Institute of Science and Technology

Mean-payoff pushdown games

Krishnendu Chatterjee, Yaron Velner

https://doi.org/10.15479/AT:IST-2012-0002

Deposited at: 12 Dec 2018 11:54 ISSN: 2664-1690

IST Austria (Institute of Science and Technology Austria)
Am Campus 1
A-3400 Klosterneuburg, Austria

Mean-Payoff Pushdown Games

Krishnendu Chatterjee and Yaron Velner

IST Austria (Institute of Science and Technology Austria)

Am Campus 1

A-3400 Klosterneuburg

Technical Report No. IST-2012-0002

http://pub.ist.ac.at/Pubs/TechRpts/2011/IST-2012-0002.pdf

Jan 10, 2012

Mean-Payoff Pushdown Games*

Krishnendu Chatterjee[1] and Yaron Velner[2]

[1] IST Austria
[2] Tel Aviv Univeristy, Israel

Abstract. Two-player games on graphs are central in many problems in formal verification and program analysis such as synthesis and verification of open systems. In this work we consider solving recursive game graphs (or pushdown game graphs) that can model the control flow of sequential programs with recursion. While pushdown games have been studied before with qualitative objectives, such as reachability and ω-regular objectives, in this work we study for the first time such games with the most well-studied quantitative objective, namely, mean-payoff objectives. In pushdown games two types of strategies are relevant: (1) global strategies, that depend on the entire global history; and (2) modular strategies, that have only local memory and thus do not depend on the context of invocation, but only on the history of the current invocation of the module. Our main results are as follows: (1) One-player pushdown games with mean-payoff objectives under global strategies are decidable in polynomial time. (2) Two-player pushdown games with mean-payoff objectives under global strategies are undecidable. (3) One-player pushdown games with mean-payoff objectives under modular strategies are NP-hard. (4) Two-player pushdown games with mean-payoff objectives under modular strategies can be solved in NP (i.e., both one-player and two-player pushdown games with mean-payoff objectives under modular strategies are NP-complete). We also establish the optimal strategy complexity showing that global strategies for mean-payoff objectives require infinite memory even in one-player pushdown games; and memoryless modular strategies are sufficient in two-player pushdown games. Finally we also show that all the problems have the same complexity if the stack boundedness condition is added, where along with the mean-payoff objective the player must also ensure that the stack height is bounded.

1 Introduction

Games on graphs. Two-player games played on finite-state graphs provide the mathematical framework to analyze several important problems in computer science as well as mathematics. In particular, when the vertices of the graph represent the states of a reactive system and the edges represent the transitions, then the synthesis problem (Church's problem) asks for the construction of a winning strategy in a game played on the graph [11, 30, 29, 28]. Game-theoretic formulations have also proved useful for the verification [3], refinement [23], and compatibility checking [14] of reactive systems. Games played on graphs are dynamic games that proceed for an infinite number of rounds. The vertex set of the graph is partitioned into player-1 vertices and player-2 vertices. The game starts at an initial vertex, and if the current vertex is a player-1 vertex, then player 1 chooses an out-going edge, and if the current vertex is a player-2 vertex, then player 2 does likewise. This process is repeated forever, and gives rise to an outcome of the game, called a *play*, that consists of the infinite sequence of states that are visited. Two-player games on finite-state graphs with qualitative objectives such as reachability, liveness, ω-regular conditions formalized as the canonical parity objectives, strong fairness objectives, etc. have been extensively studied in the literature [22, 17, 18, 35, 31, 21].

The extensions. The study of two-player finite-state games with qualitative objectives has been extended in two orthogonal directions in literature: (1) two-player infinite-state games with qualitative objectives; and (2) two-player finite-state games with quantitative objectives. One of the most well-studied model of infinite-state games with qualitative objectives are pushdown games (or games on recursive state machines) that can model reactive systems with recursion (or model the control flow of sequential programs with recursion). Pushdown games with reachability and parity objectives have been studied in [33, 32, 5, 4] (also see [19, 20, 10, 9] for sample research in stochastic pushdown games). The most well-studied quantitative objective is the *mean-payoff* objective, where a reward is associated

* The research was supported by Austrian Science Fund (FWF) Grant No P 23499-N23, FWF NFN Grant No S11407-N23 (RiSE), ERC Start grant (279307: Graph Games), Microsoft faculty fellows award, and the RICH Model Toolkit (ICT COST Action IC0901)

with every transition and the goal of one of the players is to maximize the long-run average of the rewards (and the goal of the opponent is to minimize). Two-player finite-state games with mean-payoff objectives have been studied in [16, 36, 27], and more recently applied in synthesis of reactive systems with quality guarantee [6] and robustness [7]. Moreover recently many quantitative logics and automata theoretic formalisms have been proposed with mean-payoff objectives in their heart to express properties such as reliability requirements, and resource bounds of reactive systems [12, 8, 15]. Thus pushdown games with mean-payoff objectives would be a central theoretical question for model checking of quantitative logics (specifying reliability and resource bounds) on reactive systems with recursion feature.

Pushdown mean-payoff games. In this work we study for the first time pushdown games with mean-payoff objectives (to the best of our knowledge mean-payoff objectives have not been studied in the context of pushdown games). In pushdown games two types of strategies are relevant and studied in literature. The first is the *global* strategies, where a global strategy can choose the successor state depending on the entire global history of the play (where history is the finite sequence of configurations of the current prefix of a play). The second is the *modular* strategies, and modular strategies are understood more intuitively in the model of games on recursive state machines. A *recursive state machine* (RSM) consists of a set of component machines (or modules). Each module has a set of *nodes* (atomic states) and *boxes* (each of which is mapped to a module), a well-defined interface consisting of *entry* and *exit* nodes, and edges connecting nodes/boxes. An edge entering a box models the invocation of the module associated with the box and an edge leaving the box represents return from the module. In the game version the nodes are partitioned into player-1 nodes and player-2 nodes. Due to recursion the underlying global state-space is infinite and isomorphic to pushdown games. The equivalence of pushdown games and recursive games has been established in [5]. A modular strategy is a strategy that has only local memory, and thus, the strategy does not depend on the context of invocation of the module, but only on the history within the current invocation of the module. In other words, modular strategies are appealing because they are stackless strategies, decomposable into one for each module. In this work we will study pushdown games with mean-payoff objectives for both global and modular strategies.

Previous results. Pushdown games with qualitative objectives were studied in [33, 32]. It was shown in [33] that solving pushdown games (i.e., determining the winner in pushdown games) with reachability objectives under global strategies is EXPTIME-hard, and pushdown games with parity objectives under global strategies can be solved in EXPTIME. Thus it follows that pushdown games with reachability and parity objectives under global strategies are EXPTIME-complete. The notion of modular strategies in games on recursive state machines was introduced in [5, 4]. It was shown that the modular strategies problem is NP-complete in pushdown games with reachability and parity objectives in general [5, 4]. The results of [5] also presents more refined complexity results in terms of the number of exit nodes, showing that if every module has single exit, then the problem is polynomial for reachability objectives [5] and in NP $\cap$ coNP for parity objectives [4].

Our contributions. In this work we present a complete characterization of the computational and strategy complexity of pushdown games and pushdown systems (one-player pushdown games or pushdown automata) with mean-payoff objectives. Solving a pushdown system (resp. pushdown game) with respect to a mean-payoff objective is to decide whether there exists a path that (resp. a winning strategy to ensure that every path possible given the strategy) satisfies the mean-payoff objective. Our main results for computational complexity are as follows.

1. *Global strategies.* We show that pushdown systems (one-player pushdown games) with mean-payoff objectives under global strategies can be solved in polynomial time, whereas solving pushdown games with mean-payoff objectives under global strategies is undecidable.
2. *Modular strategies.* Solving pushdown systems with single exit nodes with mean-payoff objectives under modular strategies is NP-hard, and pushdown games with mean-payoff objectives under modular strategies can be solved in NP. Thus both pushdown systems and pushdown games with mean-payoff objectives under modular strategies are NP-complete.

Our results are shown in Table 1. First observe that our hardness result for modular strategies is different from the NP-hardness of [5] because the hardness result of [5] shows hardness for *games* with reachability objectives and require that the number of modules with multiple exit nodes are not bounded (in fact if every module of the recursive game has a single exit, then the problem is in PTIME

	Global strategies	Modular strategies
Pushdown systmes	PTIME	NP-complete (NP-hard for single exit)
Pushdown games	Undecidable	NP-complete

Table 1. Computational complexity of pushdown systems and pushdown games with mean-payoff objectives.

	Global strategies	Modular strategies
Pushdown systmes	Infinite	Memoryless
Pushdown games	Infinite	Memoryless

Table 2. Strategy complexity of pushdown systems and pushdown games with mean-payoff objectives.

for reachability and NP $\cap$ coNP for parity objectives). In contrast we show that for mean-payoff objectives the problem is NP-hard even for pushdown systems (only one player), where every module has a single exit node, under modular strategies. Second we also observe the very different complexity of global and modular strategies for mean-payoff objectives in pushdown systems vs pushdown games: the global strategies problem is computationally inexpensive (in PTIME) as compared to the modular strategies problem (which is NP-complete) in pushdown systems; whereas the global strategies problem is computationaly infeasible (undecidable) as compared to the modular strategies problem (which is NP-complete) in pushdown games. Also observe that in contrast to finite-state game graphs where the complexities for mean-payoff and parity objectives match, for pushdown systems and games, the complexities of parity and mean-payoff objectives are very different. Along with the computational complexities, we also establish the optimal strategy complexity showing that global winning strategies for mean-payoff objectives in general require infinite memory even in pushdown systems; whereas memoryless or positional (independent of history) strategies suffice for modular strategies for mean-payoff objectives in pushdown games (see Table 2). Finally we also study the stack boundedness conditions where the goal of one player along with maximizing the mean-payoff objectives is also to ensure that the height of the stack is bounded. We show that all the complexities for the additional stack boundedness condition along with mean-payoff objectives are the same in pushdown systems and games as without the stack boundedness condition.

Technical contributions. Our key technical contributions are as follows. For pushdown systems under global strategies we show that the mean-payoff objective problem can be solved by only considering additional stack height that is polynomial. We then show that the stack height bounded problem can be solved in polynomial time using a dynamic programming style algorithm. For pushdown games under global strategies our undecidability result is obtained as a reduction from the universality problem of *weighted* automata (which is undecidable [26, 1]). For modular strategies we first show existence of a cycle independent modular strategies, and then show memoryless modular strategies are sufficient. Given memoryless modular strategies and polynomial time algorithm for pushdown systems, we obtain the NP upper bound for the modular strategies problem. Our NP-hardness result for modular strategies is a reduction from the 3-SAT problem.

Organization. Our paper is organized as follows. In Section 2 (resp. Section 3) we present the results for pushdown systems (resp. pushdown games) under global strategies. In Section 4 we present the results for modular strategies.

2 Mean-Payoff Pushdown Graphs

In this section we consider pushdown graphs (or pushdown systems) with mean-payoff objectives. We start with the basic notion of stack alphabet and commands.

Stack alphabet and commands. Let Γ denote a finite set of *stack alphabet*, and $\mathsf{Com}(\Gamma) = \{\mathit{skip}, \mathit{pop}\} \cup \{\mathit{push}(z) \mid z \in \Gamma\}$ denote the set of *stack commands* over Γ. Intuitively, the command *skip* does nothing, *pop* deletes the top element of the stack, $\mathit{push}(z)$ puts z on the top of the stack. For a stack command *com* and a stack string $\alpha \in \Gamma^+$ we denote by $\mathit{com}(\alpha)$ the stack string obtained by executing the command *com* on α.

Weighted pushdown systems. A *weighted pushdown system (WPS)* (or a weighted pushdown graph) is a tuple:

$$\mathcal{A} = \langle Q, \Gamma, q_0 \in Q, E \subseteq (Q \times \Gamma) \times (Q \times \mathsf{Com}(\Gamma)), w : E \to \mathbb{Z} \rangle,$$

where Q is a finite set of *states* with q_0 as the initial state; Γ the finite *stack alphabet* and we assume there is a special initial stack symbol $\perp \in \Gamma$; E describes the set of edges or transitions of the pushdown system; and w is a weight function that assigns integer weights to every edge. We assume that $\perp$ can be neither put nor removed from the stack. A *configuration* of a WPS is a pair (α, q) where $\alpha \in \Gamma^+$ is a stack string and $q \in Q$. For a stack string α we denote by $\mathsf{Top}(\alpha)$ the top symbol of the stack. The initial configuration of the WPS is $(\perp, q_0)$. We use W to denote the maximal absolute weight of the edge weights.

Successor configurations and runs. Given a WPS $\mathcal{A}$, a configuration $c_{i+1} = (\alpha_{i+1}, q_{i+1})$ is a *successor* configuration of a configuration $c_i = (\alpha_i, q_i)$, if there is an edge $(q_i, \gamma_i, q_{i+1}, com) \in E$ such that $com(\alpha_i) = \alpha_{i+1}$, where $\gamma_i = \mathsf{Top}(\alpha_i)$. A *path* π is a sequence of configurations. A path $\pi = \langle c_1, \ldots, c_{n+1} \rangle$ is a *valid path* if for all $1 \leq i \leq n$ the configuration c_{i+1} is a successor configuration of c_i (and the notation is similar for infinite paths). In the sequel we shall refer only to valid paths. Let $\pi = \langle c_1, c_2, \ldots, c_i, c_{i+1}, \ldots \rangle$ be a path. We denote by $\pi[j] = c_j$ the j-the configuration of the path and by $\pi[i_1, i_2] = \langle c_{i_1}, c_{i_1+1}, \ldots, c_{i_2} \rangle$ the segment of the path from the i_1-th to the i_2-th configuration. A path can equivalently be defined as a sequence $\langle c_1 e_1 e_2 \ldots e_n \rangle$, where c_1 is the initial configuration and e_i are valid transitions.

Average weights of paths. For a finite path π, we denote by $w(\pi)$ the sum of the weights of the edges in π and $\mathsf{Avg}(\pi) = \frac{w(\pi)}{|\pi|}$, where $|\pi|$ is the length of π, denote the average of the weights. For an infinite path π, we denote by $\mathsf{LimSupAvg}(\pi)$ (resp. $\mathsf{LimInfAvg}(\pi)$) the limit-sup (resp. limit-inf) of the averages (long-run average or mean-payoff objectives), i.e., $\limsup(\mathsf{Avg}(\pi[0, i]))_{i \geq 0}$ (resp. $\liminf(\mathsf{Avg}(\pi[0, i]))_{i \geq 0}$).

Notations. We shall use (i) γ or γ_i for an element of Γ; (ii) e or e_i for a transition (equivalently an edge) from E; (iii) α or α_i for a string from Γ^*. For a path $\pi = \langle c_1, c_2, \ldots \rangle = \langle c_1 e_1 e_2 \ldots \rangle$ we denote by (i) q_i: the state of configuration c_i, and (ii) α_i: the stack string of configuration c_i.

Stack height and additional stack height of paths. For a path $\pi = \langle (\alpha_1, q_1), \ldots, (\alpha_n, q_n) \rangle$, the *stack height* of π is the maximal height of the stack in the path, i.e., $\mathsf{SH}(\pi) = \max\{|\alpha_1|, \ldots, |\alpha_n|\}$. The *additional stack height* of π is the additional height of the stack in the segment of the path, i.e., the additional stack height $\mathsf{ASH}(\pi)$ is $\mathsf{SH}(\pi) - \max\{|\alpha_1|, |\alpha_n|\}$.

Pumpable pair of paths. Let $\pi = \langle c_1 e_1 e_2 \ldots \rangle$ be a finite or infinite path. A *pumpable pair of paths* for π is a pair of non-empty sequence of edges: $(p_1, p_2) = (e_{i_1} e_{i_1+1} \ldots e_{i_1+n_1}, e_{i_2} e_{i_2+1} \ldots e_{i_2+n_2})$, for $n_1, n_2 \geq 0$, $i_1 \geq 0$ and $i_2 > i_1 + n_1$ such that for every $j \geq 0$ the path $\pi^j_{(p_1,p_2)}$ obtained by pumping the pair p_1 and p_2 of paths j times each is a valid path, i.e., for every $j \geq 0$ we have

$$\pi^j_{(p_1,p_2)} = \langle c_1 e_1 e_2 \ldots e_{i_1-1} (e_{i_1} e_{i_1+1} \ldots e_{i_1+i_n})^j e_{i_1+i_n+1} \ldots e_{i_2-1} (e_{i_2} e_{i_2+1} \ldots e_{i_2+n_2})^j e_{i_2+n_2} \ldots \rangle$$

is a valid path. We will show that large additional stack height implies the existence of pumpable pair of paths. To prove the results we need the notion of *local minima* of paths.

Local minima of a path. Let $\pi = \langle c_1, c_2, \ldots \rangle$ be a path. A configuration $c_i = (\alpha_i, q_i)$ is a *local minima* if for every $j \geq i$ we have $\alpha_i \sqsubseteq \alpha_j$ (i.e., the stack string α_i is a prefix string of α_j). One basic fact about local minima of a path is as follows: Every infinite path has infinitely many local minimas. We discuss the proof of the basic fact and some properties of local minima. Consider a path $\pi = \langle c_1, c_2, \ldots \rangle$. If there is a finite integer j such that from some point on (say after i-th index) the stack height is always at least j, and the stack height is j infinitely often, then every configuration after i-th index with stack height j is a local minima (and there are infinitely many of them). Otherwise, for every integer j, there exists an index i, such that for every index after i the stack height exceeds j, and then for every j, the last configuration with stack height j is a local minima and we have infinitely many local minimas. This shows the basic fact about inifinitely many local minimas of a path. We now discuss a property of consecutive local minimas in a path. If we consider a path and the sequence of local minimas, and let c_i and c_j be two consecutive local minimas. Then either c_i and c_j have the same stack height, or else c_j is obtained from c_i with one push operation. In the following proposition we establish that if the additional stack height of a path exceeds $(|Q| \cdot |\Gamma|)^2$, then there is a pumpable pair of paths.

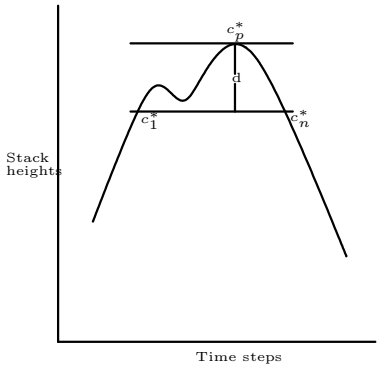

Fig. 1. Subpath construction.

Proposition 1 *Let π be a finite path such that $\mathsf{ASH}(\pi) = d \geq (|Q| \cdot |\Gamma|)^2$. Then π has a pumpable pair of paths.*

Proof. We first select a subpath of π^* of π such that $\pi^* = \langle c_1^*, \ldots, c_p^*, \ldots, c_n^* \rangle$ and the following conditions hold: (i) c_1^* is a local minima in π^*, (ii) $|\alpha_1^*| = |\alpha_n^*|$, and (iii) $|\alpha_p^*| = |\alpha_1^*| + d$. The subpath is selected as follows: consider the configuration c_p^* in π where the stack height is maximum, and c_1^* is the closest configuration before c_p^* where the stack height is exactly d less than the stack height of c_p^*, and similarly c_n^* is the closest configuration after c_p^* where the stack height is exactly d less than that of c_p^* (see Figure 1). Clearly all the three conditions are satisfied. Let c'_{i_j} (resp. c''_{i_j}) be the closest configuration before (resp. after) c_p^* such that the stack height of c'_{i_j} (resp. c''_{i_j}) is $|\alpha_1^*| + j$, for $j \geq 0$. Since $d \geq (|Q| \cdot |\Gamma|)^2$ it follows from the pigeonhole principle that there exists j_1, j_2 such that the states and the top stack symbol of $c'_{i_{j_1}}$ and $c'_{i_{j_2}}$ are identical, and the states and the top stack symbol of $c''_{i_{j_1}}$ and $c''_{i_{j_2}}$ are identical. It is straight forward to verify that the sequence p_1 of edges between $c'_{i_{j_1}}$ and $c'_{i_{j_2}}$ along with the sequence p_2 of edges between $c''_{i_{j_2}}$ and $c''_{i_{j_1}}$ are a pumpable pair, i.e., (p_1, p_2) is a pumpable pair for π. □

In the following lemma we establish the connection of additional stack height and existence of pumpable pair of paths with positive weights.

