arXiv:1404.5894v6 [cs.GT] 17 Feb 2020

Adding Negative Prices to Priced Timed Games*

Thomas Brihaye[1], Gilles Geeraerts[2**], Shankara Narayanan Krishna[3], Lakshmi Manasa[3], Benjamin Monmege[2], and Ashutosh Trivedi[3]

[1] Université de Mons, Belgium, thomas.brihaye@umons.ac.be
[2] Université libre de Bruxelles, Belgium, gigeerae,benjamin.monmege@ulb.ac.be
[3] IIT Bombay, India, krishnas,manasa,trivedi@cse.iitb.ac.in

Abstract. Priced timed games (PTGs) are two-player zero-sum games played on the infinite graph of configurations of priced timed automata where two players take turns to choose transitions in order to optimize cost to reach target states. Bouyer et al. and Alur, Bernadsky, and Madhusudan independently proposed algorithms to solve PTGs with non-negative prices under certain divergence restriction over prices. Brihaye, Bruyère, and Raskin later provided a justification for such a restriction by showing the undecidability of the optimal strategy synthesis problem in the absence of this divergence restriction. This problem for PTGs with one clock has long been conjectured to be in polynomial time, however the current best known algorithm, by Hansen, Ibsen-Jensen, and Miltersen, is exponential. We extend this picture by studying PTGs with both negative and positive prices. We refine the undecidability results for optimal strategy synthesis problem, and show undecidability for several variants of optimal reachability cost objectives including reachability cost, time-bounded reachability cost, and repeated reachability cost objectives. We also identify a subclass with bi-valued price-rates and give a pseudo-polynomialalgorithm to partially answer the conjecture on the complexity of one-clock PTGs.

1 Introduction

Timed automata [2] equip finite automata with a finite number of real-valued variables—aptly called clocks—that evolve with a uniform rate. The syntax of timed automata also permits specifying *transition guards* and *location (state) invariants* using the constraints over clock valuations, and resetting the clocks as a means to remember the time since the execution of a transition. Timed automata is a well-established formalism to specify time-critical properties of real-time systems. Priced timed automata [3,4] (PTAs) extend timed automata with price information by augmenting locations with price-rates and transitions with discrete prices. The natural reachability-cost optimization problem for PTAs is

* The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement number 601148 (CASSTING)
** Supported by a 'Crédit aux chercheurs' number 1808881 of the F.R.S./FNRS.

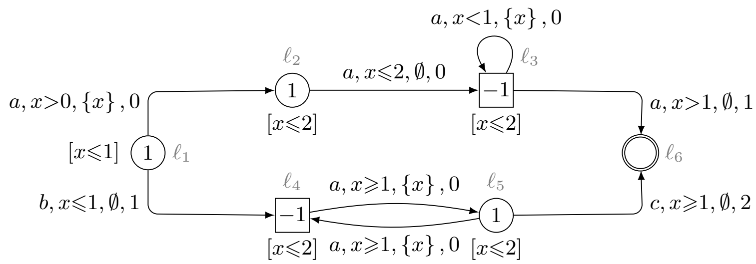


Fig. 1. A price timed game arena with one clock

known to be decidable with the same complexity [6] as the reachability problem (PSPACE-complete), and forms the backbone of many applications of timed automata including scheduling and planning.

Priced timed games (PTGs) extend the reachability-cost optimization problem to the setting of competitive optimization problem, and form the basis of optimal controller synthesis [20] for real-time systems. We study turn-based variant of these games where the game arena is a PTA with a partition of the locations between two players Player 1 and Player 2. A play of such a game begins with a token in an initial location, and at every step the player controlling the current location proposes a valid timed move, i.e., a time delay and a discrete transition, and the state of the system is modified accordingly. The play stops if the token reaches a location from a distinguished set of *target locations*, and the payoff of the play is equal to the cost accumulated before reaching the target location. If the token never reaches a target location then the game continues forever, and the payoff in this case is $+\infty$ irrespective of actual cost of the infinite play. We characterize a PTG according to the objectives of Player 1. Since we study zero-sum games, the objective of Player 2 is also implicitly defined. We study PTGs with the following objectives: (i) *Constrained-price reachability* objective $\mathsf{Reach}(\bowtie K)$ is to achieve a payoff C of the play such that $C \bowtie K$ where $\bowtie \in \{\leqslant, <, =, >, \geqslant\}$ and $K \in \mathbb{N}$; (ii) *Bounded-time reachability* objective $\mathsf{TBReach}(K, T)$ is to keep the payoff of the play less than K while keeping the total time elapsed within T units; and (iii) *Repeated reachability* objective $\mathsf{RReach}(\eta)$ is to visit target infinitely often with a payoff in the interval $[-\eta, \eta]$.

An example of PTG with clock variable x and six locations is given in Fig. 1. We depict Player 1 locations as circles and Player 2 locations as boxes. The numbers inside locations denote their price-rates, while the clock constraints next to a location depicts its invariant. We denote a transition, as usual, by an arrow between two location annotated by a tuple a, g, r, c where a is the label, g is the guard, r is the clocks reset set, and c is the cost of the transition.

Related work. PTGs with constrained-price reachability objective $\mathsf{Reach}(\leqslant K)$ were independently introduced in [9] and [1], with semi-algorithms to decide the existence of winning strategy for Player 1 in PTGs with nonnegative prices. They also showed that under the *strongly non-Zeno assumption* on prices the proposed semi-algorithms always terminate. This assumption was justified in [11] by showing that, in the absence of non-Zeno assumption, the problem of deciding the existence of winning strategy for the objective $\mathsf{Reach}(\leqslant K)$ is undecidable for PTGs with five or more clocks. This result has been later refined in [7] by showing that the problem is undecidable for PTGs with three or more clocks and nonnegative prices. In [5] is showed the undecidability of the existence of winning strategy problem for $\mathsf{Reach}(\leqslant K)$ objective over PTGs with both positive and negative price-rates and two or more clocks.

On a positive side, the existence of winning strategy for $\mathsf{Reach}(\leqslant K)$ problem for PTGs with one clock when the price-rates are restricted to values 0 and $d \in \mathbb{N}$ has been shown decidable in [11], by proving that the semi-algorithms in [9,1] always terminate. However, the authors did not provide any complexity analysis of their algorithm. One-clock PTGs with nonnegative prices are reconsidered in [10], and a 3-EXPTIME algorithm is given to solve the problem, while the best known lower bound is PTIME. A tighter analysis of the problem is presented in [21] that lowered the known complexity of this problem to EXPTIME, namely $2^{O(n^2+m)}$ where n is the number of locations and m is the number of transitions. A significant improvement over the complexity $(m12^n n^{O(1)})$ was given in [15] by improving the analysis of the semi-algorithms by [9,1].

Contributions. We consider PTGs with both negative and positive prices. We show that deciding the existence of a winning strategy for reachability objective $\mathsf{Reach}(\bowtie K)$ is undecidable for PTGs with two or more clocks. In [19], a theory of time-bounded verification has been proposed, arguing that restriction to bounded-time domain reclaims the decidability of several key verification problems. As an example, we cite [12] where authors recovered the decidability of the reachability problem for hybrid automata under time-bounded restriction. We begin studying PTGs with bounded reachability objective $\mathsf{TBReach}(K, T)$ hoping that the problem may be decidable due to time-bounded restriction. However, we answer this question negatively by showing undecidability of the existence of winning strategy problem for PTGs with six or more clocks. We also show the undecidability for the corresponding problem for repeated reachability objective $\mathsf{RReach}(\eta)$ for PTGs with three or more clocks.

On the positive side, we introduce a previously unexplored subclass of one-clock PTGs, called one-clock bi-valued priced timed games (1BPTGs), where the price-rates of locations are taken from a set of two integers from $\{-d, 0, d\}$ (with d any positive integer). None of the previously cited algorithms can be applied in this case since we do not assume non-Zenoness of prices and consider both positive and negative prices. After showing a determinacy result for 1BPTGs, we proceed to give a pseudo-polynomial time algorithm to compute the value and ε-optimal strategy for both players with $\mathsf{Reach}(\leqslant K)$ objective.

Remark on a previous version of this paper. Note that the five first arXiv revisions of the present paper (up to `https://arxiv.org/abs/1404.5894v5`), as well as the published version of the paper [13], contained a flawed claim that has been kindly pointed out to us by the authors of [16]. Namely, we were claiming that our results entail that 'the complexity [of computing the value and ε-optimal strategy for both players with $\mathsf{Reach}(\leqslant K)$ objective] drops to *polynomial* for 1BPTGs if the price-rates are non-negative integers.' While our theorems and corollaries remain correct, this conclusion was not, and we are indebted to those authors for kindly letting us know. Such claims have been removed from the present version.

2 Reachability-cost games on priced game graphs

PTGs can be considered as a succinct representation of some games on uncountable state space characterized by the configuration graph of timed automata.

We begin by introducing the concepts and notations related to such more general game arenas that we call priced game graphs.

Definition 1. *A* priced game graph *is a tuple $\mathcal{G} = (V, A, E, \pi, V_f)$ where:*
- *$V = V_1 \uplus V_2$ is the set of vertices partitioned into the sets V_1 and V_2;*
- *A is a set of labels called actions;*
- *$E\colon V \times A \to V$ is the edge function defining the set of labeled edges;*
- *$\pi\colon V \times A \to \mathbb{R}$ is the price function that assigns prices to edges; and*
- *$V_f \subseteq V$ is the set of target vertices.*

We call a game graph finite *if both V and A are finite and with rational prices.*

A reachability-cost game begins with a token placed on some initial vertex v_0. At each round, the player who controls the current vertex v chooses an action $a \in A$ and the token is moved to the vertex $E(v, a)$. The two players continue moving the token in this fashion, and give rise to an infinite sequence of vertices and actions called a play of the game. Formally, a finite play r is a finite sequence of vertices and actions $\langle v_0, a_0, v_1, a_1, \ldots, a_{n-1}, v_n\rangle$ where for each $0 \leqslant i < n$ we have that $v_{i+1} = E(v_i, a_i)$; we write $\mathsf{Last}(r)$ for the last vertex of a finite play, here $\mathsf{Last}(r) = v_n$. An infinite play is defined analogously. We write $\mathsf{FPlay}_{\mathcal{G}}$ ($\mathsf{FPlay}_{\mathcal{G}}(v)$) for the set of finite plays (starting from the vertex v) of the game graph $\mathcal{G}$. We often omit the subscript when the game arena is clear form the context. We similarly define Play and $\mathsf{Play}(v)$ for the set of infinite plays. For all $k \geqslant 0$, we let $r[k]$ be the prefix $\langle v_0, a_0, \ldots, a_{k-1}, v_k\rangle$ of r, and we denote by $\mathsf{Cost}(r[k]) = \sum_{i=0}^{k-1} \pi(v_i, a_i)$ its cost. We write $\mathsf{Stop}(r)$ for the index of the first target vertex in r, i.e., $\mathsf{Stop}(r) = \inf\{k : v_k \in F\}$. We define the cost of an infinite run $r = \langle v_0, a_1, v_1, \ldots\rangle$ as $\mathsf{Cost}(r) = +\infty$ if $\mathsf{Stop}(r) = \infty$ and $\mathsf{Cost}(r) = \mathsf{Cost}(r[\mathsf{Stop}(r)])$, otherwise.

A strategy for a Player i (for $i \in \{1, 2\}$) is a partial function $\sigma : \mathsf{FPlay} \to A$ that is defined for a run $r = \langle v_0, a_0, v_1, \ldots, a_{n-1}, v_n\rangle$ if $v_n \in V_i$ and is such that $E(v_n, \sigma(r))$ is defined, i.e., there is a $\sigma(r)$-labeled outgoing transition from v_n. We denote by $\mathsf{Strat}_i(\mathcal{G})$ (or Strat_i when the game arena is clear) the set of strategies for Player i. Given a strategy profile $(\sigma_1, \sigma_2) \in \mathsf{Strat}_1 \times \mathsf{Strat}_2$ for both players, and an initial vertex $v \in V$, the unique infinite play $\mathsf{Play}(v, \sigma_1, \sigma_2) = \langle v_0, a_0, v_1, \ldots v_k, a_k, v_{k+1}, \ldots\rangle$ is such that for all $k \geqslant 0$ if $v_k \in V_i$, for $i = 1, 2$, then $a_{k+1} = \sigma_i(r[k])$ and $v_{k+1} = E(v_k, a_{k+1})$. A strategy σ is said to be *memoryless* (or *positional*) if, for all finite plays $r, r' \in \mathsf{FPlay}$ with $\mathsf{Last}(r) =$

$\mathsf{Last}(r')$ we have that $\sigma(r) = \sigma(r')$. Similarly, *finite-memory strategies* can be defined as implementable with Moore machines, see [14] for a formal definition.

We consider optimal reachability-cost games on priced game graphs, where the goal of Player 1 is to minimize the reachability-cost, while the goal of Player 2 is the opposite. The standard concepts of upper value and lower value of the optimal reachability-cost game are defined in straightforward manner. Formally, the upper-value $\overline{\mathsf{Val}}_{\mathcal{G}}(v)$ and lower value $\underline{\mathsf{Val}}_{\mathcal{G}}(v)$ of a game starting from a vertex v is defined as $\overline{\mathsf{Val}}_{\mathcal{G}}(v) = \inf_{\sigma_1 \in \mathsf{Strat}_1} \sup_{\sigma_2 \in \mathsf{Strat}_2} \mathsf{Cost}(\mathsf{Play}(v, \sigma_1, \sigma_2))$ and $\underline{\mathsf{Val}}_{\mathcal{G}}(v) = \sup_{\sigma_2 \in \mathsf{Strat}_2} \inf_{\sigma_1 \in \mathsf{Strat}_1} \mathsf{Cost}(\mathsf{Play}(v, \sigma_1, \sigma_2))$. It is easy to see that $\underline{\mathsf{Val}}_{\mathcal{G}}(v) \leqslant \overline{\mathsf{Val}}_{\mathcal{G}}(v)$ for every vertex v. We say that a game is *determined* if the lower and the upper values match for every vertex v, and in this case, we say that the optimal value of the game exists and we let $\mathsf{Val}_{\mathcal{G}}(v) = \underline{\mathsf{Val}}_{\mathcal{G}}(v) = \overline{\mathsf{Val}}_{\mathcal{G}}(v)$. The determinacy of these games follow from Martin's determinacy theorem, and an alternative proof is given in [14].

In the following, we write $\mathsf{Cost}(v, \sigma_1)$ for the value of the strategy σ_1 of Player 1 from vertex v, i.e., $\mathsf{Cost}(v, \sigma_1) = \sup_{\sigma_2 \in \mathsf{Strat}_2} \mathsf{Cost}(\mathsf{Play}(v, \sigma_1, \sigma_2))$. A strategy σ_1^* of Player 1 is said to be optimal from v if $\mathsf{Cost}(v, \sigma_1^*) = \overline{\mathsf{Val}}_{\mathcal{G}}(v)$. Optimal strategies do not always exist, hence we also define ε-optimal strategies. For $\varepsilon > 0$, a strategy σ_1 is an ε-optimal strategy if for all vertex $v \in V$, $\mathsf{Cost}(v, \sigma_1) \leqslant \overline{\mathsf{Val}}_{\mathcal{G}}(v) + \varepsilon$. In this paper we exploit the following result from [14].

Theorem 1 ([14]). *Let $\mathcal{G}$ be a finite priced game graph.*
1. *Deciding $\mathsf{Val}_{\mathcal{G}}(v) = +\infty$ is in Polynomial Time.*
2. *Deciding $\mathsf{Val}_{\mathcal{G}}(v) = -\infty$ is in* NP $\cap$ co-NP*, can be achieved in pseudo-polynomial time*[4] *and is as hard as solving mean-payoff games [22].*
3. *Given $-\infty < \mathsf{Val}_{\mathcal{G}}(v) < +\infty$ for every vertex v, optimal strategies exist for both players. In particular, Player 2 has optimal memoryless strategies, while Player 1 has optimal finite-memory strategies. Moreover, the values $\mathsf{Val}_{\mathcal{G}}(v)$, as well as optimal strategies, can be computed in pseudo-polynomial time.*

It must be noticed that, in the presence of negative costs, and even when every vertex v has a finite value $\mathsf{Val}_{\mathcal{G}}(v) \in \mathbb{R}$, memoryless optimal strategies may not exist for Player 1, as pointed out in [14, Example 1].

3 Priced timed games

In order to formally introduce priced timed games, we need to define the concepts of clocks, clock valuations, constraints, and zones. Let $\mathcal{X}$ be a finite set of real-valued variables called *clocks*. A clock valuation on $\mathcal{X}$ is a function $\nu \colon \mathcal{X} \to \mathbb{R}_{\geqslant 0}$ and we write $V(\mathcal{X})$ for the set of clock valuations. Abusing notation, we also treat a valuation ν as a point in $\mathbb{R}^{|\mathcal{X}|}$. If $\nu \in V(\mathcal{X})$ and $t \in \mathbb{R}_{\geqslant 0}$ then we write $\nu + t$ for the clock valuation defined by $(\nu + t)(c) = \nu(c) + t$ for all $c \in \mathcal{X}$. For $C \subseteq \mathcal{X}$, we write $\nu[C := 0]$ for the valuation where $\nu[C := 0](c)$ equals 0 if $c \in C$ and $\nu(c)$ otherwise. A clock constraint over $\mathcal{X}$ is a conjunction of simple constraints

[4] polynomial time if the prices are encoded in unary.

of the form $c \bowtie i$ or $c - c' \bowtie i$, where $c, c' \in \mathcal{X}$, $i \in \mathbb{N}$ and $\bowtie \in \{<, >, =, \leqslant, \geqslant\}$. A clock zone is a finite set of clock constraints that defines a convex set of clock valuations. We write $Z(\mathcal{X})$ for the set of clock zones over the set of clocks $\mathcal{X}$.

Definition 2. *A priced timed game is a tuple $\mathcal{A} = (L, \mathcal{X}, \mathrm{Inv}, \Sigma, \delta, \omega, L_f)$ where:*
- *$L = L_1 \uplus L_2$ is a finite set of locations, partitioned into the sets L_1 and L_2;*
- *$\mathcal{X}$ is a finite set of clocks;*
- *$\mathrm{Inv}\colon L \to Z(\mathcal{X})$ associates an invariant to each location;*
- *Σ is a finite set of labels;*
- *$\delta\colon L \times \Sigma \to Z(\mathcal{X}) \times 2^{\mathcal{X}} \times L$ is a transition function that maps a location $\ell \in L$ and label $a \in \Sigma$ to a clock zone $\zeta \in Z(\mathcal{X})$ representing the guard on the transition, a set of clocks $R \subseteq \mathcal{X}$ to be reset and successor location $\ell' \in L$;*
- *$\omega\colon L \cup \Sigma \to \mathbb{Z}$ is the price function; and*
- *and $L_f \subseteq L$ is the set of target locations.*

A configuration of a PTG is a tuple $(\ell, \nu) \in L \times V$ where ℓ is a location, ν is a clock valuation and $\nu \in \mathrm{Inv}(\ell)$. A timed action is a tuple $\tau = (t, a) \in \mathbb{R}_{\geqslant 0} \times \Sigma$ where t is a time delay and a is a label. In the following, for a timed move $\tau = (t, a) \in \mathbb{R}_{\geqslant 0} \times \Sigma$, we let $\mathsf{del}(\tau) = t$ be the delay part and $\mathsf{lab}(\tau) = a$ be the label part. The semantics of a PTG is given as an infinite priced game graph.

Definition 3 (Semantics). *The semantics of a PTG $\mathcal{A} = (L, \mathcal{X}, \mathrm{Inv}, \Sigma, \delta, \omega, L_f)$ is given as a priced game graph $[\![\mathcal{A}]\!] = (S, \Gamma, \Delta, \kappa, S_f)$ where*
- *$S = \{(\ell, \nu) \in L \times V \mid \nu \in \mathrm{Inv}(\ell)\}$ is the set of configurations of the PTG;*
- *$\Gamma = \mathbb{R}_{\geqslant 0} \times \Sigma$ is the set of timed moves;*
- *$\Delta\colon S \times \Gamma \to S$ is the transition function defined by $(\ell', \nu') = \Delta((\ell, \nu), (t, a))$ if $\delta(\ell, a) = (\zeta, R, \ell')$ such that $\nu + t \in \zeta$, $\nu + t' \in \mathrm{Inv}(\ell)$ for all $0 \leqslant t' \leqslant t$, and $\nu' = (\nu + t)[R := 0]$;*
- *$\kappa : S \times \Gamma \to \mathbb{R}$ is such that $\kappa((\ell, \nu), (t, a)) = \omega(\ell) \times t + \omega(a)$; and*
- *$S_f \subseteq S$ is such that $(\ell, \nu) \in S_f$ iff $\ell \in L_f$.*

The concepts of a play, its cost, and strategies of players for a PTG $\mathcal{A}$ is defined via corresponding objects for its semantic priced game graph $[\![\mathcal{A}]\!]$. In the previous section we introduced games with reachability-cost objective for priced game graphs. We also study the following winning objectives for Player 1 in the context of priced timed games; the objective for Player 2 is the opposite.

1. **Constrained-price reachability**. The constrained-price reachability objective $\mathsf{Reach}(\leqslant K)$ is to keep the payoff within a given bound $K \in \mathbb{N}$. Objectives $\mathsf{Reach}(\bowtie K)$ for constrains $\bowtie \in \{<, =, >, \geqslant\}$ are defined analogously.
2. **Bounded-time reachability**. Given constants $K, T \in \mathbb{N}$, the bounded-time reachability objective $\mathsf{TBReach}(K, T)$ is to keep the payoff of the play less than or equal to K while keeping the total time elapsed within T units.
3. **Repeated reachability**. For this objective, we consider slightly different semantics of the game where the play continues forever, and the repeated reachability objective $\mathsf{RReach}(\eta)$, $\eta \in \mathbb{R}_{\geqslant 0}$ is to visit target locations infinitely often each time with a payoff in a given interval $[-\eta, \eta]$.

In Section 4, we sketch the proof of the following negative result regarding the decidability of PTGs with these objectives. This result is particularly surprising for bounded-time reachability objective, since bounded-time restriction has been shown to recover decidability in many related problems [19,12].

Theorem 2. *Let $\mathcal{A}$ be a priced timed game arena. The decision problems corresponding to the existence of winning strategy for following objectives are undecidable:*

1. $\mathsf{Reach}(\bowtie K)$ *objective for PTGs with two or more clocks and arbitrary prices;*
2. $\mathsf{TBReach}(K,T)$ *objective for PTGs with five or more clocks; and prices 0,1;*
3. $\mathsf{RReach}(\eta)$ *objective for PTGs with three or more clocks and arbitrary prices.*

To recover decidability, we consider a subclass of one-clock PTGs. In this subclass, the set of clocks $\mathcal{X}$ is a singleton $\{x\}$, and price-rates of the locations come from a doubleton set $\{p^-,p^+\}$ with $p^- < p^+$ two distinct elements of $\{-1,0,1\}$ (no condition is made on the prices $\omega(a)$ of labels $a \in \Sigma$). We call these restricted games *one-clock bi-valued priced timed games*, abbreviated as $1\mathrm{PTG}(p^-,p^+)$, or 1BPTG if p^- and p^+ do not matter. All our results may easily be extended to the case where p^- and p^+ are taken from the set $\{-d,0,d\}$ with $d \in \mathbb{N}$. We devote Section 5 to the proof of the following decidability results.

Theorem 3. *We have the following results:*

1. 1BPTG*s are determined.*
2. *The value of a* 1BPTG *can be computed in pseudo-polynomial time.*
3. *Given that a* 1BPTG *has a finite value, an ε-optimal strategy for Player 1 can be computed in pseudo-polynomial time.*
4. *Aforementioned complexities drop to polynomial time for* $1\mathrm{PTG}(0,1)$ *with prices of labels taken from* $\mathbb{N}$.

4 Undecidability results

In this section we provide a proof sketch of our undecidability result (Theorem 2) by reducing the halting problem for two counter machines (see [18]) to the existence of a winning strategy for Player 1 for the desired objective. For all the three objectives, given a two counter machine, we construct a PTG $\mathcal{A}$ whose building blocks are the modules for instructions. In these reductions the objective of Player 1 is linked to a faithful simulation of various increment, decrement, and zero-test instructions of the machine by choosing appropriate delays to adjust the clocks to reflect changes in counter values. The goal of Player 2 is then to verify the simulation performed by Player 1. Proofs of correctness of the reductions, as well as more details can be found in the appendix.

Constrained-price reachability objectives $\mathsf{Reach}(\bowtie K)$. The result in the case $\mathsf{Reach}(\leqslant K)$ is a consequence of the result in [5]. Undecidability for other comparison operators $\bowtie$ is a new contribution. We only consider the objective $\mathsf{Reach}(=1)$ in this section, since proofs for other constraints are similar.

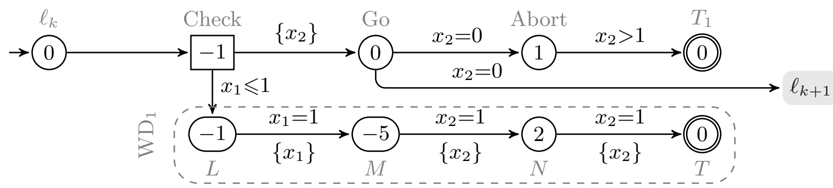


Fig. 2. Decrement module for the objection $\mathsf{Reach}(=1)$

Our reduction uses a PTG with two clocks x_1 and x_2, arbitrary price-rates for locations and no prices for labels. Each counter machine instruction (increment, decrement, and test for zero value) is specified using a PTG module. The main invariant in our reduction is that upon entry into a module, we have that $x_1 = \frac{1}{5^{c_1}7^{c_2}}$ and $x_2 = 0$ where c_1 (respectively, c_2) is the value of counter C_1 (respectively, C_2). We outline the simulation of a decrement instruction for counter C_1 in Fig. 2. Let us denote by $x_{old} = \frac{1}{5^{c_1}7^{c_2}}$ the value of x_1 while entering the module. At the location ℓ_{k+1} of the module, $x_1 = x_{new}$ should be $5x_{old}$ to correctly decrement counter C_1. At location ℓ_k, Player 1 spends a non-deterministic amount of time $t_k = x_{new} - x_{old}$ such that $x_{new} = 5x_{old} + \varepsilon$. To correctly decrement C_1, ε should be 0, and t_k must be $\frac{4}{5^{c_1}7^{c_2}}$. At location Check, Player 2 could choose to go to Go (in order to continue the simulation of the machine) or go to the widget WD$_1$, if he suspects that $\varepsilon \neq 0$. If Player 2 spends time $t > 0$ in the location Check before proceeding to Go, then Player 1 can enter the location Abort (to abort the simulation), rather than going to ℓ_{k+1}. Player 1 spends $1+t$ time in location Abort and reaches a target T_1 with cost 1 (and thus achieve his objective). However, if $t = 0$ then entering location Abort will make the cost to be greater than 1 (which is losing for Player 1). If Player 2 decides to enter widget WD$_1$, then the cost upon reaching the target in the widget WD$_1$ is $1+\varepsilon$ which is 1 iff $\varepsilon = 0$.

