The Value Problem for Weighted Timed Games with Two Clocks is Undecidable

QUENTIN GUILMANT, Max Planck Institute for Software Systems, Germany
JOËL OUAKNINE, Max Planck Institute for Software Systems, Germany
ISA VIALARD, Max Planck Institute for Software Systems, Germany

The Value Problem for weighted timed games (WTGs) consists in determining, given a two-player weighted timed game with a reachability objective and a rational threshold, whether or not the value of the game exceeds the threshold. This problem was shown to be undecidable some ten years ago for WTGs making use of at least three clocks, and is known to be decidable for single-clock WTGs. In this paper, we establish undecidability for two-clock WTGs making use of non-negative weights, even in a time-bounded setting, closing one of the last remaining major gaps in our algorithmic understanding of WTGs.

ACM Reference Format:
Quentin Guilmant, Joël Ouaknine, and Isa Vialard. 2025. The Value Problem for Weighted Timed Games with Two Clocks is Undecidable. 1, 1 (July 2025), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Real-time systems are not only ubiquitous in modern technological society, they are in fact increasingly pervasive in critical applications — from embedded controllers in automotive and avionics platforms to resource-constrained communication protocols. In such systems, exacting timing constraints and quantitative objectives must often be met simultaneously. Weighted Timed Games (WTGs), introduced over two decades ago [1–3, 11, 13], provide a powerful modelling framework for the automatic synthesis of controllers in such settings: they combine the expressiveness of Alur and Dill's clock-based timed automata with **Min**-**Max** gameplay and non-negative integer weights on both locations and transitions, enabling one to reason about quantitative aspects such as energy consumption, response times, or resource utilisation under adversarial conditions.

A central algorithmic task for WTGs is the *Value Problem*: given a two-player, turn-based WTG with a designated start configuration and a rational threshold c, determine whether Player **Min** can guarantee reaching a goal location with cumulative cost at most c, despite best adversarial play by Player **Max**. This problem lies at the heart of quantitative controller synthesis and performance analysis for real-time systems.

Unfortunately, fundamental algorithmic barriers are well known. In particular, the Value Problem is known to be undecidable in general, both for WTGs making use of three or more clocks [4], as well as for two-clock extensions of WTGs in which arbitrary integer (positive and negative) weights are allowed [8]. In fact, even approximating the value of three-clock WTGs with integer weights is known to be computationally unsolvable [10]. On the positive side, the Value Problem

Authors' Contact Information: Quentin Guilmant, Max Planck Institute for Software Systems, Saarland Informatics Campus, Saarbrücken, Saarland, Germany, quentin.guilmant@mpi-sws.org; Joël Ouaknine, Max Planck Institute for Software Systems, Saarland Informatics Campus, Saarbrücken, Saarland, Germany, joel@mpi-sws.org; Isa Vialard, Max Planck Institute for Software Systems, Saarland Informatics Campus, Saarbrücken, Saarland, Germany, vialard@mpi-sws.org.

ACM XXXX-XXXX/2025/7-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

for WTGs making use of a single clock is decidable, regardless of whether weights range over $\mathbb{N}$ or $\mathbb{Z}$ [5, 15]. There is a voluminous literature in this general area; for a comprehensive overview and discussion of the state of the art, we refer the reader to [9]. See also Fig. 1 in which we summarise some of the key existing results.

Clocks	Weights in	Value Problem
1	$\mathbb{N}$	decidable [5]
	$\mathbb{Z}$	decidable [15]
2	$\mathbb{N}$	**undecidable**
	$\mathbb{Z}$	undecidable [8]
3+	$\mathbb{N}$	undecidable [4]
	$\mathbb{Z}$	undecidable [4] inapproximable [10]

Fig. 1. State of the art on the Value Problem for weighted timed games. Approximability for 2-clock WTGs, and 3-clock WTGs with weights in $\mathbb{N}$, remain open. This paper's main contribution (undecidability for WTGs with two clocks and weights in $\mathbb{N}$) is highlighted in boldface blue.