Lemma 1 *Let c_1, c_2 be two configurations and $n \in \mathbb{Z}$. Let $d \in \mathbb{N}$ be the minimal additional stack height of all the paths between c_1 and c_2 with total weight at least n. If $d \geq (|Q| \cdot |\Gamma|)^2$, then there exists a path π^* from c_1 to c_2 with additional stack height d that has a pumpable pair (p_1, p_2) such that $w(p_1) + w(p_2) > 0$.*

Proof. The proof is by induction on the length $|\pi|$ of the witness path π (i.e., π is a path from c_1 to c_2 with minimal additional stack height with weight at least n). We prove the base case and inductive step below.

- *Base case:* Note that since $\mathsf{ASH}(\pi) = d$ we must have $|\pi| \geq d$. Hence in the base case we have $|\pi| = d$. By Proposition 1, the path π has a pumpable pair (p_1, p_2). As $|\pi| = d$ it must be that the additional stack height $\mathsf{ASH}(\pi^0_{(p_1,p_2)})$ of the path $\pi^0_{(p_1,p_2)}$ obtained from π by removing p_1 and p_2 (i.e., pumping (p_1, p_2) zero times) is less than d. Hence it must be that $w(\pi) > w(\pi^0_{(p_1,p_2)})$, as otherwise $\pi^0_{(p_1,p_2)}$ would be a witness path from c_1 to c_2 with weight at least n but smaller additional stack height contradicting the minimality of π. Hence we have that $w(p_1) + w(p_2) > 0$.
- *Inductive step:* Again by Proposition 1, we have that π has a pumpable pair (p_1, p_2). If $w(p_1) + w(p_2) > 0$, then we are done. Otherwise, $w(p_1) + w(p_2) \leq 0$, and then we have $w(\pi^0_{(p_1,p_2)}) \geq w(\pi)$. Hence $\mathsf{ASH}(\pi^0_{(p_1,p_2)})$ is d, since, otherwise if $\mathsf{ASH}(\pi^0_{(p_1,p_2)}) < d$, then $\pi^0_{(p_1,p_2)}$ would be a witness path from c_1 to c_2 with weight at least n but smaller additional stack height than π (contradicting minimality of π). Since $|\pi| > |\pi^0_{(p_1,p_2)}|$, by the inductive hypothesis it follows that $\pi^0_{(p_1,p_2)}$ has a pumpable pair (p_3, p_4) such that $w(p_3) + w(p_4) > 0$.

The desired result follows □

Mean-payoff objectives with strict and non-strict inequalities. For a given integer r, the mean-payoff objective $\mathsf{LimInfAvg} \bowtie r$ (resp. $\mathsf{LimSupAvg} \bowtie r$) defines the set of infinite paths π such that $\mathsf{LimInfAvg}(\pi) \bowtie r$ (resp. $\mathsf{LimSupAvg}(\pi) \bowtie r$), where $\bowtie \in \{\geq, >\}$. The mean-payoff objectives with integer threshold r can be transformed to threshold 0 by subtracting r from all transition weights. Hence in this work w.l.o.g we will consider mean-payoff objectives (i) $\mathsf{LimInfAvg} > 0$ (resp. $\mathsf{LimSupAvg} > 0$), and call them mean-payoff objectives with strict inequality; and (ii) $\mathsf{LimInfAvg} \geq 0$ (resp. $\mathsf{LimSupAvg} \geq 0$), and call them mean-payoff objectives with non-strict inequality. We are interested in solving WPSs with mean-payoff objectives, i.e., to decide if there is a path that satisfies the objective.

2.1 Objectives $\mathsf{LimInfAvg} > 0$ and $\mathsf{LimSupAvg} > 0$

In this section we consider limit-average (or mean-payoff) objectives with strict inequality. We will show that WPSs with such objectives can be solved in polynomial time. A crucial concept in the proof is the notion of good cycles, and we define them below.

Good cycle. A finite path $\pi = \langle c_1, \ldots, c_n \rangle$ is a *good cycle* if the following conditions hold:

1. $w(\pi) > 0$ (the weight of the path is positive);
2. c_1 is a local minima;
3. let $c_1 = (\alpha_1, q_1)$ and $c_n = (\alpha_n, q_n)$, then $q_1 = q_n$ and $\mathsf{Top}(\alpha_1) = \mathsf{Top}(\alpha_n)$.

We first prove two propositions and the intuitive descriptions of them are as follows: In the first proposition we show that for every WPS, for every natural number d there exists a natural number n such that if there is path with weight at least n and additional stack height at most d, then there is a good cycle in the WPS. The second proposition is similar to the first proposition, and shows that if the additional stack depth is large, then it is possible to construct paths with arbitrarily large weights. Using the above two propositions we then show that if the weight of a finite path is sufficiently large, then either a good cycle exists or paths with arbitrarily large weights can be constructed. Finally we prove the key lemma that establishes the equivalence of the existence of a path satisfying mean-payoff objectives with strict inequality and the existence of a good cycle.

Proposition 2 *Let $\mathcal{A}$ be a WPS. For every $d \in \mathbb{N}$ there exists $n_{\mathcal{A},d} \in \mathbb{N}$ such that the following assertion holds: If there exists a path $\pi = \langle c_1, \ldots, c_r \rangle$ such that (i) c_1 is a local minima, (ii) $w(\pi) \geq n_{\mathcal{A},d}$ and (iii) $\mathsf{ASH}(\pi) \leq d$; then $\mathcal{A}$ has a reachable good cycle.*

Proof. Let G_d be a graph that contains all the paths that begin in c_1 and end in c_r with additional stack height at most d and for which c_1 is a local minima. Clearly, the graph G_d is a finite graph, and the maximal absolute weight W is the same as that for $\mathcal{A}$. A reachable positive cycle in G_d implies the existence of a reachable good cycle in $\mathcal{A}$, and if no positive cycle is reachable, then the weight of each path is bounded by $|G_d| \cdot W$. Thus with $n_{\mathcal{A},d} = |G_d| \cdot W$ we obtain the desired result. □

Proposition 3 *Let $\mathcal{A}$ be a WPS. Let $n \in \mathbb{Z}$ and let $\pi = \langle c_1, \ldots, c_r \rangle$ be a path with weight at least n, with minimal additional stack height, such that c_1 is a local minima. If $\mathsf{ASH}(\pi) \geq (|Q| \cdot |\Gamma|)^2$, then for every $m \in \mathbb{N}$ there exists a path π_m from c_1 to c_r such that c_1 is a local minima and $w(\pi_m) \geq m$.*

Proof. By Lemma 1 there exists a path $\overline{\pi}$ from c_1 to c_r that has a pumpable pair (p_1, p_2) such that $w(p_1) + w(p_2) > 0$. Hence for every $i \in \mathbb{N}$ we get that $w(\overline{\pi}^{i+1}_{(p_1,p_2)}) > w(\overline{\pi}^{i}_{(p_1,p_2)})$ (i.e., the weight after pumping $i+1$ times the pair of paths exceeds than the weight of pumping i times). Hence for $i = m - w(\overline{\pi})$ we get that $w(\overline{\pi}^{i}_{(p_1,p_2)}) \geq m$. The desired result follows. □

Lemma 2 *Let $\mathcal{A}$ be a WPS. There exists $n_{\mathcal{A}} \in \mathbb{N}$ such that if there exists a path π from configuration c_1 to configuration c_r with c_1 as a local minima and $w(\pi) \geq n_{\mathcal{A}}$, then one of the following holds:*

1. *$\mathcal{A}$ has a reachable good cycle.*
2. *For every $n' \in \mathbb{N}$ there exists a path π' from c_1 to c_r such that c_1 is a local minima of π' and $w(\pi') > n'$.*

Proof. Observe that the $n_{\mathcal{A}}$ is our choice and we will choose it sufficiently large for the proof. Let $d^* = (|Q| \cdot |\Gamma|)^2$, and our choice of $n_{\mathcal{A}}$ is $|Q| \cdot |\Gamma| \cdot n_{\mathcal{A},d^*}$ (where $n_{\mathcal{A},d^*}$ is as defined in Proposition 2). Let $\pi = \langle c_1, c_2, \ldots, c_r \rangle$ be a path such that c_1 is a local minima and $w(\pi) \geq n_{\mathcal{A}}$. Let $m_1, \ldots, m_j$ be

the local minimas along the path. Note that $m_1 = c_1$ and $c_r = m_j$. Also note that $j = |\alpha_r| - |\alpha_1|$. Note that if $m_{i_1} = (\alpha_{i_1}, q)$ and $m_{i_2} = (\alpha_{i_2}, q)$, then if a good cycle does not exist we get that the weight of the path between m_{i_1} and m_{i_2} is non positive. Hence, since Q and Γ are finite, by the pigeonhole principle, either a good cycle exists or there exists m_i, m_{i+1} such that $\alpha_{i+1} = \alpha_i\gamma$ for some $\gamma \in \Gamma$ and there exists a path from m_i to m_{i+1} such that m_i is a local minima and the weight of the path is at least $n_{\mathcal{A},d^*}$. Let π^* be such a path with minimal additional stack height between m_i and m_{i+1}. We consider two cases to complete the proof.

1. If the additional stack height of π^* is smaller than d^*, then by Proposition 2 we have a reachable good cycle from m_i and since m_i is reachable from c_1 we have reachable good cycle from c_1 (condition 1 of the lemma holds).
2. If the additional stack of π^* is at least d^*, then by Proposition 3 for every n_0 we can construct a path π_{n_0} between m_i and m_{i+1} with weight $w(\pi_{n_0})$ at least n_0, and m_1 is a local minima of π_{n_0}. For $n' \in \mathbb{N}$, let $n_0 = n' + W \cdot |\pi|$, and let π' be the path constructed using the segment from c_1 to m_i, then the path π_{n_0}, and then the segment of π from m_{i+1} to c_r. The configuration c_1 is a local minima of π' and the weight of π' is at least $n_0 - W \cdot |\pi| \geq n'$. Hence it follows that for every n' we can construct a path from c_1 to c_r with c_1 as a local minima and weight at least n' (condition 2 of the lemma holds).

This completes the proof of the lemma. □

Lemma 3 *Let $\mathcal{A}$ be WPS. The following statements are equivalent: (i) There exists a path π_1 with $\mathsf{LimSupAvg}(\pi_1) > 0$; (ii) there exists a path π_2 with $\mathsf{LimInfAvg}(\pi_2) > 0$; and (iii) there exists a path π that contains a good cycle.*

Proof. The direction from right to left is immediate. Let $\pi = \pi_1\pi_2$ be a finite path in $\mathcal{A}$ such that π_2 is a good cycle. Let $\pi_1 = c_1 e_1^1 e_2^1 \dots e_{n_1}^1$ and $\pi_2 = c_2 e_1^2 e_2^2 \dots e_{n_2}^2$. The infinite path $\pi' = \pi_1 c_2 (e_1^2 e_2^2 \dots e_{n_2}^2)^\omega$ obtained by repeating the good cycle forever is a valid path which satisfies that $\mathsf{LimSupAvg}(\pi') \geq \mathsf{LimInfAvg}(\pi') > 0$.

In order to prove the opposite direction, we consider an infinite path π such that $\mathsf{LimSupAvg}(\pi) > 0$. Let $q \in Q$ and $\gamma \in \Gamma$ be such that the sequence $m_1 = (\alpha_{i_1}, q), m_2 = (\alpha_{i_2}, q), \dots$ is an infinite sequence of local minima of π and $\mathsf{Top}(\alpha_{i_j}) = \gamma$ (note that such state and symbol are guaranteed to exist due to the existence of infinitely many local minimas and finiteness of $|Q|$ and $|\Gamma|$). If there exists $j > 1$ such that $w(\pi[i_1, i_j]) > 0$ then by definition $\pi[i_1, i_j]$ is a good cycle and the result follows. Otherwise let us assume that for every $j > 1$ we have $w(\pi[i_1, i_j]) \leq 0$. As $\mathsf{LimSupAvg}(\pi) > 0$ it follows that for every $n \in \mathbb{N}$ there exists $i_n > 1$ such that the path $\pi[i_1, i_n]$ contains a prefix with weight at least n (otherwise $\mathsf{LimSupAvg}(\pi) \leq 0$). We now use Lemma 2 to complete the proof. Let $n = n_{\mathcal{A}}$ (where $n_{\mathcal{A}}$ is as used in Lemma 2). Let $\pi' = m_1, \dots, c^*$ be the prefix of $\pi[i_1, i_n]$ such that $w(\pi') \geq n$. If the first condition of Lemma 2 holds (i.e., $\mathcal{A}$ has a good cycle), then we are done with the proof. Otherwise, by condition 2 of Lemma 2 it follows that for every $n_0 \in \mathbb{N}$ there exists a path π_{n_0} from m_1 to c^* such that m_1 is a local minima and $w(\pi_{n_0}) \geq n_0$. Let us choose $n_0 = W \cdot |\pi[i_1, i_n]| + 1$. Then consider the path $\overline{\pi} = \pi_{n_0}\pi[i + |\pi'|, i_n]$ that is obtained by concatenating the witness path π_{n_0} for n_0 from m_1 to c^*, and then the part of π from c^* to $\pi[i_n]$. For the path $\overline{\pi}$ we have (i) the sum of weights is at least $n_0 - W \cdot |\pi[i_1, i_n]| \geq 1 > 0$; (ii) $\pi[i_1]$ is a local minima; and (iii) the state and the top stack symbol of $\pi[i_1]$ and $\pi[i_n]$ are the same. Thus $\overline{\pi}$ is a witness good cycle. For conclusion we get that if $\mathsf{LimSupAvg}(\pi) > 0$, then there exists a good cycle, which also implies that there exists a path π' such that $\mathsf{LimInfAvg}(\pi') > 0$. This concludes the proof of the lemma. □

In the above key lemma we have established the equivalence of the decision problems for WPSs with mean-payoff objectives with strict inequality and the problem of determining good cycles. We will now present a polynomial time algorithm for detecting good cycles. To this end we introduce the notion of non-decreasing paths and summary functions.

Non-decreasing paths. A path from configuration $(\alpha\gamma, q_1)$ to configuration $(\alpha\gamma\alpha_2, q_2)$ is a *non-decreasing α-path* if $(\alpha\gamma, q_1)$ is a local minima. Note that if π is a non-decreasing α-path for some $\alpha \in \Gamma^*$, then it is a non-decreasing β-path for every $\beta \in \Gamma^*$. Hence we say that π is a non-decreasing path if there exists $\alpha \in \Gamma^*$ such that π is a non-decreasing α-path.

Summary function. Let $\mathcal{A}$ be a WPS. For $\alpha \in \Gamma^*$ we define $s_\alpha : Q \times \Gamma \times Q \to \{-\infty\} \cup \mathbb{Z} \cup \{\omega\}$ as following.

1. $s_\alpha(q_1,\gamma,q_2)=\omega$ iff for every $n\in\mathbb{N}$ there exists a non-decreasing path from $(\alpha\gamma,q_1)$ to $(\alpha\gamma,q_2)$ with weight at least n.
2. $s_\alpha(q_1,\gamma,q_2)=z\in\mathbb{Z}$ iff the weight of the maximum weight non-decreasing path from configuration $(\alpha\gamma,q_1)$ to configuration $(\alpha\gamma,q_2)$ is z.
3. $s_\alpha(q_1,\gamma,q_2)=-\infty$ iff there is no non-decreasing path from $(\alpha\gamma,q_1)$ to $(\alpha\gamma,q_2)$.

Remark 1 *For every $\alpha_1,\alpha_2\in\Gamma^*$: $s_{\alpha_1}\equiv s_{\alpha_2}$.*

Due to Remark 1 it is enough to consider only $s\equiv s_\perp$. The computation of the summary function will be achieved by considering stack height bounded summary functions defined below.

Stack height bounded summary function. For every $d\in\mathbb{N}$, the *stack height bounded summary function* $s_d:Q\times\Gamma\times Q\to\{-\infty\}\cup\mathbb{Z}\cup\{\omega\}$ is defined as follows: (i) $s_d(q_1,\gamma,q_2)=\omega$ iff for every $n\in\mathbb{N}$ there exists a non-decreasing path from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ with weight at least n and additional stack height at most d; (ii) $s_d(q_1,\gamma,q_2)=z$ iff the weight of the maximum weight non-decreasing path from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ with additional stack height at most d is z; and (iii) $s_d(q_1,\gamma,q_2)=-\infty$ iff there is no non-decreasing path with additional stack height at most d from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$.

Basic facts of summary functions. We have the following basic facts: (i) for every $d\in\mathbb{N}$, we have $s_{d+1}\geq s_d$ (monotonicity); and (ii) s_{d+1} is computable in polynomial time from s_d and $\mathcal{A}$ (we will show this fact in Lemma 4). We first present a proposition that shows that from s_d, with $d=(|Q|\cdot|\Gamma|)^2$, we obtain the values of function s for all values in $\mathbb{Z}\cup\{-\infty\}$.

Proposition 4 *Let $d=(|Q|\cdot|\Gamma|)^2$. For all $q_1,q_2\in Q$ and $\gamma\in\Gamma$, if $s(q_1,\gamma,q_2)\in\mathbb{Z}\cup\{-\infty\}$, then $s(q_1,\gamma,q_2)=s_d(q_1,\gamma,q_2)$.*

Proof. By definition we have $s(q_1,\gamma,q_2)\geq s_d(q_1,\gamma,q_2)$. Assume towards contradiction that $s(q_1,\gamma,q_2)>s_d(q_1,\gamma,q_2)$, then there exists a non-decreasing path π with minimal additional stack height from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ with weight $n>s_d(q_1,\gamma,q_2)$ and additional stack height $d'>(|Q|\cdot|\Gamma|)^2$. Hence by Proposition 3 for every $m\in\mathbb{N}$ there exists a non-decreasing path from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ with weight at least m (note that in Proposition 3 the witness path constructed by pumping the positive pumpable pair yields a non-decreasing path). Hence $s(q_1,\gamma,q_2)=\omega$ in contradiction to the assumption that $s(q_1,\gamma,q_2)\in\mathbb{Z}\cup\{-\infty\}$. The desired result follows. □

Our goal now is the computation of the ω values of the summary function. To achieve the computation of ω values we will define another summary function s^* and a new WPS $\mathcal{A}^*$ such that certain cycles in $\mathcal{A}^*$ will characterize the ω values of the summary function. We now define the summary function s^* and the pushdown system $\mathcal{A}^*$. Let $d=(|Q|\cdot|\Gamma|)^2$. The new summary function s^* is defined as follows: if the values of s_d and s_{d+1} are the same then it is assigned the value of s_d, and otherwise the value ω. Formally,

$$s^*(q_1,\gamma,q_2)=\begin{cases}s_d(q_1,\gamma,q_2) & \text{if } s_d(q_1,\gamma,q_2)=s_{d+1}(q_1,\gamma,q_2)\\ \omega & \text{if } s_d(q_1,\gamma,q_2)<s_{d+1}(q_1,\gamma,q_2).\end{cases}$$

The new WPS $\mathcal{A}^*$ is constructed from $\mathcal{A}$ by adding the following set of ω-edges: $\{(q_1,\gamma,q_2,\mathit{skip})\mid s^*(q_1,\gamma,q_2)=\omega\}$.

Proposition 5 *For all $q_1,q_2\in Q$ and $\gamma\in\Gamma$, the following assertion holds: the original summary function $s(q_1,\gamma,q_2)=\omega$ iff there exists a non-decreasing path in $\mathcal{A}^*$ from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ that goes through an ω-edge.*

Proof. The direction from right to left is easy: if there is a non-decreasing path in $\mathcal{A}^*$ that goes through an ω-edge, it means that there exists (q_1',γ',q_2') with either $s_d(q_1',\gamma,q_2')=\omega$ or $s_d(q_1',\gamma',q_2')<s_{d+1}(q_1',\gamma',q_2')$. If $s_d(q_1',\gamma,q_2')=\omega$, then clearly $s(q_1',\gamma,q_2')=\omega$. Otherwise we have $s_d(q_1',\gamma',q_2')<s_{d+1}(q_1',\gamma',q_2')$, and then the proof of Proposition 4 shows that $s(q_1',\gamma',q_2')=\omega$. Since there exists finite path from $(\perp\gamma,q_1)$ to $(\perp\gamma,q_2)$ with the ω-edge it follows that $s(q_1,\gamma,q_2)=\omega$.