Bounded-time reachability objective. We sketch the reduction for objective $\mathsf{TBReach}(K, T)$. Our reduction uses a PTG with price-rates 0 or 1 on locations, and zero prices on labels, along with five clocks x_1, x_2, z, a, b. On entry into a module for the $(k+1)$th instruction, we always have one of the two clocks x_1, x_2 with value $\frac{1}{2^{k+c_1}3^{k+c_2}}$ and other is 0. Clock z keeps track of the total time elapsed during simulation of an instruction: we always have $z = 1 - \frac{1}{2^k}$ at the end of simulating kth instruction. Thus, time $\frac{1}{2}$ is spent simulating the first instruction, $\frac{1}{4}$ for the second instruction and so on, so that the total time spent in simulating the main modules is less than 1. The main challenge here is to ensure that only a bounded time is spent along the entire simulation, along with updating the counter values correctly. Clocks a, b are used for rough work. For instance, if the $(k+1)$th instruction ℓ_{k+1} is an increment of C_1, and we have $x_1 = \frac{1}{2^{k+c_1}3^{k+c_2}}$, while $a = b = x_2 = 0$, and $z = 1 - \frac{1}{2^k}$, then at the end of the module simulating ℓ_{k+1}, we want $x_2 = \frac{1}{2^{k+1+c_1+1}3^{k+1+c_2}}$ and $x_1 = 0$ and $z = 1 - \frac{1}{2^{k+1}}$.

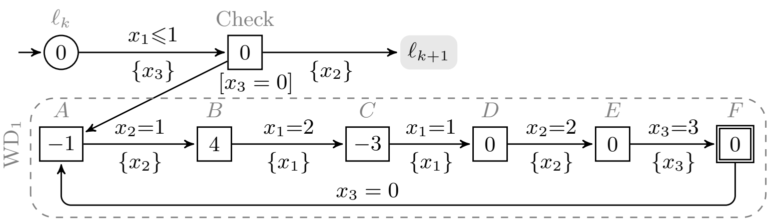


Fig. 3. Decrement module for Repeated reachability objective.

Repeated reachability objective. Finally, we consider the repeated reachability objective $\mathsf{RReach}(\eta)$. Our reduction uses a PTG with 3 clocks, and arbitrary price-rates, but zero prices for labels. On entry into a module, we have $x_1 = \frac{1}{5^{c_1}7^{c_2}}$, $x_2 = 0$ and $x_3 = 0$, where c_1, c_2 are the values of C_1 and C_2. Fig. 3 shows module to simulate decrement C_1. Location ℓ_k is entered with $x_1 = \frac{1}{5^{c_1}7^{c_2}}, x_2 = 0$ and $x_3 = 0$. To correctly decrement C_1, Player 1 should choose a delay of $\frac{4}{5^{c_1}7^{c_2}}$ in location ℓ_k. At location Check, no time can elapse because of the invariant. If Player 1 makes an error, and delays $\frac{4}{5^{c_1}7^{c_2}} + \varepsilon$ at ℓ_k ($\varepsilon \neq 0$) then Player 2 can jump in widget WD_1. The cost of going from location A to F is ε; each time we come back to A, the clock values with which A was entered are restored. Clearly, if $\varepsilon \neq 0$, Player 2 can incur a cost that is not in $[-\eta, \eta]$ by taking the loop from A to F a large number of times.

5 One-clock bi-valued priced timed games

This section is devoted to the proof of Theorem 3. First of all, let us assume that all 1BPTGs $\mathcal{A}$ we consider are *bounded*, i.e., that there is a global invariant in every location, of the form $x \leqslant M_K$ (where M_K denotes the greatest constant appearing in the clock guards and invariants of $\mathcal{A}$). This restriction comes w.l.o.g since every 1BPTG arena can be made bounded with a polynomial algorithm.[5]

Our proof of Theorem 3 is based on an extension of the classical notion of regions in timed automata, in the spirit of the regions introduced to define the corner point abstraction [8]. Indeed, to take the price into account, ε-optimal strategies do not take uniform decisions on the classical regions. That is why we need to subdivide each classical region into three parts: two small parts around the corners of the region (that we will call *borders* in the following, considering our one-clock setting), and a big part in-between. We will show that considering only strategies that never jump into those big parts is sufficient (Lemma 1). Lemma 2, later, shows a stronger result that one can restrict attention to strategies that play closer and closer to the borders of the regions as time elapses. Finally, we combine these results to show that a finite abstraction of 1BPTGs is sufficient to compute the value as well as ε-optimal strategies (Lemma 3). This

[5] By introducing auxiliary states in order to reset the clock x at every time unit once its value goes beyond M_k. The polynomial complexity holds only for one-clock PTGs.

not only yields the desired result, but also provides us further insight into the shape of ε-optimal strategies for both players.

5.1 Reduction to η-region-uniform strategies

Since we only consider one-clock PTGs, we need not consider the standard Alur-Dill regional equivalence relation. Instead, we consider special region equivalence relation characterized by the intervals with constants appearing in guards and invariants of $\mathcal{A}$ inspired by Laroussinie, Markey, and Schnoebelen construction [17]. Let $0{=}M_0{<}M_1{<}\cdots{<}M_K$ be the integers appearing in guards and invariants of $\mathcal{A}$. We say that two valuations $\nu, \nu' \in \mathbb{R}_{\geqslant 0}$ are region-equivalent (or lie in the same region), and we write $\nu \sim \nu'$, if for every $k \in \{0, \dots, K\}$, $\nu \leqslant M_k$ iff $\nu' \leqslant M_k$, and $\nu \geqslant M_k$ iff $\nu' \geqslant M_k$. We define the set of regions to be the set of equivalence classes of $\sim$. We extend the equivalence relation $\sim$ from valuations to configurations in a straightforward manner. We also generalize the regional equivalence relation to the plays. For two (finite or infinite) plays $r = \langle(\ell_0, \nu_0), (t_0, a_0), \dots\rangle$ and $r' = \langle(\ell'_0, \nu'_0), (t'_0, a'_0), \dots\rangle$ we say that $r \sim r'$ if the lengths of r and r' are equal, and they define sequences of regional equivalent states (i.e., $(\ell_i, \nu_i) \sim (\ell'_i, \nu'_i)$ for all $i \geqslant 0$) and follow equivalent timed actions (i.e., $a_i = a'_i$ and $\nu_i + t_i \sim \nu'_i + t'_i$ for all $i \geqslant 0$). We also consider a refinement of region equivalence relation that we call the η-region equivalence relation, and we write $\sim_\eta$, for a given $\eta \in (0, \frac{1}{3})$. Intuitively, $\nu \sim_\eta \nu'$ if both valuations are close or far from any borders of the regions, with respect to the distance η.

Definition 4 (η-regions). *For valuations $\nu, \nu' \in \mathbb{R}_{\geqslant 0}$ we say that $\nu \sim_\eta \nu'$ if $\nu \sim \nu'$ and for every $k \in \{0, \dots, K-1\}$, $|\nu - M_k| \leqslant \eta$ iff $|\nu' - M_k| \leqslant \eta$, and $\nu \geqslant M_K - \eta$ iff $\nu' \geqslant M_K - \eta$. We assume the natural order $\preceq$ over η-regions by their lower bounds. We call η-regions the equivalence classes of $\sim_\eta$. We also extend the relation $\sim_\eta$ to configurations and runs.*

For instance, if $M_1 = 2$ and $M_2 = 3$, the set of η-regions is given by $\{\{0\}, (0, \eta], (\eta, 2{-}\eta), [2{-}\eta, 2), \{2\}, (2, 2{+}\eta], (2{+}\eta, 3{-}\eta), [3{-}\eta, 3), \{3\}, (3, +\infty)\}$. We next introduce the strategies of a restricted shape with the properties that they depend only on the η-region abstraction of runs; their decision is uniform over each η-region; and they play η-close to the borders of the regions.

Definition 5 (η-region uniform strategies). *Let $\eta \in (0, \frac{1}{3})$ be a constant. A strategy $\sigma \in \mathsf{Strat}_1 \cup \mathsf{Strat}_2$ is said to be* η-region-uniform *if*

- *for all finite run $r \sim_\eta r'$ ending respectively in (ℓ, ν) and (ℓ, ν') (in particular $\nu \sim_\eta \nu'$) we have $\nu + \mathsf{del}(\sigma(r)) \sim_\eta \nu' + \mathsf{del}(\sigma(r'))$ and $\mathsf{lab}(\sigma(r)) = \mathsf{lab}(\sigma(r'))$;*
- *for every finite run r ending in (ℓ, ν), if $\nu + \mathsf{del}(\sigma(r)) \in (M_k, M_{k+1})$, we have $\nu + \mathsf{del}(\sigma(r)) \in (M_k, M_k + \eta] \cup [M_{k+1} - \eta, M_{k+1})$.*

We write UStrat_1^η and UStrat_2^η for the set of η-region-uniform strategies for Players 1 and 2. We also define upper-value $\overline{\mathsf{UVal}}^\eta(s)$ when both players are restricted to use only η-region-uniform strategies. Formally,

$$\overline{\mathsf{UVal}}^\eta(s) = \inf_{\sigma_1 \in \mathsf{UStrat}_1^\eta} \sup_{\sigma_2 \in \mathsf{UStrat}_2^\eta} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)), \text{ for all } s \in S.$$

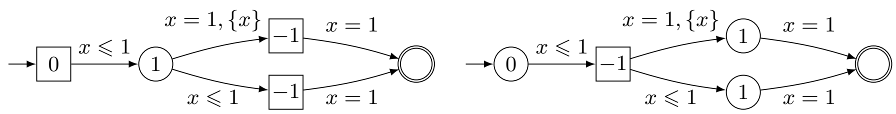


Fig. 4. The value in the left-side one-clock PTG $\mathcal{A}_1$ with price-rates in $\{-1, 0, 1\}$ is $-\frac{1}{2}$, while the value in the right-side PTG $\mathcal{A}_2$ is $\frac{1}{2}$.

Example 1. Consider PTG $\mathcal{A}_1$ shown in Fig. 4 (that is not a 1BPTG since there are three distinct price-rates). A strategy of Player 2 is entirely described by the time spent in the initial location with initial valuation 0. For example, Player 2 can choose to delay 1/2 time units before jumping in the next location. Indeed, the lower and upper value of the game is $-\frac{1}{2}$. However, this strategy is not η-region-uniform. Instead, an η-region-uniform strategy will delay t time units with $t \in [0, \eta] \cup [1-\eta, 1]$. Hence, the upper value when players can only use η-region-uniform strategies is equal to -1.

Contrary to this example, the next lemma shows that, in 1BPTGs, the upper value of the game increases when we restrict ourselves to η-region-uniform strategies. Intuitively, every cost that Player 2 can secure with general strategies, it can also secure it with η-region-uniform strategies against η-region-uniform strategies of Player 1.

Lemma 1. $\overline{\mathsf{Val}}(s) \leqslant \overline{\mathsf{UVal}}^\eta(s)$ *, for every* 1BPTG $\mathcal{A}$*,* $s \in S$ *and* $\eta \in (0, \frac{1}{3})$*,*

5.2 Reduction to η-convergent strategies

A similar result concerning the lower values of the games can be shown in case of η-region-uniform strategies. In subsequent proofs, we need a stronger result to avoid situations detailed in Example 2, where player 2 needs infinite precision to play incrementally closer to borders (as well as an infinite memory). For this reason, we restrict the shape of strategies to force them to play at distance $\frac{\eta}{2^n}$ of borders when playing the nth round of the game. The slight asymmetry in the definitions for the two players is exploited in proving subsequent results.

Definition 6 **(η-convergent strategies).** *Let* $\eta \in (0, \frac{1}{3})$ *be a constant. A strategy* $\sigma \in \mathsf{Strat}_1 \cup \mathsf{Strat}_2$ *is said to be* η-convergent *if* σ *is* η*-region-uniform and for all finite run* r *of length* n *ending in* (ℓ, ν)*:*

- *if* $\sigma \in \mathsf{Strat}_1$*, there exists* k *such that either* $|\nu + \mathsf{del}(\sigma(r)) - M_k| \leqslant \frac{\eta}{2^{n+1}}$*, or* $\mathsf{del}(\sigma(r)) = 0$ *and* $\nu \in (M_k + \frac{\eta}{2^{n+1}}, M_k + \eta]$*;*
- *if* $\sigma \in \mathsf{Strat}_2$*, there exists* k *such that either* $\nu + \mathsf{del}(\sigma(r)) \in \{M_k + \frac{\eta}{2^{n+1}}\} \cup [M_k - \frac{\eta}{2^{n+1}}, M_k)$*, or* $\mathsf{del}(\sigma(r)) = 0$ *and* $\nu \in (M_k + \frac{\eta}{2^{n+1}}, M_k + \eta]$*.*

We let CStrat_1^η *and* CStrat_2^η *be respectively the set of* η*-convergent strategies for Player 1 and Player 2, and we define, for every configuration* $s \in S$*,* $\underline{\mathsf{CVal}}^\eta(s) = \sup_{\sigma_2 \in \mathsf{CStrat}_2^\eta} \inf_{\sigma_1 \in \mathsf{CStrat}_1^\eta} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))$.

Example 2. Consider the 1BPTG $\mathcal{A}_3$ composed of a vertex per player, on top of the target vertex. In its vertex, having price-rate 0, Player 1 must choose between going to the target vertex, or going to the vertex of Player 2 by resetting clock x. In its vertex, having price-rate -1, Player 2 must go back to the vertex of Player 1, with a guard $x > 0$: hence, Player 2 would like to exit as soon as possible, but because of the guard, he must spend some time before exiting. If Player 2 plays according to a finite-memory strategy, there must be a bound ε such that Player 2 always stays in his state for a duration bounded from below by ε, and Player 1 can exploit it by letting the game continue for an arbitrarily long time to achieve an arbitrarily small payoff. On the other hand, if Player 2 plays an infinite-memory η-convergent strategy by staying in his location for a duration $\varepsilon/2^n$ in his n-th visit to its location, Player 2 ensures a payoff $-\varepsilon$ for an arbitrarily small $\varepsilon > 0$, resulting in the value 0 of the game.

It is clear from the previous example that Player 2 needs infinite-memory strategies to optimize his objective. The following lemma formalizes our intuition that the lower value of the game decreases when we restrict ourselves to η-convergent strategies. Intuitively, every cost that Player 1 can secure with general strategies, it can also secure it with η-convergent strategies against an η-convergent strategy of Player 2.

Lemma 2. $\underline{\mathsf{CVal}}^\eta(s) \leqslant \underline{\mathsf{Val}}(s)$ *, for every* 1BPTG $\mathcal{A}$*,* $s \in S$ *and* $\eta \in (0, \frac{1}{3})$*.*

Observe that this lemma fails to hold when location price-rates can take more than two values as exemplified by arena $\mathcal{A}_2$ in Fig. 4. It shows a game with three distinct prices with lower and upper value equal to 1/2. However, when restricted to η-convergent strategies, the lower value equals 1.

Our next goal is to find a common bound being both a lower bound on $\underline{\mathsf{CVal}}^\eta(s)$ and an upper bound on $\overline{\mathsf{UVal}}^\eta(s)$ by studying the value of a reachability-cost game on a finitary abstraction of 1BPTGs.

5.3 Finite abstraction of 1BPTGs

We now construct a finite price game graph $\tilde{\mathcal{A}}$ from any 1BPTG $\mathcal{A}$, as a finite abstraction of the infinite weighted game $[\![\mathcal{A}]\!]$, based on η-regions. Since we have learned that η-region-uniform strategies suffice, we limit ourselves to playing at a distance at most η from the borders of regions. Observe that only η-regions close to the borders are of interest, and moreover η-regions after the maximal constant M_K are not useful since $\mathcal{A}$ is bounded. Let $\mathcal{I}^\eta_\mathcal{A}$ be the set of remaining "useful" η-regions. For example, if constant appearing in the PTG are $M_1 = 2$ and $M_2 = 3$, we have $\mathcal{I}^\eta_\mathcal{A} = \{\{0\}, (0, \eta], [2-\eta, 2), \{2\}, (2, 2+\eta], [3-\eta, 3), \{3\}\}$. We next define the *delay* between two such η-regions $I \preceq J$, denoted by $d(I, J)$, as the closest integer of $q' - q$, where q (respectively, q') is the lower bound of interval I (respectively, J). For example, $d((2, 2+\eta], [3-\eta, 3)) = 1$ and $d(\{0\}, [2-\eta, 2)) = 2$.

Definition 7. *For every* 1BPTG $\mathcal{A}$ *we define its border abstraction as a finite priced game graph* $\tilde{\mathcal{A}} = (V = V_1 \uplus V_2, A, E, \pi, V_f)$ *where:*

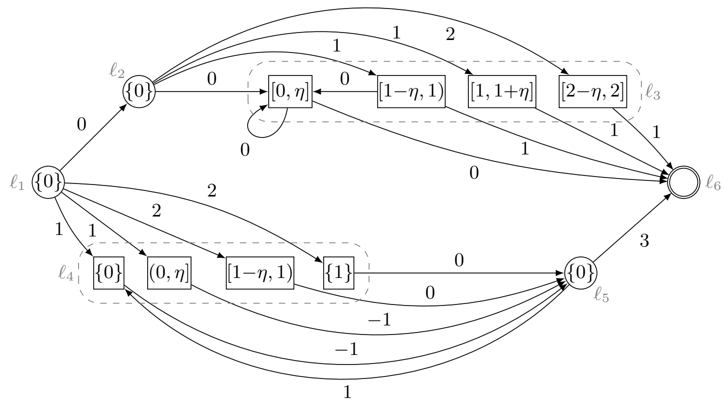


Fig. 5. Finite weighted game associated with the 1BPTG of Fig. 1.

- $V_i = \{(\ell, I) \mid \ell \in L_i, I \in \mathcal{I}^{\eta}_{\mathcal{A}}, I \subseteq \mathrm{Inv}(\ell)\}$ *for* $i \in \{1,2\}$*;*
- $A = \mathcal{I}^{\eta}_{\mathcal{A}} \times \Sigma$*;*
- E *is the set of tuples* $((\ell, I),(J,a),(\ell', J'))$ *such that* $I \preceq J$ *and for all* $I \preceq K \preceq J$ *we have* $K \subseteq \mathrm{Inv}(\ell)$ *and* $J \subseteq \zeta$ *and* $J' = J[R := 0]$ *with* $(\zeta, R, \ell') = \delta(\ell, a)\}$*;*
- $\pi((\ell, I),(J,a),(\ell', J')) = \omega(\ell) \times d(I,J) + \omega(a)$*; and*
- $V_f = \{(\ell, I) \mid \ell \in L_f, I \in \mathcal{I}^{\eta}_{\mathcal{A}}\}$.

In a border abstraction game $\tilde{\mathcal{A}}$, the meaning of action (J,a) is that the player wants to let time elapse until it reaches the η-region J, then playing label a. It simulates any timed move (t,a) with t any delay reaching a point in J.

Example 3. Consider the border abstraction of the 1BPTG of Fig. 1 shown in Fig. 5. Observe that we depict only a succinct representation of the real abstraction, since we only show the reachable part of the game from $(\ell_1, 0)$, and we have removed multiple edges (introduced due to label hiding) and kept only the most useful ones for the corresponding player. For example, consider the location $(\ell_5, \{0\})$. There are edges labelled by (J,a) for every interval $J \in \mathcal{I}^{\eta}_{\mathcal{A}}$, all directed to $(\ell_4, \{0\})$ due to a reset being performed there. We only show the best possible edge—the one with lowest price—since location ℓ_5 belongs to Player 1, who seeks to minimise cost. Each vertex contains the η-region it represents. Thanks to Theorem 1, it is possible to compute the optimal value as well as optimal strategies for both players. Here, the value of state $(\ell_1, 0)$ is 1, and an optimal strategy for Player 1 is to follow action $(\{0\}, a)$ (i.e., jump to ℓ_2 immediately), and then action $(\{1\}, a)$ (i.e., to delay 1 time unit, before jumping in ℓ_3).

Lemma 3. *Let* $\mathcal{A}$ *be a* 1BPTG *and* $\tilde{\mathcal{A}}$ *be its border abstraction. Suppose that for all* $0 \leqslant k \leqslant K$ *and* $\ell \in L$ *we have that* $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$ *is finite. Then, for all* $\varepsilon > 0$*, there is* $\eta > 0$ *s.t.* $\overline{\mathsf{UVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) - \varepsilon \leqslant \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) \leqslant \underline{\mathsf{CVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) + \varepsilon$.

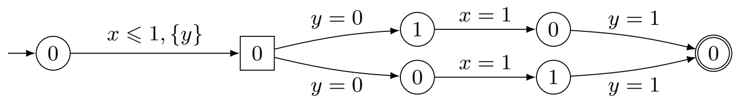


Fig. 6. A two-clock PTG with prices of locations in $\{0, +1\}$ and value 1/2

Combining this result with Theorem 1 we obtain the following.

Corollary 1. 1BPTG*s are determined and we can compute their values in pseudo-polynomial time. Moreover, in case the values are finite, ε-optimal strategies exist for both players: Player 2 may require infinite memory strategies, whereas finite memory is sufficient for Player 1. Finally, ε-optimal strategies can also be computed in pseudo-polynomial time.*

Proof. In case of infinite values $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$, we can show directly that $\overline{\mathsf{Val}}_{\mathcal{A}}((\ell, M_k)) = \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) = \underline{\mathsf{Val}}_{\mathcal{A}}((\ell, M_k))$. Otherwise, let $\varepsilon > 0$. By Lemma 3, we know that there exists $\eta > 0$ such that for every location $\ell \in L$ and integer $0 \leqslant k \leqslant K$:

$$\overline{\mathsf{UVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) - \varepsilon \leqslant \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) \leqslant \underline{\mathsf{CVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) + \varepsilon\,.$$

Moreover Lemma 1 and 2 show that:

$$\underline{\mathsf{CVal}}^{\eta}((\ell, M_k)) \leqslant \underline{\mathsf{Val}}((\ell, M_k)) \leqslant \overline{\mathsf{Val}}((\ell, M_k)) \leqslant \overline{\mathsf{UVal}}^{\eta}((\ell, M_k))\,.$$

Both inequalities combined permit to obtain

$$\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) - \varepsilon \leqslant \underline{\mathsf{Val}}((\ell, M_k)) \leqslant \overline{\mathsf{Val}}((\ell, M_k)) \leqslant \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) + \varepsilon\,.$$

Taking the limit when ε tends to 0, we obtain that $\underline{\mathsf{Val}}((\ell, M_k)) = \overline{\mathsf{Val}}((\ell, M_k)) = \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$. Therefore, 1BPTG are determined. Moreover, in case of finite values, the proof of Lemma 3 permits to construct ε-optimal η-region-uniform strategies σ_1^* (with finite memory) and σ_2^* (which is moreover η-convergent). $\square$

In the case of 1BPTGs, the finite values are integers. This property fails if we allow more than one clock, as shows Fig. 6 with a two-clock PTG with price-rates in $\{0, 1\}$ and optimal value $\frac{1}{2}$. It also fails if we allow more than two price-rates as was shown in Fig. 4. However for 1PTG$(0, 1)$ with prices of labels in $\mathbb{N}$, the value of the game is necessarily nonnegative disallowing the case $-\infty$. The case $+\infty$ can be detected in polynomial time. If the value is not $+\infty$, the exact computation in the finite abstraction $\tilde{\mathcal{A}}$ can be performed in polynomial time (see [14] or [15]), resulting in a polynomial algorithm for PTGs. The sketch of Theorem 3 is now complete. Contrary to what it has been said in [13], our results do not allow to ensure that the number of cutpoints is polynomial in this subclass. It can indeed be exponential as shown in [16].

6 Conclusion

We revisited games with reachability objective on PTGs with both positive and negative price-rates. We showed undecidability of all classes of constrained-price reachability objectives with two or more clocks. We also observed that adding bounded-time restriction does not recover decidability, even with nonnegative prices. We also partially answer the question regarding polynomial-time algorithm for one-clock PTGs by showing that for a bi-valued variant the problem is in pseudo-polynomial time. However, the existence of a polynomial-time algorithm for multi-priced one-clock PTGs with nonnegative price-rates, and the existence of algorithm for computing ε-optimal strategies for PTGs with arbitrary number of clocks remain open problems.

References

1. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *ICALP'04*, *LNCS* 3142, pages 122–133. Springer, 2004.
2. R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.
3. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. *TCS*, 318(3):297–322, 2004.
4. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, J. Romijn, and F. W. Vaandrager. Minimum-cost reachability for priced timed automata. In *HSCC'01*, *LNCS* 2034, pages 147–161. Springer, 2001.
5. J. Berendsen, T. Chen, and D. Jansen. Undecidability of cost-bounded reachability in priced probabilistic timed automata. In *Theory and Applications of Models of Computation*, *LNCS* 5532, pages 128–137. Springer, 2009.
6. P. Bouyer, T. Brihaye, V. Bruyère, and J.-F. Raskin. On the optimal reachability problem of weighted timed automata. *FMSD*, 31(2):135–175, 2007.
7. P. Bouyer, T. Brihaye, and N. Markey. Improved undecidability results on weighted timed automata. *IPL*, 98(5):188–194, 2006.
8. P. Bouyer, E. Brinksma, and K. G. Larsen. Optimal infinite scheduling for multi-priced timed automata. *FMSD*, 32(1):3–23, 2008.
9. P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal strategies in priced timed game automata. In *FSTTCS'04*, *LNCS* 3328, pages 148–160. Springer, 2004.
10. P. Bouyer, K. G. Larsen, N. Markey, and J. I. Rasmussen. Almost optimal strategies in one-clock priced timed games. In *FSTTCS'06*, *LNCS* 4337, pages 345–356. Springer, 2006.
11. T. Brihaye, V. Bruyère, and J.-F. Raskin. On optimal timed strategies. In *FORMATS'05*, *LNCS* 3829, pages 49–64. Springer, 2005.
12. T. Brihaye, L. Doyen, G. Geeraerts, J. Ouaknine, J.-F. Raskin, and J. Worrell. Time-bounded reachability for monotonic hybrid automata: Complexity and fixed points. In *ATVA'13*, *LNCS* 8172, pages 55–70. Springer, 2013.
13. T. Brihaye, G. Geeraerts, S. N. Krishna, L. Manasa, B. Monmege and A. Trivedi Adding Negative Prices to Priced Timed Games In *CONCUR'14*, *LNCS* 8704, pages 560–575, Springer, 2014.
14. T. Brihaye, G. Geeraerts, and B. Monmege. Reachability-cost games with negative weights. Technical report, `http://www.ulb.ac.be/di/verif/monmege/download/priced-games.pdf`, 2014.

15. T. D. Hansen, R. Ibsen-Jensen, and P. B. Miltersen. A faster algorithm for solving one-clock priced timed games. In *CONCUR'13*, *LNCS* 8052, pages 531–545. Springer, 2013.
16. J. Fearnley, R. Ibsen-Jensen and R. Savani. One-Clock Priced Timed Games are PSPACE-hard CoRR abs/2001.04458, `https://arxiv.org/abs/2001.04458`, 2020.
17. F. Laroussinie, N. Markey, and P. Schnoebelen. Model checking timed automata with one or two clocks. In *CONCUR'04*, pages 387–401, 2004.
18. M. L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.
19. J. Ouaknine and J. Worrell. Towards a theory of time-bounded verification. In *Automata, Languages and Programming*, *LNCS* 6199, pages 22–37. Springer, 2010.
20. P. J. Ramadge and W. M. Wonham. The control of discrete event systems. In *IEEE*, volume 77, pages 81–98, 1989.
21. M. Rutkowski. Two-player reachability-price games on single-clock timed automata. In *QAPL'11*, volume 57 of *EPTCS*, pages 31–46, 2011.
22. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *TCS*, 158:343–359, 1996.

