory strategy is an ultimately periodic path in G^σ. Hence, there exists an infinite path π in G^σ for which E assigns a value greater than ν. We claim that for every $\epsilon > 0$ there is an ultimately periodic path ρ in G^σ such that in every dimension $\mathit{LimInfAvg}_i(\rho_\epsilon) \ge \mathit{LimInfAvg}_i(\pi) - \epsilon$. Indeed, let s be a state that is visited infinitely often by π, and let π_s be a suffix of π that begins in state s, and we observe that $\mathit{LimInfAvg}(\pi_s) = \mathit{LimInfAvg}(\pi)$. By the definition of $\mathit{LimInfAvg}$ and by the finiteness of the graph it follows that for every $\epsilon > 0$ there exists a path π_ϵ that is a prefix of π_s, ends in state s, and $\mathit{LimInfAvg}_i(\pi_\epsilon) \ge \mathit{LimInfAvg}_i(\pi_s) - \epsilon$. We denote by π_0 the shortest path from the initial state to s, and we get that the ultimately periodic path $\rho_\epsilon = \pi_0(\pi_\epsilon)^\omega$ satisfies the assertion of the claim. To complete the proof of the lemma, we denote the number of sum operators in E by $\#\,\mathrm{sum}$ and we set $\epsilon = \frac{E(\pi)-\nu}{2\#\,\mathrm{sum}}$. It is easy to verify that the ultimately periodic path ρ_ϵ satisfies $E(\rho_\epsilon) \ge E(\pi) - \frac{E(\pi)-\nu}{2} = \frac{E(\pi)+\nu}{2} > \nu$, and the proof follows. ■

The proof of Theorem 4 follows immediately from the fact that we only consider $\mathit{LimInfAvg}$-only expressions and from Lemma 18 and Theorems 2 and 3.