For the converse direction, we consider the case that $s(q_1,\gamma,q_2)=\omega$. If $s^*(q_1,\gamma,q_2)=\omega$, then the proof follows immediately. Otherwise it follows that $s_d(q_1,\gamma,q_2)\in\mathbb{Z}$. Hence there exists a weight $n\in\mathbb{Z}$ such that the non-decreasing path with the minimal additional stack height with weight n has additional stack height $d'\geq d+1$. Let π be that path. Then there exists a non-decreasing subpath that starts at $(\alpha\gamma',q_1')$ and ends at $(\alpha\gamma',q_2')$ with additional stack height exactly $d+1$. If $s_{d+1}(q_1',\gamma',q_2')=s_d(q_1',\gamma',q_2')$, then π is not the path with the minimal additional stack height. Hence, as $s_{d+1}(q_1',\gamma',q_2')>s_d(q_1',\gamma',q_2')$, by definition $s^*(q_1',\gamma',q_2')=\omega$ and the proof follows. □

We are now ready to show that the summary function s can be computed polynomial time.

Lemma 4 *For a WPS $\mathcal{A}$, the summary function s is computable in polynomial time.*

Proof. There are two key steps of the proof: (i) computation of s_d, for $d = (|Q| \cdot |\Gamma|)^2$, and we will argue how to compute s_{i+1} from s_i in polynomial time; (ii) computation of a non-decreasing path in $\mathcal{A}^*$ that goes through an ω-edge. We first argue how the key steps give us the desired result and then present the details of the key steps. Given the computation of (i), we construct s_d, s_{d+1} in polynomial time, and hence also s^*. Given s^* we construct $\mathcal{A}^*$ in polynomial time. By computation (ii) we can assign the ω values for the summary function, and all other have values as defined by s_d. Thus with the computation of key steps (i) and (ii) in polynomial time, we can compute the summary function s in polynomial time. We now describe the key steps:

1. *Computation of s_{i+1} from s_i and $\mathcal{A}$.* Let $G_{\mathcal{A}}$ be the finite weighted graph that is formed by all the configurations of $\mathcal{A}$ with stack height either one or two, that is, the vertices are of the form (α, q) where $q \in Q$ and $\alpha \in \{\bot \cdot \gamma, \bot \cdot \gamma_1 \cdot \gamma_2 \mid \gamma, \gamma_1, \gamma_2 \in \Gamma\}$. The edges (and their weights) are according to the transitions of $\mathcal{A}$: formally, (i) (Skip edges): for vertices $(\bot \cdot \alpha, q)$ we have an edge to $(\bot \cdot \alpha, q')$ iff $e = (q, \mathsf{Top}(\alpha), \mathit{skip}, q')$ is an edge in $\mathcal{A}$ (and the weight of the edge in $G_{\mathcal{A}}$ is $w(e)$) where $\alpha = \gamma$ or $\alpha = \gamma_1 \cdot \gamma_2$ for $\gamma, \gamma_1, \gamma_2 \in \Gamma$; (ii) (Push edges): for vertices $(\bot \cdot \gamma, q)$ we have an edge to $(\bot \cdot \gamma \cdot \gamma', q')$ iff $e = (q, \gamma, \mathit{push}(\gamma'), q')$ is an edge in $\mathcal{A}$ (and the weight of the edge in $G_{\mathcal{A}}$ is $w(e)$) for $\gamma, \gamma' \in \Gamma$; and (iii) (Pop edges): for vertices $(\bot \cdot \gamma \cdot \gamma', q)$ we have an edge to $(\bot \cdot \gamma, q')$ iff $e = (q, \gamma', \mathit{pop}, q')$ is an edge in $\mathcal{A}$ (and the weight of the edge in $G_{\mathcal{A}}$ is $w(e)$) for $\gamma, \gamma' \in \Gamma$. Intuitively, $G_{\mathcal{A}}$ allows skips, push pop pairs, and only one additional push. Note that $G_{\mathcal{A}}$ has at most $2 \cdot |Q| \cdot |\Gamma|^2$ vertices, and can be constructed in polynomial time.
   For every $i \geq 1$, given the function s_i, the graph $G^i_{\mathcal{A}}$ is constructed from $G_{\mathcal{A}}$ as follows: adding edges $((\bot\gamma_1\gamma_2, q_1), (\bot\gamma_1\gamma_2, q_2))$ (if the edge does not exist already) and changing its weight to $s_i(q_1, \gamma_2, q_2)$ for every $\gamma_1, \gamma_2 \in \Gamma$ and $q_1, q_2 \in Q$. The value of $s_{i+1}(q_1, \gamma, q_2)$ is exactly the weight of the maximum weight path between $(\bot\gamma, q_1)$ and $(\bot\gamma, q_2)$ in $G^i_{\mathcal{A}}$ (with the following convention: $-\infty < z < \omega$, $z + \omega = \omega$ and $z + -\infty = \omega + -\infty = -\infty$ for every $z \in \mathbb{Z}$). If in $G^i_{\mathcal{A}}$ there is a path from $(\bot\gamma, q_1)$ to $(\bot\gamma, q_2)$ that contains a cycle with positive weight, then we set $s_{i+1}(q_1, \gamma, q_2) = \omega$. Hence, given s_i and $\mathcal{A}$, the construction of $G^i_{\mathcal{A}}$ is achieved in polynomial time, and the computation of s_{i+1} is achieved using the Bellman-Ford algorithm [13] in polynomial time (the maximum weight path is the shortest weight if we define the edge length as the negative of the edge weight). Also note that the Bellman-Ford algorithm reports cycles with positive weight (that is, negative length) which is required to set ω values of s_{i+1}. It follows that we can compute s_{i+1} given s_i and $\mathcal{A}$ in polynomial time.
2. *Non-decreasing ω-edge path in $\mathcal{A}^*$.* We reduce the problem of checking if there exists a non-decreasing path from $(\bot\gamma, q_1)$ to $(\bot\gamma, q_2)$ in $\mathcal{A}^*$ that goes through an ω-edge to the problem of pushdown reachability in pushdown systems (or pushdown graphs), which is known to be in PTIME [34, 2]. The reduction is as follows: for every state $q \in Q$ we add a fresh (new) state q^ω, add a transition (or edge) $(q^\omega_1, \gamma, q^\omega_2, \mathit{com})$ for every $(q_1, \gamma, q_2, \mathit{com}) \in \Delta$ (i.e., the freshly added states follow the transition in the fresh copy as in the original WPS), and a transition $(q_1, \gamma, q^\omega_2, \mathit{com})$ for every transition $(q_1, \gamma, q_2, \mathit{com})$ that has an ω weight (i.e., there is a transition to the fresh copy only for an ω-edge). It follows that there exists an ω-edge non-decreasing path in $\mathcal{A}^*$ from $(\bot\gamma, q_1)$ to $(\bot\gamma, q_2)$ iff the configuration $(\bot\gamma, q^\omega_2)$ is pushdown reachable from the configuration $(\bot\gamma, q_1)$. Hence it follows that existence of non-decreasing ω-edge path in $\mathcal{A}^*$ can be determined in polynomial time.

This completes the proof of polynomial time algorithm to compute the summary function. □

Given the computation of the summary function, we will construct a summary graph, and show the equivalence of the existence of good cycles in a WPS with the existence of positive cycles in the summary graph.

Summary graph and positive simple cycles. Given a WPS $\mathcal{A} = \langle Q, \Gamma, q_0 \in Q, E \subseteq (Q \times \Gamma) \times (Q \times \mathsf{Com}(\Gamma)), w : E \to \mathbb{Z}\rangle$ and the summary function s, we construct the *summary graph* $\mathsf{Gr}(\mathcal{A}) = (\overline{V}, \overline{E})$ of $\mathcal{A}$ with an weight function $\overline{w} : \overline{E} \to \mathbb{Z} \cup \{\omega\}$ as follows: (i) $\overline{V} = Q \times \Gamma$; and (ii) $\overline{E} = E_{\mathit{skip}} \cup E_{\mathit{push}}$ where $E_{\mathit{skip}} = \{((q_1, \gamma), (q_2, \gamma)) \mid s(q_1, \gamma, q_2) > -\infty\}$, and $E_{\mathit{push}} = \{((q_1, \gamma_1), (q_2, \gamma_2)) \mid (q_1, \gamma_1, q_2, \mathit{push}(\gamma_2)) \in E\}$; and (iii) for all $e = ((q_1, \gamma), (q_2, \gamma)) \in E_{\mathit{skip}}$ we have $\overline{w}(e) = s(q_1, \gamma, q_2)$, and for all $e \in E_{\mathit{push}}$ is

according to weight function of $\mathcal{A}$, i.e., $\overline{w}(e) = w(e)$. A simple cycle C in $\mathsf{Gr}(\mathcal{A})$ is a *positive simple cycle* iff one of the following conditions hold: (i) either C contains an ω-edge (i.e., edge labeled ω by $\overline{w}$); or (ii) the sum of the weights of the edges of the cycles according to $\overline{w}$ is positive.

Proposition 6 *A WPS $\mathcal{A}$ has a good cycle iff the summary graph $\mathsf{Gr}(\mathcal{A})$ has a positive simple cycle.*

Proof. If $\mathcal{A}$ has a good cycle, then let π be a good cycle. The good cycle π is a non-decreasing path $\langle c_1, \ldots, c_n \rangle$ such that $c_1 = (\alpha_1\gamma, q)$ and $c_n = (\alpha_1\gamma\alpha_2\gamma, q)$ and $w(\pi) > 0$. Let $m_1, \ldots, m_r$ be the local minimas along the path. Note that for every $i < r$, either m_i and m_{i+1} have the same stack height or m_{i+1} is reachable from m_i via one push transition. For configuration $c = (\alpha\gamma, q)$, let us denote $\mathsf{Top}(c) = (\gamma, q)$. Hence the path $\mathsf{Top}(m_1), \ldots, \mathsf{Top}(m_r)$ is a cycle in $\mathsf{Gr}(\mathcal{A})$. If the cycle contains an ω-edge, then it is a positive cycle (by definition of positive cycles in $\mathsf{Gr}(\mathcal{A})$). Otherwise, the weight of the cycle in $\mathsf{Gr}(\mathcal{A})$ is at least $w(\pi)$, and therefore $\mathsf{Gr}(\mathcal{A})$ has a positive cycle (and therefore positive simple cycle).

The other direction is as follows. Consider a positive cycle in $\mathsf{Gr}(\mathcal{A})$. If the cycle does not contain an ω-edge, then there exists a non-decreasing path in $\mathcal{A}$ with the same weight that forms a good cycle. Otherwise, let (γ, q) be a vertex in the cycle, and $((\gamma_1, q_1), (\gamma_1, q_2))$ be an ω-edge in the cycle of $\mathsf{Gr}(\mathcal{A})$. From the construction of $\mathsf{Gr}(\mathcal{A})$, it follows that there exist $\alpha_1, \alpha_2, \alpha_3$ in $\mathcal{A}$ such that the following non-decreasing paths exist:

- A non-decreasing path π_1 from $(\alpha_1\gamma, q)$ to $(\alpha_1\gamma\alpha_2\gamma_1, q_1)$ (due to the path of the cycle).
- For every $m \in \mathbb{N}$: a non-decreasing path π^m from $(\alpha_1\gamma\alpha_2\gamma_1, q_1)$ to $(\alpha_1\gamma\alpha_2\gamma_1, q_2)$ with weight at least m (due to the ω-edge).
- A non-decreasing path π_2 from $(\alpha_1\gamma\alpha_2\gamma_1, q_2)$ to $(\alpha_1\gamma\alpha_2\gamma_1\alpha_3, q)$ (due to the path of the cycle).

Hence, for $m = W \cdot (|\pi_1| + |\pi_2|) + 1$, we get that the path $\pi_1\pi^m\pi_2$ is a good cycle. This completes both directions of the proof and gives us the result. □

Since the summary function and summary graph can be constructed in polynomial time, and the existence of a positive cycle in a graph can be checked in polynomial time (for example, first checking existence of a cycle with an ω-edge, and then applying Karp's mean-cycle algorithm [25] after removing all ω edges), we have the following lemma.

Lemma 5 *Given a WPS $\mathcal{A}$, whether $\mathcal{A}$ has a good cycle can be decided in polynomial time.*

Lemma 3 and Lemma 5 give us the following theorem.

Theorem 1 *Given a WPS $\mathcal{A}$, whether there exists an infinite path π such that $\mathsf{LimInfAvg}(\pi) > 0$ (or $\mathsf{LimSupAvg}(\pi) > 0$) can be decided in polynomial time. If there exists an infinite path π such that $\mathsf{LimSupAvg}(\pi) > 0$, then there exists an ultimately periodic infinite path π' such that both $\mathsf{LimSupAvg}(\pi') > 0$ and $\mathsf{LimInfAvg}(\pi') > 0$.*

2.2 Objectives $\mathsf{LimInfAvg} \geq 0$ and $\mathsf{LimSupAvg} \geq 0$

In this section we consider mean-payoff objectives with non-strict inequality. We will assume that the input WPS $\mathcal{A}$ has integer weights, but we will consider certain transformations that produce rational weight functions.

Transformed weight functions and weighted graphs. Let $w : E \to \mathbb{Q}$ be a weight function, and $r \in \mathbb{Q}$ be a rational value, then the weight function $w + r : E \to \mathbb{Q}$ is defined as follows: for all $e \in E$ we have $(w + r)(e) = w(e) + r$. Let $G = (V, E)$ be a (possibly infinite)[3] graph with a weight function $w : E \to \mathbb{Q}$. In order to emphasize that w is the weight function for G, we use w_G. We denote by G^r the same infinite graph with weight function $w_G + r$. We first show that if the lim-inf-average objective can be satisfied for all $\epsilon > 0$, then the non-strict lim-inf-average objective can also be satisfied.

Proposition 7 *Let $\mathcal{A}$ be a WPS. There exists a path π with $\mathsf{LimInfAvg}(\pi) \geq 0$ iff for every $\epsilon > 0$ there exists a path π_ϵ with $\mathsf{LimInfAvg}(\pi_\epsilon) > -\epsilon$.*

[3] In this subsection we often look at a WPS as an infinite graph of the configurations

Proof. The direction from left to right is trivial.

In order to prove the converse direction let us assume that for every $n \in \mathbb{N}$ there exists a path π_n with $\mathsf{LimInfAvg}(\pi_n) > -\frac{1}{n}$. Hence for every $n \in \mathbb{N}$ there exists a path π^* which leads to a path C_n that is a good cycle with respect to the weight function $w + \frac{1}{n}$. Since there are infinitely many values of $n \in \mathbb{N}$, and Q and Γ are finite, w.l.o.g all the good cycles (with respect to $w + \frac{1}{n}$) starts at the same top configuration (γ, q). Hence for $\pi = \pi^* C_1^{2^{W \cdot |C_2|^2}} \dots C_i^{2^{W \cdot |C_{i+1}|^i}} \dots$ we get that $\mathsf{LimInfAvg}(\pi) \geq 0$. This completes the proof. □

Lemma 6 *Let $\mathcal{A}$ be a WPS with integer weights (weight function w). Let $\ell = |\Gamma| \cdot |Q|$, and fix $\epsilon = \frac{1}{\ell^{(\ell+1)^2} \cdot 2 \cdot \ell}$. Then the WPS $\mathcal{A}^\epsilon$ (with weight function $w + \epsilon$) has a good cycle iff for every $\delta > 0$ the WPS $\mathcal{A}^\delta$ (with weight function $w + \delta$) has a good cycle.*

Proof. The direction from right to left is trivial. For the converse direction we first prove the following proposition.

Proposition 8 *Let s^ϵ be the summary function for $\mathcal{A}^\epsilon$.*

1. *If $s^\epsilon(q_1, \gamma, q_2) \neq \omega$, then $s^\epsilon(q_1, \gamma, q_2) \leq s(q_1, \gamma, q_2) + \frac{1}{2 \cdot \ell}$.*
2. *If $s^\epsilon(q_1, \gamma, q_2) = \omega$, then for every $\delta > 0$ we have $s^\delta(q_1, \gamma, q_2) = \omega$, where s^δ is the summary function for $\mathcal{A}^\delta$.*

Proof. We prove both the items below.

1. If $s^\epsilon(q_1, \gamma, q_2) \neq \omega$, then maximum weight non-decreasing path with minimal additional stack height from $(\bot\gamma, q_1)$ to $(\bot\gamma, q_2)$ has an additional stack height of at most $(|Q| \cdot |\Gamma|)^2 = \ell^2$. Note that this path does not contain positive cycles (since $s^\epsilon(q_1, \gamma, q_2) \neq \omega$). Hence there exists a path π with the same weight and with stack height at most ℓ^2 which does not contain any cycles. Hence $|\pi| \leq \ell^{\ell^2}$, and therefore

$$w_{\mathcal{A}^\epsilon}(\pi) = w_{\mathcal{A}}(\pi) + \epsilon \cdot |\pi| \leq w_{\mathcal{A}}(\pi) + \epsilon \cdot \ell^{\ell^2} \leq w_{\mathcal{A}}(\pi) + \frac{1}{2 \cdot \ell}.$$

   Since $s(q_1, \gamma, q_2) \geq w_{\mathcal{A}}(\pi)$ (as π is a non-decreasing path we have $s(q_1, \gamma, q_2) \geq w_{\mathcal{A}}(\pi)$), we obtain the result of the first item.
2. In order to prove the second item of the proposition, it is enough to prove that if an edge weight is ω in $(\mathcal{A}^\epsilon)^*$ (where $(\mathcal{A}^\epsilon)^*$ is the WPS constructed with the function $(s^\epsilon)^*$), then for every $\delta > 0$ the weight of the edge is ω also in the summary graph $\mathsf{Gr}(\mathcal{A}^\delta)$ of $\mathcal{A}^\delta$. We consider two cases to complete the proof.
   - *Case 1.* If $s^\epsilon_\ell(q_1, \gamma, q_2) = \omega$, then the infinite graph $\mathcal{A}^\epsilon$ has a positive cycle C with stack height at most ℓ^2, hence there exists positive cycle C' such that $|C'| \leq \ell^{\ell^2}$. Towards contradiction, let us assume that $w_{\mathcal{A}}(C') < 0$. As all the weights in $\mathcal{A}$ are integers we get that $w_{\mathcal{A}}(C') \leq -1$. As $w_{\mathcal{A}}(C') + \epsilon \cdot |C'| = w_{\mathcal{A}^\epsilon}(C') \geq 0$ we get that $|C'| \geq \frac{1}{\epsilon}$ which is a contradiction. Thus $w_{\mathcal{A}}(C') \geq 0$, and hence for every $\delta > 0$ we have $w_{\mathcal{A}^\delta}(C') > 0$, hence $s^\delta_\ell(q_1, \gamma, q_2) = \omega$.
   - *Case 2.* Otherwise, we have $s^\epsilon_{\ell+1}(q_1, \gamma, q_2) > s^\epsilon_\ell(q_1, \gamma, q_2)$. Let π be a path from $(\bot\gamma, q_1)$ to $(\bot\gamma, q_2)$ with additional stack height $\ell + 1$ and weight $s^\epsilon_{\ell+1}(q_1, \gamma, q_2)$. As $s^\epsilon_{\ell+1}(q_1, \gamma, q_2) > s^\epsilon_\ell(q_1, \gamma, q_2)$, by Lemma 1 it follows that π has a pumpable pair (p_1, p_2) with $w_{\mathcal{A}^\epsilon}(p_1) + w_{\mathcal{A}^\epsilon}(p_2) > 0$. If p_1 (resp. p_2) contains a cycle with positive weight, then by the same arguments presented in the proof of the first item of the proposition this cycle will be positive also in $\mathcal{A}^\delta$, for every $\delta > 0$, and hence $s^\delta(q_1, \gamma, q_2) = \omega$. Therefore w.l.o.g both p_1 and p_2 do not contain any cycles and thus $|p_1|, |p_2| \leq \ell^{\ell+1}$. Again by the same arguments presented in the proof of the first item we obtain that $w_{\mathcal{A}}(p_1) + w_{\mathcal{A}}(p_2) \geq 0$ and hence for every $\delta > 0$ we have $w_{\mathcal{A}^\delta}(p_1) + w_{\mathcal{A}^\delta}(p_2) > 0$. As (p_1, p_2) is a positive pumpable pair in $\mathcal{A}^\delta$ it follows that $s^\delta(q_1, \gamma, q_2) = \omega$.