A Motivation: A case-study from Project Cassting

Our work is partly motivated by possible applications to real case study, like energy-aware houses, taken from the EU FP7 project Cassting. It consists in houses equiped with solar panels and energy storage capacities (e.g., with a water tank), that can produce some energy, use or store it, and possibly sell it on a local grid, for use by other houses. Priced timed games may permit to model such house, and strategy synthesis allows us to build optimal controller for the house, aiming at energy and cost savings: player 1 models the house and wants to minimize its cost, whereas player 2 models other houses and the environment, modelling the worst possible situation. Pricing policy implies that selling energy on the grid is more profitable during the day than the night, whereas it is more profitable to use energy from the grid during the night than the day. Moreover, weather conditions (sun or clouds, e.g.) imply that the solar panels are not constantly producing energy. Hence, we can model the situation with three independant phases: sunny day, cloudy day and night. In each of these phases, we can suppose that only two rates are indeed available. During a sunny day, selling energy will be rewarded α euros per time unit (assuming that the energy production or consumption is constant), whereas consumption or storing of energy does not cost anything. During a cloudy day, solar panels are off: because of storage capacities, it is however possible to sell energy at the same cost than previously, but the consumption may now cost α euros per time unit. Finally, during the night, selling energy rewards β euros per time unit, whereas consuming costs β euros per time unit. Notice that in each of the phases, we can use bi-valued priced timed games to model the arena.

B Detailed decidability proofs

B.1 Proof of Lemma 1

With respect to rechability of the target locations, (η-)equivalent plays are indistinguishable. Moreover, with respect to weights, we show that plays may avoid regions $(M_k + \eta, M_{k+1} + \eta)$ without loss of generality. Formally, a play $r = (\ell_0, \nu_0), (t_0, a_0), \dots$ is said to *stay η-close to borders* if for all $i \geqslant 0$, there exists k such that $|\nu_i + t_i - M_k| \leqslant \eta$. Before proving Lemma 1, we first study more precisely the relationship between general plays and plays that stay η-close to borders. More precisely, we now explain how to construct, from any finite play r, a play r^+ such that (i) r^+ stays η-close to borders; (ii) r and r^+ are region-equivalent; and (iii) $\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$. Intuitively, the idea is to consider the steps of play r and to shift them towards one of the closest borders, unless the current step is already η-close to a border: when the current location has price p^+, since we want the weight of r^+ to be greater than or equal to the weight of r, we will spend more time in this location, hence shifting the step 'to the right', and symetrically in case of a location of price p^-. Moreover, the construction will trivially verify that if r' and r are two plays that coincide on their prefix of length n, i.e., $r'[n] = r[n]$, then $(r')^+[n] = r^+[n]$.

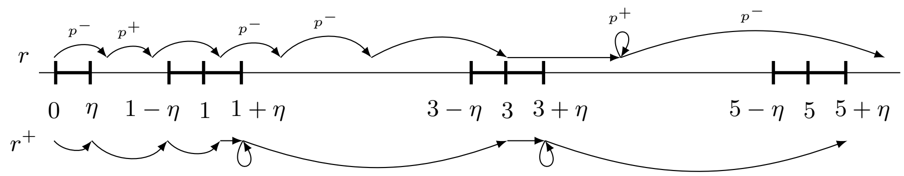


Fig. 7. A play r and its associated play r^+. The price of the location in play r, when it matters, is denoted on the transition exiting this location: for instance the first location of r has price p^-, whereas the second has price p^+. On the first transition, the second rule of the definition applies, and less time is spent in the location of price p^-. On the second transition, the first rule applies, and more time is spent in the location of price p^+. On the third transition, the first rule applies and both plays synchronize. The transition of time duration $t = 0$ (denoted as a loop) in r is supposed to be taken in a location owned by player 1, of price p^+. In particular, in r^+, it is taken when the valuation is $3 + \eta$, which implies a time duration $t^+ = 0$ as prescribed in the second case of the last rule of the definition.

The construction is by induction on the length of the play. Henceforth, let $r = (\ell_0, \nu_0), (t_0, a_0), \ldots, (\ell_{i+1}, \nu_{i+1})$ and suppose that ν_0 is η-close to some M_k. We construct a play $r^+ = (\ell_0^+, \nu_0^+), (t_0^+, a_0^+), \ldots, (\ell_{i+1}^+, \nu_{i+1}^+)$. In case $i + 1 = 0$, we simply let $\ell_0^+ = \ell_0$ and $\nu_0^+ = \nu_0$. Otherwise, we consider the play r^+ constructed by induction up to its configuration of index i, and we now explain how to construct the next timed action (t_i^+, a_i^+). First, we let $a_i^+ = a_i$. Then, we distinguish between three cases:

- if there exists k such that $|\nu_i + t_i - M_k| \leqslant \eta$, then we let $t_i^+ = \nu_i + t_i - \nu_i^+$;
- if there exists k such that $\nu_i + t_i \in (M_k + \eta, M_{k+1} - \eta)$ and $\omega(\ell_i) = p^-$, then we let $t_i^+ = \max(M_k + \eta - \nu_i^+, 0)$. Indeed, in this case, we want to spend as little time as possible in the location in order to ensure $\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$;
- if there exists k such that $\nu_i + t_i \in (M_k + \eta, M_{k+1} - \eta)$ and $\omega(\ell_i) = p^+$, then there are two cases. In case $\ell_i \in L_2$ or $t_i > 0$ or $\nu_i^+ \neq M_k + \eta$, we let $t_i^+ = M_{k+1} - \eta - \nu_i^+$. The intution is that we try to spend as much time as possible in the location to enforce $\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$. Otherwise, i.e., if $\ell_i \in L_1$ and $t_i = 0$ and $\nu_i^+ = M_k + \eta$, then we let $t_i^+ = 0$. Because $t_i = 0 = t_i^+$, we will still ensure that $\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$ in that case.

Notice that this definition is purely syntactic, and we will show later that all t_i^+'s are non-negative, ensuring that we are indeed constructing a play of the game. We then let $\ell_{i+1}^+ = \ell_{i+1}$, and valuation ν_{i+1}^+ is defined according to the semantics of the game.

An example of construction of r^+ is given in Fig. 7, for a play r without reset. We have supposed that the sequence of borders is $0, 1, 3$ and 5.

Lemma 4. *Let r be a play that starts from some state η-close to a border. Then, the play r^+ verifies the following properties:*

1. *if for some j, there exists k such that $|\nu_j + t_j - M_k| \leqslant \eta$, then $\nu_j^+ + t_j^+ = \nu_j + t_j$;*

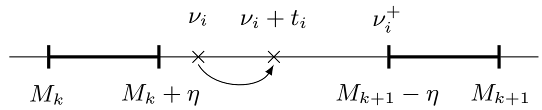


Fig. 8. Illustration for the proof of Lemma 4-2

2. *if for some j, there exists k such that $\nu_j + t_j \in (M_k + \eta, M_{k+1} - \eta)$, then $\nu_j^+ + t_j^+ \in \{M_k + \eta, M_{k+1} - \eta\}$;*
3. *r^+ is a play that stays η-close to borders;*
4. *$r^+ \sim r$;*
5. *$\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$.*

In the rest of this section, we call the five points of this Lemma 'property 1', 'property 2', and so forth.

Remark 1. Before starting the proof, notice that in case $\delta(\ell_j, a_j) = (\zeta, \emptyset, \ell_{j+1})$, i.e., the clock is not reset at step j, we have $\nu_{j+1} = \nu_j + t_j$ (and we will also have $\nu_{j+1}^+ = \nu_j^+ + t_j^+$). In particular, in that case, property 1 implies that if $|\nu_{j+1} - M_k| \leqslant \eta$, then $\nu_{j+1}^+ = \nu_{j+1}$, whereas property 2 implies that if $\nu_{j+1} \in (M_k + \eta, M_{k+1} - \eta)$, then $\nu_{j+1}^+ \in \{M_k + \eta, M_{k+1} - \eta\}$.

Proof. All properties are shown by a simultaneous induction on the length of the play. All properties are clearly true in case of a play r reduced to a single state (which is assumed to be η-close to a border). We now consider a play $r = (\ell_0, \nu_0), (t_0, a_0), \ldots, (\ell_{i+1}, \nu_{i+1})$ with $i + 1 \geqslant 1$, and prove the properties for the play r^+ whose construction has been given before.

1. The first property is true directly by construction.
2. For the second property, by induction hypothesis, it is sufficient to prove that if $\nu_i + t_i \in (M_k + \eta, M_{k+1} - \eta)$, then $\nu_i^+ + t_i^+ \in \{M_k + \eta, M_{k+1} - \eta\}$. Suppose first that $\omega(\ell_i) = p^-$: then, $t_i^+ = \max(M_k + \eta - \nu_i^+, 0)$. In case $t_i^+ > 0$, we have $t_i^+ = M_k + \eta - \nu_i^+$, hence, $\nu_i^+ + t_i^+ = M_k + \eta$. Otherwise, we have $t_i^+ = 0$ so that $\nu_i^+ + t_i^+ = \nu_i^+$ and $\nu_i^+ > M_k + \eta$: we have depicted an example of the situation in Fig. 8. Since ν_i^+ must be η-close to a border, we have $\nu_i^+ \geqslant M_{k+1} - \eta$. Then, $\nu_i^+ > \nu_i + t_i \geqslant \nu_i$. But, ν_i^+ and ν_i are in the same region by induction hypothesis of property 4 (no reset has been performed in the previous transition since $\nu_i \neq \nu_i^+$, so that they are respectively equal to $\nu_{i-1}^+ + t_{i-1}^+$ and $\nu_{i-1} + t_{i-1}$). This implies that $\nu_i^+ \in [M_{k+1} - \eta, M_{k+1})$ and $\nu_i \in (M_k, M_{k+1} - \eta)$. However, by induction hypothesis, if $\nu_i \in (M_k, M_k + \eta]$, then $\nu_i^+ = \nu_i$ which is forbidden in this case. Hence, we have $\nu_i \in (M_k + \eta, M_{k+1} - \eta)$. Hence, by induction hypothesis again, we have $\nu_i^+ \in \{M_k + \eta, M_{k+1} - \eta\}$, which finally implies that $\nu_i^+ = M_{k+1} - \eta$.
   Suppose then that $\omega(\ell_i) = p^+$. The result is again immediate in case $t_i^+ = M_{k+1} - \eta - \nu_i^+$. Otherwise, we have $t_i^+ = 0$, $\ell_i \in L_1$, $t_i = 0$ and $\nu_i^+ = M_k + \eta$. We directly obtain $\nu_i^+ + t_i^+ = M_k + \eta$ which permits to conclude.

3. We now prove that r^+ is indeed a play. The only non-trivial property is that t_i^+ is a non negative delay, especially when $t_i^+ = \nu_i + t_i - \nu_i^+$ or $t_i^+ = M_{k+1} - \eta - \nu_i^+$.
   Consider first the case $t_i^+ = \nu_i + t_i - \nu_i^+$. It has to be shown that if there exists k such that $|\nu_i + t_i - M_k| \leqslant \eta$, then ν_i^+ is not greater than $\nu_i + t_i$. Notice first that $\nu_i \leqslant \nu_i + t_i$. Hence, if ν_i is η-close to a border, then by property 1, $\nu_i^+ = \nu_i \leqslant \nu_i + t_i$. Otherwise, we know that $\nu_i \notin [M_k - \eta, M_k + \eta]$, hence, either $\nu_i < M_k - \eta$, or $\nu_i > M_k + \eta$. Since $\nu_i \leqslant \nu_i + t_i \leqslant M_k + \eta$, we can rule out the case $\nu_i > M_k + \eta$, and conclude that $\nu_i < M_k - \eta$. By property 2 (applied to $\nu_i^+ = \nu_{i-1}^+ + t_{i-1}^+$, since no reset has just been performed, knowing that ν_i is not η-close to 0), this implies that $\nu_i^+ \leqslant M_k - \eta \leqslant \nu_i + t_i$.
   Consider then the case $t_i^+ = M_{k+1} - \eta - \nu_i^+$, which holds when $\nu_i + t_i \in (M_k + \eta, M_{k+1} - \eta)$ and $\omega(\ell_i) = p^+$. Then, $\nu_i \leqslant \nu_i + t_i < M_{k+1} - \eta$. Hence, either ν_i is η-close to a border (in particular when the clock has just been reset), in which case, by property 1, we have $\nu_i^+ = \nu_i < M_{k+1} - \eta$, so that $t_i^+ = M_{k+1} - \eta - \nu_i^+ > 0$. Or $\nu_i \in (M_{k'} + \eta, M_{k'+1} - \eta)$ with $k' \leqslant k$. By property 2, this implies that $\nu_i^+ \leqslant M_{k'+1} - \eta \leqslant M_{k+1} - \eta$, once again implying that $t_i^+ \geqslant 0$.
   The fact that r^+ stays η-close to the borders is directly implied by properties 1 and 2.
4. Property $r^+ \sim r$ is also a direct consequence of properties 1 and 2.
5. It only remains to prove that $\mathsf{Cost}(r) \leqslant \mathsf{Cost}(r^+)$. Notice that the weights $\mathsf{Cost}(r)$ and $\mathsf{Cost}(r^+)$ can be decomposed as sums of weights of subplays that start and end in the same configuration, but with intermediate configurations that do no match. Hence, it is sufficient to prove the inequality for subplays r and r^+ that start in the same configuration (at step j, $\nu_j + t_j = \nu_j^+ + t_j^+$) and that do not contain other identical configurations, unless possibly the last one. For the sake of simplicity, we suppose in the following that $j = 0$. In particular, we may suppose that r does not contain reset transitions or positions j such that ν_j is η-close to borders, except possibly the very last one (otherwise, the two plays would again contain matching configurations). Since there are no resets, we have $\nu_j = \nu_{j-1} + t_{j-1}$ and $\nu_j^+ = \nu_{j-1}^+ + t_{j-1}^+$ for every $j > 0$.
   We now consider separately the possibility of sets $\{p^-, p^+\}$.
   - As a first case, consider that $p^- = -1$ and $p^+ = +1$. We prove by induction over $0 \geqslant j \geqslant i$ that

     $$\mathsf{Cost}(r^+[j+1]) \geqslant \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu_{j+1}^+| \,.$$

     For the sake of brevity, we omit the weights of the actions in this proof, but notice that the same actions occur in r and r^+ since these two plays are equivalent (by property 4).
     - **Base case**. If $j = 0$, then we have supposed that ν_0 is η-close to a border, so that $\nu_0^+ = \nu_0$. Let k be such that[6] $\nu_1 \in (M_k + \eta, M_{k+1} - \eta)$.

[6] Remember that we suppose that there is no synchronization for now, so that no valuation in r is η-close to a border.

Then, if $\omega(\ell_0) = -1$, $t_0^+ = \max(M_k + \eta - \nu_0, 0)$. However, $\nu_0 \leqslant \nu_1$ and ν_0 is η-close to a border, so that $\nu_0 \leqslant M_k + \eta$. Hence, $t_0^+ = M_k + \eta - \nu_0$, which implies $\nu_1^+ = \nu_0 + t_0^+ = M_k + \eta$. Hence,

$$\begin{aligned}\mathsf{Cost}(r^+[1]) = -(\nu_1^+ - \nu_0) &= \mathsf{Cost}(r[1]) + \nu_1 - \nu_1^+ \\ &= \mathsf{Cost}(r[1]) + |\nu_1 - \nu_1^+|\,.\end{aligned}$$

Consider then the case where $\omega(\ell_0) = +1$. Notice that we supposed that ν_1 is not η-close to a border, contrary to ν_0, so that $t_0 > 0$. Hence, we are sure that $t_0^+ = M_{k+1} - \eta - \nu_0$. This implies that $\nu_1^+ = \nu_0^+ + t_0^+ = \nu_0 + t_0^+ = M_{k+1} - \eta > \nu_1$ and

$$\begin{aligned}\mathsf{Cost}(r^+[1]) = \nu_1^+ - \nu_0 &= \mathsf{Cost}(r[1]) + \nu_1^+ - \nu_1 \\ &= \mathsf{Cost}(r[1]) + |\nu_1 - \nu_1^+|\,.\end{aligned}$$

- **Inductive case**. Let us suppose that the property is proved for all indices less than or equal to j, and prove it for $j+1$. We let k be such that $\nu_{j+1} = \nu_j + t_j \in (M_k + \eta, M_{k+1} - \eta)$. We will distinguish four possible cases depending on $\omega(\ell_j)$ and the relative order between ν_{j+1}^+ and ν_{j+1}.

  (a) We first suppose that $\omega(\ell_j) = +1$. Then,

  $$\begin{aligned}\mathsf{Cost}(r^+[j+1]) &= \mathsf{Cost}(r^+[j]) + \nu_{j+1}^+ - \nu_j^+ \\ &\geqslant \mathsf{Cost}(r[j]) + |\nu_j - \nu_j^+| + \nu_{j+1}^+ - \nu_j^+ \quad \text{(Ind. Hyp.)}\end{aligned}$$

  Hence, since $\mathsf{Cost}(r[j+1]) = \mathsf{Cost}(r[j]) + (\nu_{j+1} - \nu_j)$:

  $$\begin{aligned}\mathsf{Cost}(r^+[j+1]) \geqslant{}& \mathsf{Cost}(r[j+1]) - (\nu_{j+1} - \nu_j) \\ &+ |\nu_j - \nu_j^+| + \nu_{j+1}^+ - \nu_j^+\,. \qquad (1)\end{aligned}$$

  i. In the case where $\nu_{j+1}^+ > \nu_{j+1}$, we have $|\nu_{j+1} - \nu_{j+1}^+| = \nu_{j+1}^+ - \nu_{j+1}$ so that (1) can be rewritten

  $$\begin{aligned}\mathsf{Cost}(r^+[j+1]) \geqslant{}& \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu_{j+1}^+| \\ &+ (\nu_j - \nu_j^+) + |\nu_j - \nu_j^+|\end{aligned}$$

  which is greater than or equal to $\mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu_{j+1}^+|$ since $\nu_j - \nu_j^+ \geqslant -|\nu_j - \nu_j^+|$.

  ii. Similarly, in the case where $\nu_{j+1}^+ < \nu_{j+1}$, we have $|\nu_{j+1} - \nu_{j+1}^+| = \nu_{j+1} - \nu_{j+1}^+$. Notice that this necessarily implies that $\ell_i \in L_1$ and $t_j = 0$ and $\nu_j^+ = M_k + \eta$: otherwise, we would have $t_j^+ = M_{k+1} - \eta - \nu_j^+$ and thus $\nu_{j+1}^+ = M_{k+1} - \eta > \nu_{j+1}$ that contradicts the hypothesis. In particular, we have $t_j^+ = 0$ and $\nu_j^+ = \nu_{j+1}^+ < \nu_{j+1} = \nu_j$, so that (1) becomes

  $$\begin{aligned}\mathsf{Cost}(r^+[j+1]) &\geqslant \mathsf{Cost}(r[j+1]) + |\nu_j - \nu_j^+| \\ &= \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu_{j+1}^+|\,.\end{aligned}$$

(b) Suppose then that $\omega(\ell_j) = -1$. Then, a similar calculation gives

$$\begin{aligned}\mathsf{Cost}(r^+[j+1]) &= \mathsf{Cost}(r^+[j]) - (\nu^+_{j+1} - \nu^+_j)\\ &\geqslant \mathsf{Cost}(r[j]) + |\nu_j - \nu^+_j| - (\nu^+_{j+1} - \nu^+_j) \quad \text{(Ind. Hyp.)}\end{aligned}$$

Hence, since $\mathsf{Cost}(r[j+1]) = \mathsf{Cost}(r[j]) - (\nu_{j+1} - \nu_j)$:

$$\begin{aligned}\mathsf{Cost}(r^+[j+1]) \geqslant \mathsf{Cost}(r[j+1]) &+ (\nu_{j+1} - \nu_j)\\ &+ |\nu_j - \nu^+_j| - (\nu^+_{j+1} - \nu^+_j)\,. \end{aligned} \tag{2}$$

i. Once again, if $\nu^+_{j+1} < \nu_{j+1}$, we have $|\nu_{j+1} - \nu^+_{j+1}| = \nu_{j+1} - \nu^+_{j+1}$ so that (2) becomes

$$\begin{aligned}\mathsf{Cost}(r^+[j+1]) \geqslant \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu^+_{j+1}|\\ - (\nu_j - \nu^+_j) + |\nu_j - \nu^+_j|\end{aligned}$$

which is greater than or equal to $\mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^+_{j+1}|$ since $\nu_j - \nu^+_j \leqslant |\nu_j - \nu^+_j|$.

ii. Similarly, if $\nu^+_{j+1} > \nu_{j+1}$, we know by property 2 that $\nu^+_{j+1} = M_{k+1} - \eta$. In particular, since $t^+_j = \max(M_k + \eta - \nu^+_j, 0)$ and $\nu^+_{j+1} = \nu^+_j + t^+_j \neq M_k + \eta$, we know that $t^+_j = 0$, and $\nu^+_j \geqslant M_k + \eta$. This implies $\nu^+_{j+1} = \nu^+_j = M_{k+1} - \eta > \nu_{j+1} \geqslant \nu_j$. Knowing that $|\nu_{j+1} - \nu^+_{j+1}| = \nu^+_{j+1} - \nu_{j+1}$, we obtain from (2)

$$\begin{aligned}\mathsf{Cost}(r^+[j+1]) &\geqslant \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu^+_{j+1}|\\ &\qquad + 2(\nu_{j+1} - \nu^+_{j+1}) - 2(\nu_j - \nu^+_j)\\ &= \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu^+_{j+1}| + 2(\nu_{j+1} - \nu_j)\\ &\geqslant \mathsf{Cost}(r[j+1]) + |\nu_{j+1} - \nu^+_{j+1}|\,.\end{aligned}$$

We finally have proved the property by induction. Notice in particular that this shows that $\mathsf{Cost}(r^+[j+1]) \geqslant \mathsf{Cost}(r[j+1])$ for every j with $\nu^+_{j+1} \neq \nu_{j+1}$. To conclude the proof of $\mathsf{Cost}(r^+) \geqslant \mathsf{Cost}(r)$, it remains to deal with the case of a possible last transition ending with $\nu^+_{i+1} = \nu_{i+1}$. Unless $i = 0$, in which case we have $\mathsf{Cost}(r^+) = \mathsf{Cost}(r)$, we know by hypothesis that $\nu^+_i \neq \nu_i$. By the previous property, we have $\mathsf{Cost}(r^+[i]) \geqslant \mathsf{Cost}(r[i]) + |\nu_i - \nu^+_i|$. Moreover, $\mathsf{Cost}(r^+) = \mathsf{Cost}(r^+[i]) + \omega(\ell_i)(\nu^+_{i+1} - \nu^+_i)$ and $\mathsf{Cost}(r) = \mathsf{Cost}(r[i]) + \omega(\ell_i)(\nu_{i+1} - \nu_i)$. In the overall (using the fact that $\nu^+_{i+1} = \nu_{i+1}$), we get

$$\mathsf{Cost}(r^+) \geqslant \mathsf{Cost}(r) + \omega(\ell_i)(\nu_i - \nu^+_i) + |\nu_i - \nu^+_i|\,.$$

In all cases, we verify that $-\omega(\ell_i)(\nu_i - \nu^+_i) \leqslant |\nu_i - \nu^+_i|$, so that we have proved that $\mathsf{Cost}(r^+) \geqslant \mathsf{Cost}(r)$.

- We now consider the case where $p^- = 0$ and $p^+ = +1$ (the case $p^- = -1$ and $p^+ = 0$ is very similar, and not explained in details here). We prove another inequality by induction over $0 \geqslant j \geqslant i$, namely that

$$\mathsf{Cost}(r^+[j+1]) \geqslant \mathsf{Cost}(r[j+1]) + \max(\nu_{j+1} - \nu^+_{j+1}, 0)\,.$$

The proof is very similar to the previous case, and we conclude as previously. $\square$

We now go to the proof of Lemma 1. In case, $\overline{\mathsf{Val}}(s) = -\infty$ the Lemma is trivially true. We first consider the case $\overline{\mathsf{Val}}(s) < +\infty$. Let $\sigma'_1 \in \mathsf{UStrat}^\eta_1$. We now explain how to construct a strategy $\sigma_1 \in \mathsf{Strat}_1$ such that for all states s

$$\sup_{\sigma'_2 \in \mathsf{UStrat}^\eta_2} \mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \geqslant \sup_{\sigma_2 \in \mathsf{Strat}_2} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))\,.$$

To prove such an inequality, we will consider any strategy $\sigma_2 \in \mathsf{Strat}_2$ and construct a strategy $\sigma'_2 \in \mathsf{UStrat}^\eta_2$ such that

$$\mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \geqslant \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))\,.$$

Strategy σ_1 follows σ'_1 in case of plays staying η-close to borders. We must however extend it to deal with the other plays faithfully. Let $r = (\ell_0, \nu_0), (t_0, a_0), \ldots, (\ell_i, \nu_i)$ be any finite play ending in a location ℓ_i of player 1, and $r^+ = (\ell^+_0, \nu^+_0), (t^+_0, a^+_0), \ldots, (\ell^+_i, \nu^+_i)$ the play constructed as before. By Lemma 4, we know that r^+ is a play that stays η-close to borders. Hence, $\sigma'_1(r^+) = (t'_i, a)$, for some $t'_i \in \mathbb{R}_{\geqslant 0}$, with $\nu^+_i + t'_i$ being η-close to a border. We let $t_i = \max(\nu^+_i + t'_i - \nu_i, 0)$ and $\sigma_1(r) = (t_i, a)$. Let $\tilde{r}$ (respectively, r') be the play r (respectively, r^+) extended with the step prescribed by σ_1 (respectively, σ'_1). Then, we prove that r' matches the construction above starting from the run $\tilde{r}$, i.e., $\tilde{r}^+ = r'$. By construction, we only have to verify that the value of t'_i is consistent with the previous constructions, i.e., $t'_i = t^+_i$.

Lemma 5. *We have $t'_i = t^+_i$.*

Proof. In case $\nu_i \leqslant \nu^+_i + t'_i$, since $t_i = \max(\nu^+_i + t'_i - \nu_i, 0)$, we have $\nu_i + t_i = \nu^+_i + t'_i$ which is η-close to borders, and $t'_i = \nu_i + t_i - \nu^+_i$ that fits with the definition of t^+_i in $\tilde{r}^+$: hence $t'_i = t^+_i$ in that case.