   This completes the proof of the second item.

We obtain the desired result of the proposition. □

We are now ready to prove Lemma 6. Let us assume that there exists a good cycle in $\mathcal{A}^\epsilon$. Then by Proposition 6 there exists a positive simple cycle C in the summary graph $\mathsf{Gr}(\mathcal{A}^\epsilon)$. We consider two cases:

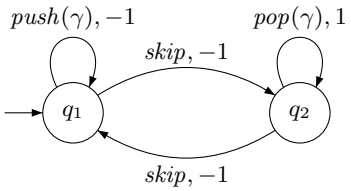

Fig. 2. WPS $\mathcal{A}$ to witness ultimately periodic words might not suffice for mean-payoff objectives with non-strict inequality

- If C contains an ω-edge e, then by Proposition 8 for every $\delta > 0$ the same cycle in $\mathsf{Gr}(\mathcal{A}^\delta)$ will also contain an ω-edge. Therefore C is a positive cycle also in $\mathsf{Gr}(\mathcal{A}^\delta)$ and hence $\mathcal{A}^\delta$ has a good cycle.
- Otherwise C does not contain an ω-edge. Towards contradiction assume that the weight of C in $\mathsf{Gr}(\mathcal{A})$ is negative. As the weight of $\mathcal{A}$ are integers it follows that the weight of C is at most -1. By Proposition 8, for every $e \in C$ we have $w_{\mathsf{Gr}(\mathcal{A}^\epsilon)}(e) \leq w_{\mathsf{Gr}(\mathcal{A})}(e) + \frac{1}{2\cdot\ell}$; and thus $w_{\mathsf{Gr}(\mathcal{A}^\epsilon)}(C) \leq w_{\mathsf{Gr}(\mathcal{A})}(C) + \frac{|C|}{2\cdot\ell}$. As C is a simple cycle (in $\mathsf{Gr}(\mathcal{A}^\epsilon)$) we get that $|C| \leq \ell$, and hence we have $w_{\mathsf{Gr}(\mathcal{A}^\epsilon)}(C) \leq w_{\mathsf{Gr}(\mathcal{A})}(C) + \frac{1}{2} \leq -\frac{1}{2}$, which contradicts the assumption that C is a positive cycle. Therefore we have $w_{\mathsf{Gr}(\mathcal{A})}(C) \geq 0$, and therefore for every $\delta > 0$ we get that $w_{\mathsf{Gr}(\mathcal{A}^\delta)}(C) > 0$ and hence $\mathcal{A}^\delta$ has a good cycle.

This completes the proof of the lemma. □

Theorem 2 *Given a WPS $\mathcal{A}$, whether there exists an infinite path π such that* $\mathsf{LimInfAvg}(\pi) \geq 0$ *(or* $\mathsf{LimSupAvg}(\pi) \geq 0$*) can be decided in polynomial time. There exists a WPS $\mathcal{A}$ such that there exists a path π with* $\mathsf{LimInfAvg}(\pi) = 0$ *but for every ultimately periodic path π we have both* $\mathsf{LimInfAvg}(\pi) < 0$ *and* $\mathsf{LimSupAvg}(\pi) < 0$.

Proof. From Proposition 7 it follows that if there is a path π such that $\mathsf{LimInfAvg}(\pi) \geq 0$, then for every $\epsilon_1 > 0$ there is a path π such that $\mathsf{LimInfAvg}(\pi) > -\epsilon_1$. By Lemma 6 it follows that it suffices to check for ϵ (for the ϵ described by Lemma 6). Given a WPS $\mathcal{A}$, the WPS $\mathcal{A}^\epsilon$ can be constructed in polynomial time (as ϵ has only polynomial number of bits). Then applying the polynomial time algorithm to find good cycles (as given in the previous subsection) we answer the decision problems in polynomial time. We observe that Proposition 7 and Lemma 6 also hold for $\mathsf{LimSupAvg}$ objectives, and thus the result also follows for $\mathsf{LimSupAvg}$ objectives.

We now present the example to show that the witness paths for non-strict inequality mean-payoff objectives are not necessarily ultimately periodic. Consider the WPS $\mathcal{A}$ with two states $Q = \{q_1, q_2\}$ with two symbol stack alphabet $\Gamma = \{\bot, \gamma\}$ and the edge set $E = \{e_1, e_2, \ldots, e_5\}$ is described as follows: $e_1 = (q_1, \bot, q_1, \mathit{push}(\gamma)), e_2 = (q_1, \gamma, q_1, \mathit{push}(\gamma)), e_3 = (q_1, \gamma, q_2, \mathit{skip}), e_4 = (q_2, \gamma, q_2, \mathit{pop}), e_5 = (q_2, \bot, q_1, \mathit{skip})$. The weight function is as follows: $w(e_4) = 1$, and all other edge weights are -1. (See Figure 2 for pictorial description). For $i \geq 1$, consider the path segment $\rho_i = e_1 e_2^{i-1} e_3 e_4^i e_5$ that executes the edge e_1, followed by $(i-1)$-times the edge e_2, then the edge e_3, followed by i-times the edge e_4 and finally the edge e_5. It is straight forward to verify that for the infinite path $\pi = (\bot, q_1)\rho_1\rho_2\rho_3\ldots$ we have that get that $\mathsf{LimSupAvg}(\pi) = \mathsf{LimInfAvg}(\pi) = 0$. However for every valid path $\pi = \xi_1\xi_2^\omega$, where $\xi_1 \in E^*$ and $\xi_2 \in E^+$ it must be the case that either (i) $\xi_2 = e_2$ and then $\mathsf{LimInfAvg}(\pi) = \mathsf{LimSupAvg}(\pi) = -1$ or that (ii) ξ_2 is a cycle with length at most $|\xi_2|$ has weight at most -1, and hence $\mathsf{LimInfAvg}(\pi) \leq \mathsf{LimSupAvg}(\pi) \leq -\frac{1}{|\xi_2|} < 0$. This completes the proof of the result. □

2.3 Mean-payoff objectives with stack boundedness

In this section we consider WPSs with mean-payoff objectives along with the *stack boundedness* condition that requires the height of the stack to be bounded. An infinite path $\pi = \langle c_1, c_2, \ldots c_i \ldots \rangle$ is a *stack bounded path* if there exists $n \in \mathbb{N}$ such that $|\alpha_i| \leq n$ for every $i \in \mathbb{N}$ (recall that α_i is the stack string of configuration c_i).

Theorem 3 *Given a WPS $\mathcal{A}$, the following problems can be solved in PTIME.*

1. *Does there exist a stack bounded infinite path π such that* $\mathsf{LimInfAvg}(\pi) \bowtie 0$ *(resp.* $\mathsf{LimSupAvg}(\pi) \bowtie 0$*), for* $\bowtie \in \{\geq, >\}$*?*
2. *Is* $\sup\{\mathsf{LimInfAvg}(\pi) \mid \pi \text{ is a stack bounded path}\} \geq 0$ *(resp.* $\sup\{\mathsf{LimSupAvg}(\pi) \mid \pi \text{ is a stack bounded path}\} \geq 0$*)?*

Proof. The results for each item are proved with a proposition of the proof below.

Proposition 9 *There exists a stack bounded infinite path π in $\mathcal{A}$ such that* $\mathsf{LimSupAvg}(\pi) > 0$ *(resp.* $\mathsf{LimSupAvg}(\pi) \geq 0$*) iff the summary graph* $\mathsf{Gr}(\mathcal{A})$ *has a vertex with self-loop that has a positive (resp. non-negative) weight.*

Proof. If there exists a stack bounded infinite path π in $\mathcal{A}$ such that $\mathsf{LimSupAvg}(\pi) > 0$ (resp. $\mathsf{LimSupAvg}(\pi) \geq 0$), then it contains a cycle that begins and ends at configuration $(\alpha\gamma, q)$ with positive (resp. non negative) weight. Hence in the summary graph $\mathsf{Gr}(\mathcal{A})$ the vertex (γ, q) will have a self-loop with positive (resp. non negative) weight. The other direction is straight forward. □

It is straight forward to verify that Proposition 9 also holds for $\mathsf{LimInfAvg}(\pi)$ objective. This gives us the first item of the theorem. The next proposition proves the last item of the theorem.

Proposition 10 *Let ϵ be the constant from Lemma 6. Then there exists a stack bounded infinite path π such that* $\mathsf{LimInfAvg}(\pi) > -\epsilon$ *iff* $\sup\{\mathsf{LimInfAvg}(\pi) \mid \pi \text{ is a stack bounded path}\} \geq 0$.

Proof. The direction from right to left is immediate. In order to prove the other direction let us assume that there exists a stack bounded infinite path π such that $\mathsf{LimInfAvg}(\pi) > -\epsilon$. Hence by Proposition 9 the summary graph $\mathsf{Gr}(\mathcal{A}^\epsilon)$ contains a self-loop for vertex (γ, q) with positive weight. By the same argument used in the proof of Lemma 6 it follows that for every $\delta > 0$ the self-loop of vertex (γ, q) will have a positive weight in graph $\mathsf{Gr}(\mathcal{A}^\delta)$. Hence for every $\delta > 0$ there exists a stack bounded path π_δ such that $\mathsf{LimInfAvg}(\pi_\delta) > -\delta$, which implies that $\sup\{\mathsf{LimInfAvg}(\pi) \mid \pi \text{ is a stack bounded path}\} \geq 0$. □

The proof of Proposition 10 straight forwardly extends to $\mathsf{LimSupAvg}(\pi)$ objective, and hence we have the desired result of the theorem. □

Thus we have the following result summarizing the computational complexity.

Theorem 4 *Given a WPS $\mathcal{A}$, the following questions can be solved in polynomial time: (1) Whether there exists a path π in $\Phi \bowtie 0$, where $\Phi \in \{\mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$ and $\bowtie \in \{\geq, >\}$; and (2) whether there exists a path π in $\Phi \bowtie 0$ such that π is stack bounded, where $\Phi \in \{\mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$ and $\bowtie \in \{\geq, >\}$.*

3 Mean-Payoff Pushdown Games

In this section we consider pushdown games with mean-payoff objectives. We will show that the problem of deciding the existence of a strategy (or a finite-memory strategy) to ensure mean-payoff objectives in pushdown games is undecidable. The undecidability results will be obtained by a reduction from the *universality problem* of weighted sum automata, which is known to be undecidable [26, 1]. We start with the definition of weighted pushdown games.

Weighted pushdown games (WPGs). A *weighted pushdown game (WPG)* $\mathcal{G} = \langle \mathcal{A}, (Q_1, Q_2) \rangle$ consists of a WPS $\mathcal{A}$ and a partition (Q_1, Q_2) of the state space Q of $\mathcal{A}$ into player-1 states Q_1 and player-2 states Q_2. A WPG defines an infinite-state game graph $(\overline{V}, \overline{E})$ with partition $(\overline{V}_1, \overline{V}_2)$ of the vertex set $\overline{V}$, where $\overline{V}$ is the set of configurations of $\mathcal{A}$, and $\overline{V}_1 = \{(\alpha, q) \in \overline{V} \mid q \in Q_1\}$, $\overline{V}_2 = \{(\alpha, q) \in \overline{V} \mid q \in Q_2\}$ and $\overline{E}$ is obtained from the transitions of $\mathcal{A}$. The *initial vertex* is the configuration $(\bot, q_0)$.

Plays and strategies. A *play* on $\mathcal{G}$ (or equivalently on the infinite-state game graph) is played in the following way: a pebble (or token) is placed on the initial vertex; and in every round, if the pebble is currently on player-1 vertex (a vertex in $\overline{V}_1$), then he chooses an edge to follow, and moves the pebble accordingly; and if the current vertex is a player-2 vertex, he does likewise. The process goes on forever and generates an infinite *play* (an infinite path π in the infinite graph of the game). A *strategy* for player 1 is a recipe to extend plays; formally, a strategy for player 1 is a function $\tau : \overline{V}^* \times \overline{V}_1 \to \overline{V}$ such that for all $w \in \overline{V}^*$ and $v \in \overline{V}_1$ we have $(v, \tau(w \cdot v)) \in \overline{E}$. Equivalently a strategy for player 1

given a history of configurations (i.e., the sequence of configurations of the finite prefix of a play) ending in a player-1 state, chooses the successor configuration according to the transition of $\mathcal{A}$. A play $\pi = v_1 v_2 \dots$ is *consistent* with a strategy τ if for every $v_i \in \overline{V}_1$ we have $v_{i+1} = \tau(v_1 v_2 \dots v_i)$, i.e., the play is possible according to the strategy τ. The definition of player-2 strategies is analogous. Informally a strategy can be viewed as a transducer that takes as input the sequence of transitions, and outputs the transitions to be taken. A strategy is *finite-memory* if there is a finite-state transducer to implement the strategy.

Winning strategies. We will consider mean-payoff objectives, as already defined in the previous section. A player-1 strategy τ is a *winning strategy* if for every play π consistent with τ we have $\mathsf{LimInfAvg}(\pi) \geq 0$ (resp. $\mathsf{LimInfAvg}(\pi) > 0, \mathsf{LimSupAvg}(\pi) \geq 0, \mathsf{LimSupAvg}(\pi) > 0$). In other words, a winning strategy for player 1 ensures the mean-payoff objective against all strategies of player 2. We are interested in the question of existence of a winning strategy, and the existence of a finite-memory winning strategy for player 1 in WPGs with mean-payoff objectives. Our undecidability results for WPGs with mean-payoff objectives will be a reduction from the non-universality of weighted finite automata. We define the problem below.

Weighted finite automata (WFA). A *weighted finite automaton (WFA)* is a tuple $\mathcal{A} = \langle \Sigma, Q, q_0, \Delta, w : \Delta \to \mathbb{Z} \rangle$, where Σ is a finite input alphabet, Q is a finite set of states, $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation, $w : \Delta \to \mathbb{Z}$ is a weight function and $q_0 \in Q$ is the initial state. For a word $\rho = \sigma_1 \sigma_2 \dots \sigma_n$, a *run* of $\mathcal{A}$ on ρ is a sequence $r = r_0 r_1 \dots r_n \in Q^+$, where $r_0 = q_0$, and for all $1 \leq i \leq n$ we have $d_i = (r_{i-1}, \sigma_i, r_i) \in \Delta$. The weight of the run r is $w(r) = \sum_{i=1}^{n} w(d_i)$. Since the automaton is non-deterministic there maybe several runs for a word, and the weight of a finite word $\rho \in \Sigma^*$ over $\mathcal{A}$ is the minimal weight over all runs on ρ, i.e., $L_{\mathcal{A}}(\rho) = \min\{w(r) \mid r \text{ is a run of } \mathcal{A} \text{ on } \rho\}$. The *non-universality* problem asks, given $\nu \in \mathbb{Z}$, whether there exists a word $\rho \in \Sigma^*$ for which $L_{\mathcal{A}}(\rho) \geq \nu$?

Theorem 5 ([1]) *Given a WFA with weight function $w : \Delta \to \{-1, 0, 1\}$, the non-universality problem is undecidable for every $\nu \in \mathbb{Z}$ for the following question: (a) Does there exist a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq \nu$? (Equivalently, is it not the case that for every $\rho \in \Sigma^*$ we have $L_{\mathcal{A}}(\rho) \leq \nu - 1$?)*

Informally, given a WFA $\mathcal{A}$ we will construct a WPG in such way that in the first rounds player-1 fills the stack with letters that construct a word ρ of $\mathcal{A}$, and then player-2 simulates the WFA minimal run on ρ and then the game returns to the initial state. If for all $\rho \in \Sigma^*$ we have $L_{\mathcal{A}}(\rho) \leq 0$, then the mean-payoff of the play will be at most 0, otherwise, there exists a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) > 0$, and then by playing according to ρ, player-1 can ensure positive mean-payoff.

Reduction: WFA to WPGs. We first prove that WPGs are undecidable for $\mathsf{LimInfAvg}(\pi) > 0$ and $\mathsf{LimSupAvg}(\pi) > 0$ objectives. This proof will immediately show the undecidability also for $\mathsf{LimInfAvg}(\pi) \geq 0$ and $\mathsf{LimSupAvg}(\pi) \geq 0$ objectives, as $\mathsf{LimInfAvg}(\pi) \geq 0$ (resp. $\mathsf{LimSupAvg}(\pi) \geq 0$) is dual to the objective of player 2 when the objective of player 1 is $\mathsf{LimSupAvg}(\pi) > 0$ (resp. $\mathsf{LimInfAvg}(\pi) > 0$).

Reduction. The reduction from the non-universality problem of a weighted automaton is as follows. Given a WFA $\mathcal{A} = \langle \Sigma, Q, q_0, \Delta, w : \Delta \to \{-1, 0, 1\} \rangle$ we construct a WPG $\mathcal{G}$ with the aid of *five* gadgets, and we describe the gadgets below. WLOG we assume that there is a special symbol \$ that does not belong to Σ.

1. *Gadget 1.* The first gadget contains only one state, namely $q_{\$}$, which is a player-1 state. The state has two possible transitions. In the first transition it pushes \$ into the stack and remains in the same state. In the second transition it pushes \$ and goes to the second gadget. All the weights in this gadget are -10.
2. *Gadget 2.* The second gadget also contains one state, namely q_{Σ}, which is also a player-1 state. For every $\sigma \in \Sigma$ the state has a transition that pushes σ into the stack and remains in the same state. In addition there is one more transition, which leads to the third gadget keeping the stack unchanged with weight 0. All the weights in this gadget (other than the skip transition) are -1. Informally, in this gadget player 1 needs to construct a word ρ such that the reverse of ρ has value at least 1 in $\mathcal{A}$. For a word ρ, let $\mathsf{rev}(\rho)$ denote the reverse of the word.
   - In this gadget player 1 should construct a word $\rho \in \Sigma^*$ for which $L_{\mathcal{A}}(\mathsf{rev}(\rho)) \geq 1$.
   - The WPG $\mathcal{G}$ will be constructed in such way that player 1 must play in a way so that the number of \$ in the stack will be greater than the number of letters from σ to ensure the mean-payoff objectives.

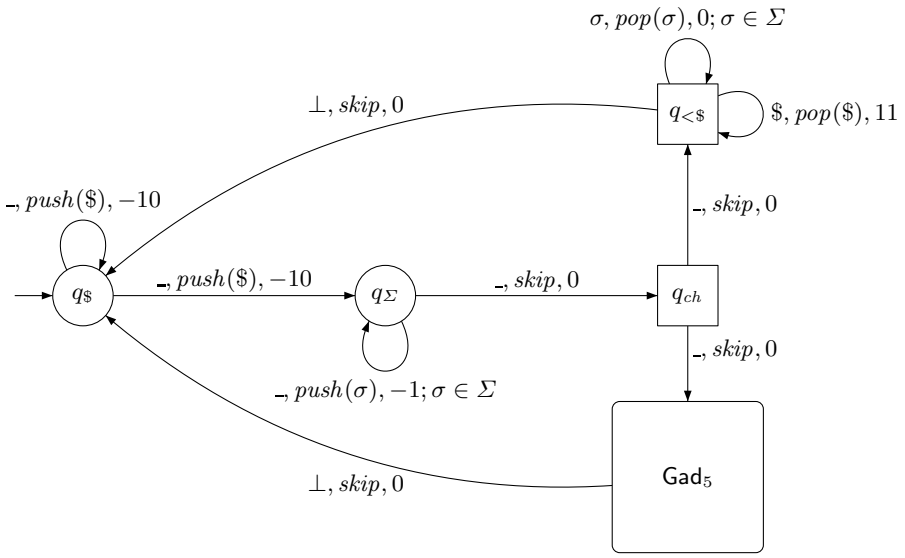

Fig. 3. The WPG $\mathcal{G}$ from WFA $\mathcal{A}$

3. *Gadget 3.* The third gadget is the *choice* gadget with only one player-2 state q_{ch}, which either leads to the fourth gadget or the fifth gadget. The weights of the transitions are 0 and the stack is not changed. Informally, player-2 should go to the fifth gadget if the word that player 1 pushed into the stack has non-positive weight, and should go to the fourth gadget if the number of \$ symbols in the stack is less than the number of symbols from Σ.
4. *Gadget 4.* The fourth gadget consists of only one player-2 state $q_{<\$}$ (to denote that there is not enough \$ symbols). It has a transition $pop(\sigma)$ with 0 weight, for all $\sigma \in \Sigma$; and a transition $pop(\$)$ with $+11$ weight. If the stack is empty, then there is a transition to the initial state.
   A pictorial description of the first four gadgets is shown in Figure 3, where ◯ denote player-1 states, □ denote player-2 states, and edges are labeled by stack top; followed by the stack command; and then the weight; and if the stack top is irrelevant (i.e., the transition is valid for all stack tops), then it is denoted as $\_$. We now describe the fifth gadget.
5. *Gadget 5.* The fifth gadget is the *simulate run* gadget. The states in this gadgets are essentially the set Q of states of the automaton $\mathcal{A}$; and all the states are player-2 states. The transitions and edge weights are as follows: (i) for every $(q, \sigma, q') \in \Delta$ we have a transition $(q, \sigma, q', pop(\sigma))$, with weight $w_{\mathcal{A}}(q, \sigma, q') + 1$ (1 plus the weight in $\mathcal{A}$); and (ii) in addition there exists a transition $(q, \$, q, pop(\$))$ with weight $+10$ and a transition $(q, \bot, q_\$, skip)$ to the initial state for empty stack with weight 0.