Otherwise, we have $\nu_i > \nu^+_i + t'_i$ and $t_i = 0$. In particular, $\nu_i > \nu^+_i$ so that ν_i cannot be η-close to a border (by Lemma 4-1). By Lemma 4-2, since $\nu^+_i < \nu_i$, there exists k such that $\nu^+_i = M_k + \eta$ and $\nu_i \in (M_k + \eta, M_{k+1} - \eta)$. We are thus in the situation $\ell_i \in L_1$ and $t_i = 0$ and $\nu^+_i = M_k + \eta$ that prescribes a choice of the next time delay $t^+_i = 0$. Hence, we must prove that $t'_i = 0$. It is necessarily the case, since $M_k + \eta = \nu^+_i \leqslant \nu^+_i + t'_i < \nu_i$ with $\nu^+_i + t'_i$ being η-close to a border and $\nu_i \in (M_k + \eta, M_{k+1} - \eta)$. Finally, we obtain $t'_i = 0 = t^+_i$. $\square$

We now consider any strategy $\sigma_2 \in \mathsf{Strat}_2$, and construct a strategy $\sigma'_2 \in \mathsf{UStrat}^\eta_2$ such that $\mathsf{Play}(s, \sigma'_1, \sigma'_2) = \mathsf{Play}(s, \sigma_1, \sigma_2)^+$ for every state s η-close to a border. From Lemma 4, we will then get

$$\mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \geqslant \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))\,,$$

which will enable us to conclude. We assume $\mathsf{Play}(s,\sigma_1,\sigma_2)^+ = (\ell_0^+,\nu_0^+),(t_0^+,a_0^+), \dots,(\ell_n^+,\nu_n^+),\dots$ with $s=(\ell_0^+,\nu_0^+)$ η-close to a border. Then, we first define σ_2' over the finite plays $\mathsf{Play}(s,\sigma_1,\sigma_2)^+[n]$ with $n\in\mathbb{N}$, by letting

$$\sigma_2'(\mathsf{Play}(s,\sigma_1,\sigma_2)^+[n]) = (t_n^+,a_n^+)\,.$$

Notice first that this strategy verifies $\mathsf{Play}(s,\sigma_1',\sigma_2')[n] = \mathsf{Play}(s,\sigma_1,\sigma_2)[n]^+$ for every state s η-close to a border, by induction on $n\in\mathbb{N}$. In fact, in case $\mathsf{Play}(s,\sigma_1,\sigma_2)^+[n]$ ends with a state of player 1, the equation holds by construction of σ_1, and in case it ends with a state of player 2, by construction of σ_2'.

Once built on these finite plays, it is possible to extend σ_2' as an η-region-uniform strategy defined over every play: in particular, if $\mathsf{Play}(s,\sigma_1,\sigma_2)^+[n] \sim_\eta \mathsf{Play}(s',\sigma_1,\sigma_2)^+[n]$ (with s and s' different states η-close to a border), we have that $\sigma_2'(\mathsf{Play}(s,\sigma_1,\sigma_2)^+[n]) \sim_\eta \sigma_2'(\mathsf{Play}(s',\sigma_1,\sigma_2)^+[n])$ (induced by Lemma 4-4) validating the definition of η-region-uniform strategies.

This concludes the proof of $\overline{\mathsf{Val}}(s) \leqslant \overline{\mathsf{UVal}}^\eta(s)$ in case $\overline{\mathsf{Val}}(s) < +\infty$.

Finally, the case $\overline{\mathsf{Val}}(s)=+\infty$, corresponds to two possible situations: either player 2 has a way to ensure that the goal is never reached, or he cannot have such a guarantee, but still is able to make the price go bigger and bigger, i.e. he has a family of strategies that do not forbid from reaching the goal but ensure a price which is not bounded over the family[7]. It only remains to prove that player 2 can do it so with η-region-uniform strategies too. Let $\sigma_1\in\mathsf{UStrat}_1^\eta$ be an η-region-uniform strategy for player 1. We know that

$$\sup_{\sigma_2\in\mathsf{Strat}_2} \mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2)) = +\infty\,.$$

The first case corresponds to the one where there exists a strategy $\sigma_2\in\mathsf{Strat}_2$ such that $\mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2)) = +\infty$, i.e., in this outcome, the goal is not reached. As previously, it is possible to reconstruct from σ_2 an η-region-uniform strategy σ_2' achieving the very same goal (notice that the goal is definable with regions): the only difference is the fact that σ_2' must now mimic an infinite number of prefixes since the outcome is no longer finite. The second case finally corresponds to the one where there is no strategy $\sigma_2\in\mathsf{Strat}_2$ such that $\mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2))=+\infty$. We construct as previously a strategy $\sigma_1\in\mathsf{Strat}_2$ for player 1 from strategy σ_1'. From the fact that $\sup_{\sigma_2\in\mathsf{Strat}_2}\mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2)) > M$, we know the existence of a strategy $\sigma_2\in\mathsf{Strat}_2$ so that $\mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2)) > M$. Since this price is finite by hypothesis, the previous construction allows us to obtain a region-uniform strategy σ_2' verifying:

$$\mathsf{Cost}(\mathsf{Play}(s,\sigma_1',\sigma_2')) \geqslant \mathsf{Cost}(\mathsf{Play}(s,\sigma_1,\sigma_2)) > M$$

for all $M\in\mathbb{R}$. This proves that $\overline{\mathsf{UVal}}(s)=+\infty$.

[7] Indeed, we will show in Appendix B.4 that only the first alternative is possible.

B.2 Proof of Lemma 2

The proof of Lemma 2 is a refinement of the proof of Lemma 1 (where, moreover, the roles of both players are switched). To avoid the divergence phenomenon of Example 2, we restrict our attention to plays staying η-close to borders, that moreover jump closer and close to borders. More formally, a play $r = (\ell_0, \nu_0), (t_0, a_0), \dots$ is said to be *η-convergent* if for all $i \geqslant 0$, there exists k such that either $|\nu_i + t_i - M_k| \leqslant \eta/2^{i+1}$, or $t_i = 0$ and $\nu_i \in (M_k + \eta/2^{i+1}, M_k + \eta]$. Notice in particular that η-convergent runs stay η-close to borders. Unfortunately, it is not possible to define η-convergent runs as runs such that the first property ($|\nu_i + t_i - M_k| \leqslant \eta/2^{i+1}$) always holds since it would forbid a player to delay 0 time units when, in the round i, its valuation is $\nu_i \in (M_k + \eta/2^{i+1}, M_k + \eta]$. The second property is there to fix this issue.

We first study more precisely the relationship between general plays and η-convergent plays, like we did for runs staying η-close to borders. More precisely, we now explain how to construct from any finite play r, a play r^- such that (*i*) r^- is an η-convergent play; (*ii*) r and r^- are region-equivalent; and (*iii*) $\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r)$. Moreover, the construction will trivially verify that if r' and r are two plays that coincide on their prefix of length n, i.e., $r'[n] = r[n]$, then $(r')^-[n] = r^-[n]$. Indeed, the construction is by induction on the length of the play, and very similar to the construction of r^+ in the previous section. The main difference with the case of r^+, apart from the fact that we look for a play with a smaller weight rather than a greater weight, belongs in the fact that r^- must jump closer and closer to the borders.

Henceforth, let $r = (\ell_0, \nu_0), (t_0, a_0), \dots, (\ell_{i+1}, \nu_{i+1})$ and suppose that ν_0 is η-close to some M_k. We construct a play $r^- = (\ell_0^-, \nu_0^-), (t_0^-, a_0^-), \dots, (\ell_{i+1}^-, \nu_{i+1}^-)$. In case $i + 1 = 0$, we simply let $\ell_0^- = \ell_0^+ = \ell_0$ and $\nu_0^- = \nu_0^+ = \nu_0$. Otherwise, we consider the play r^- constructed by induction up to its configuration of index i, and we now explain how to construct the next timed action (t_i^-, a_i^-). First, we let $a_i^- = a_i$. Then, we distinguish between three cases:

- if there exists k such that $|\nu_i + t_i - M_k| \leqslant \eta/2^{i+1}$, then we let $t_i^- = \nu_i + t_i - \nu_i^-$;
- if there exists k such that $\nu_i + t_i \in (M_k + \eta/2^{i+1}, M_{k+1} - \eta/2^{i+1})$ and $\omega(\ell_i) = p^+$, then we let $t_i^- = \max(M_k + \eta/2^{i+1} - \nu_i^-, 0)$;
- if there exists k such that $\nu_i + t_i \in (M_k + \eta/2^{i+1}, M_{k+1} - \eta/2^{i+1})$ and $\omega(\ell_i) = p^-$, then there are two cases. In case $\ell_i \in L_1$ or $t_i > 0$ or $\nu_i^- > M_k + \eta$, in which case we let $t_i^- = M_{k+1} - \eta/2^{i+1} - \nu_i^-$. Otherwise, i.e., if $\ell_i \in L_2$ and $t_i = 0$ and $\nu_i^- \leqslant M_k + \eta$, then we let $t_i^- = \max(M_k + \eta/2^{i+1} - \nu_i^-, 0)$.

Once again, we will verify in the next lemma that t_i^- is always non-negative, ensuring that we indeed construct a valid play. Once defined t_i^-, we let $\ell_{i+1}^- = \ell_{i+1}$, and valuation ν_{i+1}^- is defined to be consistent with the semantics $[\![\mathcal{A}]\!]$ of the game.

Lemma 6. *Let r be a play that starts from some state η-close to a border. Then, the play r^- constructed before verifies the following properties:*

1. *if for some j, there exists k such that $|\nu_j + t_j - M_k| \leqslant \eta/2^{j+1}$, then $\nu_j^- + t_j^- = \nu_j + t_j$;*

2. *if for some j, there exists k such that $\nu_j + t_j \in (M_k + \eta/2^{j+1}, M_{k+1} - \eta/2^{j+1})$, then*
   (a) $\nu_j^- + t_j^- \in \{M_k + \eta/2^{j+1}, M_{k+1} - \eta/2^{j+1}, \nu_j^-\}$;
   (b) $\nu_j^- + t_j^- \leqslant M_{k+1} - \eta/2^{j+1}$;
   (c) *if* $t_j^- = 0$ *and* $\nu_j^- \leqslant M_k + \eta$, *then* $\nu_j^- \leqslant \nu_j + t_j$;
3. r^- *is an η-convergent play;*
4. $r^- \sim r$;
5. $\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r)$.

Remark 2. As for Lemma 1, notice that if the clock is not reset at step j, we have $\nu_{j+1} = \nu_j + t_j$, and $\nu_{j+1}^- = \nu_j^- + t_j^-$. In particular, in that case, property 1 implies that if $|\nu_{j+1} - M_k| \leqslant \eta/2^{j+1}$, then $\nu_{j+1}^- = \nu_{j+1}$. Similarly, property 2 implies that if $\nu_{j+1} \in (M_k + \eta/2^{j+1}, M_{k+1} - \eta/2^{j+1})$, then $\nu_{j+1}^- \in \{M_k + \eta/2^{j+1}, M_{k+1} - \eta/2^{j+1}, \nu_j^-\}$. Moreover, $\nu_{j+1}^- \leqslant M_{k+1} - \eta/2^{j+1}$ and if $t_j^- = 0$ and $\nu_j^- \leqslant M_k + \eta$, then $\nu_j^- \leqslant \nu_{j+1}$. Properties (b) and (c) will be useful in the proof by induction of subsequent properties.

Proof. All properties are shown by induction on the length of the play. All properties are clearly true in case of a play r reduced to a single state (which is assumed to be η-close to a border). We now consider a play $r = (\ell_0, \nu_0), (t_0, a_0), \dots, (\ell_{i+1}, \nu_{i+1})$ with $i + 1 \geqslant 1$ and we prove the properties for the play r^- constructed before.

1. The first property is true directly by construction.
2. For the second property, by induction hypothesis, it is sufficient to prove the property for $j = i$. Hence, suppose that $\nu_i + t_i \in (M_k + \eta/2^{i+1}, M_{k+1} - \eta/2^{i+1})$.
   In case $\omega(\ell_i) = p^+$, we have $t_i^- \in \{M_k + \eta/2^{i+1} - \nu_i^-, 0\}$. In case $\omega(\ell_i) = p^-$, we have $t_i^- \in \{M_k + \eta/2^{i+1} - \nu_i^-, M_{k+1} - \eta/2^{i+1} - \nu_i^-, 0\}$. This implies that $\nu_i^- + t_i^- \in \{M_k + \eta/2^{i+1}, M_{k+1} - \eta/2^{i+1}, \nu_i^-\}$, i.e., property (a). Property (b), namely $\nu_i^- + t_i^- \leqslant M_{k+1} - \eta/2^{i+1}$, needs only to be proved when $\nu_i^- + t_i^- = \nu_i^-$ (otherwise the property is trivially verified since $M_k + \eta/2^{i+1} < M_{k+1} - \eta/2^{i+1}$). In that case, there are two possibilities. If ν_i is $\eta/2^i$-close to borders, since $\nu_i \leqslant \nu_i + t_i$, we have $\nu_i \leqslant M_{k+1} - \eta/2^{i+1}$, and by property 1, $\nu_i^- = \nu_i \leqslant M_{k+1} - \eta/2^{i+1}$ (as $\nu_i^- = \nu_{i-1}^- + t_{i-1}^- = \nu_{i-1} + t_{i-1} = \nu_i$). If ν_i is not $\eta/2^i$-close to borders, then $\nu_i \in (M_{k'} + \eta/2^i, M_{k'+1} - \eta/2^i)$, with $k' \leqslant k$. By induction, $\nu_i^- \leqslant M_{k'+1} - \eta/2^i$, so that $\nu_i^- + t_i^- = \nu_i^- \leqslant M_{k+1} - \eta/2^{i+1}$.
   Finally, let us prove property (c). Assume that $t_i^- = 0$ and $\nu_i^- \leqslant M_k + \eta$. We now prove that $\nu_i^- \leqslant \nu_i$. Whatever the value of $\omega(\ell_i)$, we have that $\nu_i^- \geqslant M_k + \eta/2^{i+1}$. There are then four cases.
   - In a first case, we have $\nu_i \leqslant M_k$. Whatever ν_i is $\eta/2^i$-close to a border or not, we obtain (by property 1 or by induction), that $\nu_i^- \leqslant M_k \leqslant \nu_i + t_i$.
   - In a second case, we have $\nu_i \in (M_k, M_k + \eta/2^i]$. By property 1, we deduce that $\nu_i^- = \nu_i \leqslant \nu_i + t_i$.
   - The third case corresponds to $\nu_i \in (M_k + \eta/2^i, M_{k+1} - \eta/2^i)$, which implies that $\nu_{i-1} + t_{i-1} = \nu_i \in (M_k + \eta/2^i, M_{k+1} - \eta/2^i)$. By induction,

we obtain that $\nu_i^- = \nu_{i-1}^- + t_{i-1}^- \in \{M_k + \eta/2^i, M_{k+1} - \eta/2^i, \nu_{i-1}^-\}$. If $\nu_i^- = M_k + \eta/2^i < \nu_i \leqslant \nu_i + t_i$, we conclude directly. The case $\nu_i^- = M_{k+1} - \eta/2^i > M_k + \eta$ leads to a contradiction. Finally, if $\nu_i^- = \nu_{i-1}^-$, since we assume $\nu_{i-1}^- \leqslant M_k + \eta$, we obtain by induction that $\nu_{i-1}^- \leqslant \nu_{i-1} + t_{i-1} = \nu_i \leqslant \nu_i + t_i$.

   - The fourth case is $\nu_i \in [M_{k+1} - \eta/2^i, M_{k+1} - \eta/2^{i+1})$, but then $\nu_i^- \leqslant M_k + \eta \leqslant M_{k+1} - \eta/2^i \leqslant \nu_i \leqslant \nu_i + t_i$.

3. We now prove that r^- is indeed a play. The only non-trivial property is that t_i^- is a non negative delay, especially when $t_i^- = \nu_i + t_i - \nu_i^-$ or $t_i^- = M_{k+1} - \eta/2^{i+1} - \nu_i^-$. Consider first the case $t_i^- = \nu_i + t_i - \nu_i^-$. It has to be shown that if there exists k such that $|\nu_i + t_i - M_k| \leqslant \eta/2^{i+1}$, then ν_i^- is not greater than $\nu_i + t_i$. Notice that $\nu_i \leqslant \nu_i + t_i$. Hence, if ν_i is $\eta/2^i$-close to a border, then by property 1, $\nu_i^- = \nu_i \leqslant \nu_i + t_i$, and we are done. Otherwise (i.e. if ν_i is not $\eta/2^i$-close to a border), since $\nu_i \leqslant \nu_i + t_i \leqslant M_k + \eta/2^{i+1}$, we even know that $\nu_i < M_k - \eta/2^i$. Since no reset has been possibly performed during action a_{i-1} (otherwise, $\nu_i = 0$ is $\eta/2^i$-close to a border), we have $\nu_i = \nu_{i-1} + t_{i-1} \in (M_{k'} + \eta/2^i, M_{k'+1} - \eta/2^i)$ with $k' < k$. By property 2, this implies that $\nu_i^- = \nu_{i-1}^- + t_{i-1}^- \leqslant M_{k'+1} - \eta/2^i$. In consequence, $\nu_i^- \leqslant M_k - \eta/2^i \leqslant \nu_i + t_i$.
   Consider then the case $t_i^- = M_{k+1} - \eta/2^{i+1} - \nu_i^-$, which holds when $\nu_i + t_i \in (M_k + \eta/2^{i+1}, M_{k+1} - \eta/2^{i+1})$ and $\omega(\ell_i) = p^-$. Then, $\nu_i \leqslant \nu_i + t_i < M_{k+1} - \eta/2^{i+1}$. Hence, either ν_i is $\eta/2^i$-close to a border (in particular when the clock has just been reset), in which case, by property 1, we have $\nu_i^- = \nu_i < M_{k+1} - \eta/2^{i+1}$, so that $t_i^- > 0$. Or $\nu_i \in (M_{k'} + \eta/2^i, M_{k'+1} - \eta/2^i)$ with $k' \leqslant k$. By property 2, this implies that $\nu_i^- \leqslant M_{k'+1} - \eta/2^i < M_{k+1} - \eta/2^{i+1}$, once again implying that $t_i^- > 0$.
   The fact that r^- is an η-convergent play is then directly implied by properties 1 and 2.
4. Property $r^- \sim r$ is also a direct consequence of properties 1 and 2.
5. It only remains to prove that $\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r)$. Notice that, by induction, only matters the weight since the last index j where plays r and r^- have synchronized, i.e., where $\nu_j + t_j = \nu_j^- + t_j^-$. For the sake of simplicity, we suppose in the following that $j = 0$. In particular, we may suppose that r does not contain reset transitions or positions j such that ν_j is $\eta/2^j$-close to borders, except possibly the very last one. Since there are no resets, we have $\nu_j = \nu_{j-1} + t_{j-1}$ for every $j > 0$.
   We now consider separately the possibility of sets $\{p^-, p^+\}$.
   - As a first case, consider that $p^- = -1$ and $p^+ = +1$. We prove by induction over $0 \leqslant j \leqslant i$ that

$$\mathsf{Cost}(r^-[j+1]) \leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu_{j+1}^-| .$$

     To simplify the notations, we forget the weights of the actions in this proof, but notice that the same weights occur in r and r^- since these two plays are equivalent.

- If $j = 0$, then we have supposed that ν_0 is η-close to a border, so that $\nu_0^- = \nu_0$. Let k be such that $\nu_1 \in (M_k + \eta/2, M_{k+1} - \eta/2)$. Then, if $\omega(\ell_0) = +1$, $t_0^- = \max(M_k + \eta/2 - \nu_0, 0)$. If $\nu_0 \geqslant M_k + \eta/2$, then $\nu_1^- = \nu_0^- = \nu_0$ so that

  $$\mathsf{Cost}(r^-[1]) = 0 = \mathsf{Cost}(r[1]) - |\nu_1 - \nu_0| = \mathsf{Cost}(r[1]) - |\nu_1 - \nu_1^-|\,.$$

  If $\nu_0 < M_k + \eta/2$, $\nu_1^- = M_k + \eta/2 \leqslant \nu_1$ so that

  $$\mathsf{Cost}(r^-[1]) = \nu_1^- - \nu_0 = \mathsf{Cost}(r[1]) - \nu_1 + \nu_1^- = \mathsf{Cost}(r[1]) - |\nu_1 - \nu_1^-|\,.$$

  Consider then the case where $\omega(\ell_0) = -1$. In case $\nu_1^- = M_{k+1} - \eta/2 \geqslant \nu_1$, we have

  $$\begin{aligned}\mathsf{Cost}(r^-[1]) = -(\nu_1^- - \nu_0) &= \mathsf{Cost}(r[1]) - (\nu_1^- - \nu_1)\\ &= \mathsf{Cost}(r[1]) - |\nu_1 - \nu_1^-|\,.\end{aligned}$$

  Otherwise, we know that $t_0 = 0$ and $\nu_0^- \leqslant M_k + \eta$. Since ν_0 is not $\eta/2$-close from a border, but $\nu_1 = \nu_0$ should be η-close from a border, we know that $\nu_0 \in (M_k + \eta/2, M_k + \eta] \cup [M_{k+1} - \eta, M_{k+1} - \eta/2)$. Since $\nu_0^- = \nu_0 \leqslant M_k + \eta$, we know that $\nu_0 \in (M_k + \eta/2, M_k + \eta]$. Then, we obtain $\nu_1^- = M_k + \eta/2 \leqslant \nu_0 = \nu_1$, so that

  $$\mathsf{Cost}(r^-[1]) = -(\nu_1^- - \nu_0) = \mathsf{Cost}(r[1]) - |\nu_1^- - \nu_1|\,.$$

- Let us suppose that the property is proved for all indices less than or equal to j, and prove it for $j+1$. We let k be such that $\nu_{j+1} = \nu_j + t_j \in (M_k + \eta/2^{j+1}, M_{k+1} - \eta/2^{j+1})$. We will distinguish four possible cases depending on $\omega(\ell_j)$ and the relative order between ν_{j+1}^- and ν_{j+1}.
  (a) We first suppose that $\omega(\ell_j) = -1$. Then,

  $$\begin{aligned}\mathsf{Cost}(r^-[j+1]) &= \mathsf{Cost}(r^-[j]) - (\nu_{j+1}^- - \nu_j^-)\\ &\leqslant \mathsf{Cost}(r[j]) - |\nu_j - \nu_j^-| - \nu_{j+1}^- + \nu_j^- \quad \text{(Ind. Hyp.)}\\ \mathsf{Cost}(r^-[j+1]) &\leqslant \mathsf{Cost}(r[j+1]) + (\nu_{j+1} - \nu_j)\\ &\qquad - |\nu_j - \nu_j^-| - \nu_{j+1}^- + \nu_j^-\,. \end{aligned} \tag{3}$$

     i. In the case where $\nu_{j+1}^- > \nu_{j+1}$, we have $|\nu_{j+1} - \nu_{j+1}^-| = \nu_{j+1}^- - \nu_{j+1}$ so that (3) becomes

     $$\begin{aligned}\mathsf{Cost}(r^-[j+1]) \leqslant \mathsf{Cost}(r[j+1]) &- |\nu_{j+1} - \nu_{j+1}^-|\\ &+ (\nu_j^- - \nu_j) - |\nu_j^- - \nu_j|\end{aligned}$$

     which is less than or equal to $\mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu_{j+1}^-|$ since $\nu_j^- - \nu_j \leqslant |\nu_j^- - \nu_j|$.

ii. Similarly, in the case where $\nu^-_{j+1} < \nu_{j+1}$, we have $|\nu_{j+1} - \nu^-_{j+1}| = \nu_{j+1} - \nu^-_{j+1}$. Notice that this necessarily implies that $\ell_i \in L_2$ and $t_j = 0$, $\nu^-_j \leqslant M_k + \eta$ and $t^-_j = \max(M_k + \eta/2^{j+1} - \nu^-_j, 0)$: otherwise, we would have $\nu^-_{j+1} = M_{k+1} - \eta/2^{i+1} > \nu_{j+1}$ that contradicts the hypothesis. If $t^-_j = 0$, this implies that $\nu^-_j = \nu^-_{j+1} < \nu_{j+1} = \nu_j$, so that (3) can be rewritten

$$\begin{aligned}\mathsf{Cost}(r^-[j+1]) &\leqslant \mathsf{Cost}(r[j+1]) - \nu_j + \nu^-_j \\ &= \mathsf{Cost}(r[j+1]) - |\nu_{j+1} + \nu^-_{j+1}| \,.\end{aligned}$$

Otherwise, $t^-_j > 0$ and we have $t^-_j = M_k + \eta/2^{j+1} - \nu^-_j$. This is possible only if $\nu^-_j \leqslant M_k + \eta/2^{j+1} < \nu_{j+1} = \nu_j$. Then, $|\nu^-_j - \nu_j| = \nu_j - \nu^-_j$ so that

$$\begin{aligned}\mathsf{Cost}(r^-[j+1]) &\leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} + \nu^-_{j+1}| - 2|\nu^-_j - \nu_j| \\ &\leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} + \nu^-_{j+1}| \,.\end{aligned}$$

(b) Suppose then that $\omega(\ell_j) = +1$. Then, a similar calculation gives

$$\begin{aligned}\mathsf{Cost}(r^-[j+1]) &= \mathsf{Cost}(r^-[j]) + \nu^-_{j+1} - \nu^-_j \\ &\leqslant \mathsf{Cost}(r[j]) - |\nu_j - \nu^-_j| + \nu^-_{j+1} - \nu^-_j \quad \text{(Ind. Hyp.)} \\ \mathsf{Cost}(r^-[j+1]) &\leqslant \mathsf{Cost}(r[j+1]) - (\nu_{j+1} - \nu_j) \\ &\qquad - |\nu_j - \nu^-_j| + \nu^-_{j+1} - \nu^-_j \,. \end{aligned} \tag{4}$$

i. Once again, if $\nu^-_{j+1} < \nu_{j+1}$, we have $|\nu_{j+1} - \nu^-_{j+1}| = \nu_{j+1} - \nu^-_{j+1}$ so that (4) is rewritten

$$\mathsf{Cost}(r^-[j+1]) \leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^-_{j+1}| + \nu_j - \nu^-_j - |\nu_j - \nu^-_j|$$

which is less than or equal to $\mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^-_{j+1}|$ since $\nu_j - \nu^-_j \leqslant |\nu_j - \nu^-_j|$.

ii. Similarly, if $\nu^-_{j+1} > \nu_{j+1}$, we know by property 2 that $\nu^-_{j+1} \in \{M_{k+1} - \eta/2^{j+1}, \nu^-_j\}$. If $\nu^-_{j+1} = M_{k+1} - \eta/2^{j+1}$, since $t^-_j = \max(M_k + \eta/2^{j+1} - \nu^-_j, 0)$, we know that $t^-_j = 0$, i.e., in all case $\nu^-_{j+1} = \nu^-_j$. Then, $\nu^-_{j+1} = \nu^-_j > \nu_{j+1} \geqslant \nu_j$. Knowing that $|\nu_{j+1} - \nu^-_{j+1}| = \nu^-_{j+1} - \nu_{j+1}$, (4) becomes

$$\begin{aligned}\mathsf{Cost}(r^-[j+1]) &\leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^-_{j+1}| \\ &\qquad + 2(\nu^-_{j+1} - \nu_{j+1}) + 2(\nu_j - \nu^-_j) \\ &= \mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^-_{j+1}| + 2(\nu_j - \nu_{j+1}) \\ &\leqslant \mathsf{Cost}(r[j+1]) - |\nu_{j+1} - \nu^-_{j+1}| \,.\end{aligned}$$

We finally have proved the property by induction. Notice in particular that this shows that $\mathsf{Cost}(r^-[j+1]) \leqslant \mathsf{Cost}(r[j+1])$ for every j with $\nu^-_{j+1} \neq \nu_{j+1}$. To conclude the proof of $\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r)$, it remains to deal with the case of a possible last transition ending with $\nu^-_{i+1} = \nu_{i+1}$. Unless $i = 0$, in which case we have $\mathsf{Cost}(r^-) = \mathsf{Cost}(r)$, we know by hypothesis that $\nu^-_i \neq \nu_i$. By the previous property, we have $\mathsf{Cost}(r^-[i]) \leqslant \mathsf{Cost}(r[i]) - |\nu_i - \nu^-_i|$. Then, $\mathsf{Cost}(r^-) = \mathsf{Cost}(r^-[i]) + \omega(\ell_i)(\nu^-_{i+1} - \nu^-_i)$ and $\mathsf{Cost}(r) = \mathsf{Cost}(r[i]) + \omega(\ell_i)(\nu_{i+1} - \nu_i)$. In the overall, we get

$$\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r) + \omega(\ell_i)(\nu_i - \nu^-_i) - |\nu_i - \nu^-_i|\,.$$

In all cases, we verify that $\omega(\ell_i)(\nu_i - \nu^-_i) \leqslant |\nu_i - \nu^-_i|$, so that we have proved that $\mathsf{Cost}(r^-) \leqslant \mathsf{Cost}(r)$.

– We now consider the case where $p^- = 0$ and $p^+ = +1$ (the case $p^- = -1$ and $p^+ = 0$ is very similar, and not explained in details here). We prove another inequality by induction over $0 \leqslant j \leqslant i$, namely that

$$\mathsf{Cost}(r^-[j+1]) \leqslant \mathsf{Cost}(r[j+1]) - \max(\nu_{j+1} - \nu^-_{j+1}, 0)\,.$$

The proof is very similar to the previous case, and we conclude as previously. ☐

We now go to the proof of Lemma 2. In case, $\underline{\mathsf{Val}}(s) = +\infty$ nothing has to be done. We then consider the case $\underline{\mathsf{Val}}(s) < +\infty$. Let $\sigma'_2 \in \mathsf{CStrat}^\eta_2$. We now explain how to construct a strategy $\sigma_2 \in \mathsf{Strat}_2$ such that for all state s

$$\inf_{\sigma'_1 \in \mathsf{CStrat}^\eta_1} \mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \leqslant \inf_{\sigma_1 \in \mathsf{Strat}_1} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))\,.$$

To prove such an inequality, we will consider any strategy $\sigma_1 \in \mathsf{Strat}_1$ and construct a strategy $\sigma'_1 \in \mathsf{CStrat}^\eta_1$ such that

$$\mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \leqslant \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2))\,.$$

Strategy σ_2 follows σ'_2 in case of η-convergent plays. We must however extend it to deal with the other plays faithfully. Let $r = (\ell_0, \nu_0), (t_0, a_0), \ldots, (\ell_i, \nu_i)$ be any finite play ending in a location ℓ_i of player 2, and $r^- = (\ell^-_0, \nu^-_0), (t^-_0, a^-_0), \ldots, (\ell^-_i, \nu^-_i)$ the play constructed as before. By Lemma 6, we know that r^- is an η-convergent play. Hence, $\sigma'_2(r^-) = (t'_i, a)$, for some $t'_i \in \mathbb{R}_{\geqslant 0}$, and there exists k such that either $\nu^-_i + t'_i \in \{M_k + \eta/2^{i+1}\} \cup [M_k - \eta/2^{i+1}, M_k)$, or $t'_i = 0$ and $\nu^-_i \in (M_k + \eta/2^{i+1}, M_k + \eta]$. We let $t_i = \max(\nu^-_i + t'_i - \nu_i, 0)$ and $\sigma_2(r) = (t_i, a)$. Let $\tilde{r}$ (respectively, r') be the play r (respectively, r^-) extended with the step prescribed by σ_2 (respectively, σ'_2). Then, we prove that r' matches the construction above starting from the run $\tilde{r}$, i.e., $\tilde{r}^- = r'$. By construction, we only have to verify that the value of t'_i is consistent with the previous constructions, i.e., $t'_i = t^-_i$.

Lemma 7. *We have $t'_i = t^-_i$.*

Proof. The proof considers several cases.

- Suppose first that there exists k' such that $|\nu_i + t_i - M_{k'}| \leqslant \eta/2^{i+1}$. In case $\nu_i^- + t'_i = \nu_i + t_i$, we have $t'_i = \nu_i + t_i - \nu_i^-$ which is equal to t_i^- (since $\nu_i + t_i$ is $\eta/2^{i+1}$-close to a border implying that the first rule of the construction of $\tilde{r}^-$ applies). Otherwise (i.e. when $\nu_i^- + t'_i \neq \nu_i + t_i$), we know that $t_i = 0$ (by definition of t_i as $\max(\nu_i^- + t'_i - \nu_i, 0)$) and that $\nu_i > \nu_i^- + t'_i \geqslant \nu_i^-$. In particular, we have $\nu_i \neq \nu_i^-$. However, since $t_i = 0$, $|\nu_i - M_{k'}| \leqslant \eta/2^{i+1} \leqslant \eta/2^i$ so that we should have $\nu_i^- = \nu_i$, causing a contradiction.
- Suppose then that there exists k' such that $\nu_i + t_i \in (M_{k'} + \eta/2^{i+1}, M_{k'+1} - \eta/2^{i+1})$.
  - In case $\nu_i + t_i = \nu_i^- + t'_i$, since σ'_2 is η-convergent and $\nu_i^- + t'_i$ not $\eta/2^{i+1}$-close to a border, we have $t'_i = 0$, and $\nu_i^- \in (M_k + \eta/2^{i+1}, M_k + \eta]$: in particular, $k' = k$ and $\nu_i + t_i = \nu_i^- \in (M_k + \eta/2^{i+1}, M_k + \eta]$. We know that $\nu_i \sim \nu_i^-$, hence there are two possibilities for the position of $\nu_i \in (M_k, M_k + \eta]$.
    - The first case is $\nu_i \in (M_k, M_k + \eta/2^i)$: by Lemma 6-1 (applied on $\nu_i = \nu_{i-1} + t_{i-1}$ since no reset transition may have been taken), we know that $\nu_i^- = \nu_i$. Since $\nu_i^- = \nu_i + t_i$, we have $t_i = 0$. This shows that $t'_i = 0$ is consistent with the construction of $\tilde{r}^-$ that sets $t_i^- = \nu_i + t_i - \nu_i^- = 0$ in that case (since $\nu_i + t_i \in (M_k, M_k + \eta/2^i)$).
    - The second case is $\nu_i \in (M_k + \eta/2^i, M_k + \eta]$: by Lemma 6-2-($a$), we know that $\nu_i^- \in \{M_k + \eta/2^i, M_{k+1} - \eta/2^i, \nu_{i-1}^-\}$. It is not possible neither that $\nu_i^- = M_k + \eta/2^i < \nu_i$ (because $\nu_i^- = \nu_i + t_i \geqslant \nu_i$), nor that $\nu_i^- = M_{k+1} - \eta/2^i > M_k + \eta$ (because we know that $\nu_i^- \leqslant M_k + \eta$). Hence, we have $\nu_i^- = \nu_{i-1}^-$ (and thus $t_{i-1}^- = 0$). Moreover, by Lemma 6-2-(c), since $t_{i-1}^- = 0$ and $\nu_{i-1}^- = \nu_i^- \leqslant M_k + \eta$, we have $\nu_i^- = \nu_{i-1}^- \leqslant \nu_{i-1} + t_{i-1} = \nu_i$. Since, we also have $\nu_i \leqslant \nu_i + t_i = \nu_i^-$, we obtain that $\nu_i = \nu_i^-$, and $t_i = \nu_i^- + t'_i - \nu_i = 0$. In case $\omega(\ell_i) = p^+$, this shows that $t'_i = 0$ is consistent with the construction of $\tilde{r}^-$ that sets $t_i^- = \max(M_k + \eta/2^{i+1} - \nu_i^-, 0) = 0$ (since $\nu_i^- = \nu_i > M_k + \eta/2^i > M_k + \eta/2^{i+1}$). In case $\omega(\ell_i) = p^-$, since we are in the case where $\ell_i \in L_2$ (since ℓ_i is a location where σ_2 needs to be defined), $t_i = 0$ and $\nu_i^- \leqslant M_k + \eta$, the choice $t'_i = 0$ is also consistent with the construction of $\tilde{r}^-$ which sets $t_i^- = \max(M_k + \eta/2^{i+1} - \nu_i^-, 0) = 0$ (once again, because $\nu_i^- > M_k + \eta/2^{i+1}$).
  - The last case is when $\nu_i + t_i \neq \nu_i^- + t'_i$, implying that $t_i = 0$ (by definition of t_i as $\max(\nu_i^- + t'_i - \nu_i, 0)$). This implies that $\nu_i > \nu_i^- + t'_i \geqslant \nu_i^-$. It is not possible that $\nu_i \in (M_{k'} + \eta/2^{i+1}, M_{k'} + \eta/2^i] \cup [M_{k'+1} - \eta/2^i, M_{k'+1} - \eta/2^{i+1})$, since otherwise we would have $\nu_i^- = \nu_i$, by Lemma 6-1 (applied to $\nu_i^- = \nu_{i-1}^- + t_{i-1}^- = \nu_{i-1} + t_{i-1} = \nu_i$). Hence, we have $\nu_i \in (M_{k'} + \eta/2^i, M_{k'+1} - \eta/2^i)$. In particular, by Lemma 6-2-(a), we know that $\nu_i^- \in \{M_{k'} + \eta/2^i, M_{k'+1} - \eta/2^i, \nu_{i-1}^-\}$. There are two possibilities now, depending on t'_i taken from the fact that σ'_2 is η-convergent.

* The first possibility is $\nu_i^- + t'_i \in \{M_k + \eta/2^{i+1}\} \cup [M_k - \eta/2^{i+1}, M_k)$. Notice that $\nu_i^- \sim \nu_i$ and $\nu_i^- + t'_i \in [\nu_i^-, \nu_i)$, so that $\nu_i^- + t'_i \sim \nu_i$. Hence, $\nu_i^- + t'_i \in (M_{k'}, M_{k'+1})$. Knowing that $\nu_i^- + t'_i < \nu_i < M_{k'+1} - \eta/2^i$, and that $|\nu_i^- + t'_i - M_k| \leqslant \eta/2^{i+1}$, we conclude that $k' = k$ and $\nu_i^- + t'_i \in (M_{k'}, M_{k'} + \eta/2^{i+1}]$. It is therefore only possible that $\nu_i^- + t'_i = M_{k'} + \eta/2^{i+1}$, i.e., $t'_i = M_{k'} + \eta/2^{i+1} - \nu_i^-$. This choice is consistent with the construction of $\tilde{r}^-$, whatever the price of ℓ_i, which sets $t_i^- = \max(M_{k'} + \eta/2^{i+1} - \nu_i^-, 0) = M_{k'} + \eta/2^{i+1} - \nu_i^-$ (since $\nu_i^- \leqslant M_{k'} + \eta/2^{i+1}$): in particular if $\omega(\ell_i) = p^-$, we are indeed in the case $\ell_i \in L_2$, $t_i = 0$ and $\nu_i^- \leqslant M_{k'} + \eta$.
* The second possibility is that $t'_i = 0$ and $\nu_i^- \in (M_k + \eta/2^{i+1}, M_k + \eta]$, in which case we again deduce from $\nu_i \sim \nu_i^-$ that $k' = k$. We have $t_i^- = \max(M_{k'} + \eta/2^{i+1} - \nu_i^-, 0) = 0 = t'_i$ (since $\nu_i^- \geqslant M_{k'} + \eta/2^{i+1}$), whatever the price of ℓ_i: once again, if $\omega(\ell_i) = p^-$, we are indeed in the case $\ell_i \in L_2$, $t_i = 0$ and $\nu_i^- \leqslant M_{k'} + \eta$. □

We now consider any strategy $\sigma_1 \in \mathsf{Strat}_1$, and construct a strategy $\sigma'_1 \in \mathsf{CStrat}_1^\eta$ such that $\mathsf{Play}(s, \sigma'_1, \sigma'_2) = \mathsf{Play}(s, \sigma_1, \sigma_2)^-$ for every state s η-close to a border. From Lemma 6, we will then get

$$\mathsf{Cost}(\mathsf{Play}(s, \sigma'_1, \sigma'_2)) \leqslant \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)),$$

which will enable us to conclude. We let $\mathsf{Play}(s, \sigma_1, \sigma_2)^- = (\ell_0^-, \nu_0^-), (t_0^-, a_0^-), \ldots, (\ell_n^-, \nu_n^-), \ldots$ with $s = (\ell_0^-, \nu_0^-)$ η-close to a border. Then, we first define σ'_1 over the finite plays $\mathsf{Play}(s, \sigma_1, \sigma_2)^-[n]$ with $n \in \mathbb{N}$, by letting

$$\sigma'_1(\mathsf{Play}(s, \sigma_1, \sigma_2)^-[n]) = (t_n^-, a_n^-).$$

Notice first that this strategy verifies $\mathsf{Play}(s, \sigma'_1, \sigma'_2)[n] = \mathsf{Play}(s, \sigma_1, \sigma_2)[n]^-$ for every state s η-close to a border, by induction on $n \in \mathbb{N}$. In fact, in case $\mathsf{Play}(s, \sigma_1, \sigma_2)^-[n]$ ends with a state of player 2, the equation holds by construction of σ_2, and in case it ends with a state of player 1, by construction of σ'_1.

Once built on these finite plays, it is possible to extend σ'_1 as an η-convergent strategy defined over every play: in particular, the η-region-uniformity is possible, since, if $\mathsf{Play}(s, \sigma_1, \sigma_2)^-[n] \sim_\eta \mathsf{Play}(s', \sigma_1, \sigma_2)^-[n]$ (with s and s' η-close to a border), we have $\sigma'_1(\mathsf{Play}(s, \sigma_1, \sigma_2)^-[n]) \sim_\eta \sigma'_1(\mathsf{Play}(s', \sigma_1, \sigma_2)^-[n])$ (induced by Lemma 6-4) validating the definition of η-region-uniform strategies. The η-convergence is ensured by Lemma 6-3.

This concludes the proof of $\underline{\mathsf{Val}}(s) \geqslant \underline{\mathsf{UVal}}^\eta(s)$ in case $\underline{\mathsf{Val}}(s) > -\infty$.

Finally, in case $\underline{\mathsf{Val}}(s) = -\infty$, we have to show that $\underline{\mathsf{CVal}}^\eta(s) = -\infty$ too. Notice that $\underline{\mathsf{Val}}(s) = -\infty$ means that for all strategy $\sigma_2 \in \mathsf{Strat}_2$ of player 2, we have

$$\inf_{\sigma_1 \in \mathsf{Strat}_1} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)) = -\infty,$$

i.e., player 1 has a sequence of strategies ensuring the reachability of the goal with smaller and smaller prices. Hence, let $\sigma'_2 \in \mathsf{CStrat}_2^\eta$ be an η-convergent strategy for player 2, and $M \in \mathbb{R}$ be any constant. We construct as previously a strategy $\sigma_2 \in \mathsf{Strat}_2$ for player 2. From the fact that $\inf_{\sigma_1 \in \mathsf{Strat}_1} \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)) < M$, we know the existence of a strategy $\sigma_1 \in \mathsf{Strat}_1$ so that $\mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)) < M$. In particular, this price is finite so that the previous construction allows us to obtain an η-convergent strategy σ_1' verifying that

$$\mathsf{Cost}(\mathsf{Play}(s, \sigma_1', \sigma_2')) \leqslant \mathsf{Cost}(\mathsf{Play}(s, \sigma_1, \sigma_2)) < M\,.$$

This proves that $\underline{\mathsf{CVal}}(s) = -\infty$.

B.3 Proof of Lemma 3

The proof uses the fact that we can translate a strategy in $\tilde{\mathcal{A}}$ into η-region-uniform/η-convergent strategies in the original game $\mathcal{A}$ and vice versa.

Since we assume that $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$ is finite, let ξ_1^* and ξ_2^* the optimal strategies for both players provided by Theorem 1. Strategy ξ_1^* uses a finite memory whereas ξ_2^* is memoryless. They verify, for all ℓ, and $0 \leqslant k \leqslant K$:

$$\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) = \mathsf{Cost}(\mathsf{Play}((\ell, \{M_k\}), \xi_1^*, \xi_2^*))\,.$$

We first show inequality $\overline{\mathsf{UVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) - \varepsilon \leqslant \mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$. In case $\overline{\mathsf{UVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) = -\infty$, the inequality is trivially verified. Since $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\}))$ is finite for every ℓ, and $0 \leqslant k \leqslant K$, we know that every play where player 1 follows strategy ξ_1^* reaches the target set of states. Let $L \in \mathbb{N}$ be the maximum $\mathsf{Length}(r)$ for every such play r: notice that L is finite since the game arena is finite and that ξ_1^* has finite memory. For all $\varepsilon > 0$, we let η be a positive rational number less than $\varepsilon/(2L)$ and we explain how to construct an η-region-uniform strategy $\sigma_1^{(\varepsilon)}$ of $\mathcal{A}$ so that

$$\sup_{\sigma_2 \in \mathsf{UStrat}_2^{\eta}(\mathcal{A})} \mathsf{Cost}(\mathsf{Play}((\ell, M_k), \sigma_1^{(\varepsilon)}, \sigma_2)) \leqslant \mathsf{Cost}(\mathsf{Play}((\ell, \{M_k\}), \xi_1^*, \xi_2^*)) + \varepsilon \quad (5)$$

Since $\sigma_1^{(\varepsilon)}$ will only have to play against an η-region-uniform strategy σ_2, we may define only $\sigma_1^{(\varepsilon)}$ on finite plays r only visiting η-regions in $\mathcal{I}^{\eta}_{\mathcal{A}}$, i.e., that only stops at distance at most η from the integers M_k. Such a play r can indeed be translated in a play ρ of $\tilde{\mathcal{A}}$, by simply translating each state (ℓ_i, ν_i) (with ν_i at distance at most η from some M_k by assumption) it encounters into (ℓ_i, I_i) with $I_i \in \mathcal{I}^{\eta}_{\mathcal{A}}$ the interval containing ν_i. Then, we let $(J, a) = \xi_1^*(\rho)$. Let also denote by (ℓ', ν) the last state of r, as well as I the interval of $\mathcal{I}^{\eta}_{\mathcal{A}}$ containing ν. Notice that $I \preceq J$. In case $J = I$, we let $\sigma_1^{(\varepsilon)}(r) = (0, a)$ forcing player 1 to play immediately. In case J is a singleton $\{M_k\}$, we let $\sigma_1^{(\varepsilon)}(r) = (M_k - \nu, a)$. In case $J = [M_k - \eta, M_k)$, we let $\sigma_1^{(\varepsilon)}(r) = (M_k - \eta - \nu, a)$. Finally, in case $J = (M_k, M_k + \eta]$, we let $\sigma_1^{(\varepsilon)}(r) = (M_k + \eta - \nu, a)$. Notice that in all case, the fact that $\nu \in I$ implies that the delay prescribed in $\sigma_1^{(\varepsilon)}(r)$ is not less than 0.

We now prove (5). For that purpose, we consider a strategy $\sigma_2 \in \mathsf{UStrat}_2^\eta(\mathcal{A})$. We reconstruct from it a strategy $\xi_2 \in \mathsf{Strat}_2(\tilde{\mathcal{A}})$ such that $\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2)$ is the sequence of η-regions visited by the play $\mathsf{Play}((\ell,M_k),\sigma_1^{(\varepsilon)},\sigma_2)$, which stays η-close to borders, since $\sigma_1^{(\varepsilon)}$ and σ_2 are η-region-uniform strategies. The following lemma compares the weight of these two plays.

Lemma 8. *Let r be a play of $\mathcal{A}$ staying η-close to borders and $\tilde{r}$ its abstraction in terms of η-regions. Then,*

$$|\mathsf{Cost}(\tilde{r}) - \mathsf{Cost}(r)| \leqslant 2\eta\,\mathsf{Length}(\tilde{r})\,.$$

Proof. Nothing has to be done in case $\mathsf{Cost}(r)$ (or equivalently $\mathsf{Cost}(\tilde{r})$) is equal to $+\infty$. Hence, we can suppose that these two runs have a finite length. We show by induction on $0 \leqslant n \leqslant \mathsf{Length}(\tilde{r}) = \mathsf{Length}(r)$ that

$$|\mathsf{Cost}(\tilde{r}[n]) - \mathsf{Cost}(r[n])| \leqslant 2\eta\, n\,.$$

For $n = 0$, nothing has to be proved since $\mathsf{Cost}(\tilde{r}[0]) = \mathsf{Cost}(r[0]) = 0$.

Suppose that the property holds until step $n < \mathsf{Length}(\tilde{r}) = \mathsf{Length}(r)$. We now prove it for index $n+1$. Let (ℓ,ν) and (ℓ, I) be the last states of $r[n]$ and $\tilde{r}[n]$, respectively. We denote by t the delay taken in r at step n, and J the choice of successor in $\tilde{r}$ at step n. Since the same action occurs at step n, we have

$$\mathsf{Cost}(\tilde{r}[n+1]) - \mathsf{Cost}(r[n+1]) = \mathsf{Cost}(\tilde{r}[n]) - \mathsf{Cost}(r[n]) + \omega(\ell)(d(I,J) - t)\,.$$

We clearly have $|d(I,J) - t| \leqslant 2\eta$ so that, by induction hypothesis,

$$|\mathsf{Cost}(\tilde{r}[n+1]) - \mathsf{Cost}(r[n+1])| \leqslant 2\eta\, n + 2\eta = 2\eta\,(n+1)\,.$$

Hence, the property is proved by induction. □

By Lemma 8, we have

$$\begin{aligned}
\mathsf{Cost}(\mathsf{Play}((\ell,M_k),\sigma_1^{(\varepsilon)},\sigma_2)) &\leqslant \mathsf{Cost}(\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2)) + \\
&\qquad\qquad 2\eta\,\mathsf{Length}(\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2)) \\
&\leqslant \mathsf{Cost}(\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2)) + 2\eta\, L \\
&\leqslant \mathsf{Cost}(\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2^*)) + \varepsilon
\end{aligned}$$

where the last inequality comes from the definition of η, and ξ_2^* is the optimal strategy for player 2 in $\tilde{\mathcal{A}}$. In particular, notice that this shows that $\mathsf{Cost}(\mathsf{Play}((\ell,M_k),\sigma_1^{(\varepsilon)},\sigma_2))$, and hence $\overline{\mathsf{UVal}}_{\mathcal{A}}^\eta((\ell,M_k))$, is less than $+\infty$.

We then show inequality $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell,\{M_k\})) \leqslant \underline{\mathsf{CVal}}_{\mathcal{A}}^\eta((\ell,M_k)) + \varepsilon$. In case $\underline{\mathsf{CVal}}_{\mathcal{A}}^\eta((\ell,M_k)) = +\infty$, the inequality is trivially verified. For all $\varepsilon > 0$, we let η be a positive rational number less than $\varepsilon/3$ and we explain how to construct (from ξ_2^*) an η-convergent strategy $\sigma_2^{(\varepsilon)}$ of $\mathcal{A}$ so that

$$\mathsf{Cost}(\mathsf{Play}((\ell,\{M_k\}),\xi_1^*,\xi_2^*)) - \varepsilon \leqslant \inf_{\sigma_1 \in \mathsf{CStrat}_1^\eta(\mathcal{A})} \mathsf{Cost}(\mathsf{Play}((\ell,M_k),\sigma_1,\sigma_2^{(\varepsilon)}))\,. \quad (6)$$

The construction of $\sigma_2^{(\varepsilon)}$ is inspired from the previous case, but special care has to be made to prevent player 1 from decreasing its cost in $\mathcal{A}$ by accumulating errors made by player 2. Once again, since $\sigma_2^{(\varepsilon)}$ will be evaluated against an η-convergent play of player 1, we only define it on finite plays r that are η-convergent. Let (ℓ',ν) be the last configuration of r, n its length and $I \in \mathcal{I}^\eta_{\mathcal{A}}$ the interval containing ν. Let $(J,a) = \xi_2^*(\ell', I)$ (remember that ξ_2^* is memoryless so that only the last configuration of the play is useful). Once again, we necessarily have $I \preceq J$. In case J is a singleton $\{M_k\}$, we let $\sigma_2^{(\varepsilon)}(r) = (M_k - \nu, a)$. In case $J = [M_k - \eta, M_k)$, we let $\sigma_2^{(\varepsilon)}(r) = (\max(M_k - \eta/2^{n+1} - \nu, 0), a)$. In case $J = (M_k, M_k + \eta]$, we let $\sigma_2^{(\varepsilon)}(r) = (\max(M_k + \eta/2^{n+1} - \nu, 0), a)$. The use of a maximum operator to define the delay in the two last cases permits to delay 0 in case ν is either too close or too far from M_k to go exactly at distance $\eta/2^{n+1}$ from M_k.

Then, to prove inequality (6), we consider any η-convergent strategy σ_1 of player 1. As in the previous case, we can construct from σ_1 a strategy $\xi_1 \in \mathsf{Strat}_1(\tilde{\mathcal{A}})$ verifying that $\mathsf{Play}((\ell, \{M_k\}), \xi_1, \xi_2^*)$ is the sequence of η-regions visited by the play $\mathsf{Play}((\ell, M_k), \sigma_1, \sigma_2^{(\varepsilon)})$. The following lemma permits to compare the weight of these two plays.

Lemma 9. *Let $r = (\ell_0,\nu_0),(t_0,a_0),\ldots,(\ell_{i+1},\nu_{i+1}),\ldots$ be a play of $\mathcal{A}$, such that $\nu_0 = M_{k'}$ for some k', and $\tilde r$ its abstraction in terms of η-regions. Suppose that r is η-convergent. Then,*

$$|\mathsf{Cost}(\tilde r) - \mathsf{Cost}(r)| \leqslant 3\eta\,.$$

Proof. We define the *latency* λ_n of an index $n \leqslant \mathsf{Length}(\tilde r) = \mathsf{Length}(r)$ as the greatest index $0 < m \leqslant n$ such that $t_{m-1} \neq 0$ or the clock has been reset by action a_{m-1} in location ℓ_{m-1}, or 0 if such an index m does not exist. Notice that if $n < n'$ and $\lambda_n = \lambda_{n'}$, then we have $t_m = 0$ for every $m \in \{n,\ldots,n'-1\}$, and $\nu_n = \nu_{n'}$. We also know that ν_{λ_n} is $\eta/2^{\lambda_n+1}$-close to a border: indeed, either $t_{\lambda_n-1} \neq 0$ (which permits to conclude by definition of η-convergent plays), or the clock has been reset by action a_{λ_n-1}, in which case $\nu_{\lambda_n} = 0$.

We then prove by induction on $0 \leqslant n \leqslant \mathsf{Length}(\tilde r) = \mathsf{Length}(r)$ that

$$|\mathsf{Cost}(\tilde r[n]) - \mathsf{Cost}(r[n])| \leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_n}}\right)\,.$$

The property is trivially verified for $n = 0$, since $\lambda_n = 0$, and $\mathsf{Cost}(\tilde r[n]) = \mathsf{Cost}(r[n]) = 0$.

Suppose now that the property holds for $n < \mathsf{Length}(\tilde r) = \mathsf{Length}(r)$. We now prove it for $n+1$. We split our study with respect to the value of λ_{n+1}.

- If $\lambda_{n+1} = n+1$, then we know that $t_n \neq 0$ or that the clock has just been reset. Denote by I the η-region containing ν_n and J the η-region containing $\nu_n + t_n$.