Correctness of reduction. We will now prove the correctness of the reduction by showing that there is a winning strategy (also a finite-memory winning strategy) in the WPG $\mathcal{G}$ for mean-payoff objectives with strict inequality iff there is a finite word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq 1$. Let π be a play on the above WPG $\mathcal{G}$. The *i-th iteration* of the play are the rounds between the i-th visit and the $(i+1)$-th visit to the initial state.

Lemma 7 *If there is a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq 1$, then there exists a finite-memory strategy τ_1^* for player 1 to ensure that for all plays π consistent with τ_1^* we have $\mathsf{LimSupAvg}(\pi) > 0$ and $\mathsf{LimInfAvg}(\pi) > 0$.*

Proof. The finite-memory strategy τ_1^* for player 1 is to play in every iteration $\$^{n+1}$ in $q_\$$ followed by $\mathsf{rev}(\rho)$ in q_Σ, where $n = |\rho|$ is the length of the word ρ. In every iteration, the sum of the weights is at least 1 as $L_{\mathcal{A}}(\rho) \geq 1$, and the length of play in every iteration is at most $4 \cdot n$. It follows that for all plays π consistent with τ_1^* we have both $\mathsf{LimSupAvg}(\pi) > 0$ and $\mathsf{LimInfAvg}(\pi) > 0$. □

Lemma 8 *If for all words $\rho \in \Sigma^*$ we have $L_{\mathcal{A}}(\rho) \leq 0$, then there exists a counter strategy τ_2^* for player 2 to ensure that for all strategies τ_1 of player 1, for the play given τ_2^* and τ_1: for all iterations i, for every round between i-th iteration and $(i+1)$-th iteration, the sum of the weights from the beginning of the iteration to the current round of the iteration is at most 0.*

Proof. The counter strategy τ_2^* is as follows: consider an iteration i, and let the strategy of player 1 in this iteration produce the sequence $\$^n\rho$, for $\rho \in \Sigma^*$. Note that if the state q_{ch} is never reached, then all the weights are negative (in $q_\$$ and q_Σ all weights are negative). The strategy τ_2^* is described considering the following two cases.

1. If $n \leq |\rho|$, then the strategy τ_2^* chooses the state $q_{<\$}$ at the state q_{ch} (since there are not enough \$ in the stack). For any round of the play till state q_{ch} is reached, the sum of the weights is negative. Once $q_{<\$}$ is reached, for any round the payoff is at most $-10 \cdot n - |\rho| + 11 \cdot n = -|\rho| + n \leq 0$, as $n \leq |\rho|$.
2. Otherwise, we have $n > |\rho|$ and $L_{\mathcal{A}}(\mathsf{rev}(\rho)) \leq 0$. There exists a run r on $\mathsf{rev}(\rho)$ such that for every prefix β of $\mathsf{rev}(\rho)$ the sum of the weights is at most $2 \cdot |\beta| \leq 2 \cdot |\rho| < 2 \cdot n$ (since the absolute value of the weights of $\mathcal{A}$ are bounded by 1) and in the end of the run sum of the weights is at most 0. The counter strategy τ_2^* follows the run r. Hence the sum of the weights for any prefix β is at most $-10 \cdot n + 2 \cdot |\beta| \leq -10 \cdot n + 2 \cdot n < 0$, until the letter \$ is the top symbol of the stack. Once \$ is the top symbol, the sum of the weights is at most $-10 \cdot n$, since the sum of the weights of the run is at most 0. Since with each pop of \$ the weight is 10, and there are n pops, it follows that in every round of iteration the sum of the weights is at most 0. Finally, once the iteration is completed the sum of the weights is also at most 0.

The desired result follows. □

Lemma 9 *Given WFA $\mathcal{A}$ and the WPG $\mathcal{G}$ constructed by the reduction the following the assertions hold:*

1. *If there is a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq 1$, then there is a finite-memory winning strategy τ_1^* for player 1 for the objectives $\mathsf{LimSupAvg}(\pi) > 0$ and $\mathsf{LimInfAvg}(\pi) > 0$.*
2. *If for all words $\rho \in \Sigma^*$ we have $L_{\mathcal{A}}(\rho) \leq 0$, then there is no winning strategy for player 1 for the objectives $\mathsf{LimSupAvg}(\pi) > 0$ and $\mathsf{LimInfAvg}(\pi) > 0$.*
3. *There exists a winning strategy (resp. a finite-memory winning strategy) for player 1 for the objectives $\mathsf{LimSupAvg}(\pi) > 0$ and $\mathsf{LimInfAvg}(\pi) > 0$ iff there is a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq 1$.*

Proof. Note that the third item is a consequence of the first two items. The first item follows from Lemma 7. We now use Lemma 8 to prove the second item. Given the condition of the second item, let us consider the strategy τ_2^* for player 2 as described in Lemma 8. Let π be a play consistent with τ_2^*. We consider two cases to complete the proof.

- If π does not have infinite number of iterations, then from some point on only states $q_\$$ or q_Σ are visited, and they both have only negative weights. Hence all the weights occur in π from some point on are non-positive and hence $\mathsf{LimInfAvg}(\pi) \leq \mathsf{LimSupAvg}(\pi) \leq 0$.
- Otherwise, π has infinite number of iterations. Given τ_2^* it follows from Lemma 8 that for all iterations, in every round of an iteration, the sum of the weights from the beginning of the iteration to the current round is non-positive. Hence $\mathsf{LimInfAvg}(\pi) \leq \mathsf{LimSupAvg}(\pi) \leq 0$.

The desired result follows. □

Undecidability for related decision problems. It follows from Lemma 9 that the existence of winning strategies (resp. finite-memory winning strategies) for mean-payoff objectives with strict inequality is undecidable for WPGs. For general strategies the result also follows for non-strict inequality by duality. We now show the undecidability for finite-memory strategy for the non-strict inequality as well as undecidability for stack boundedness. This is done by showing a reduction from the non-universality problem for WFA with threshold $\nu = 0$. The reduction is identical to the original reduction presented in this section. If there exists a word $\rho \in \Sigma^*$ such that $L_{\mathcal{A}}(\rho) \geq 0$, then playing $\$^{|\rho|+1}\mathsf{rev}(\rho)$ in every iteration is a finite-memory winning strategy for player-1 (also for the stack boundedness condition). Otherwise, for every $\rho \in \Sigma^*$ we have $L_{\mathcal{A}}(\rho) \leq -1$. In this case, against every player-1 finite-memory strategy, with memory size M, player-2 has a strategy that ensures that the mean-payoff is at most $-\frac{1}{2M}$. Similarly, against every player-1 strategy that ensures stack height at most M, player-2 has a strategy that ensures that the mean-payoff at most $-\frac{1}{2M}$.

Theorem 6 *Given a WPG $\mathcal{G}$, the following questions are undecidable: (1) Whether there exists a winning strategy (resp. finite-memory winning strategy) to ensure $\Phi \bowtie 0$, where $\Phi \in$*

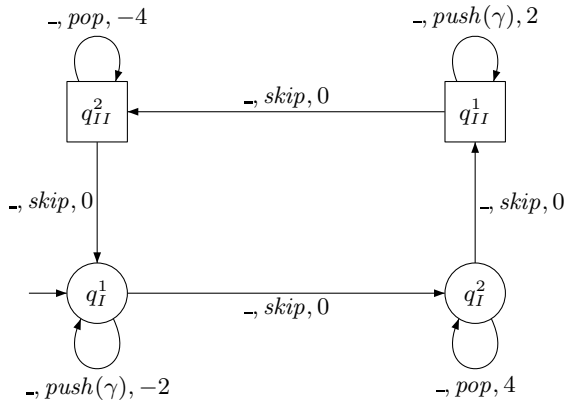

Fig. 4. Example WPG $\mathcal{G}$.

$\{\mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$ *and* $\bowtie \in \{\geq, >\}$*; and (2) whether there exists a winning strategy (resp. finite-memory winning strategy) to ensure* $\Phi \bowtie 0$ *along with stack boundedness, where* $\Phi \in \{\mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$ *and* $\bowtie \in \{\geq, >\}$.

Distinguishing facts. We now show some interesting facts about WPGs with mean-payoff objectives that distinguish from finite game graphs with mean-payoff objectives.

1. *Fact 1.* It follows from the proof of Theorem 2 (the example of ultimately periodic words are not sufficient to witness values) that in general, positional (or memoryless) strategies are not sufficient, and infinite-memory strategies are required in general (in contrast, in finite game graphs, memoryless winning strategies are guaranteed to exist).
2. *Fact 2.* The objectives $\mathsf{LimSupAvg}$ and $\mathsf{LimInfAvg}$ do not coincide in general for WPGs. We show this in Example 1.
3. *Fact 3.* We also note that pushdown mean-payoff games are very different as compared to parity games. For finite-state games both parity and mean-payoff conditions have the same complexity (both lie in NP ∩ coNP and also in UP ∩ coUP [24]), whereas for pushdown games the mean-payoff problem is undecidable, whereas the parity problem is EXPTIME-complete [33]. Moreover, for countably infinite games with finitely many priorities, for parity objectives memoryless winning strategies exist [31], whereas as we show (Fact 1) for mean-payoff pushdown games infinite-memory strategies are required.

Example 1 *We show that there exists a WPG such that player-1 can ensure that* $\mathsf{LimInfAvg}(\pi) \geq 2$ *and player-2 can ensure that* $\mathsf{LimInfAvg}(\pi) \leq -2$*. The WPG is described as follows: Let* $Q_1 = \{q^1_I, q^2_I\}$ *and* $Q_2 = \{q^1_{II}, q^2_{II}\}$*, and let* E*, the set of transitions, be as follows*

- $(q^1_I, \bot, q^1_I, push(\gamma))$ *with weight* -2*;*
- $(q^1_I, \gamma, q^1_I, push(\gamma))$ *with weight* -2*;*
- $(q^1_I, \gamma, q^2_I, skip)$ *with weight* 0*;*
- $(q^2_I, \gamma, q^2_I, pop)$ *with weight* $+4$*;*
- $(q^2_I, \bot, q^1_{II}, skip)$ *with weight* 0*;*
- $(q^1_{II}, \bot, q^1_{II}, push(\gamma))$ *with weight* $+2$*;*
- $(q^1_{II}, \gamma, q^1_{II}, push(\gamma))$ *with weight* $+2$*;*
- $(q^1_{II}, \gamma, q^2_{II}, skip)$ *with weight* 0*;*
- $(q^2_{II}, \gamma, q^2_{II}, pop)$ *with weight* -4*;*
- $(q^2_{II}, \bot, q^1_I, skip)$ *with weight* 0.

The WPG is shown in Figure 4. It is straight forward to verify that player-1 can ensure $\mathsf{LimSupAvg}(\pi) \geq 2$*, and player-2 can assure* $\mathsf{LimInfAvg}(\pi) \leq -2$. □

4 Recursive Games and Modular Strategies

In this section we will consider *modular* strategies in pushdown games, and modular strategies are more intuitive in the equivalent model of recursive game graphs. We first present the definition of recursive game graphs from [5].

Weighted recursive game graphs (WRGs). A *recursive game graph* $\mathcal{A}$ consists of a tuple $\langle A_1, \ldots, A_n \rangle$ of *game modules*, where each game module $A_i = (N_i, B_i, V_i^1, V_i^2, En_i, Ex_i, \delta_i)$ consists of the following components:

- A finite nonempty set of *nodes* N_i.
- A nonempty set of *entry* nodes $En_i \subseteq N_i$ and a nonempty set of *exit* nodes $Ex_i \subseteq N_i$.
- A set of *boxes* B_i.
- Two disjoint sets V_i^1 and V_i^2 that partition the set of nodes and boxes into two sets, i.e., $V_i^1 \cup V_i^2 = N_i \cup B_i$ and $V_i^1 \cap V_i^2 = \emptyset$. The set V_i^1 (resp. V_i^2) denotes the places where it is the turn of player 1 (resp. player 2) to play (i.e., choose transitions).
- A labeling $Y_i : B_i \to \{1, \ldots, n\}$ that assigns to every box an index of the game modules $A_1 \ldots A_n$.
- Let $\mathsf{Calls}_i = \{(b, e) \mid b \in B_i, e \in En_j, j = Y_i(b)\}$ denote the set of *calls* of module A_i and let $\mathsf{Retns}_i = \{(b, x) \mid b \in B_i, x \in Ex_j, j = Y_i(b)\}$ denote the set of *returns* in A_i. Then, $\delta_i \subseteq (N_i \cup \mathsf{Retns}_i) \times (N_i \cup \mathsf{Calls}_i)$ is the *transition relation* for module A_i.

A *weighted recursive game graph* (for short WRG) is a recursive game graph, equipped with a weight function w on the transitions. We also refer the readers to [5] for detailed description and illustration with figures of recursive game graphs. WLOG we shall assume that the boxes and nodes of all modules are disjoint. Let $B = \bigcup_i B_i$ denote the set of all boxes, $N = \bigcup_i N_i$ denote the set of all nodes, $En = \bigcup_i En_i$ denote the set of all entry nodes, $Ex = \bigcup_i Ex_i$ denote the set of all exit nodes, $V^1 = \bigcup_i V_i^1$ (resp. $V^2 = \bigcup_i V_i^2$) denote the set of all places under player 1's control (resp. player 2's control), and $V = V^1 \cup V^2$ denote the set of all vertices. We will also consider the special case of one-player WRGs, where either V^2 is empty (player-1 WRGs) or V^1 is empty (player-2 WRGs).

Configurations, paths and local history. A *configuration* c consists of a sequence $(b_1, \ldots, b_r, u)$, where $b_1, \ldots, b_r \in B$ and $u \in N$. Intuitively, $b_1, \ldots, b_r$ denote the current stack (of modules), and n is the current node. A sequence of configurations is *valid* if it does not violate the transition relation. The *configuration stack height* of c is r. Let us denote by $\mathbb{C}$ the set of all configurations, and let $\mathbb{C}_1$ (resp. $\mathbb{C}_2$) denote the set of all configurations under player 1's control (resp. player 2's control). A *path* $\pi = \langle c_1, c_2, c_3, \ldots \rangle$ is a valid sequence of configurations. Let $\rho = \langle c_1, c_2, \ldots, c_k \rangle$ be a valid finite sequence of configurations, such that $c_i = (b_1^i, \ldots, b_{d_i}^i, u_i)$, and the stack height of c_i is d_i. Let c_i be the first configuration with stack height $d_i = d_k$, such that for every $i \leq j \leq k$, if c_j has stack height d_i, then $u_j \notin Ex$ (u_j is not an exit node). The *local history* of ρ, denoted by $\mathsf{LocalHistory}(\rho)$, is the sequence $(u_{j_1}, \ldots, u_{j_m})$ such that $c_{j_1} = c_i$, $c_{j_m} = c_k$, $j_1 < j_2 < \cdots < j_m$, and the stack height of $c_{j_1}, \ldots, c_{j_m}$ is exactly d_i. Intuitively, the local history is the sequence of nodes in a module. Note that by definition, for every $\rho \in \mathbb{C}^*$, there exists $i \in \{1, \ldots, n\}$ such that all the nodes that occur in $\mathsf{LocalHistory}(\rho)$ belongs to V_i. We say that $\mathsf{LocalHistory}(\rho) \in A_i$ if all the nodes in $\mathsf{LocalHistory}(\rho)$ belongs to V_i.

Global game graph and isomorphism to pushdown game graphs. The *global game graph* corresponding to a WRG $\mathcal{A} = \langle A_1, \ldots, A_n \rangle$ is the graph of all valid configurations, with an edge (c_1, c_2) between configurations c_1 and c_2 if there exists a transition from c_1 to c_2. It follows from the results of [5] that every recursive game graph has an isomorphic pushdown game graph that is computable in polynomial time.

Plays, strategies and modular strategies. A play is played in the usual sense over the global game graph (which is possibly an infinite graph). A (finite) play is a (finite) valid sequence of configurations $\langle c_1, c_2, c_3, \ldots \rangle$ (i.e., a path in the global game graph). A *strategy* for player 1 is a function $\tau : \mathbb{C}^* \times \mathbb{C}_1 \to \mathbb{C}$ respecting the edge relationship of the global game graph, i.e., for all $w \in \mathbb{C}^*$ and $c_1 \in \mathbb{C}_1$ we have that $(c_1, \tau(w \cdot c_1))$ is an edge in the global game graph. A *modular strategy* τ for player 1 is a set of functions $\{\tau_i\}_{i=1}^n$, one for each module, where for every i, we have $\tau_i : (N_i \cup \mathsf{Retns}_i)^* \to \delta_i$. The function τ is defined as follows: For every play prefix ρ we have $\tau(\rho) = \tau_i(\mathsf{LocalHistory}(\rho))$, where $\mathsf{LocalHistory}(\rho) \in A_i$. The function τ_i is the *local strategy* of module A_i. Intuitively, a modular strategy only depends on the local history, and not on the context of invocation of the module. A modular strategy $\tau = \{\tau_i\}_{i=1}^n$ is a *finite-memory* modular strategy if τ_i is a finite-memory strategy for every

$i \in \{1, \ldots, n\}$. A *memoryless* modular strategy is defined in similar way, where every component local strategy is memoryless.

Mean-payoff objectives and winning modular strategies. The weight of a finite path π, denoted by $w(\pi)$ is the sum of all weights along the path. For an infinite path π (as in the previous sections) we denote $\mathsf{LimInfAvg}(\pi) = \liminf_{n\to\infty} \frac{w(\pi[1,n])}{n}$ (resp. $\mathsf{LimSupAvg}(\pi) = \limsup_{n\to\infty} \frac{w(\pi[1,n])}{n}$), where $\pi[1,n]$ is the initial prefix of length n. The *modular winning strategy problem* asks if player 1 has a modular strategy τ such that for every play ρ consistent with τ we have $\mathsf{LimInfAvg}(\rho) \geq 0$ (note that the counter strategy of player 2 is a general strategy), and similarly for other mean-payoff objectives.

Basic properties. We now present some basic properties of recursive game graphs.

Non-decreasing cycles and proper cycles. A *non-decreasing cycle* in a recursive game graph $\mathcal{A} = \langle A_1, \ldots, A_n \rangle$ is a path segment from a module A_i and vertex $v_i \in A_i$ to the same module and the same vertex (possibly at different stack level), such that the first occurrence of module A_i in the path segment does not return (i.e., does not reach an exit node) during the path segment. A non-decreasing cycle C is a *proper cycle* if the stack heights at the beginning and the end of the path segment are the same.

Proposition 11 *Consider a one-player WRG $\mathcal{A} = \langle A_1, \ldots, A_n \rangle$ (i.e., consists of only one-player). The following assertions hold:*

- *The WRG $\mathcal{A}$ has a path π with $\mathsf{LimInfAvg}(\pi) > 0$ (resp. $\mathsf{LimSupAvg}(\pi) > 0$) iff there exists a non-decreasing cycle with positive weight.*
- *The WRG $\mathcal{A}$ has a path π with $\mathsf{LimInfAvg}(\pi) < 0$ (resp. $\mathsf{LimSupAvg}(\pi) < 0$) iff there exists a non-decreasing cycle with negative weight.*

Proof. The first item follows from (i) the isomorphism of one-player WRGs and weighted push-down systems (WPSs), (ii) the correspondence positive non-decreasing cycles and good cycles for WPSs, and (iii) the results established in Section 2 showing equivalence of existence of a path π with $\mathsf{LimInfAvg}(\pi) > 0$ (resp. $\mathsf{LimSupAvg}(\pi) > 0$) and existence of good cycles in a WPS. The second item follows from the duality of $\mathsf{LimInfAvg}(\pi) > 0$ and $\mathsf{LimSupAvg}(\pi) < 0$. □

WRG given finite-memory strategies. Given a WRG $\mathcal{A}$, let $\tau = \{\tau_i\}_{i=1}^n$ be a finite-memory modular strategy. Let M_i be the set of memory states of strategy τ_i, i.e., τ_i is described as a deterministic transducer with state space M_i. The *one-player WRG (player-2 WRG)* given τ is the tuple $\mathcal{A}^\tau = \langle A_1^{\tau_1} = A_1 \times M_1, \ldots, A_n^{\tau_n} = A_n \times M_n \rangle$, where each $A_i^{\tau_i} = A_i \times M_i$ is obtained as the synchronous product of A_i and the deterministic transducer describing the local strategy τ_i. The weights of the transitions are specified according to the weight function of $\mathcal{A}$. Note that if τ is a memoryless modular strategy, then $\mathcal{A}^\tau$ is obtained as a sub-gamegraph of $\mathcal{A}$.

Proposition 12 *Given a WRG $\mathcal{A}$ and a modular strategy τ, every (finite or infinite) path in the one-player WRG $\mathcal{A}^\tau$ is a (finite or infinite) play in $\mathcal{A}$ consistent with τ, and vice versa.*

4.1 Decidability of the modular winning strategy problem

In this section we will establish the decidability of the existence of modular winning strategy problem. In the following section we will establish the NP upper bound, and finally show NP-hardness. We start with objective $\mathsf{LimInfAvg} \geq 0$, and then show the result for objective $\mathsf{LimSupAvg} \geq 0$. The result for mean-payoff objectives with strict inequality will also easily follow from our results.

Objective $\mathsf{LimInfAvg} \geq 0$. For the decidability result, we will show the existence of *cycle independent* modular winning strategies, and the result will also be useful to establish the complexity results. We start with the notion of a cycle free path in a graph.

Cycle free path. Let $G = (V, E)$ be a simple (no parallel edges) directed graph. We define the operator $\mathsf{CycleFree} : V^* \to V^*$ in the following way: let $\pi = \langle v_1, v_2, \ldots, v_n \rangle$ be a finite path in G.

- $\mathsf{CycleFree}(\pi) = \pi$ if π is a simple path (i.e., with no cycles).
- Otherwise we define $\mathsf{CycleFree}$ inductively as follows. Let $\mathsf{CycleFree}(v_1 \ldots v_{n-1}) = u_1 u_2 \ldots u_m$. Let i be the first index such that $v_n = u_i$. If such an index does not exist, then $\mathsf{CycleFree}(\pi) = u_1 u_2 \ldots u_m v_n$. Otherwise $\mathsf{CycleFree}(\pi) = u_1 u_2 \ldots u_i$. Intuitively, the $\mathsf{CycleFree}$ operator takes a finite path and returns a simple path by removing simple cycles according to order of appearance.

Cycle independent modular strategy. Given a recursive game graph, a local strategy τ_i for module A_i is a *cycle independent local strategy*, if for every $\rho \in V_i^*$ we have $\tau_i(\rho) = \tau_i(\mathsf{CycleFree}(\rho))$. A modular strategy $\tau = \{\tau_i\}_{i=1}^n$ is a *cycle independent modular strategy* if τ_i is a cycle independent local strategy for every $i \in \{1, \ldots, n\}$.

Observation 1 *For a recursive game graph $\mathcal{A} = \langle A_1, \ldots, A_n \rangle$, there exists at most $|V|^{|V|^2}$ different cycle independent modular strategies, where $|V|$ is the number of vertices in $\mathcal{A}$.*

The main result of this section is that if there is a modular winning strategy, then there is a cycle independent modular winning strategy. To establish the result we introduce the notion of *manipulated* paths, using *rewind*, *fast forward* and *simulation* operations.

Manipualted paths, rewind, fast forward and simulation operations. Let $\tau = \{\tau_i\}_{i=1}^n$ be a modular winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$, and let $\epsilon > 0$ be an arbitrary constant. Let $\pi_m = \pi_{m-1} \cdot n_i$ be a play prefix at round m, that ends at node $n_i \in A_i$. The *manipulated play prefix* of π_m according to τ and ϵ, denoted by $\mathsf{Man}_\epsilon^\tau(\pi_m)$, is defined inductively as follows: Let $\mathsf{Man}_\epsilon^\tau(\pi_{m-1})$ be the manipulated play prefix at round $m-1$. Then $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is obtained from $\mathsf{Man}_\epsilon^\tau(\pi_{m-1})$ and n_i by one of the following operations.

1. *Rewind operation*: The condition for the rewind operation is $\mathsf{CycleFree}(\mathsf{Man}_\epsilon^\tau(\pi_{m-1})) \cdot n_i$ closes a proper cycle in the top module A_i. If the rewind condition holds, then $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is formed from $\mathsf{Man}_\epsilon^\tau(\pi_{m-1}) \cdot n_i$ by removing the proper cycle suffix from $\mathsf{Man}_\epsilon^\tau(\pi_{m-1}) \cdot n_i$. Intuitively the rewind operation *rewinds* the path by chopping off the cycle in the end.
2. *Fast forward operation:* Let $h_0 = \mathsf{Man}_\epsilon^\tau(\pi_{m-1}) \cdot n_i$. The fast forward condition for history h that ends at node n_i is as follows: there exists a play prefix $h \cdot \pi'(h)$ consistent with τ such that $n_i \cdot \pi'(h)$ is a proper cycle with average payoff less than $-\epsilon$. In order to be precise, we define $\pi'(h)$ as the first such prefix according to lexicographic ordering of the prefixes. If the rewind condition does not hold, and the fast forward condition holds for h_0, then construct $h_1 = h_0 \cdot \pi'(h_0)$. Continue the process and build $h_i = h_{i-1} \cdot \pi'(h_{i-1})$, as long as h_{i-1} satisfies the fast forward condition. If there exists a minimal index $i \in \mathbb{N}$ such that h_i does not satisfy the fast forward condition, then we define $\mathsf{Man}_\epsilon^\tau(\pi_m) = h_i$. Otherwise, $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is undefined (not well defined), and we say that the process is stuck in the fast forward operation.
3. *Simulation operation:* Else, if the rewind and fast forward conditions do not hold, then we have $\mathsf{Man}_\epsilon^\tau(\pi_m) = \mathsf{Man}_\epsilon^\tau(\pi_{m-1}) \cdot n_i$.

In the following proposition we establish consistency and well-definedness of the manipulated operation for a winning strategy.

Proposition 13 *Let τ be a winning strategy (a general winning strategy, not necessarily modular) for the objective $\mathsf{LimInfAvg} \geq 0$. Let $\epsilon > 0$ be an arbitrary constant. We define a strategy σ in the following way: for a history π we have $\sigma(\pi) = \tau(\mathsf{Man}_\epsilon^\tau(\pi))$. Let π_m be a play prefix of length m that is consistent with σ. Then the following assertions hold:*

- *$\mathsf{Man}_\epsilon^\tau(\pi_m)$ is well defined, i.e., the process does not get stuck in the fast forward operation.*
- *$\mathsf{Man}_\epsilon^\tau(\pi_m)$ is consistent with τ.*

Proof. We shall prove both the items by induction on m. In the base case when $m = 0$ (i.e., empty play prefix), all the claims are trivially satisfied. We now consider the inductive case with $m > 0$. Let $\pi_m = \pi_{m-1} \cdot n_i$, for some $n_i \in A_i$, be a play prefix consistent with σ. By the inductive hypothesis $\mathsf{Man}_\epsilon^\tau(\pi_{m-1})$ is well defined and consistent with τ. Then $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is computed by performing one of the following operations

- *Rewind operation:* In this case clearly $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is well defined. In addition $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is a prefix of $\mathsf{Man}_\epsilon^\tau(\pi_{m-1})$, which is by the inductive hypothesis consistent with τ, hence also $\mathsf{Man}_\epsilon^\tau(\pi_m)$ is consistent with τ.
- *Fast forward operation:* Towards contradiction, let us assume that the fast forward process enters infinite loop. We consider the prefix $h_0 = \overline{h}_0 \cdot n_i$, where $\overline{h}_0 = \mathsf{Man}_\epsilon^\tau(\pi_{m-1})$, and let $\overline{v}_0$ be the last vertex in $\overline{h}_0$. The prefix h_0 is consistent with τ for the following reason: $\mathsf{Man}_\epsilon^\tau(\pi_{m-1})$ is consistent with τ by the inductive hypothesis, and if $\overline{v}_0$ is under player 1's control, then $\tau(\mathsf{Man}_\epsilon^\tau(\pi_{m-1})) = n_i$ (as π_m consistent with σ), and otherwise $\overline{v}_0$ is under player 2's control, and since there is a

transition from $\overline{v}_0$ to n_i (as π_m is a play prefix) the consistency of h_0 follows. The prefix h_0 has an infinite sequence of extensions $\pi^1, \pi^2, \ldots$ such that the infinite play $h = h_0\pi^1\pi^2 \ldots \pi^j\pi^{j+1} \ldots$ is consistent with τ and $\mathsf{Avg}(\pi^j) < -\epsilon$ for every $j \in \mathbb{N}$ (by the infinite loop of the fast forward operation). Hence, by definition, $\mathsf{LimInfAvg}(h) \leq -\epsilon < 0$.[4] Thus we get that there exists a play consistent with τ that is not winning for the objective $\mathsf{LimInfAvg} \geq 0$, which contradicts the assumption that τ is a winning strategy. Hence, the fast forward process always terminates. It follows that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ is well defined and also by definition of the fast forward operation it is consistent with τ.

– *Simulation operation:* By definition, $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m) = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1}) \cdot n_i$. Let $\overline{h}_0 = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})$, and let $\overline{v}_0$ be the last vertex in $\overline{h}_0$. The prefix $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ is consistent with τ for the following reason: $\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})$ is consistent with τ by the inductive hypothesis, and if $\overline{v}_0$ is under player 1's control, then $\tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})) = n_i$, and otherwise $\overline{v}_0$ is under player 2's control, and there is a transition from $\overline{v}_0$ to n_i. Thus we have the consistency of $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$, and the well-definedness is trivial.

Hence we have that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ is both consistent with τ and well defined. □

In the following lemma we obtain a bound of the average of the play prefixes obtained from the manipulated operation of a winning strategy.

Lemma 10 *Let τ be a winning strategy (a general winning strategy, not necessarily modular) for the objective $\mathsf{LimInfAvg} \geq 0$. Let $\epsilon > 0$ be an arbitrary constant. We define a strategy σ in the following way: for a history π we have $\sigma(\pi) = \tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi))$. Let π_m be a play prefix of length m that is consistent with σ. Then we have $w(\pi_m) \geq w(\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)) - \epsilon \cdot |\pi_m|$.*

Proof. The following claim is the key for the proof.

Claim. Every time a rewind operation is done, the cycle C, for which $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m) \cdot C = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1}) \cdot n_i$, satisfies that $\mathsf{Avg}(C) \geq -\epsilon$.

We first prove the claim. Towards contradiction, assume that $\mathsf{Avg}(C) < -\epsilon$. Let $j < m$ be the first index for which $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j) = \mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$. Note that such index must exist. We first argue that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m) \cdot C = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1}) \cdot n_i$ is consistent with τ: (i) $\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})$ is consistent with τ (by Proposition 13) and (ii) let $\overline{h}_0 = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})$, and let $\overline{v}_0$ be the last vertex in $\overline{h}_0$; if $\overline{v}_0$ is under player 1's control, then $\tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1})) = n_i$, and otherwise $\overline{v}_0$ is under player 2's control, and there is a transition from $\overline{v}_0$ to n_i. Thus we have the consistency of $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m) \cdot C = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{m-1}) \cdot n_i$, and it follows that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j) \cdot C$ is also consistent with τ. Hence, since $\mathsf{Avg}(C) < -\epsilon$, a fast forward operation had occurred in round j (note that it is not possible that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j)$ was obtained after a rewind operation, since j is the first index for which $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j) = \mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$). Hence it is not possible that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j) = \mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$, since at the very least, $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m) \cdot C$ is a prefix of $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j)$. Thus, for every $j < m$ we have $\mathsf{Man}^{\tau}_{\epsilon}(\pi_j) \neq \mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$, and the contradiction is obtained.

We now complete the proof of the lemma using the claim. We note that difference between π_m and $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ contains only (i) cycles with negative weight that were added to $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ (by fast forward operation) or (ii) cycles with average weight at most $-\epsilon$ and length at most $|\pi_m|$ that were chopped from $\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)$ (by rewind operation). The desired result follows. □

We now show that from a winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$, the strategy obtained using manipulated operation is winning for the objective $\mathsf{LimSupAvg} \geq -2 \cdot \epsilon$.

Lemma 11 *Let τ be a winning strategy (a general winning strategy, not necessarily modular) for the objective $\mathsf{LimInfAvg} \geq 0$. Let $\epsilon > 0$ be an arbitrary constant. We define a strategy σ in the following way: for a history π we have $\sigma(\pi) = \tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi))$. Then σ is a winning strategy for the objective $\mathsf{LimSupAvg} \geq -2 \cdot \epsilon$.*

Proof. Let π be a play consistent with σ, and let π_m be the play prefix until round m. We consider two cases to complete the proof.

1. In the first case, there exists a constant $n_0 \in \mathbb{N}$ such that for infinitely many indices $m_1, m_2, \ldots$, we have $|\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m_i})| \leq n_0$. In this case, due to Lemma 10, in rounds $m_1, m_2, \ldots$ we get that $w(\pi_{m_i}) \geq -n_0 \cdot W - \epsilon \cdot |\pi_{m_i}|$. Hence, by definition, $\mathsf{LimSupAvg}(\pi) \geq -\epsilon > -2 \cdot \epsilon$.

[4] only this inequality need not hold for $\mathsf{LimSupAvg}(h)$

2. In the second case, for every $i > 0$ there exists $\ell_i \in \mathbb{N}$, such that for every $m > \ell_i$ we have $|\mathsf{Man}^{\tau}_{\epsilon}(\pi_m)| \geq i$. By the definition of the manipulation operations, we get that $\mathsf{Man}^{\tau}_{\epsilon}(\pi_{\ell_i})[0,i] = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{\ell_i+1})[0,i]$, i.e., the prefix upto length i coincides. Denote $\rho_i = \mathsf{Man}^{\tau}_{\epsilon}(\pi_{\ell_i})[i]$ the i-th position of $\mathsf{Man}^{\tau}_{\epsilon}(\pi_{\ell_i})$. Due to Proposition 13 the infinite play $\rho = \rho_1\rho_2\dots$ is consistent with τ. Since τ is a winning strategy we get that $\mathsf{LimInfAvg}(\rho) \geq 0$. Hence there exists infinitely many indices $m_1, m_2, \dots$ for which $\mathsf{Avg}(\mathsf{Man}^{\tau}_{\epsilon}(\pi_{m_i})) \geq -\epsilon$, and therefore, due to Lemma 10, we get that $\mathsf{Avg}(\pi_{m_i}) \geq -2 \cdot \epsilon$. Hence by definition of $\mathsf{LimSupAvg}$, we obtain $\mathsf{LimSupAvg}(\pi) \geq -2 \cdot \epsilon$.

This concludes the proof of the lemma. □

Proposition 14 *Given a WRG $\mathcal{A}$, let τ be a modular winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$. Let $\epsilon > 0$ be an arbitrary constant. We define a strategy σ in the following way: for a history π we have $\sigma(\pi) = \tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi))$. Then σ is a cycle independent modular strategy.*

Proof. In order to verify that σ is a modular strategy, we observe that $\mathsf{LocalHistory}(\mathsf{Man}^{\tau}_{\epsilon}(\pi))$ is computable (maybe not effectively) from $\mathsf{LocalHistory}(\pi)$ and that τ is a modular strategy.

To verify that σ is a cycle independent strategy, we observe that if π and $\pi \cdot \pi_c$ are consistent with σ and π_c is a cycle in $\mathsf{CycleFree}(\pi) \cdot \pi_c$, then $\mathsf{Man}^{\tau}_{\epsilon}(\pi \cdot \pi_c) = \mathsf{Man}^{\tau}_{\epsilon}(\pi)$ as π_c will be chopped by the rewind operation. □

Lemma 12 *Given a WRG $\mathcal{A}$, if there exists a modular winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$ for player 1, then there exists a cycle independent modular winning strategy for the objective for player 1.*

Proof. Let τ be a modular winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$. For every $\epsilon > 0$, define the strategy σ_ϵ in the following way: for a history π we have $\sigma_\epsilon(\pi) = \tau(\mathsf{Man}^{\tau}_{\epsilon}(\pi))$. By Lemma 11 and Proposition 14, for every $\epsilon > 0$ we have σ_ϵ is a cycle independent modular winning strategy for the objective $\mathsf{LimSupAvg} > -2 \cdot \epsilon$. Since there are only a bounded number of cycle independent modular strategies (by Observation 1), it must be the case that one of the modular strategies is a winning strategy for the $\mathsf{LimSupAvg} \geq 0$ objective. Let σ be that strategy. Let $\mathcal{A}^{\sigma}$ be the player-2 WRG obtained given the strategy σ. As σ is a winning strategy for the objective $\mathsf{LimSupAvg} \geq 0$, then due to Proposition 12 and Proposition 11, the graph $\mathcal{A}^{\sigma}$ does not have a negative non-decreasing cycle. Hence due to Proposition 12 and Proposition 11, the strategy σ is winning also for the objective $\mathsf{LimInfAvg} \geq 0$. This completes the proof of the result. □

The next theorem is an immediate consequence of Lemma 12.

Theorem 7 *Given a WRG $\mathcal{A}$, the problem of deciding if player 1 has a modular winning strategy for the objective $\mathsf{LimInfAvg} \geq 0$ is decidable.*

Proof. By Lemma 12 it is enough to check if player 1 has a cycle independent modular strategy. As the number of such strategies is bounded (by Observation 1), it is enough to construct the graph $\mathcal{A}^{\tau}$ for every cycle independent modular strategy τ and check if in $\mathcal{A}^{\tau}$ there exists a path π with $\mathsf{LimInfAvg}(\pi) < 0$ (this check is achieved using the algorithms of Section 2). □

Objective $\mathsf{LimSupAvg} \geq 0$. The proof for the objective $\mathsf{LimSupAvg} \geq 0$ will reuse many parts of the proof for the objective $\mathsf{LimInfAvg} \geq 0$, however, some parts of the proof are different and we present them below. In fact we will show for modular winning strategies the objective $\mathsf{LimSupAvg} \geq 0$ coincides with the objective $\mathsf{LimInfAvg} \geq 0$. For the proof we need the notion of non-negative cycle free local history, which we define below.

Non-negative cycle free local history operator. Consider a WRG $\mathcal{A}$ and let τ be a modular strategy, and π be a path in $\mathcal{A}$, consistent with τ, that begins at the entry of module A_i and ends at module A_i (in the same stack height). The *non-negative cycle free local history operator* is defined as follows:

1. For $|\mathsf{LocalHistory}(\pi)| = 1$ we have $\mathsf{NonNegCFLocalHistory}^{\tau}(\pi) = \mathsf{LocalHistory}(\pi)$.
2. For $|\mathsf{LocalHistory}(\pi)| > 1$, let $\pi = \pi_0 v_i$ such that $\mathsf{NonNegCFLocalHistory}^{\tau}(\pi_0) = u_0 u_1 \dots u_m$. Let $j \in \{0, \dots, m\}$ be the first index such that $u_j = v_i$, and every sub-play π^* consistent with τ with local history $u_j u_{j+1} \dots u_m v_i$ is a cycle with non-negative total weight. If such index j exists, then $\mathsf{NonNegCFLocalHistory}^{\tau}(\pi) = u_0 \dots u_j$, otherwise $\mathsf{NonNegCFLocalHistory}^{\tau}(\pi) = u_0 u_1 \dots u_m v_i$.