- If $t_n = 0$, then $d(I,J) = 0$, and

$$\begin{aligned}|\mathsf{Cost}(\tilde{r}[n+1]) - \mathsf{Cost}(r[n+1])| &= |\mathsf{Cost}(\tilde{r}[n]) - \mathsf{Cost}(r[n])| \\ &\leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_n}}\right) \quad \text{(Ind. Hyp.)} \\ &\leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_{n+1}}}\right)\end{aligned}$$

since $\lambda_{n+1} = n+1 > n \geqslant \lambda_n$.

- Otherwise, $t_n \neq 0$ so that, $\nu_n + t_n$ is $\eta/2^{n+1}$-close to a border (since the play is supposed to be η-convergent). By reasoning on the transition from step $\lambda_n - 1$ to step λ_n, we also know that ν_n is $\eta/2^{\lambda_n}$-close to a border. By definition of $d(I,J)$, we obtain $|d(I,J) - t_n| \leqslant \eta(1/2^{n+1} + 1/2^{\lambda_n})$. In the overall, we get

$$\begin{aligned}&|\mathsf{Cost}(\tilde{r}[n+1]) - \mathsf{Cost}(r[n+1])| \\ &\qquad \leqslant |\mathsf{Cost}(\tilde{r}[n]) - \mathsf{Cost}(r[n])| + |\omega(\ell_n)(d(I,J) - t_n)| \\ &\qquad \leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_n}}\right) + \eta\left(\frac{1}{2^{n+1}} + \frac{1}{2^{\lambda_n}}\right) \\ &\qquad = \eta\left(3 - \frac{1}{2^{\lambda_n - 1}} + \frac{1}{2^{n+1}}\right) \\ &\qquad \leqslant \eta\left(3 - \frac{3}{2^{n+1}}\right) = 3\eta\left(1 - \frac{1}{2^{\lambda_{n+1}}}\right)\end{aligned}$$

the last inequality coming from the fact that $\lambda_n \leqslant n$, so that $1/2^{n-1} \leqslant 1/2^{\lambda_n - 1}$.

– If $\lambda_{n+1} < n$, then we know that $\lambda_{n+1} = \lambda_n$ and $t_n = 0$, so that

$$\begin{aligned}|\mathsf{Cost}(\tilde{r}[n+1]) - \mathsf{Cost}(r[n+1])| &= |\mathsf{Cost}(\tilde{r}[n]) - \mathsf{Cost}(r[n])| \\ &\leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_n}}\right) \quad \text{(Ind. Hyp.)} \\ &\leqslant 3\eta\left(1 - \frac{1}{2^{\lambda_{n+1}}}\right).\end{aligned}$$

The property is proved by induction, and we conclude as in Lemma 8. □

Since $\mathsf{Play}((\ell, M_k), \sigma_1, \sigma_2^{(\varepsilon)})$ is an η-convergent play by construction, this lemma permits to conclude that

$$\mathsf{Cost}(\mathsf{Play}((\ell, \{M_k\}), \xi_1, \xi_2^*)) - \varepsilon \leqslant \mathsf{Cost}(\mathsf{Play}((\ell, M_k), \sigma_1, \sigma_2^{(\varepsilon)}))\,.$$

Once again, notice that this shows that $\mathsf{Cost}(\mathsf{Play}((\ell, M_k), \sigma_1, \sigma_2^{(\varepsilon)}))$, and hence also $\underline{\mathsf{CVal}}^\eta_{\mathcal{A}}((\ell, M_k))$, is greater than $-\infty$.

B.4 Proof of the infinite case of Corollary 1

If $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) = -\infty$, then it is sufficient to prove that $\overline{\mathsf{Val}}_{\mathcal{A}}((\ell, M_k)) = -\infty$. To do so, let $M \in \mathbb{R}$ and ξ_1 a strategy of player 1 such that $\mathsf{Cost}((\ell, \{M_k\}), \xi_1) \leqslant M$. From the construction in [14], we know that we can choose ξ_1 with the following structure: the strategy follows the mean-payoff strategy, storing in its memory the cost accumulated so far, and stops the mean-payoff execution to reach a target state whenever the cost is low enough. In particular, the length of this computation is bounded by a linear function over M of the shape CM with C only depending on $\mathcal{A}$. Following the same reconstruction as the one given in Lemma 3, with $\eta \leqslant \varepsilon/(CM \max|\pi|)$, we can map strategy ξ_1 into an η-region-uniform strategy $\sigma_1^{(\varepsilon)}$ of $\mathcal{A}$. In the overall, whatever strategy σ_2 of player 2, we can build ξ_2 in $\tilde{\mathcal{A}}$ that mimicks the play following the profile (σ_1, σ_2) of strategies in $\mathcal{A}$. However, the length of this play will necessarily be bounded by CM, so that the difference of weight between the two plays is bounded by ε. This suffices to prove that $\overline{\mathsf{UVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) \leqslant M$. By Lemma 1, this shows that $\overline{\mathsf{Val}}_{\mathcal{A}}((\ell, M_k)) \leqslant M$. Since this holds for every M, we conclude that $\overline{\mathsf{Val}}_{\mathcal{A}}((\ell, M_k)) = -\infty$.

Finally, if $\mathsf{Val}_{\tilde{\mathcal{A}}}((\ell, \{M_k\})) = +\infty$, then we prove that $\underline{\mathsf{CVal}}^{\eta}_{\mathcal{A}}((\ell, M_k)) = +\infty$, that is sufficient for the result by Lemma 2. We will indeed show that player 2 has a strategy σ_2 to ensure a value $+\infty$, i.e., ensuring that player 1 cannot reach the target set of states. We again use our knowledge on the strategy ξ_2 ensuring a cost $+\infty$ in $\tilde{\mathcal{A}}$: following the construction of Theorem 3, we know that a memoryless strategy is enough. Similarly to the previous cases, we can reconstruct a strategy σ_2 in $\mathcal{A}$. In particular, it automatically ensure that the target set of states is not reachable by any strategy of player 1, which permits to conclude easily.

C Detailed undecidability proofs

C.1 Counter machines

A two-counter machine M is a tuple (L, C) where $L = \{\ell_0, \ell_1, \ldots, \ell_n\}$ is the set of instructions—including a distinguished terminal instruction ℓ_n called HALT—and $C = \{c_1, c_2\}$ is the set of two *counters*. The instructions L are one of the following types:

1. (increment c) $\ell_i : c := c + 1$; goto ℓ_k,
2. (decrement c) $\ell_i : c := c - 1$; goto ℓ_k,
3. (zero-check c) ℓ_i : if $(c > 0)$ then goto ℓ_k else goto ℓ_m,
4. (Halt) ℓ_n : HALT.

where $c \in C$, $\ell_i, \ell_k, \ell_m \in L$. A configuration of a two-counter machine is a tuple (l, c, d) where $l \in L$ is an instruction, and c, d are natural numbers that specify the value of counters c_1 and c_2, respectively. The initial configuration is $(\ell_0, 0, 0)$. A run of a two-counter machine is a (finite or infinite) sequence of configurations $\langle k_0, k_1, \ldots \rangle$ where k_0 is the initial configuration, and the relation between subsequent configurations is governed by transitions between respective instructions. The run is a finite sequence if and only if the last configuration is the terminal instruction ℓ_n. Note that a two-counter machine has exactly one run starting from the initial configuration. The *halting problem* for a two-counter machine asks whether its unique run ends at the terminal instruction ℓ_n. It is well known ([18]) that the halting problem for two-counter machines is undecidable.

C.2 Constrained-price reachability

Theorem 4. *Deciding the existence of a strategy for constrained-reachability objective* $\mathsf{Reach}(\bowtie 1)$ *with* $\bowtie \in \{\leqslant, <, =, >, \geqslant\}$ *is undecidable for PTGs with two clocks or more.*

Proof. We prove that the existence of a strategy for constrained-reachability objective $\mathsf{Reach}(=1)$ is undecidable. The proofs for other objectives follow a similar approach: we outline the changes at the end of this proof. In order to obtain the undecidability result, we use a reduction from the halting problem for two counter machines. Our reduction uses a PTG with arbitrary price-rates, and zero prices on labels, along with two clocks x_1, x_2.

We specify a module for each instruction of the counter machine. On entry into a module for increment/decrement/zero check, we always have that $x_1 = \frac{1}{5^{c_1}7^{c_2}}$ and $x_2 = 0$ where c_1 (resp. c_2) is the value of counter C_1 (resp. C_2). Given a two counter machine, we construct a PTG $\mathcal{A}$ whose building blocks are the modules for instructions. The role of Player 1 will be to simulate faithfully the machine by choosing appropriate delays to adjust the clocks to reflect changes in counter values. Player 2 will have the opportunity to verify that Player 1 did not cheat while simulating the machine. We shall now present the modules for decrement, increment and zero-check instructions of the two counter machine.

Simulate decrement instruction. Fig. 2 gives the complete module for the instruction to decrement C_1. Let us denote by $x_{old} = \frac{1}{5^{c_1}7^{c_2}}$ the value of x_1 while entering the module. At the location ℓ_{k+1} of the module, $x_1 = x_{new}$ should be $5x_{old}$ to correctly decrement counter C_1.

At location ℓ_k, Player 1 spends a nondeterministic amount of time $k = x_{new} - x_{old}$ such that $x_{new} = 5x_{old} + \varepsilon$. To correctly decrement C_1, ε should be 0, and k must be $\frac{4}{5^{c_1}7^{c_2}}$. At location Check, Player 2 could choose to go to *Go* (in order to continue the simulation of the machine) or to go to the widget WD_1, if he suspects that $\varepsilon \neq 0$. If Player 2 spends t time in the location Check before proceeding to ℓ_{k+1}, then Player 1 can enter the location Abort from *Go* (to abort the simulation), spend $1 + t$ time in location Abort and reach a target T_1 with cost $= 1$ (and thus achieve his objective). However, if $t = 0$ then entering location Abort will make the cost to be > 1 (which is losing for Player 1). In this case, player 1 will prefer entering ℓ_{k+1} from *Go*.

If player 2 spends t time in location Check, and enters widget WD_1, then the cost upon reaching the target in the widget WD_1 is $1 + \varepsilon$ which is 1 iff $\varepsilon = 0$ (see Table 1).

Let us summarize the construction. Let us assume that when entering ℓ_k (see Fig. 2) the value of x_1 (resp. x_2) is $\frac{1}{5^{c_1}7^{c_2}}$ (resp. 0). First, let us consider

Table 1. Cost Incurred in WD_1

Location $\rightarrow$	L	M	N	T
x_1 while entering	$x_{new}+t$ delay in Check is t	0		
x_2 while entering	$x_{new}-x_{old}+t$	$1-x_{old}$	0	1
Time elapsed at	$1-x_{new}-t$	x_{old}	1	-
Cost incurred at	$-1+x_{new}+t$	$-5x_{old}$	2	-
Total cost upon reaching target				cost of $-t$ due to delay in Check and $x_{new}=5x_{old}+\varepsilon$ gives a total cost $-t+1+x_{new}+t-5x_{old}=1+\varepsilon=1$ iff $\varepsilon=0$

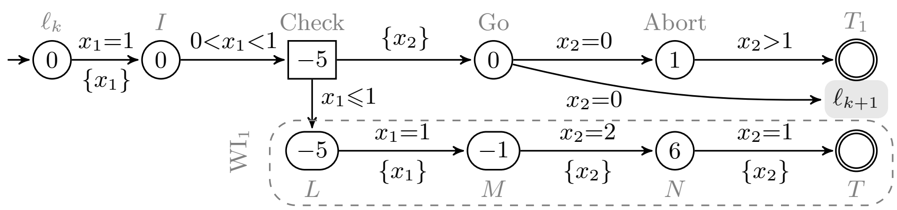


Fig. 9. Reach$(=1)$: Widget to simulate increment counter C_1 instructions

the case where Player 1 simulates correctly the machine (i.e., if Player 1 spends $\frac{4}{5^{c_1}7^{c_2}}$ time units in ℓ_k). In this case, Player 2 has three possibilities: (i) either Player 2 goes immediately to *Go*, in this case Player 1 has no incentive to go to the location Abort and thus the simulation of the machines goes on; (ii) or Player 2 goes to *Go* after delaying some time $t > 0$ in the location Check, in this case Player 1 will go to the location Abort in order to immediately achieve his objective; (iii) or Player 2 goes to the widget WD_1, in this case, as Player 1 has correctly simulated the machine, Player 1 achieves his objective. Let us now turn to the case where Player 1 does not simulate the machine faithfully. In this case, Player 2 has the possibility to reach the target location with a cost different from 1 by using the widget WD_1. In conclusion, if Player 1 simulates the machine properly either the simulation continues or he achieves his objective immediately; and if Player 1 does not simulate the machine properly, then Player 2 has a strategy to reach a target location with a cost different from 1.

Simulate Increment instruction. Fig. 9 gives the complete module for increment instruction. The construction is similar to that as decrement.

Simulate Zero-check instruction. The module for the Zero check instruction (ℓ_k : if $C_2 = 0$ goto ℓ_{k+1} else goto ℓ_{k+2}) is depicted in Fig. 10 and its widgets are in Fig. 11 and 12.

Player 1 guesses whether C_2 is zero or not and Player 2 has the possibility to check if the guess of Player 1 is correct by entering the corresponding widget.

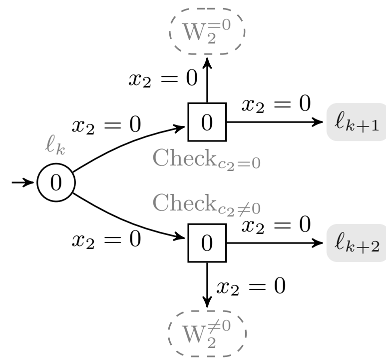


Fig. 10. Widget WZ_2 simulating zero-check for C_2

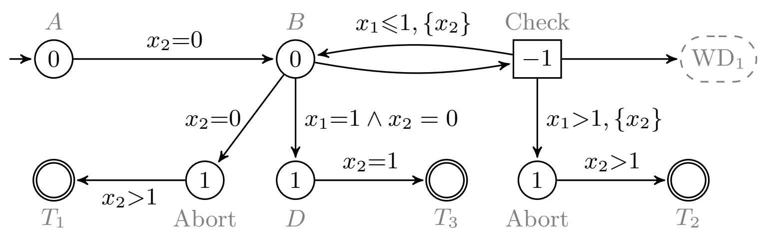


Fig. 11. Widget $\mathrm{W}_2^{=0}$. Note that if the delay in Check is > 0 then Player 1 will go to Abort and reach a target with cost $= 1$. Widget WD_1 is the same as in the decrement module. Delay at B is $k = \frac{4\times 5^i}{5^{c_1}7^{c_2}} + \varepsilon$. If $\varepsilon \neq 0$ then Player 2 will enter the widget WD_1 and the target in WD_1 is reached with cost $\neq 1$.

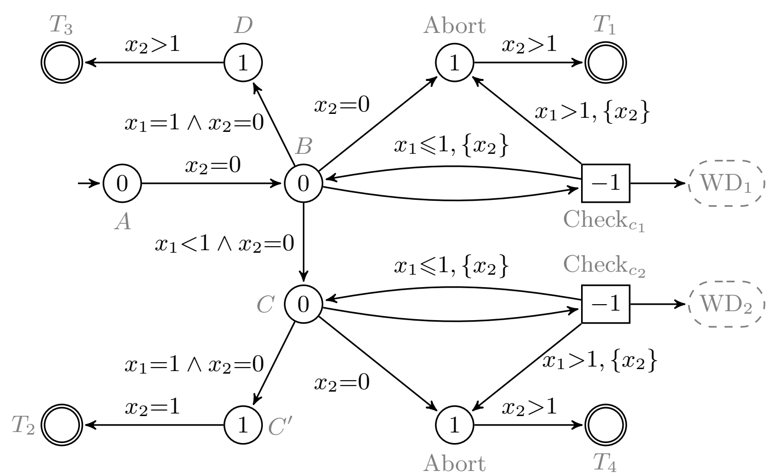


Fig. 12. Widget $\mathrm{W}_2^{\neq 0}$. Widget WD_2 is similar to WD_1 in the decrement module shown earlier, except that the prices are adjusted to verify decrement of counter C_2 instead of C_1.

In the widget depicted in Fig. 11, in order to verify if C_2 is indeed zero, C_1 is repeatedly decremented (using a construction similar to Decrement module in Fig. 2) until C_1 becomes 0. At that point, if x_1 is not 1 then clearly C_2 was nonzero, and Player 1 has made an error in guessing. In this case, the total cost incurred will be greater than 1. However, if indeed C_2 was zero, then the total cost incurred is 1 for the following reasons:

- If some time has elapsed at location Check, and still x_1 is less than 1, Player 1 can go to Abort from location B. However, if time has elapsed at Check and $x_1 > 1$, the only option is to go to Abort from Check; then Abort goes to T_2. If Player 1 cannot control x_1 not exceeding 1, then in that case, Player 2 will spend 0 units of time in Check, and then Player 1 will pay a cost > 1 on reaching T_2.

In a similar manner, if Player 1 guesses that $C_2 \neq 0$, then Player 2 verifies it by entering the widget in Fig. 12. In this case, C_1 is first decremented (the decrement module may be called several times) until it reaches 0, and then C_2 is decremented until it also reaches 0 (the decrement module is called at least once):

- Entry to C happens with $x_1 < 1$ and $x_2 = 0$. To get to C', some time elapse is needed at C. However, then $x_2 > 0$, so directly there is no access to C', without going to Check_{c_2} (ensuring that the decrement module for C_2 is taken at least once). To not get punished through Check_{c_2}, Player 1 has to elapse a time of the form $\frac{6}{5^i 7^j}$: hence, if $C_2 = 0$, this time elapse must be $\frac{6}{5^i}$, if C was entered with $x_1 = \frac{1}{5^i}$. Due to $C_2 = 0$, at some point, $\frac{6}{5^i}$ will exceed 1; at that time, Player 2 will spend 0 unit of time in Check, and go to Abort. That will punish Player 1.

Note that Player 2 can not delay in the zero check module due to clock constraints. In the widgets $\mathrm{W}_1^{=0}$ and $\mathrm{W}_1^{\neq 0}$ however, Player 2 can delay at the Check locations. As in the decrement module, we offer to Player 1 the option to abort (via the Abort location) : if t units of time was spent at location Check, then the cost incurred will be $1 + p - t$ where $p > 0$ is chosen by Player 1. Table 2 shows the cost incurred in the widget $\mathrm{W}_2^{\neq 0}$ when Player 1 guesses $C_2 \neq 0$.

To summarize the zero check instruction for C_2: assume that at location ℓ_k, we enter with $x_1 = \frac{1}{5^i}$ (hence $C_2 = 0$) and $x_2 = 0$. First let us consider the case when Player 1 guesses correctly that C_2 is zero: in this case, Player 2 has 2 possibilities: (i) continue with the next instruction, (ii) simulate the widget $\mathrm{W}_2^{=0}$. In both cases, no time elapse is possible at location Check_{c_2}. Player 1 achieves his objective even on entering widget $\mathrm{W}_2^{=0}$ due to his correct guess. Let us now turn to the case when Player 1 made a wrong guess. Then, Player 2 has the capability to invoke widget $\mathrm{W}_2^{\neq 0}$ and in this case, Player 1 will incur a cost different from 1. A similar analysis applies to the case when we start in ℓ_k with $x_1 = \frac{1}{5^i 7^j}$.

Correctness of the construction. On entry into the location ℓ_n (encoding the HALT instruction of the two-counter machine), we reset clock x_1 to 0; ℓ_n has cost 1, and the edge coming out of ℓ_n goes to a Goal location, with guard x_1=1.

Table 2. Cost incurred in widget $W_2^{\neq 0}$

Location →	A	B	D	T_3	C	C'	T_2
x_1 while entering	$\frac{1}{5^{c_1}7^{c_2}}$	$\frac{5^i}{5^{c_1}7^{c_2}}$ loop taken i times	1 $\frac{5^i}{5^{c_1}7^{c_2}} = 1$ $\Rightarrow i = c_1 \wedge$ $c_2 = 0$		$\frac{7^j}{7^{c_2}}$ loop taken j times	1 $\frac{5^i 7^j}{5^{c_1}7^{c_2}} = 1$ $\Rightarrow i = c_1 \wedge$ $j = c_2 > 0$	
x_2 while entering	0	0	0	> 1	0	0	1
Time elapsed at	0	$\frac{4\times 5^i}{5^{c_1}7^{c_2}} + \varepsilon$	> 1	0	$\frac{6\times 7^j}{7^{c_2}} + \varepsilon$	1	0
Cost incurred at	0	0	> 1	0	0	1	
Total cost				> 1			1

1. Assume that the two counter machine halts. If Player 1 simulates all the instructions correctly, he will incur a cost 1, by either reaching the goal location after ℓ_n, or by entering a widget (the second case only occurs if Player 2 decides to check whether Player 1 simulates the machine faithfully. If Player 1 makes an error in his computation, Player 2 can always enter an appropriate widget, making the cost different from 1. In summary, if the two counter machine halts, Player 1 has a strategy to achieve his goal (i.e., reaching a target location with a cost equal to 1).
2. Assume that the two counter machine does not halt.
   - If Player 1 simulates all the instructions correctly, and if Player 2 never enters a widget, then Player 1 incurs cost ∞ as the path never reaches a target.
   - Suppose now that Player 1 makes an error. In this case, Player 2 always has the capability to reach a target set with a cost different from 1.

   In summary, if the two counter machine does not halt, Player 1 does not have a strategy to achieve his goal.

Thus, Player 1 incurs a cost 1 iff he chooses the strategy of faithfully simulating the two counter machine, when the machine halts. When the machine does not halt, the cost incurred by Player 1 is different from 1 if Player 1 made a simulation error and Player 2 entered a widget. Else if a widget is not entered then the run does not end and cost is $+\infty$. □

We briefly discuss the difference with the proofs for other constrained-price reachability objectives.

- $\mathsf{Reach}(<1)$. We use 2 clocks x_1, x_2, and a module for each instruction of the two counter machine. On entry into a module for increment/decrement/zero check, $x_1 = \frac{1}{5^{c_1}7^{c_2}}$ and $x_2 = 0$.

  We discuss the case of decrementing counter C_1. Fig. 13 gives the complete module for the instruction to decrement C_1. To simulate the decrementation correctly, Player 1 has to elapse $\frac{4}{5^{c_1}7^{c_2}}$ units of time at ℓ_k. At the location Check, Player 2 can either continue to ℓ_{k+1} after elapsing $t \geqslant 0$ units of

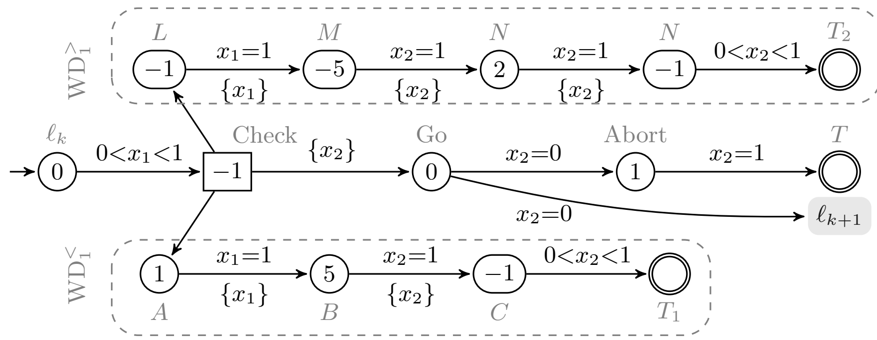


Fig. 13. $\mathsf{Reach}(<1)$: Simulation to decrement counter C_1

time, or enter a widget to check the choice of Player 1. Consider the case when Player 2 proceeds to ℓ_{k+1} after elapsing a time $t > 0$. This incurs a cost $-t$. Then Player 1 can go to location Abort, and spend 1 unit of time there, reaching a target with cost $1 - t < 1$. Assume now that Player 1 spends $\frac{4}{5^{c_1}7^{c_2}} + \varepsilon$ units of time in ℓ_k, and Player 1 spends $t > 0$ units of time at Check. Since Player 1 has elapsed more time than what he should have, Player 2 enters the widget $\mathrm{WD}_1^>$. The cost incurred so far is $-t$. On entry into $\mathrm{WD}_1^>$, we have $x_1 = \frac{5}{5^{c_1}7^{c_2}} + \varepsilon + t$, $x_2 = \frac{4}{5^{c_1}7^{c_2}} + \varepsilon + t$. Then, the total accumulated cost becomes $-1 + \frac{5}{5^{c_1}7^{c_2}} + \varepsilon$ on coming out of L, and then becomes $-1 + \varepsilon$ on coming out of M; this further becomes $1 + \varepsilon$ on entering O. At location O, time $0 < p < 1$ has to be spent, making the total cost to be $1 - p + \varepsilon$. Time p is chosen by Player 2. Thus, if $\varepsilon = 0$ (hence, Player 1 made no error), the cost incurred is less than 1; however, when $\varepsilon > 0$, p can always be chosen to be at most ε, thereby making the total cost at least 1.
A similar analysis can be done when Player 1 incurs a delay $\frac{4}{5^{c_1}7^{c_2}} - \varepsilon$, $\varepsilon > 0$ at location ℓ_k. In this case, Player 2 enters widget $\mathrm{WD}_1^<$.
The increment and zero-check instructions are obtained by a similar approach.

– $\mathsf{Reach}(\leqslant 1)$. We use 2 clocks x_1, x_2, and a module for each instruction of the two counter machine. On entry into a module for increment/decrement/zero check, $x_1 = \frac{1}{5^{c_1}7^{c_2}}$ and $x_2 = 0$ where $c_1(c_2)$ is the value of counter $C_1(C_2)$.
We discuss the case of decrementing counter C_1. Fig. 14 gives the complete module for the instruction to decrement C_1. As in the case of the objective $\mathsf{Reach}(<1)$, Player 1 has to elapse a time $\frac{4}{5^{c_1}7^{c_2}}$ at ℓ_k. If Player 2 elapses $t > 0$ units of time in Check and proceeds to ℓ_{k+1}, then Player 1 has the option to go to Abort, and reach a target with a total cost of $-t + 1 + k$, where $1 + k$ is the time elapsed at Abort. k can be chosen by Player 1 so that $1 + k - t \leqslant 1$.
Let us consider the case when Player 1 spends $\frac{4}{5^{c_1}7^{c_2}} - \varepsilon$ units of time in ℓ_k, with $\varepsilon \geqslant 0$ and Player 1 spends $t > 0$ units of time at Check. Player 2 then will proceed to the widget $\mathrm{WD}_1^<$. On entry into $\mathrm{WD}_1^<$, we have $x_1 =$

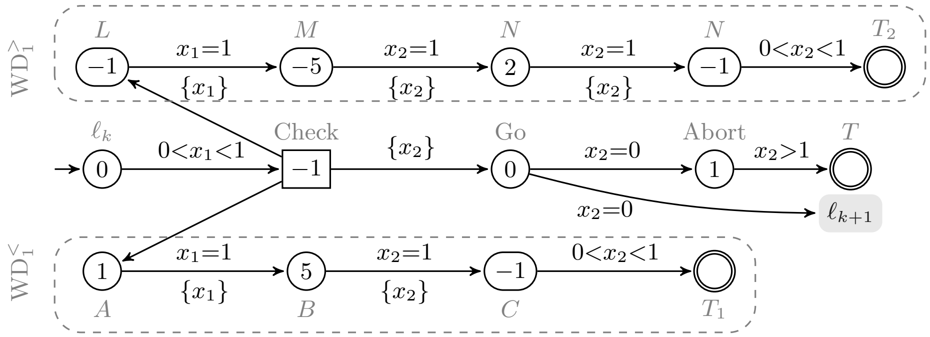


Fig. 14. Reach($\leqslant$1): Simulation to decrement counter C_1

$\frac{5}{5^{c_1}7^{c_2}} - \varepsilon + t$, $x_2 = \frac{4}{5^{c_1}7^{c_2}} - \varepsilon + t$, and an incurred cost of $-t$. On entering M, we have an accumulated cost of $1 - \frac{5}{5^{c_1}7^{c_2}} + \varepsilon - 2t$, and further on entering N, the accumulated cost becomes $1 + \varepsilon - 2t$. Finally, when T is entered, the total cost is $1 + \varepsilon - 2t - p$, where $0 < p < 1$ is a delay chosen by Player 2. Clearly, $1 + \varepsilon - 2t - p < 1$ if $\varepsilon = 0$. However, if $\varepsilon \neq 0$, Player 2 can adjust the values of p, t in such a way that $2t + p < \varepsilon$, there by making the total cost > 1.