Informally, the $\mathsf{NonNegCFLocalHistory}^{\tau}$ operator removes cycles that are assured to have non-negative weight from the local history.

Non-negative cycle independent modular strategy. Given a modular strategy τ, *the non-negative cycle independent modular strategy* σ of τ, is defined as follows: For a local history ρ we have $\sigma(\rho) = \tau(\mathsf{NonNegCFLocalHistory}^{\tau}(\rho))$. We now present some notations required for the proofs.

Sure non-negative cycle and proper simple cycle. Given a modular strategy τ, a path $\pi = v_0 v_1 \dots v_m$ is a *sure non-negative cycle* if π is a proper cycle consistent with τ, and every path π' consistent with τ such that $\mathsf{LocalHistory}(\pi) = \mathsf{LocalHistory}(\pi')$ is a proper cycle with non-negative weight. A path π is a *proper simple cycle* if π is a proper cycle, and $\mathsf{LocalHistory}(\pi)$ is a simple cycle.

Lemma 13 *If τ is a modular winning strategy for the objective $\mathsf{LimSupAvg} \geq 0$, then the non-negative cycle independent modular strategy σ of τ is also a winning strategy for the objective $\mathsf{LimSupAvg} \geq 0$.*

Proof. The proof is essentially similar to the proof of the result for the objective $\mathsf{LimInfAvg} \geq 0$, and we present a succinct argument (as it is very similar to the previous proofs for $\mathsf{LimInfAvg} \geq 0$). As previously, we define manipulated operations on history such that every sure non-negative cycle is chopped (by the rewind operation) from the history. By definition, the non-negative cycle independent strategy σ of τ makes choices according to the choices of τ on the manipulated history. The weight of the manipulated history, in every round, is at most the weight of the original history, since only non-negative cycles are chopped. By similar arguments to those presented in Lemma 11, we get that the $\mathsf{LimSupAvg}$ of the original history is at most 0, and thus the non-negative cycle independent strategy σ of τ is a winning strategy. □

In the following lemmas we establish that for modular strategies the objectives $\mathsf{LimSupAvg} \geq 0$ and $\mathsf{LimInfAvg} \geq 0$ coincide.

Lemma 14 *Given a WRG $\mathcal{A}$, let σ be a non-negative cycle independent modular strategy. Let A_i be a module in $\mathcal{A}$. Then there exists $n_\sigma \in \mathbb{N}$ and $\delta_\sigma > 0$, such that for every possible non-negative cycle free local histories h_0 and h_1 of the module A_i, and for every sub-play ρ, consistent with σ such that*

- *ρ is a proper cycle with negative weight; and*
- *the non-negative cycle free local history of A_i before (resp. after) ρ was played is h_0 (resp. h_1);*

there exists a sub-play ρ_{h_0,h_1}, consistent with σ and satisfies the items above, such that every play-prefix of ρ_{h_0,h_1} with length at least n_σ has an average weight of at most $-\delta_\sigma$.

Proof. Let $(b_1, n_1), \dots, (b_m, n_m)$ be the pairs of all boxes and their return nodes that appear in module A_i. For every pair (b_j, n_j), let π_{b_j,n_j} be the shortest play, consistent with σ, with minimal weight from b_j to n_j. If such path exists we denote $w_{(b_j,n_j)} = w(\pi_{b_j,n_j})$. Let X be the maximal such weight. W.l.o.g all the weights of the edges occur in A_i are at most X, and $X \geq 0$.

For every b_j, n_j such that there exists a play from b_j to n_j consistent with σ, but shortest minimal play does not exist, we denote by π_{b_j,n_j} the shortest play consistent with σ that leads from b_j to n_j with weight at most $-20 \cdot |V_i| \cdot X$, (where $|V_i|$ is the size of the vertex set in A_i). Let Π be the longest path among all π_{b_j,n_j}. Let ρ be a sub-play consistent with σ such that ρ is a proper cycle in module A_i with negative weight. Let h_0 (resp. h_1) be the non-negative cycle free history of A_i before (resp. after) playing ρ.

First, we form ρ' from ρ by removing all the sure non-negative cycles from ρ. Clearly, (i) ρ' has negative weight; and (ii) ρ' is consistent with σ, because ρ is consistent with σ and σ is a non-negative cycle independent strategy. Moreover, the non-negative cycle free local history after playing ρ' is the same as after playing ρ. Next we form ρ'' from ρ' by replacing every sub-play from b_j to n_j (such that n_j is the first time the sub-play enters A_i) in ρ' with π_{b_j,n_j}. Again, ρ'' is consistent with σ, since σ is a modular strategy. In addition, the local history of A_i was not changed at all.

We claim that ρ'' does not contain simple proper non-negative cycles in module A_i. Indeed, towards contradiction let $v_1 v_2 \dots v_m$ be the local history of the first such cycle (note that $m \leq |V_i|$). Note that if the cycle contains a sub-play π_{b_j,n_j} such that $w_{(b_j,n_j)} = -20 \cdot |V_i| \cdot X$, then the cycle cannot be with non-negative weight. Hence it follows that this cycle is the cycle with minimal weight among all simple cycles with local history $v_1 v_2 \dots v_m$. Hence this cycle is sure non-negative, which contradicts the fact that $v_1 v_2 \dots v_m$ is a local history of sub-play of ρ''. Thus every simple proper cycle in ρ'' has a negative weight. Moreover, the length of every simple proper cycle in ρ'' is at most $|V_i| \cdot |\Pi|$. Hence every sub-play of ρ'' with length at least $n_\sigma = (|V_i| \cdot |\Pi| \cdot X)^2$ will have an average weight of at most $-\delta_\sigma = -\frac{1}{|V_i| \cdot |\Pi| \cdot X}$. Note that n_σ and δ_σ do not depend on ρ, hence the desired result follows. □

Lemma 15 *Let σ be a non-negative cycle independent modular strategy. If there exists a play ρ consistent with σ such that the suffix of ρ is an infinite sequence of proper cycles $C_1, C_2, \ldots$ with negative weights, then σ is not a winning strategy for the objective* $\mathsf{LimSupAvg} \geq 0$.

Proof. Let $\rho = \rho_0 C_1 C_2 \ldots C_i \ldots$, and let A_i and n_i be the module and the vertex, respectively, that all the cycles begin and end in. Let n_σ, δ_σ be the constants from Lemma 14. Let h_0^i be the non-negative cycle free local history of A_i before cycle C_i is played, and h_1^i be the non-negative cycle free local history of A_i after cycle C_i is played. By Lemma 14, for every cycle C_i there exists a cycle $C_{h_0^i,h_1^i}$ consistent with σ, with negative weight, and the average weight of every sub-play of $C_{h_0^i,h_1^i}$ longer than n_σ is at most $-\delta_\sigma$. The play $\rho' = \rho_0 C_1 C_2 \ldots C_{i-1} C_{h_0^i,h_1^i} C_{i+1} \ldots$ is consistent with σ, since $C_{h_0^i,h_1^i}$ is consistent with the initial non-negative cycle free local history h_0^i, and the non-negative cycle free local history after playing $C_{h_0^i,h_1^i}$ is h_1^i. Since σ is a non-negative cycle independent strategy, the play $\rho^* = \rho_0 C_{h_0^0,h_1^0} C_{h_0^1,h_1^1} \ldots C_{h_0^i,h_1^i} \ldots$ is consistent with σ. On the other hand, it is straight forward to verify that we have $\mathsf{LimSupAvg}(\rho^*) \leq \max\{-\frac{1}{n_\sigma}, -\delta_\sigma\} < 0$. Hence σ is not a winning strategy for the objective $\mathsf{LimSupAvg} \geq 0$, and the desired result follows. □

Lemma 16 *If player 1 has a modular winning strategy for the objective* $\mathsf{LimSupAvg} \geq 0$*, then for every $\epsilon > 0$, player 1 has a cycle independent modular winning strategy σ for the objective* $\mathsf{LimSupAvg} \geq -\epsilon$.

Proof. Let τ^* be a modular winning strategy for the objective $\mathsf{LimSupAvg} \geq 0$, and let τ be the non-negative cycle independent strategy of τ^*. For $\epsilon > 0$, consider the cycle independent modular strategy σ such that for history π we have $\sigma(\pi) = \mathsf{Man}_\epsilon^\tau(\pi)$. By Lemma 14 the strategy τ is also a winning strategy. We note that all the arguments for the objective $\mathsf{LimInfAvg} \geq 0$ also hold for the objective $\mathsf{LimSupAvg} \geq 0$, other than the inequality (mentioned as footnote) in Proposition 13. We replace the inequality (mentioned as footnote) of Proposition 13 with Lemma 15, and repeat exactly the same arguments for the objective $\mathsf{LimInfAvg} \geq 0$ (up to Lemma 12) to obtain the desired result. □

We obtain the following result as a corollary, and all the desired results follow for the objective $\mathsf{LimSupAvg} \geq 0$.

Corollary 1 *Given a WRG $\mathcal{A}$, there exists a modular winning strategy for the objective* $\mathsf{LimSupAvg} \geq 0$ *iff there exists a modular winning strategy for the objective* $\mathsf{LimInfAvg} \geq 0$.

4.2 Modular winning strategy problem in NP

In this section we will show that the modular winning strategy problem is in NP. To show the result we introduce the notion of a signature game.

The signature game. Let $G = ((V, E), (V_1, V_2))$ be a finite two-player game graph (on finite directed graph (V, E)) with vertex set V, edge set E, and partition (V_1, V_2) of the vertex set into player-1 (resp. player-2) vertex set V_1 (resp. V_2). Let the game graph be equipped a weight function $w : E \to \mathbb{Z} \cup \{-\omega\}$. Let the initial vertex be $v_0 \in V$. Let $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{|V|})$ be a threshold vector such that all $\nu_i \in \mathbb{Z} \cup \{+\infty, -\omega\}$. The weight of a finite path in G is the sum of the edge weights of the path, according to the following convention: $-\omega + z = -\omega$, for any $z \in \mathbb{Z} \cup \{-\omega\}$. A *signature game* consists of a tuple $(G, \boldsymbol{\nu})$, where G is a two-player game graph, and $\boldsymbol{\nu}$ is the threshold vector. For a play $\rho = \rho_0 \rho_1 \rho_2 \ldots \rho_j \rho_{j+1} \ldots$, player 1 is the winner if both the following two conditions hold:

- The play ρ does not contain a negative cycle, or a cycle that has an edge with weight $-\omega$.
- For every $j \in \mathbb{N}$, if $\rho_j = v_i$ (i.e., the j-index of the play is the i-th vertex), then $w(\rho_0 \rho_1 \rho_2 \ldots \rho_j) \geq \nu_i$, according to the convention (i) $-\omega < z$ for every $z \in \mathbb{Z}$, and (ii) $z, -\omega < +\infty$ for every $z \in \mathbb{Z}$. In other words, the sum of the weights upto any index j (with i-th vertex in index j) must be at least ν_i, and no $-\omega$-edge must be visited unless $\nu_i = -\omega$.

We will consider signature games such that if $\nu_i \in \mathbb{Z}$, then $\nu_i \geq -2 \cdot W \cdot |V|$, where W is the maximum absolute values of the integer weights in graph G. If for a vertex v, the threshold value is $+\infty$, then to ensure winning player 1 must ensure that v is never visited. In other words, vertices with $+\infty$ threshold must be avoided (it can be interpreted as a safety objective with $+\infty$ vertices as non-safe vertices to be avoided). Our first goal is to show that memoryless winning strategies exist in signature games, and the result will be obtained by a reduction to finite-state mean-payoff games. Given a signature game $(G, \boldsymbol{\nu})$ we define an auxiliary finite-state mean-payoff game as follows.

From signature game $(G, \boldsymbol{\nu})$ to auxiliary mean-payoff game $\overline{G}_{\boldsymbol{\nu}}$. Given a signature game $(G, \boldsymbol{\nu})$ we construct a finite-state auxiliary mean-payoff game $\overline{G}_{\boldsymbol{\nu}} = ((\overline{V}, \overline{E}), (\overline{V}_1, \overline{V}_2))$, with an weight function $\overline{w}$ as follows: let the signature game graph be $G = ((V, E), (V_1, V_2))$ and the weight function in G be w_G. Then we have the following components in the auxiliary game:

- *(Vertex set and partition).* $\overline{V} = V \times \{1, 2\}$; and $\overline{V}_1 = V_1 \times \{1\}$.
- *(Edges).* $\overline{E} = \{((u, 1), (v, 2)) \mid (u, v) \in E\} \cup \{((v, 2), (v, 1)) \mid v \in V\} \cup \{((v_i, 2), (v_0, 1)) \mid v_i \in V, \nu_i \neq -\omega\}$.
- *(Weight function).* If $w_G(u, v) \neq -\omega$, then $\overline{w}((u, 1), (v, 2)) = w_G(u, v)$; otherwise (we have $w_G(u, v) = -\omega$) we set $\overline{w}((u, 1), (v, 2)) = -10 \cdot W \cdot |V|$. Moreover, $\overline{w}((v_i, 2), (v_0, 1)) = -\nu_i$ and all the other edges are assigned with zero weight. Note that if $\nu_i = +\infty$, then $\overline{w}$ assigns weight $-\infty$, and to win player 1 must avoid such edges (can be interpreted as a safety condition).

Informally, the auxiliary mean-payoff game is constructed from the signature game by adding for every vertex v_i a fresh copy (vertex $(v_i, 2)$ and the original vertex is represented as $(v_i, 1)$), and an option for player 2 to return to the initial vertex $(v_0, 1)$ "paying" cost $-\nu_i$ (whenever $\nu_i \neq -\omega$). Thus, if at any round, the play is on vertex $(v_i, 2)$, and the sum of the weights since the last visit to $(v_0, 1)$ is less than ν_i, then player 2 can ensure that a negative cycle is completed. The mean-payoff objective of player 1 is to ensure non-negative average payoff. Also note that player 1 must avoid the $-\infty$ edge weights, and equivalently it can be treated as a mean-payoff safety game. In the following proposition we establish the relation of the signature game and the auxiliary game.

Proposition 15 *Let $(G, \boldsymbol{\nu})$ be a signature game such that $\nu_i \geq -2 \cdot W \cdot |V|$ for every $\nu_i \in \mathbb{Z}$, and let $\overline{G}_{\boldsymbol{\nu}}$ be the corresponding auxiliary mean-payoff game. Then the following statements are equivalent:*

1. *Player 1 is the winner in the auxiliary mean-payoff game $\overline{G}_{\boldsymbol{\nu}}$ (i.e., player 1 can ensure non-negative mean-payoff).*
2. *Player 1 has a memoryless winning strategy in the signature game $(G, \boldsymbol{\nu})$.*
3. *Player 1 is the winner in the signature game $(G, \boldsymbol{\nu})$.*

Proof. We first prove that item 1 implies item 2.

1. In order to prove that item 1 implies item 2, let us assume that player 1 is the winner in the auxiliary mean-payoff game. Note that the auxiliary mean-payoff game is equivalently a mean-payoff safety game, and therefore player 1 has a memoryless winning strategy τ in the auxiliary mean-payoff game [16] (a mean-payoff safety game is easily transformed to a mean-payoff game by making the non-safe vertices absorbing with negative weights). Hence the memoryless strategy τ ensures that edges with weight $-\infty$ are never visited. Towards contradiction, let us assume that τ is not a winning strategy for the signature game (note that τ is also a well defined player 1 strategy in the signature game choosing edges in copy 1 according to τ). Therefore one of the following two cases occur.
   - Case 1: There exists a finite play prefix ρ that is consistent with τ, which starts from the initial v_0 to some vertex $v_i \in V$ with sum of weights less than ν_i. In this case, either ρ goes through an $-\omega$ edge, or $w_G(\rho) < \nu_i$. If ρ goes through an $-\omega$ edge in G, then the weight of ρ in $\overline{G}_{\boldsymbol{\nu}}$ is at most $-9 \cdot W \cdot |V|$, since w.l.o.g we can assume that ρ does not have positive cycles (as τ is memoryless). As $-9 \cdot W \cdot |V| < \nu_i$, it follows that the path $(\rho \cdot ((v_i, 2), (v_0, 1)))^\omega$ is consistent with τ and has a negative mean-payoff in the auxiliary game. This contradicts the assumption that τ is a winning strategy. If ρ does not go through an $-\omega$ edge, then $w_G(\rho) < \nu_i$ and again $(\rho \cdot ((v_i, 2), (v_0, 1)))^\omega$ is consistent with τ and has a negative mean-payoff in the auxiliary game. This is again a contradiction that τ is a winning strategy, and concludes the proof of the first case.
   - Case 2: There exists a finite play prefix $\rho = \rho_1 \cdot \rho_2$ that is consistent with τ, such that ρ_2 is a negative cycle (or a cycle with $-\omega$ edge) in the signature game graph. If ρ_2 does not contain an $-\omega$ edge e, then by definition, ρ_2 is a negative cycle also in the auxiliary game. Otherwise, ρ_2 contains an $-\omega$ edge e, and then again ρ_2 is a negative cycle in the auxiliary game, as w.l.o.g we can assume that ρ_2 does not contain positive cycles, and since $\overline{w}(e) \leq -10 \cdot W \cdot |V|$. Thus the play $\rho_1 \cdot (\rho_2)^\omega$ is consistent with τ and has a negative mean-payoff in the auxiliary game. This contradicts the assumption that τ is a winning strategy, and completes the proof.
2. Item 2 trivially implies item 3.

3. We now show that item 3 immediately implies item 1. It is straight forward to verify that if player 1 plays according to the signature game winning strategy in every round, then a negative cycle will not be formed in the auxiliary game (as a negative cycle is not formed in the signature game, and the threshold vector is always satisfied in the signature game) and a vertex with threshold $+\infty$ will never be reached. Hence the mean-payoff of the play in the auxiliary mean-payoff game will be non-negative and $-\infty$ edges will never be visited. This shows that item 3 implies item 1.

This completes the proof. □

Lemma 17 *Let $(G, \boldsymbol{\nu})$ be a signature game such that $\nu_i \geq -2 \cdot W \cdot |V|$ for every $\nu_i \in \mathbb{Z}$. There is a winning strategy for player 1 in the signature game $(G, \boldsymbol{\nu})$ iff player 1 has a memoryless winning strategy.*

Proof. Follows from Proposition 15. □

Signature games to memoryless modular strategies. We will now use the existence of cycle independent modular winning strategies, and memoryless strategies in signature games to show existence of memoryless modular strategies. For simplicity we will consider recusrive game graphs where every module has a single entry, and a simple polynomial reduction from multi-entry recursive game graphs to single entry recursive game graphs is established in [5]. To prove the result of memoryless modular strategies we define the signature games for modular strategies.

Signature games for modular strategies. Consider a WRG $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ and let $\tau = \{\tau_i\}_{i=1}^n$ be a modular strategy. Consider a module A_i in $\mathcal{A}$. Let $b \in B_i$ be a box in module A_i, which invokes the module A_j, and let $n_i \in N_i$ be a node in module A_i that is connected to one exit of A_j, which is reachable according to the strategy τ. We denote by w^{τ}_{b,n_i} the minimal weight of all plays according to τ that begins at the call to box b and ends at n_i (in the same stack height), and do not visit any other vertices in A_i (in the same stack height). If such minimal weight does not exist, then let $w^{\tau}_{b,n_i} = -\omega$. For every module A_i, we form a finite two-player game graph G_{A_i}, with a weight function as follows: (i) in the module A_i we add an edge from every box b to every return node n_i with weight w^{τ}_{b,n_i}, and add a self loop, with weight 0, to every exit node; and (ii) every box is now interpreted as a player-2 vertex. Note that the local strategy τ_i is a well defined player-1 strategy in the game graph G_{A_i}. For a vertex $v_j \in V_i$, (i) if v_j is visited along a play consistent with τ_i, then let η_j denote the maximal value such that in every round of a play according to τ_i on G_{A_i}, that begins in the entry node of A_i, and is currently at vertex $v_j \in V_i$, the sum of weights from the beginning of the play is at least η_j; and (ii) otherwise, v_j is never visited along all plays consistent with τ_i, then $\eta_j = +\infty$ (note that this is like a safety condition to ensure v_j is not visited). The *signature game for τ on module A_i* consists of the game graph G_{A_i} and the threshold vector $\boldsymbol{\nu}^i \in (\{-\omega, +\infty\} \cup \mathbb{Z})^{|V_i|}$, such that $\nu^i_j = \eta_j$ for all vertices $v_j \in V_i$. We denote by $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$ the signature game obtained given the modular strategy τ on module A_i. We first establish some basic properties in the following proposition, and then in the following lemma we establish properties of the winning strategies in the signature games from modular strategies.