A similar analysis can be done when Player 1 incurs a delay $\frac{4}{5^{c_1}7^{c_2}} + \varepsilon$, $\varepsilon > 0$ at location ℓ_k. In this case, Player 2 enters widget $\mathrm{WD}_1^{>}$.

The increment and zero-check instructions are obtained by a similar approach.

– Reach($>$1). We use 2 clocks x_1, x_2, and a module for each instruction of the two counter machine. On entry into a module for increment/decrement/zero check, $x_1 = \frac{1}{5^{c_1}7^{c_2}}$ and $x_2 = 0$ where $c_1(c_2)$ is the value of counter $C_1(C_2)$.

  We discuss the case of decrementing counter C_1. Fig. 15 gives the complete module for the instruction to decrement C_1. As in the previous two objectives, Player 1 has to spend a time $\frac{4}{5^{c_1}7^{c_2}}$ at ℓ_k. If Player 2 elapses $t > 0$ units of time in Check and proceeds to ℓ_{k+1}, then Player 1 has the option to go to Abort, and reach a target with a total cost of $t + k$, where $k < 1$ is the time elapsed at Abort by Player 1. k can be chosen by Player 1 so that $t + k > 1$.

  Let us consider the case when Player 1 spends $\frac{4}{5^{c_1}7^{c_2}} - \varepsilon$ units of time in ℓ_k, with $\varepsilon \geqslant 0$ and Player 1 spends $t > 0$ units of time at Check. Player 2 then will proceed to the widget $\mathrm{WD}_1^{<}$. On entry into $\mathrm{WD}_1^{<}$, we have $x_1 = \frac{5}{5^{c_1}7^{c_2}} - \varepsilon + t$, $x_2 = \frac{4}{5^{c_1}7^{c_2}} - \varepsilon + t$, and an incurred cost of t. On entering M, we have an accumulated cost of $-1 + \frac{5}{5^{c_1}7^{c_2}} - \varepsilon + 2t$, and further on entering N, the accumulated cost becomes $-1 - \varepsilon + 2t$. On entering O, the accumulated cost becomes $1 - \varepsilon + 2t$, and finally, on entering T, the total cost is $1 - \varepsilon + 2t + p$, where $0 < p < 1$ is a delay chosen by Player 2. Clearly, $1 - \varepsilon + 2t + p > 1$ if $\varepsilon = 0$. However, if $\varepsilon \neq 0$, Player 2 can adjust the values of p, t in such a way that $2t + p \leqslant \varepsilon$, there by making the total cost $\leqslant 1$.

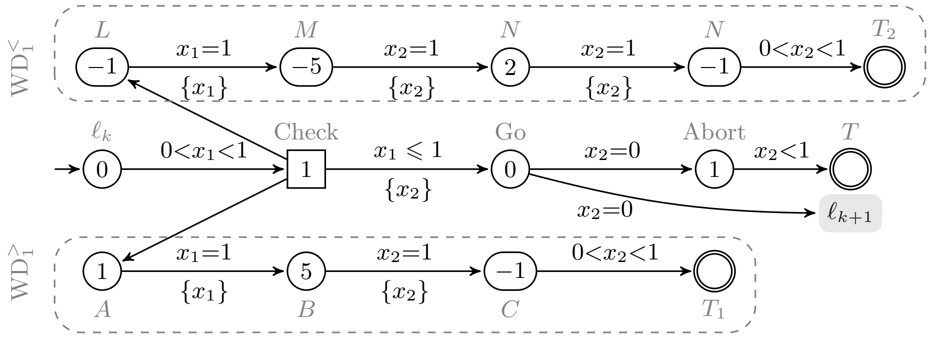


Fig. 15. Reach(> 1) : Simulation to Decrement Counter C_1

– Reach($\geqslant$1). It can be seen that a proof along similar lines as in the case of Reach($>$1) can be given.

C.3 Bounded-time reachability objective

Lemma 10. *The existence of a strategy for bounded-time reachability objective* TBReach$(18, 40)$ *is undecidable for PTGs with price-rates taken from* $\{0, 1\}$ *and 5 clocks or more.*

Proof. We prove that the existence of a strategy for bounded-time reachability objective ensuring a cost at most 18 within 40 time units of total elapsed time is undecidable. In order to obtain the undecidability result, we use a reduction from the halting problem for two counter machines. Our reduction uses PTGs with price-rates in $\{0, 1\}$, zero prices of labels, and 6 clocks.

We specify a module for each instruction of the two counter machine. On entry into a module for the $(k+1)$th instruction, we have one of the clocks x_1, x_2 having the value as $\frac{1}{2^{k+c_1}3^{k+c_2}}$, and the other one having value 0. Values c_1 and c_2 represent the values of the two counters, after simulation of k instructions. A clock z keeps track of the total time elapsed during simulation of an instruction: we always have $z = 1 - \frac{1}{2^k}$ at the end of simulating k instructions. Thus, a time of $\frac{1}{2}$ is spent simulating the first instruction, a time of $\frac{1}{4}$ is spent simulating the second instruction and so on, so that the total time spent in simulating the main modules corresponding to increment/decrement/zero check is less than 1 at any point of time. Two clocks a and b are used for rough work, and for enforcing urgency in some locations.

Again, the role of Player 1 is to simulate the machine faithfully by choosing appropriate delays to adjust clock values in order to reflect the changes in counter values, and also to reflect the total time elapsed. Player 2 will have the opportunity to check if Player cheated while simulating the machine. We now present the modules for decrement, increment and zero-check instructions.

Simulate increment instruction. Fig. 16 gives the complete increment module with respect to counter C_1. Assume this is the $(k+1)$th instruction that we are simulating. Also, as mentioned earlier, one of x_1, x_2 will have the value of $\frac{1}{2^{k+c_1}3^{k+c_2}}$ on entry, while the other will be zero. Without loss of generality, assume $x_1 = \frac{1}{2^{k+c_1}3^{k+c_2}}$, while $a = b = x_2 = 0$, and $z = 1 - \frac{1}{2^k}$. At the end of the module we want $x_2 = \frac{1}{2^{k+1+c_1+1}3^{k+1+c_2}}$ and $x_1 = 0$ and $z = 1 - \frac{1}{2^{k+1}}$.

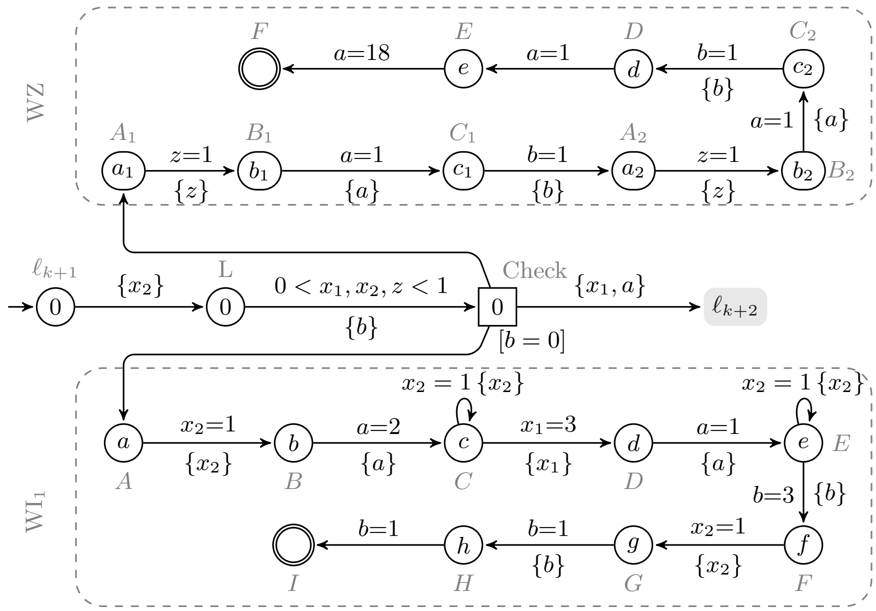
Let t_1 and t_2 be respectively the time spent at locations ℓ_{k+1} and L. We want to check that $t_1 + t_2 = \frac{1}{2^{k+1}}$, $t_2 = \frac{1}{2^{k+1+c_1+1}3^{k+1+c_2}}$. This will ensure that the clocks keep track of the total time elapsed, as well as the increment of C_1. Player 2 has two widgets at his disposal to check each of these:

1. WZ is a widget (that can be used by Player 2) in order to check that the total time elapsed in any module corresponding is correct. More precisely Player 2 has the opportunity to check (by means of the module WZ) that the execution of the module corresponding to the $(k+1)$th instruction takes exaclty $\frac{1}{2^{k+1}}$ time units (recall that this time is recorded in clock z).
2. WI_1 is a widget (that can be used by Player 2) in order to check that counter C_1 is indeed incremented properly.

Upon entering the location Check, the values of clocks are $a = t_1 + t_2$, $z = 1 - \frac{1}{2^k} + t_1 + t_2$, $x_1 = t_2$, $x_2 = \frac{1}{2^{k+c_1}3^{k+c_2}} + t_1 + t_2$ and $b = 0$.

- Widget WZ: The role of widget WZ is to check if the value of the clock z is $1 - \frac{1}{2^{k+1}}$ when the location Check is reached. The PTG corresponding to WZ is depicted in Fig. 16, and a cost analysis of WZ is presented in Table 3.[8] WZ is a general template for two widgets, based on the actual values of the price-rates. These two widgets are $\mathrm{WZ}^<$ and $\mathrm{WZ}^>$. In widget $\mathrm{WZ}^>$, price-rates $b_1 = c_1 = c_2 = e = 1$ and the other ones are zero, while in widget $\mathrm{WZ}^<$, price-rates $a_1 = d = e = 1$ and the other ones are zero.
  It can be seen that if $t_1 + t_2 < \frac{1}{2^{k+1}}$, then the total cost incurred in $\mathrm{WZ}^<$ is strictly more than 18; similarly, if $t_1 + t_2 > \frac{1}{2^{k+1}}$, then the total cost incurred in $\mathrm{WZ}^>$ is strictly more than 18. Player 2 can choose one these widgets appropriately. In case $t_1 + t_2 = \frac{1}{2^{k+1}}$, then the total cost incurred in either widget is exactly 18.
- Widget WI_1: The widget WI_1 in Fig. 16 ensures that upon entering the location Check, the value of the clock $x_2 = t_2 = \frac{1}{2^{k+1+c_1+1}3^{k+1+c_2}} = \frac{1}{12} \times \frac{1}{2^{k+c_1}3^{k+c_2}} = \frac{n}{12}$. This indeed accounts for increment of counter C_1, and also for reaching the end of $k + 1^{th}$ instruction while keeping the value of C_2 unchanged.
  The widget WI_1 in Fig. 16 is again a general template for two widgets $\mathrm{WI}_1^<$ and $\mathrm{WI}_1^>$, obtained by fixing the prices at various locations. The prices a, b, c, d, e, f are values > 0. It must be noted that we only need prices in $\{0, 1\}$ for each location; using general prices is a short hand notation for a longer path which will use only prices from $\{0, 1\}$. In widget $\mathrm{WI}_1^<$, (in the shorthand notation), we have prices $a = d = 1, f = 11, h = 6$ and the rest

[8] Notice that in the table, the line "time elapse" represents time elapsed in the current location, and not the time elapsed before reaching the current location.



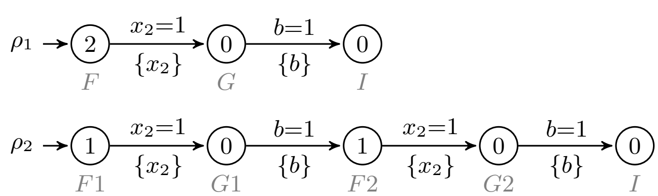


Fig. 16. $\mathsf{TBReach}(18, 40)$: Simulation of instruction : ℓ_{k+1}: increment C_1. WZ is a template for two widgets $\mathrm{WZ}^<$ and $\mathrm{WZ}^>$, based on the actual values of price-rate parameters. $\mathrm{WZ}^>$ has prices $b_1 = c_1 = c_2 = e = 1$ and rest are 0, while $\mathrm{WZ}^<$ has $a_1 = d = e = 1$ and rest are zero. Similarly, widget WI_1 is template for two widgets $\mathrm{WI}_1^<$ $(a = d = 1, f = 11, h = 6)$ and $\mathrm{WI}_1^>$ $(c = 1, g = 12, h = 17)$. Path ρ_1 is the shorthand notation with larger prices for longhand notation of path ρ_2 using prices 0,1 only.

Table 3. $\mathsf{TBReach}(18, 40)$: Cost Incurred in WZ: Total time elapsed= $19 - t < 20$

Location $\rightarrow$ $i \in \{1, 2\}$	A_i	B_i	C_i	D	E
z on entry	$1 - \frac{1}{2^k} + t_1 + t_2$ $= 1 - \frac{1}{2^k} + t$ $t = t_1 + t_2$	0	$1 - \frac{1}{2^k}$	-	-
a on entry	t	$\frac{1}{2^k}$	0	1	
b on entry	0	$\frac{1}{2^k} - t$	$1 - t$	0	-
time elapsed at	$\frac{1}{2^k} - t$	$1 - \frac{1}{2^k}$	t	$1 - t$	17
Widget $\mathrm{WZ}^{>}$ check $t > \frac{1}{2^{k+1}}$ prices $b_1, c_1, c_2, e : 1$ rest : 0					
cost incurred at	0	$1 - \frac{1}{2^k}$ at B_1	$2t$ t at C_1 t at C_2	0	17
Total Cost at target	-	-	-	-	$1 - \frac{1}{2^k} + 2t + 17$ $= 18\ if\ t = \frac{1}{2^{k+1}}$ $> 18\ if\ t > \frac{1}{2^{k+1}}$
Widget $\mathrm{WZ}^{<}$ check $t < \frac{1}{2^{k+1}}$ prices $a_1, d, e : 1$ rest : 0					
cost incurred at	$\frac{1}{2^k} - t$ at A_1	0	0	$1 - t$	17
Total Cost at target	-	-	-	-	$18 + \frac{1}{2^k} - 2t$ $= 18\ if\ t = \frac{1}{2^{k+1}}$ $> 18\ if\ t < \frac{1}{2^{k+1}}$

are zero. Likewise, in widget $\mathrm{WI}_1^{>}$, we have (in shorthand notation), prices $c = 1, g = 12, h = 17$ and the rest of the prices 0. Player 2 uses $\mathrm{WI}_1^{<}$ to check if $t_2 < \frac{n}{12}$, and uses $\mathrm{WI}_1^{>}$ to check if $t_2 > \frac{n}{12}$. Table 4 runs the reader through the widgets $\mathrm{WI}_1^{<}$ and $\mathrm{WI}_1^{>}$. While reading the table, keep in mind that $n = \frac{1}{2^{k+c_1}3^{k+c_2}}$ and that $t = t_1 + t_2$. As can be seen from the table, the total cost incurred is exactly 18 iff $t_2 = \frac{n}{12}$.

To summarize the simulation of the $(k+1)$th instruction, which is an increment C_1 instruction: assume we enter ℓ_{k+1} with values $x_1 = \frac{1}{2^{k+c_1}3^{k+c_2}}$, while $a = b = x_2 = 0$, and $z = 1 - \frac{1}{2^k}$. First let us consider the case when Player 1 correctly simulates the machine, respecting the time limit: then Player 1 spends a total of $\frac{1}{2^{k+1}}$ time across ℓ_{k+1} and L, and a time $\frac{1}{2^{k+1+c_1+1}3^{k+1+c_2}}$ at L. Player 2 has 3 possibilities : (i)either Player 2 directly goes to the next instruction ℓ_{k+2}

Table 4. $\mathsf{TBReach}(18, 40)$: Cost incurred in WI_1. Recall $t_1 + t_2 = t$. Also, total time elapsed in $\mathrm{WI}_1^{>}$ and $\mathrm{WI}_1^{<}$ in the long hand notation is $\leqslant 3 + 12 + 17 = 32$

Loc $\rightarrow$	A	B	C	D	E	F	G	H	I
x_1 on entry	$n+t$	$1+n+t_1$	$2+n$	0	-	-	-	-	-
x_2 on entry	t_2	0	$1-t_1$	$1-t_1-n$	$1-t_1$	t_2	0	-	-
a on entry	t	$1+t_1$	0	$1-n$	0	-	-	-	-
b on entry	0	$1-t_2$	$2-t$	$3-t-n$	$3-t$	0	$1-t_2$	0	1
time elapsed at	$1-t_2$	$1-t_1$	$1-n$	n	t	$1-t_2$	t_2	1	0
$\mathrm{WI}_1^{<}$ checks if $t_2 < \frac{n}{12}$ prices $a, d{=}1$ $f{=}11, h{=}6$ rest : 0									
cost incurred at	$1-t_2$	0	0	n	0	$11-11t_2$	0	6	-
Total cost at target	-	-	-	-	-	-	-	-	$18-12t_2+n$ $=18\ if\ t_2{=}\frac{n}{12}$ $>18\ if\ t_2{<}\frac{n}{12}$
$\mathrm{WI}_1^{>}$ checks if $t_2 > \frac{n}{12}$ prices $c{=}1, g{=}12,$ $h{=}17$ rest : 0									
cost incurred at	0	0	$1-n$	0	0	0	$12t_2$	17	-
Total cost at target	-	-	-	-	-	-	-	-	$18+12t_2-n$ $=18\ if\ t_2{=}\frac{n}{12}$ $>18\ if\ t_2{>}\frac{n}{12}$

and thus the simulation of the machine goes on, or (ii) Player 2 goes to one of the widgets $\mathrm{WZ}^{>}$ or $\mathrm{WZ}^{<}$; in this case, since Player 1 has spent the right amount of time in ℓ_{k+1} and L together, he achieves his objective, or (iii) Player 2 goes to one of the widgets $\mathrm{WI}_1^{>}$ or $\mathrm{WI}_1^{<}$; again, in this case, since Player 1 has spent the right amount of time at L, he achieves his objective. Let us now turn to the case when Player 1 does not spend the right amount of time in ℓ_k and L together; in this case, Player 2 can always enter one of the widgets $\mathrm{WZ}^{>}$ or $\mathrm{WZ}^{<}$ and reach a target with a cost > 18. Similarly, if Player 1 does not spend

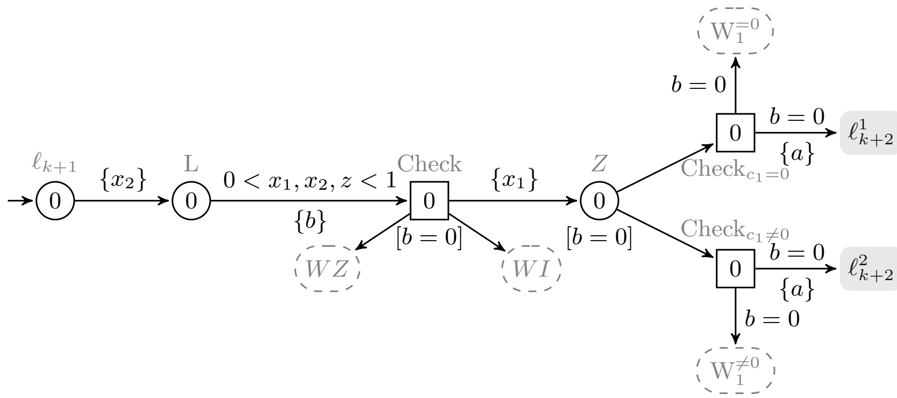


Fig. 17. $\mathsf{TBReach}(18, 40)$: Simulation of instruction zero check $C_1 = 0$. Widget WZ given in Fig. 16, and WI is similar to WI_1 differing only in prices. WI verifies if $x_2 = \frac{1}{6}\frac{1}{2^{k+c_1}3^{k+c_2}}$.

the right amount of time in L (and therefore did not increment C_1 properly), then again Player 2 has the possibility to reach a target with a cost > 18 using one of the widgets $\mathrm{WI}_1^{>}$ or $\mathrm{WI}_1^{<}$.

Simulate decrement instruction. The module to decrement counter C_1 is the same as the module to increment C_1 in Fig. 16. We replace only the widget WI_1 by the widget WD_1. This widget ensures that the time spent t_2 at L (and hence the value of x_2) is $\frac{1}{2^{k+1+c_1-1}3^{k+1+c_2}} = \frac{1}{2^{k+c_1}3^{k+1+c_2}} = \frac{1}{3}\frac{1}{2^{k+c_1}3^{k+c_2}}$. WD_1 can be obtained by a simple modification of the prices in WI_1.

Simulate Zero-check instruction. Consider ℓ_{k+1}: if $C_1 = 0$ then goto l^1_{k+2} else goto l^2_{k+2}. During the simulation of this instruction, we need to ensure that all the clocks are updated to account for reaching the end of $(k+1)$th instruction and the counter values remain unchanged.

The module for zero check is given in Fig. 17. At the location labelled Check, the widgets WZ and WI ensure that the clocks z and x_2 are updated to account for reaching the end of the $(k+1)$th instruction. This is similar to the increment module. If the values of clocks on entering ℓ_{k+1} are $z = 1 - \frac{1}{2^k}$, $x_1 = \frac{1}{2^{k+c_1}3^{k+c_2}}$ then upon entering Check they are $z = 1 - \frac{1}{2^{k+1}}$, $x_2 = \frac{1}{2^{k+1+c_1}3^{k+1+c_2}}$. No time elapses at the Check location. At location Z, no time elapses, and Player 1 guesses the value of counter C_1 and goes to either of the locations $\mathrm{Check}_{c_1=0}$ or $\mathrm{Check}_{c_1\neq0}$. Based on the choice of Player 1, Player 2 can go to one of the widgets $\mathrm{W}_1^{=0}$ (in Fig. 18) or $\mathrm{W}_1^{\neq0}$ (similar to $\mathrm{W}_1^{=0}$), if he suspects that Player 1 has made a wrong guess.

- Widget $\mathrm{W}_1^{=0}$ given in Fig. 18. We have $x_2 = \frac{1}{2^{k+1+c_1}3^{k+1+c_2}}$, on entering the L_1 of Widget $\mathrm{W}_1^{=0}$. To check if $c_1 = 0$, we first convert x_2 to be of the form $\frac{1}{2^{c_1}3^{c_2}}$ by multiplying x_2 by 6^{k+1}.

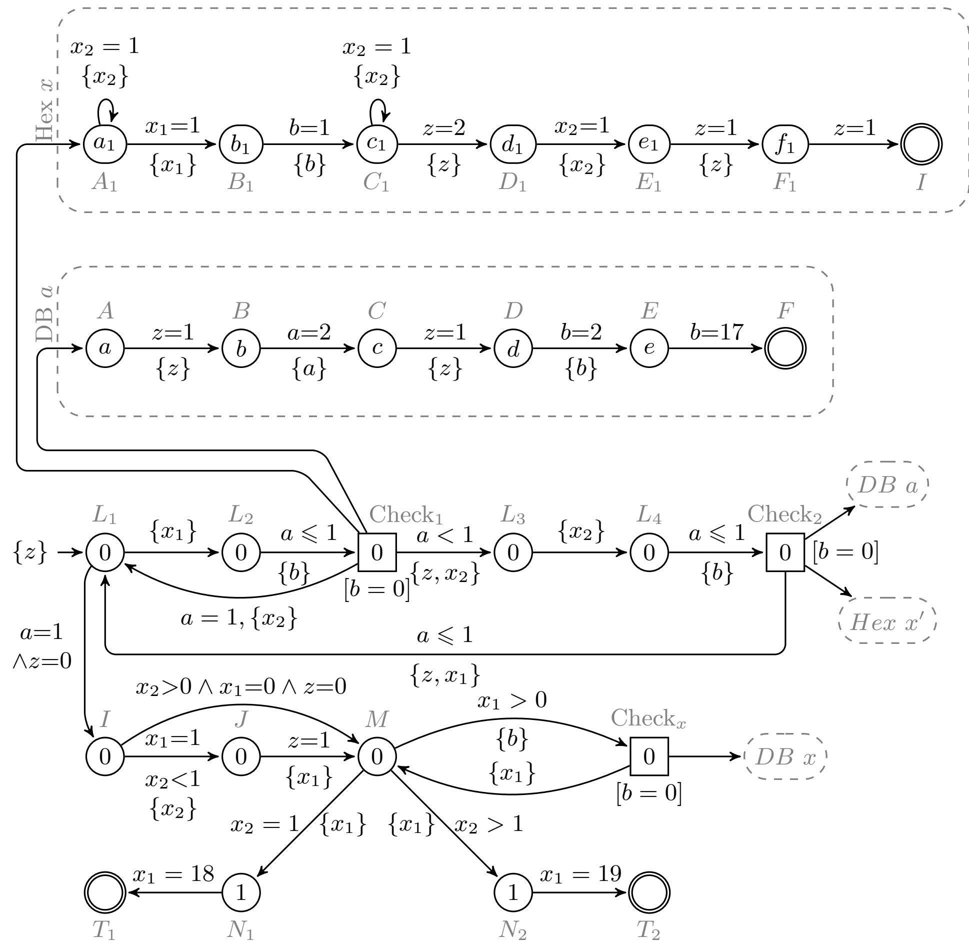


Fig. 18. $\mathsf{TBReach}(18, 40)$: Widget $\mathrm{W}_1^{=0}$ entered with $a = \frac{1}{2^{k+1}} = \alpha$ and $x_1 = x_3 = \frac{1}{2^{k+1+c_1}3^{k+1+c_2}} = \beta$. Widget DB_a checks if upon entry $a = \alpha + t$ and $z = t$ then $t = \alpha$. Times spent : $1 - t$ at A, $1 - \alpha$ at B, α at C, t at D and 17 at E. DB_a stands for 2 widgets, to check $\alpha > t$ $(a, c, e = 1)$ and to check $\alpha < t$ $(d, b, e = 1)$. Widget Hex x checks that if $x_1 = t$, $x_2 = \beta + t + k$ and $z = t + k$ then $t = 6\beta$. Again, this stands for two widgets, one when $t < 6\beta$ ($a_1 = 1, e_1 = 6, f_1 = 17$), and the other when $t > 6\beta$ $(d_1 = 6, b_1 = 1, f_1 = 12)$. DB_x is same as DB_a where a, z are replaced by x_2 and x_1 respectively. Widget Hex x' is the same as widget Hex x with roles of x_1 and x_2 reversed.

Let $\alpha = \frac{1}{2^{k+1}}$ and $\beta = \frac{1}{2^{k+1+c_1}3^{k+1+c_2}}$. The location L_1 is entered with $a = \frac{1}{2^{k+1}} = \alpha$ and $x_2 = \frac{1}{2^{k+1+c_1}3^{k+1+c_2}} = \beta$. Let t_1, t_2 be the times spent at locations L_1, L_2 respectively. Then, on entering Check_1, we have $a = \alpha + t_1 + t_2$, $z = t_1 + t_2$, $x_2 = \beta + t_1 + t_2$ and $x_1 = t_2$. The widget DB a (Fig. 18) ensures that $t_1 + t_2 = \alpha$, i.e; a has been doubled. Similarly, the widget Hex x (Fig. 18) ensures that $t_2 = 6\beta$. No time is spent at Check_1.
We repeatedly keep multiplying a by 2 until a becomes equal to 1. For this, we once take the path L_1 to Check_1, using clock x_1, and the next time, use the path L_3 to Check_2, using clock x_2. Note that when a becomes 1 after $k+1$ iterations, we have also multiplied β with 6^{k+1}. Due to the alternation of clocks x_1, x_2 in paths L_1 to Check_1 and Check_1 to Check_2, the value $\beta * 6^{k+1}$ could be either in x_1 or x_2 while the other clock is 0. We ensure (via locations I, J) that $x_2 = 6^{k+1}\beta$ and $x_1 = 0$ upon entering M. Hence, we get after $k+1$ iterations, $a = 1 = 2^{k+1}\alpha$ and $x_2 = 6^{k+1}\beta = 6^{k+1}\frac{1}{2^{k+1+c_1}3^{k+1+c_2}} = \frac{1}{2^{c_1}3^{c_2}}$. At this point, we are at location M.
Now each time the loop between M and Check_x is taken, value in x_2 is doubled; the widget DB x checks this. Repeatedly doubling x_2 i times gives $x_2 = 2^i\frac{1}{2^{c_1}3^{c_2}}$. If this value is 1, then we know $i = c_1$, and $c_2 = 0$. When this happens we reach location N_1, from where a target is reached with cost 18. If $c_2 \neq 0$, then after some j iterations of the loop between $Check_x$ and M, we will obtain $x_2 = \frac{1}{3^{c_2}}$. Note that we can neither go to N_1 nor N_2 at this point, so the only option is to continue the loop between M and Check_x. Clearly, x_2 will never become 1; so the only option is for x_2 to grow larger than 1. At that point, the transition to N_2 is enabled, and we reach a target with a cost 19.

- Time spent in widget $\text{W}_1^{=0}$: if L_1 was entered for the first time with $a = \frac{1}{2^{k+1}} = \alpha$, then the time spent in L_1 and L_2 before Check_1 is entered is $t = \alpha$. After visiting L_1, L_2, the next time we use L_3, L_4. Since we enter L_3 for the first time with $a = 2\alpha$, the time spent in L_3, L_4 is 2α. The next time we visit L_1, we will be spending 4α and so on. Proceeding like this, we know that the total time spent in this loop before M is reached is $\alpha + 2\alpha + 4\alpha + \cdots + \frac{1}{2}$ which is always < 1. A similar argument holds for the time spent in the loop between M and Check_x. This apart, we spend 18 or 19 units of time (at N_1 or N_2) or at most 25 units of time in the widgets (Hex x), thus adding upto a total time that is at most < 28.
- The widget $\text{W}_1^{\neq 0}$ is similar to $\text{W}_1^{=0}$. The loop between M and Check_x is retained, as is. In addition, when $x_2 < 1$ and $x_1 = 0$, we go to a location M_1 from M. The idea is to first multiply x_2 repeatedly till we obtain $x_2 = \frac{1}{3^{c_2}} < 1$, at which point of time, we go to M_1. From M_1, we have a loop between M_1 and an urgent Player 2 location Check'_x, and a widget Triple x is attached to Check'_x. Each time we come back to M_1 from Check'_x, we reset x_1. Finally, if we get $x_2 = 1, x_1 = 0$, then we go from M_1 to a location M_2 having price 1. Elapsing 18 units of time in M_2, we reach a target with cost 18. However, if x_2 exceeds 1, then with the guard $x_2 > 1, x_1 = 0?$, we go from M_1 to a location M_3 having price 1. To reach a target from M_3, one

has to elapse 19 units of time, thereby incurring a cost 19. Clearly, the route via M_3 will be needed iff $c_2 = 0$.
- The total time spent in $\mathrm{W}_1^{\neq 0}$ will also be less than 28.

To summarize the zero check instruction for C_1: assume we start with $x_1 = \frac{1}{2^{k+c_1}3^{k+c_2}}$, $z = \frac{1}{2^k}$ at location ℓ_{k+1} of Fig. 17. Let us consider the case when Player 1 correctly simulates the instructions within time limits. In this case, location Check is reached with $z = \frac{1}{2^{k+1}}$ and $x_2 = \frac{1}{2^{k+c_1+1}3^{k+c_2+1}}$. Player 2 has the possibility to test if this is indeed the case, by visiting widgets WZ, WI; however Player 1 will achieve his objective in that case. At location Z, Player 1 then correctly guesses whether C_1 is zero or not, by appropriately choosing one of the locations $\mathrm{Check}_{c_1=0}$ or $\mathrm{Check}_{c_1\neq 0}$. Again, Player 2 has the possibility to check if Player 1's guess is correct by visiting widgets $\mathrm{W}_1^{=0}$ and $\mathrm{W}_1^{\neq 0}$; however, Player 1 will achieve his objective here as well. Now consider the case that Player 1 made a mistake: if he did not spend the right amount of time in ℓ_{k+1} and T_2, then Player 2 has the opportunity to punish him through the widgets WI and WZ; if he made a wrong guess regarding C_1 being zero or non-zero, then again Player 2 has a possibility to punish him through the widgets $\mathrm{W}_1^{=0}$ and $\mathrm{W}_1^{\neq 0}$.

Correctness of the construction. On entry into the location ℓ_n (for the HALT instruction), we reset clock x_1 to 0; ℓ_n has cost 1, and the edge coming out of ℓ_n goes to a Goal location, with constraint $x_1 \leqslant 18$.

1. Assume that the two counter machine halts. If Player 1 simulates all the instructions correctly, he will incur a cost $\leqslant 18$, by either reaching the goal location after ℓ_n, or by entering a widget (the second case only occurs if Player 2 decides to check whether Player 1 simulates the machine faithfully. If Player 1 makes an error in his computation, Player 2 can always enter an appropriate widget, making the cost greater than 18. In summary, if the two counter machine halts Player 1 has a strategy to achieve his goal (i.e., reaching a target location with a cost at most 18).
2. Assume that the two counter machine does not halt.
   - If Player 1 simulates all the instructions correctly, and if Player 2 never enters a check widget, then Player 1 incurs cost ∞ as the path is infinite. Notice that even in this case, the total time needed for the computations will be less than 1, due to the strictly decreasing sequence of times chosen for simulating successive instructions. In this case, Player 2 will never want to enter a widget, since he gets a higher payoff.
   - Suppose now that Player 1 makes an error. In this case, Player 2 always has the possibility to reach a target set with a cost greater than 18.

   In summary, if the two counter machine does not halts Player 1 does not have a strategy to achieve his goal.

Thus, Player 1 incurs a cost at most 18 iff he chooses the strategy of faithfully simulating the two counter machine, when the machine halts. When the machine does not halt, the cost incurred by Player 1 is greater than 18 if Player 1 made a simulation error and Player 2 entered a widget. Otherwise, if a widget is not entered, then the run does not end and cost is $+\infty$. □

Shorthand and Longhand notations used in the proof: Note that in the shorthand notation used in widgets $\mathrm{WI}_1^{>}$ and $\mathrm{WI}_1^{<}$,

- we never have consecutive locations with price-rates different from 0;
- on entering any location, there is a "free" clock with value 0;
- further, all guards are of the form $x = c$ with reset of x on all edges.

The time elapsed at a location is captured in the "free" clock which had value 0 while entering that location. Consider, for example two consecutive locations ℓ_1 and ℓ_2 where ℓ_1 has price $f > 0$, ℓ_2 has price 0, with an edge between ℓ_1 and ℓ_2 with guard $x = c$ and reset of x. Let $y = 0$ on entering ℓ_1. If t units of time was spent at ℓ_1, we get $y = t, x = 0$, and the rest of the clocks are incremented by t. A cost of ft is incurred. This can be replaced by a series of $2f - 1$ blocks, where a block looks like this: The first block contains a copy ℓ_{11} of ℓ_1 and some dummy location d_1; ℓ_{11} has price 1, d_1 has price 0, and there is an edge from ℓ_{11} to d_1 with guard $x = c$, and reset of x. A cost t is incurred. In the second block, all locations have price 0. The second block begins from d_1 and ends in the second copy ℓ_{12} of ℓ_1. The price of all copies of ℓ_1 is same as that of ℓ_1. In the second block, the clock values are adjusted to be the same as they were, when they first entered ℓ_1. The third block begins with ℓ_{12}, and is like the first block: it ends in a dummy location d_2. d_2 has price 0. The fourth block begins with d_2, and is like the second block: it ends in the 3rd copy ℓ_{13} of ℓ_1 and so on. The $2f - 1$th block will end in location d_{2f-3}, with price 0. The valuations of all clocks on entering d_{2f-3} is the same as what they were on entering ℓ_2, in the original transition from ℓ_1 to ℓ_2. With no time elapse in d_{2f-3}, we go to ℓ_2.

The total cost incurred across the $2f - 1$ blocks is ft. If the time t elapsed at ℓ_1 is such that $\lceil t \rceil = r$, then the total time elapsed across the $2f-1$ blocks is $\leqslant fr$: t time elapsed in the odd numbered blocks, and $r-t$ in the even numbered blocks, to restore clock values. The point to note is that, as long as there are sufficiently many clocks, the above trick can be done.

Consider the paths ρ_1 and ρ_2 in Figure 16. As explained above, the "free" clock with value 0 is b. Here f is 2 and $t = 1 - v$ where v is the value of clock x_2. Clearly, the cost accumulated in path ρ_1 upon reaching I is $2 * (1 - v)$. The clock values upon entering $F1$ of path ρ_2 are $x_2 = v$ and $b = 0$. Upon entering $G1$ the values are $x_2 = 0$ and $b = 1 - v$. Thus upon entering $F2$, $x_2 = v$ and $b = 0$ and so on. The costs incurred in this path are $1 - v$ at $F1$ and $1 - v$ at $F2$. Hence total cost accumulated is $2 * (1 - v)$ upon reaching I.

C.4 Repeated reachability

Lemma 11. *The existence of a strategy for the repeated reachability objective* $\mathsf{RReach}(\eta)$*, for any* $\eta \in \mathbb{R}_{\geqslant 0}$ *is undecidable for PTGs with 3 clocks or more.*

Proof. We prove the existence of a strategy with a repeated reachability objective ensuring the cost is within an interval $[-\eta, \eta]$ is undecidable, for any choice of $\eta > 0$. In order to obtain the undecidability result, we use a reduction from the halting problem of 2 counter machines. Our reduction uses a PTG with 3 clocks, and arbitrary location prices, but no edge prices.

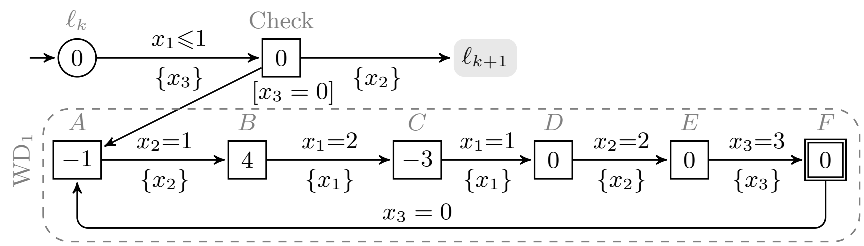


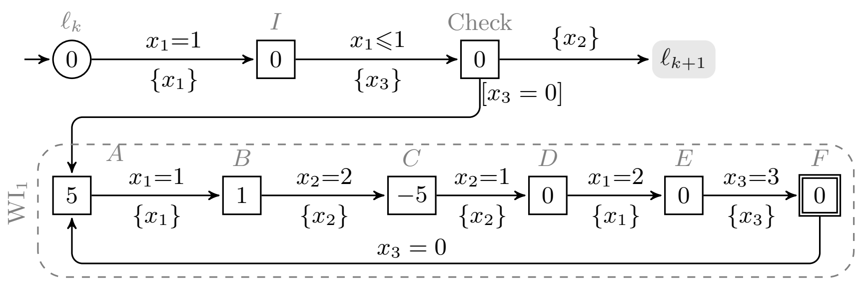
Fig. 19. RReach(η): Simulation to decrement counter C_1

We specify a module for each instruction of the two counter machine. On entry into a module, we have $x_1 = \frac{1}{5^{c_1}7^{c_2}}$,$x_2 = 0$ and $x_3 = 0$, where c_1, c_2 are the values of counters C_1, C_2. We construct a PTG whose building blocks are the modules for instructions. The role of Player 1 is to faithfully simulate the two counter machine, by choosing appropriate delays to adjust the clocks to reflect changes in counter values. Player 2 will have the opportunity to verify that Player 1 did not cheat while simulating the machine. We shall now present modules for increment, decrement and zero check instructions.

Simulation of decrement instruction. : The module to simulate the decrement of counter C_1 is given in Fig. 19. We enter location ℓ_k with $x_1 = \frac{1}{5^{c_1}7^{c_2}}, x_2 = 0$ and $x_3 = 0$. Lets denote by x_{old} the value $\frac{1}{5^{c_1}7^{c_2}}$. To correctly decrement C_1, Player 1 should choose a delay of $4x_{old}$ in location ℓ_k. At location Check, there is no time elapse. Player 2 has two possibilities : (i) to go to ℓ_{k+1}, or (ii) to enter the widget WD$_1$. If Player 1 makes an error, and delays $4x_{old} + \varepsilon$ at ℓ_k $(\varepsilon \neq 0)$then Player 2 can enter the widget WD$_1$ and punish Player 1. When we enter WD$_1$ for the first time, we have $x_1 = x_{old} + 4x_{old} + \varepsilon$, $x_2 = 4x_{old} + \varepsilon$ and $x_3 = 0$. In WD$_1$, the cost of going from location A to F is ε. Also, when we get back to A after going through the loop once, the clock values with which we entered WD$_1$ are restored; thus, each time, we come back to A, we restore the starting values with which we enter WD$_1$. The third clock is really useful for this purpose only.

Since all locations in WD$_1$ are Player 2 locations, Player 2 can continue taking this loop as long as he pleases; each time incurring a cost ε. Thus, for any choice of η, Player 2 can incur a cost that is not in $[-\eta, \eta]$ by taking the loop from A to F a large number of times. Note however that when $\varepsilon = 0$, then Player 1 will always achieve his objective: he will visit F infinitely often with a cost $0 \in [-\eta, \eta]$ for any choice of η.

Simulation of increment instruction.: The module to increment C_1 is given in Fig. 20. Again, we start at ℓ_k with $x_1 = \frac{1}{5^{c_1}7^{c_2}}, x_2 = 0$ and $x_3 = 0$. Again, call $\frac{1}{5^{c_1}7^{c_2}}$ as x_{old}. A time of $1 - x_{old}$ is spent at ℓ_k. Let the time spent at I be denoted x_{new}. To correctly increment counter 1, x_{new} must be $\frac{x_{old}}{5}$. No time is spent at Check. Player 2 can either continue simulation of the next instruction, or can enter the widget WI$_1$ to verify if x_{new} is indeed $\frac{x_{old}}{5}$. Fig. 20 gives a table detailing the values of clocks, time elapsed and cost incurred at each location



Let $x_n = x_{new}$ and $x_o = x_{old}$

Loc $\rightarrow$	ℓ_k	I	Check	ℓ_{k+1}
x_1 on entry	x_o	0	x_n	x_n
x_2 on entry	0	$1 - x_o$	$1 - x_o + x_n$	0
x_3 on entry	0	$1 - x_o$	0	0
Time elapsed	$1 - x_o$	x_n	0	0
Cost incurred	0	0	0	0

Loc $\rightarrow$	A	B	C	D	E	F
x_1 on entry	x_n	0	x_o	$1 + x_o$	0	x_n
x_2 on entry	$1 - x_o + x_n$	$2 - x_o$	0	0	$1 - x_o$	$1 - x_o + x_n$
x_3 on entry	0	$1 - x_n$	$1 - x_n + x_o$	$2 - x_n + x_o$	$3 - x_n$	0
Time elapsed	$1 - x_n$	x_o	1	$1 - x_o$	x_n	0
Cost incurred in one pass	$5 - 5x_n$	x_o	-5	0	0	0
Total cost	-	-	-	-	-	$x_n = \frac{x_o}{5} + \varepsilon$ one pass $= -5x_n + x_o$ $= -5\varepsilon$ after i passes $= -i \times 5 \times \varepsilon$ Total cost $= 0\ if\ \varepsilon = 0$ $> \eta\ if\ \varepsilon < 0$ $< -\eta\ if\ \varepsilon > 0$

Fig. 20. RReach(η): Simulation to increment counter C_1

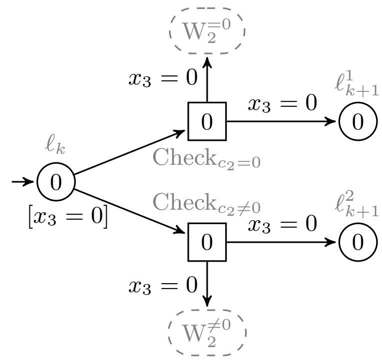


Fig. 21. $\mathsf{RReach}(\eta)$: Widget WZ_2 simulating zero-check for C_2

of the main module, as well as WI_1. It can be seen that the total cost incurred in one pass from location A to F is $x_{old} - 5x_{new}$, which is 0 iff $x_{old} = 5x_{new}$. As seen in the case of decrement, here also, each time we come back to A, we restore the clock values with which we enter WI_1; clearly, if Player 1 makes an error of the form $x_{new} = \frac{x_{old}}{5} + \varepsilon$, the cost incurred in one pass from A to F is -5ε. If $\varepsilon > 0$, then Player 2 can bring the cost less than $-\eta$ for any choice of η by taking the loop between A and F a large number of times. Similarly, if $\varepsilon < 0$, a cost $> \eta$ can be incurred for any choice of η.

Simulation of Zero-check.: Fig. 21 gives the module for zero-check instruction for counter C_2. ℓ_k is a no time elapse location, from where, Player 1 chooses one of the locations $\mathrm{Check}_{c_2=0}$ or $\mathrm{Check}_{c_2\neq 0}$. Both these are Player 2 locations, and Player 2 can either continue the simulation, or can go to the check widgets $\mathrm{W}_2^{=0}$ or $\mathrm{W}_2^{\neq 0}$ to verify the correctness of Player 1's choice.

The widgets $\mathrm{W}_2^{=0}$ and $\mathrm{W}_2^{\neq 0}$ are given in Fig. 22 and 23 respectively.

- Consider the case when Player 1 guessed that C_2 is zero, and entered the location $\mathrm{Check}_{c_2=0}$ in Fig. 21. Let us assume that Player 2 verifies Player 1's guess by entering $\mathrm{W}_2^{=0}$ (Fig. 22). No time is spent in the initial location A of $\mathrm{W}_2^{=0}$. We are therefore at B with $x_1 = \frac{1}{5^{c_1}7^{c_2}} = x_{old}$ and $x_2, x_3 = 0$. In case $c_1 = c_2 = 0$, we can directly go to the F state, and stay there forever, incurring cost 0. If that is not the case, Player 1 has to prove his claim right, by multiplying x_1 with 5 repeatedly, till x_1 becomes 1; clearly, this is possible iff $c_2 = 0$. The loop between B and Check precisely does this: each time Player 1 spends a time x_{new} in B, Player 2 can verify that $x_{new} = 5x_{old}$ by going to WD_1, or come back to B. No time is elasped in Check. Finally, if $x_1 = 1$, we can go to F, and Player 1 achieves his objective. However, if C_2 was non-zero, then x_1 will never reach 1 after repeatedly multiplying x_1 with 5; in this case, at some point, the edge from Check to C will be enabled. In this case, the infinite loop between C and T, incurs a cost $+\infty$.
- Consider now the case when Player 1 guessed that C_2 is non-zero, and hence entered the location $\mathrm{Check}_{c_2\neq 0}$ in Fig. 21. Let us assume now that Player 2

enters $\mathrm{W}_2^{\neq 0}$ (Fig. 23) to verify Player 1's guess. Similar to $\mathrm{W}_2^{=0}$, no time is spent at location A of $\mathrm{W}_2^{\neq 0}$, and the clock values at B are $x_1 = \frac{1}{5^{c_1}7^{c_2}} = x_{old}$ and $x_2, x_3 = 0$. If $c_1 = c_2 = 0$, then $x_2 = 1$, in which case, the location D is reached, from where, the loop between D, T is taken incurring a cost $+\infty$. There are two possibilities now: (i) B can go to C or (ii) to Check_{c_1}. In case B goes to Check_{c_1}, then Player 1 repeatedly multiplies x_1 by 5 till we obtain x_1 as $\frac{1}{7^{c_2}}$. Note that mistakes committed by Player1 in the multiplication by 5, can always be caught by Player 2 via WD_1.

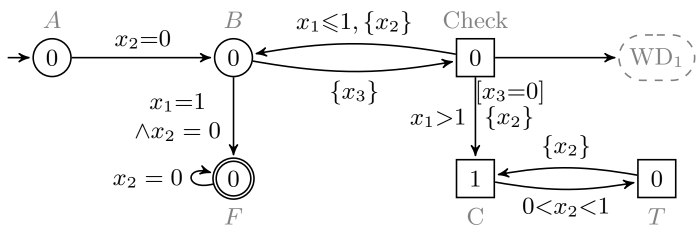
If $c_1 = 0$ already, then we can straightaway go to C spending no time at B, if $c_2 \neq 0$. In this case, Player 1 has to compulsorily go to Check_{c_2} from C atleast once, since the edge from C to F is not enabled. The loop between C and Check_{c_2} results in Player 1 multiplying x_1 of the form $\frac{1}{7^{c_2}}$ till x_1 becomes 1. Here again, if Player 1 commits a mistake during multiplication by 7, Player 2 can catch Player 1 by entering the widget WD_2. Otherwise, when x_1 reaches 1, Player 1 can go from C to F achieving his objective.

Correctness of the construction. On entry into the location ℓ_n (for HALT instruction), we reset clock x_1 to 0; from ℓ_n, we go to a state F with price 0, with a self loop $x_1 = 0$.

1. Assume that the two counter machine halts. If Player 1 simulates all the instructions correctly, he will incur a cost $= 0$, by either reaching the F after ℓ_n, or by entering a widget (the second case only occurs if Player 2 decides to check whether Player 1 simulates the machine faithfully. If Player 1 makes an error in his computation, Player 2 can always enter an appropriate widget, making the cost as large or as small, so as to not fit in $[-\eta, \eta]$ for any choice of η. In summary, if the two counter machine halts Player 1 has a strategy to achieve his goal (visiting F with a cost $0 \in [-\eta, \eta]$ for any $\eta > 0$.)
2. Assume that the two counter machine does not halt.
   - If Player 1 simulates all the instructions correctly, and if Player 2 never enters a check widget, then Player 1 incurs cost ∞ as we never reach F. In this case, Player 2 will never want to enter a widget, since he gets a higher payoff.
   - Suppose now that Player 1 makes an error. In this case, Player 2 always has the possibility to enter a loop, where Player 1 will incur cost ∞ or $-\infty$.

   In summary, if the two counter machine does not halts Player 1 does not have a strategy to achieve his goal.

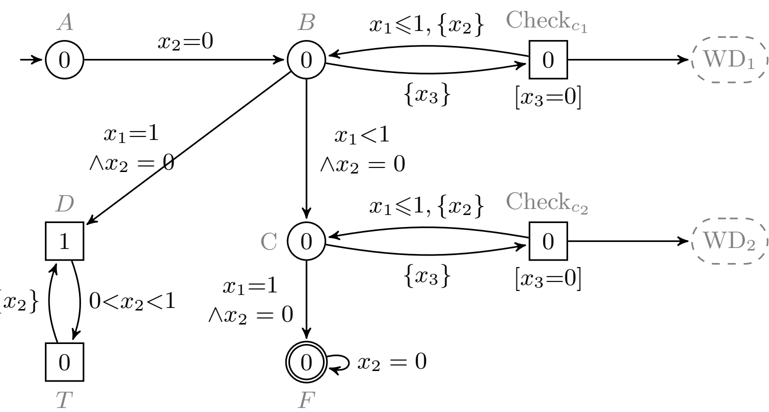
Thus, Player 1 incurs a cost in $[-\eta, \eta]$ for any η iff he chooses the strategy of faithfully simulating the two counter machine, when the machine halts. When the machine does not halt, the cost incurred by Player 1 is not in $[-\eta, \eta]$ for any chosen η if Player 1 made a simulation error and Player 2 entered a widget. Else if a widget is not entered then the run does not reach F and cost is $+\infty$. $\square$



Let $\alpha = \frac{1}{5^{c_1} 7^{c_2}}$

Location $\rightarrow$	A	B	Check	C	T	F
x_1 on entry	α	$5^i \times \alpha$ loop $B \rightarrow$ Check taken i times	$5^i \times \alpha + k$ $k = 4(5^i \times \alpha) + \varepsilon$	> 1 $5^i \times \alpha > 1$ $c_2 \neq 0$	-	1 $5^i \times \alpha = 1$ $i = c_1 \wedge c_2 = 0$
x_2 on entry	0	0	k	0	$p > 0$	0
x_3 on entry	0	0	0	-	-	0
Time elapsed	0	k	0	p	0	0
Cost incurred in one pass	0	0	0	p	0	0
Total cost	-	-	-	$> \eta$	-	0

Fig. 22. $\mathsf{RReach}(\eta)$: Widget $\mathrm{W}_2^{=0}$. Delay at B is $k = 4(5^i \times \alpha) + \varepsilon = 4(5^i \times \frac{1}{5^{c_1} 7^{c_2}}) + \varepsilon$. If $\varepsilon \neq 0$ then Player 2 will enter the widget WD_1 and where the cost incurred $\notin [-\eta, \eta]$ if $\varepsilon \neq 0$.



Let $\alpha = \frac{1}{5^{c_1} 7^{c_2}}$

Loc $\rightarrow$	A	B	D	T	C	F
x_1 on entry	α	$5^i \times \alpha$ loop $B \rightarrow$ Check taken i times	1 $5^i \times \alpha = 1$ $i = c_1 \wedge c_2 = 0$	-	$7^j \times \frac{1}{7^{c_2}}$ loop $C \rightarrow$ Check taken j times	1 $7^j \times 5^i \times \alpha = 1$ $i = c_1 \wedge$ $j = c_2 > 0$
x_2 on entry	0	0	0	$p > 0$	0	0
Time elapsed	0	$4(5^i \times \alpha) + \varepsilon$	p	0	$6(7^j \times \frac{1}{7^{c_2}}) + \varepsilon$	0
Cost incurred in one pass	0	0	p	0	0	0
Total cost	-	-	$> \eta$	-	-	0

Fig. 23. $\mathsf{RReach}(\eta)$: Widget $\mathrm{W}_2^{\neq 0}$. Widget WD_2 following Check$_{c_2}$ is similar to WD_1 shown earlier, except that the prices are adjusted to verify decrement of counter C_2.