Proposition 16 *Let $\mathcal{A}$ be a WRG. If τ is a cycle independent modular winning strategy for the objective* $\mathsf{LimInfAvg} \geq 0$*, then for every module A_i the following assertions hold:*

- *τ_i is a winning strategy for the signature game $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$; and*
- *the integer coefficients of $\boldsymbol{\nu}^i$ are at least $-2 \cdot W \cdot |V_i|$.*

Proof. The first fact of the proposition follows from the facts that every path according to τ_i has sum of weights at least $\boldsymbol{\nu}^i$ and does not contain negative cycles (by the definition of signature game given τ as τ is a winning strategy), The second fact of the proposition follows from the fact that every path according to τ_i does not contain negative cycle, as τ_i is a cycle independent modular winning strategy. □

Lemma 18 *Let $\mathcal{A}$ be a WRG. Let $\tau = \{\tau_i\}_{i=1}^n$ be a cycle independent modular winning strategy in $\mathcal{A}$ for the objective* $\mathsf{LimInfAvg} \geq 0$*. Let $\sigma = \{\sigma_i\}_{i=1}^n$ be a modular strategy such that σ_i is a winning strategy for the signature game $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$. Then for every play ρ^{σ}, consistent with σ, which starts from the entry node of a module A_i to a node n_ℓ in the same module (and possibly goes through box nodes), there exists a play ρ^{τ}, consistent with τ, from the same entry node to the same node n_ℓ, such that $w(\rho^{\sigma}) \geq w(\rho^{\tau})$. In addition, the path ρ^{σ} does not contain a negative proper cycle.*

Proof. The proof is by induction on the additional stack height of ρ^σ.

- *Base case:* Additional stack height is 0. In this case the play ρ^σ have only edges from A_i, and the weight of the play is identical to the weight of the same play in $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$. The play ρ^σ does not visit a vertex with threshold $+\infty$, otherwise σ_i would not be a winning strategy in the signature game $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$. Hence, by definition, there exists a play ρ^τ, consistent with τ from the entry node to n_ℓ with weight at most ν^i_ℓ. Since σ_i is a winning strategy in the signature game, and ρ^σ is consistent with σ_i, we get that $w(\rho^\sigma) \geq \nu^i_\ell$. Therefore $w(\rho^\sigma) \geq w(\rho^\tau)$. Since σ_i is a winning strategy in the signature game, and the path ρ^σ is consistent with σ_i also in graph G_{A_i}, we get that ρ^σ does not contain negative cycles.
- *Inductive step:* Additional stack height > 0. For simplicity, we first assume that ρ^σ goes only through one box node in the module A_i (in the first stack level). Let node b be that box, and let node $u \in A_i$ be the return node in that path. Let $\rho^\sigma_{b,u}$ be the sub-play from the entry node of b to node u. Recall that $w^\tau_{b,u}$ is the minimal weight among all plays consistent with τ between b and u. Let A_j be the module invoked by b, and let u' be the exit node that leads to the return node u in A_i. As the additional stack height from the entry node of A_j to u' is strictly smaller than the additional stack height of ρ^σ, it follows from the inductive hypothesis that there exists a path consistent with τ between these two nodes with weight at most $w(\rho^\sigma_{b,u})$. Hence $w^\tau_{b,u} \leq w(\rho^\sigma_{b,u})$. Thus, the weight of ρ^σ is bounded from below by the induced path of ρ^σ over the signature game $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$. Thus, by the definition of the signature game there exists a path ρ^τ as desired. In addition, by the inductive hypothesis, the path $\rho^\sigma_{b,u}$ does not contain proper negative cycle, and by the same arguments as above, there is also no negative proper cycle in module A_i. The case where ρ^σ goes through more then one box, is a straight forward extension of the argument presented above.

Thus we have the desired result. □

Lemma 19 *Let $\mathcal{A}$ be a WRG. Let $\tau = \{\tau_i\}_{i=1}^n$ be a cycle independent modular winning strategy in $\mathcal{A}$ for the objective* $\mathsf{LimInfAvg} \geq 0$. *Let $\sigma = \{\sigma_i\}_{i=1}^n$ be a memoryless modular strategy such that σ_i is a memoryless winning strategy for the signature game $(G_{A_i}, \boldsymbol{\nu}^i, \tau)$. Then σ is a memoryless modular winning strategy in $\mathcal{A}$ for the objective* $\mathsf{LimInfAvg} \geq 0$.

Proof. Let $\mathcal{A}^\sigma$ be the player-2 WRG obtained by fixing the memoryless modular strategy σ in $\mathcal{A}$. Assume towards contradiction that $\mathcal{A}^\sigma$ has a reachable non-decreasing negative cycle C, and let ρ be a finite path that leads to first vertex of C. By Lemma 18 it follows that C cannot be a proper cycle.

First, we argue that in $\mathcal{A}$ there exists a finite path ρ^τ from the first vertex of ρ to the last vertex of ρ that is consistent with τ. Indeed, by Lemma 18, between every entry node and box node in ρ there exists a path consistent with τ, and finally there also exists such a path between the last entry node and the last node of ρ.

Second, to achieve the contradiction we will show that τ is not a winning strategy. Let e_1 be the first entry node in C (it must exist as C is not a proper cycle), and n_i be the last (and first) node in C (note that C is not a proper cycle, and hence this node is well defined). Note that for every $m \in \mathbb{N}$, the path C^m is a non-decreasing cycle that is consistent with σ (as σ is a memoryless strategy). Let ρ_{e_1,n_i} be the path from e_1 to n_i. Let ρ_{n_i,e_1} be the path from n_i to first appearance of e_1 in C. We consider the path $\rho^* = \rho_{e_1,n_i} \cdot C^m \cdot \rho_{n_i,e_1}$, for $m = 2 \cdot W \cdot (|\rho_{e_1,n_i}| + |\rho_{n_i,e_1}|)$. This is a path that is (i) consistent with σ, (ii) begins and ends in the entry node e_1 of the same module (not necessarily in the same stack height). Let $b_1, b_2, \ldots b_\ell$ be the boxes occur in the path. By Lemma 18, for every k there exists a path $\rho^\tau_{b_i,b_{i+1}}$ consistent with τ such that $w(\rho^\tau_{b_i,b_{i+1}}) \leq w(\rho^\sigma_{b_i,b_{i+1}})$. Hence there exists a path ρ^τ_* that is consistent with τ from e_1 to e_1 such that the sum of the weights is negative. As τ is a modular strategy, and e_1 is an entry node, it follows that the path $(\rho^\tau_*)^\omega$ is also consistent with τ, and has negative mean-payoff. In conclusion, we obtain that there exists a reachable negative non-decreasing cycle in $\mathcal{A}$ consistent with τ, and this contradicts that τ is a winning strategy.

Hence, every path consistent with σ does not contain a negative non-decreasing cycle. By Proposition 11 it follows that σ is a winning strategy in $\mathcal{A}$ for the objective $\mathsf{LimInfAvg} \geq 0$. □

Lemma 20 *Let $\mathcal{A}$ be WRG. Player 1 has a modular winning strategy for the objective* $\mathsf{LimInfAvg} \geq 0$ *iff there exists a memoryless modular winning strategy for player 1 for the objective.*

Proof. The proof for the direction from right to left is trivial. The opposite direction is obtained as follows: by Lemma 12 it follows that if there is a modular winning strategy, then there is a cycle independent modular winning strategy; and by Lemma 19 it follows that if there is a cycle independent modular winning strategy, then there is a memoryless modular winning strategy. The desired result follows. □

We are now ready to prove the main result of this section.

Theorem 8 *The problem of deciding if player 1 has a modular winning strategy in a WRG $\mathcal{A}$ for objective* $\mathsf{LimInfAvg} \geq 0$ *is in NP.*

Proof. By Lemma 18 it is enough to guess a memoryless modular strategy (the memoryless modular strategy is the polynomial witness) and verify that it is indeed a winning strategy. The verification can be achieved in polynomial time using the polynomial time algorithms of Section 2 for WPSs with mean-payoff objectives. □

Strict inequalities and stack boundedness. Note that by Corollary 1 the results of Lemma 20 and Theorem 8 also hold for objective $\mathsf{LimSupAvg} \geq 0$. For modular strategies we only presented the result for mean-payoff objectives with non-strict inequalities. The results for strict inequalities follow from an adaptation of the proofs for non-strict inequalities (for which we prove memoryless modular strategies are sufficient). Moreover, the results also follow from mean-payoff objectives with the stack boundedness condition for the following reason: we observe that the manipulated operations never increase the stack height. Thus from our results it follow that if there is a modular winning strategy to ensure mean-payoff objective along with stack boundedness, then there is a memoryless modular strategy. Hence the NP upper bound follows for strict inequalities as well as for stack boundedness.

4.3 NP-hardness of the modular winning strategy problem

In this section we establish the NP-hardness of the modular winning strategy problem. Our hardness result will be for one-player WRGs (player-1 WRGs), where every module will have single exit, and the weights are $\{-1, 0, +1\}$. In other words, our hardness result show that even a very simple version of the problem (single exit one-player WRGs with constant weights) is NP-hard.

Reduction. We present a reduction from the 3-SAT problem (satisfiability of a CNF formula where every clause has exactly three distinct literals). Consider a 3-SAT formula $\varphi(x_1, x_2, \ldots, x_n) = \bigwedge_{i=1}^{m} cl_i$, over n variables $x_1, x_2, \ldots, x_n$, and m clauses $cl_1, cl_2, \ldots, cl_m$. A literal is a variable x_i or its negation $\neg x_i$. We construct a player-1 WRG as follows: $\mathcal{A}_\varphi = \langle A_0, x_1, \neg x_1, x_2, \neg x_2, \ldots, x_n, \neg x_n, cl_1, cl_2, \ldots, cl_m \rangle$ in the following way: there is an initial module A_0, there is a module for every literal and for every clause. We now describe the modules.

Module A_0. The module invokes in an infinite loop in sequence the modules $cl_1, cl_2, \ldots cl_m$, and all the transitions in this module have weight zero.

Module for clause cl_i. There is an edge from the entry node of module cl_i to a box that invokes module y, for every literal y that appears in the clause cl_i. There is also an edge from the return node of y to the exit node of cl_i. All the weights in this module are zero.

Module for literal y_i. The entry node of y_i has outdegree two (left edge and right edge). The left edge is the FALSE edge, which leads to the exit node, and has a weight -1. The right edge is the TRUE edge, which leads to a box that invokes a call for module $\neg y_i$, and its weight is -1. The return of the box leads to the exit node and the edge weight is $+2$. The reduction is illustrated pictorially in Fig 5.

Observation 2 *Every path from the entry node of the module y_i to its exit node, has a weight of at most* 0*, hence every path from the entry node of module cl_i to its exit node, has a weight of at most* 0*.*

Lemma 21 *There exists a modular winning strategy for player 1 in $\mathcal{A}_\varphi$ for the objective* $\mathsf{LimInfAvg} \geq 0$ *iff φ is satisfiable.*

Proof. We first observe that every modular strategy in $\mathcal{A}_\varphi$ is memoryless modular strategy. Observe that modular strategy for player 1 is a selection of a literal for every clause module, and selecting either the TRUE or FALSE edge for every literal module. We now present both directions of the proof.

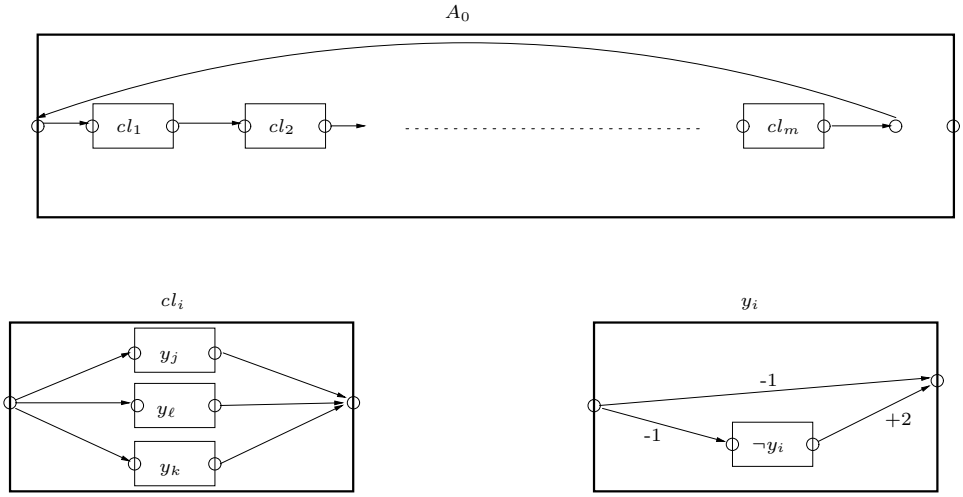

Fig. 5. NP-hardness reduction

- *Modular winning strategy implies satisfiable.* Let τ be a modular winning strategy for player 1, such that the literal y_i^τ is chosen for clause cl_i. First, towards contradiction, let us assume that there exists $i \in \{1, \ldots, m\}$ such that τ selects the FALSE edge for module y_i^τ. Hence the weight of the path inside the module cl_i is negative. Thus, due to Observation 2, the mean-payoff value according to τ is negative. Therefore τ selects the TRUE edge for the module y_i^τ. Next, towards contradiction, let us assume that there exist $i, j \in \{1, \ldots, m\}$ such that $y_i^\tau = \neg y_j^\tau$ and τ selects the TRUE edge for both modules. In this case, the play will never exit module y_i^τ, and will go forever through edges with negative weights. Therefore if τ selects the TRUE edge for y_i^τ, then it does not select the TRUE edge for $\neg y_i^\tau$. Due to the above, the assignment that assigns a true value to the literal y_i^τ in clause cl_i is a valid (non-conflicting) assignment that satisfies φ.
- *Satisfiable implies modular winning strategy.* Let $\overline{x}$ be a satisfying assignment (a non-conflicting assignment of truth values to variables) for the formula φ. We construct a modular winning strategy $\tau_{\overline{x}}$ as follows. In module cl_i, the modular strategy invokes the module $y_i^{\overline{x}}$, where $y_i^{\overline{x}}$ is a literal for which $\overline{x}$ assigns a true value (since $\overline{x}$ is a satisfying assignment such a literal must exist). In module y_i, follow the TRUE edge if $\overline{x}$ assigns a true value to the literal y_i, and follow the FALSE edge otherwise. It is straight forward to verify that the mean-payoff in a play according to $\tau_{\overline{x}}$ is zero.

This completes the proof. □

Observe that in the hardness reduction we have used positive weight +2 for simplicity, which can be split into two edges of weight +1 each. Hence we have the following theorem.

Theorem 9 *The modular winning strategy problem is NP-hard for one-player WRGs (player-1 WRGs) with single exit for every module and objective* $\mathsf{LimInfAvg} \geq 0$ *with edge weights in* $\{-1, 0, +1\}$.

Strict inequalities and stack boundedness. We first observe that the above reduction also holds for $\mathsf{LimSupAvg} \geq 0$ objective. Moreover, whenever the 3-SAT formula φ is satisfiable, then the witness memoryless modular strategy along with mean-payoff objective also ensures stack boundedness. Hence the hardness result follows from mean-payoff objectives with non-strict inequalities as well as for stack boundedness. The result for strict inequality is obtained as follows: we modify the above reduction by changing the weight of the edge back to the entry node of A_0 from 0 to 1. Then if the formula φ is satisfiable, then the average payoff for memoryless modular strategies is at least $\frac{1}{|V|}$, where $|V|$ is the number of vertices, and if the formula φ is not satisfiable, then the mean-payoff under all memoryless modular strategies is at most 0. Hence the hardness follows also for mean-payoff objectives with strict inequalities. We have the following theorem summarizing the results for modular strategies.

Theorem 10 *The following assertions hold for WRGs with objectives* $\Phi \bowtie 0$*, for* $\bowtie \in \{\geq, >\}$*,* $\Phi \in \{\mathsf{LimSupAvg}, \mathsf{LimInfAvg}\}$*, as well as objectives* Φ *along with stack boundedness.*

1. *If there is a modular winning strategy, then there is a memoryless modular winning strategy.*
2. *The decision problem of whether there is a memoryless modular winning strategy is NP-complete.*
3. *The decision problem is NP-hard for player-1 WRGs with single exit for every module and edge weights in* $\{-1, 0, +1\}$.

5 Conclusion

In this work we study for the first time mean-payoff objectives in pushdown games and present a complete characterization of computational and strategy complexity. We show that pushdown systems (one-player pushdown games) with mean-payoff objectives under global strategies can be solved in polynomial time, whereas pushdown games with mean-payoff objectives under global strategies is undecidable. For modular strategies both pushdown systems and pushdown games with mean-payoff objectives are NP-complete. We also show that global strategies for mean-payoff objectives in general require infinite memory even in pushdown systems; whereas memoryless strategies suffice for modular strategies for mean-payoff objectives.

References

1. S. Almagor, U. Boker, and O. Kupferman. What's decidable about weighted automata? In *ATVA*, pages 482–491, 2011.
2. R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. W. Reps, and M. Yannakakis. Analysis of recursive state machines. *ACM Trans. Program. Lang. Syst.*, 27(4):786–818, 2005.
3. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.
4. R. Alur, S. La Torre, and P. Madhusudan. Modular strategies for infinite games on recursive graphs. In *CAV*, pages 67–79, 2003.
5. R. Alur, S. La Torre, and P. Madhusudan. Modular strategies for recursive game graphs. *Theor. Comput. Sci.*, 354(2):230–249, 2006.
6. R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Better quality in synthesis through quantitative objectives. In *CAV*, pages 140–156, 2009.
7. R. Bloem, K. Greimel, T. A. Henzinger, and B. Jobstmann. Synthesizing robust systems. In *FMCAD*, pages 85–92, 2009.
8. U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *LICS*, pages 43–52, 2011.
9. T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive Markov decision processes. *Inf. Comput.*, 206(5):520–537, 2008.
10. T. Brázdil, V. Brozek, A. Kucera, and J. Obdrzálek. Qualitative reachability in stochastic BPA games. *Inf. Comput.*, 209(8):1160–1183, 2011.
11. J.R. Büchi and L.H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the AMS*, 138:295–311, 1969.
12. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4), 2010.
13. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.
14. L. de Alfaro and T.A. Henzinger. Interface theories for component-based design. In *EMSOFT*, LNCS 2211, pages 148–165. Springer, 2001.
15. M. Droste and I. Meinecke. Describing average- and longtime-behavior by weighted MSO logics. In *MFCS*, pages 537–548, 2010.
16. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
17. E.A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *FOCS'88*, pages 328–337. IEEE, 1988.
18. E.A. Emerson and C. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS*, pages 368–377. IEEE, 1991.
19. K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *ICALP'05*, LNCS 3580, Springer, pages 891–903, 2005.
20. K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *J. ACM*, 56(1), 2009.
21. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, LNCS 2500. Springer, 2002.

22. Y. Gurevich and L. Harrington. Trees, automata, and games. In *STOC'82*, pages 60–65. ACM Press, 1982.
23. T. A. Henzinger, O. Kupferman, and S. Rajamani. Fair simulation. *Information and Computation*, 173:64–81, 2002.
24. M. Jurdzinski. Deciding the winner in parity games is in UP ∩ co-UP. *Information Processing Letters*, 68(3):119–124, 1998.
25. R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978.
26. D Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. In *ICALP*, pages 101–112, 1992.
27. T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *Siam Review*, 11:604–607, 1969.
28. R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65:149–184, 1993.
29. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190. ACM Press, 1989.
30. P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.
31. W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
32. I. Walukiewicz. Model checking CTL properties of pushdown systems. In *FSTTCS*, pages 127–138, 2000.
33. I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001.
34. M. Yannakakis. Graph-theoretic methods in database theory. In *PODS*, pages 230–242, 1990.
35. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. In *Theoretical Computer Science*, volume 200(1-2), pages 135–183, 1998.
36. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.