The case of WTGs with exactly two clocks (and non-negative weights) has remained stubbornly open. Resolving this question is essential, since two clocks suffice to encode most practical timing constraints (e.g., deadline plus cooldown), and efficient single-clock algorithms cannot in general be lifted to richer timing scenarios. The main contribution of this paper is to close this gap by establish undecidability:

Theorem 1.1. *The Value Problem for two-player, turn-based, time-bounded, two-clock, weighted timed games with non-negative integer weights is undecidable. The same holds for weighted timed games over unbounded time otherwise satisfying the same hypotheses.*

Our reduction is from the halting problem for deterministic two-counter machines and proceeds via a careful encoding of counter values in clock valuations, combined with "punishment" gadgets that force faithful simulation or allow the adversary to drive the accumulated cost up. Key technical novelties include:

- Counter-Evolution Control (CEC) modules, which enforce precise proportional delays encoding incrementation and decrementation of counters.
- Multiplication-Control (MC) gadgets, allowing the adversary to verify whether simulated counter updates match exact multiplication factors.
- Zero and Non-Zero control schemes, enabling precise zero-testing within the two-clock framework and ensuring that any deviation from a faithful simulation triggers a cost penalty.

Together, these constructions fit within the two-clock timing structure and utilise only non-negative integer weights, thereby demonstrating that even the two-clock fragment — previously the only remaining decidability candidate — admits no algorithmic solution for the Value Problem. As a complementary result, we also show that the related Existence Problem (does **Min** have a strategy to achieve cost at most c?) is undecidable under the same hypotheses.

Let us conclude by noting that our reduction is implemented via WTGs having bounded duration by construction. This is notable in view of the fact that many algorithmic problems for real-time and hybrid systems that are known to be undecidable over unbounded time become decidable in a time-bounded setting; see, e.g., [6, 7, 12, 16, 17].

2 Weighted Timed Games

2.1 Definitions

Let $\mathcal{X}$ be a finite set of **clocks**. **Clock constraints** over $\mathcal{X}$ are expressions of the form $x \bowtie n$, where $x, y \in \mathcal{X}$ are clocks, $\bowtie \in \{<, \leq, =, \geq, >\}$ is a comparison symbol, and $n \in \mathbb{N}$ is a natural number. We write C to denote the set of all clock constraints over $\mathcal{X}$. A **valuation** on $\mathcal{X}$ is a function $\nu : \mathcal{X} \to \mathbb{R}_+$. For $d \in \mathbb{R}_+$ we denote by $\nu + d$ the valuation such that, for all clocks $x \in \mathcal{X}$, $(\nu + d)(x) = \nu(x) + d$. Let $X \subseteq \mathcal{X}$ be a set of clocks. We write $\nu[X := 0]$ for the valuation such that, for all clocks $x \in X$, $\nu[X := 0](x) = 0$, and $\nu[X := 0](y) = \nu(y)$ for all other clocks $y \notin X$. For $C \subseteq \mathcal{C}$ a set of clock constraints over $\mathcal{X}$, we say that the valuation ν **satisfies** C, denoted $\nu \models C$, if and only if all the comparisons in C hold when replacing each clock x by its corresponding value $\nu(x)$. Finally, we write $\mathbf{0}$ to denote the valuation that assigns 0 to every clock.

Definition 2.1. A **weighted timed game** (WTG) is a tuple $\mathcal{G} = (L, G, \mathcal{X}, T_{\text{Min}}, T_{\text{Max}}, w)$, where:

- L is a set of **locations**.
- $G \subseteq L$ are the **goal locations**.
- $\mathcal{X}$ is a set of clocks.
- $T_{\text{Min}}, T_{\text{Max}} \subseteq (L \setminus G) \times 2^{C} \times 2^{\mathcal{X}} \times L$ are sets of **(discrete) transitions** belonging to players **Min** and **Max** respectively. We denote $T_{\mathcal{G}} = T_{\text{Min}} \cup T_{\text{Max}}$ the set of all transitions. Transition $\ell \xrightarrow{C,X} \ell'$ enables moving from location ℓ to location ℓ', provided all clock constraints in C are satisfied, and afterwards resetting all clocks in X to zero.
- $w : (L \setminus G) \cup T \to \mathbb{Z}$ is a **weight function**.

In the above, we assume that all data (set of locations, set of clocks, set of transitions, set of clock constraints) are finite.

Definition 2.2. A WTG $\mathcal{G} = (L, G, \mathcal{X}, T_{\text{Min}}, T_{\text{Max}}, w)$ is said to be **turn-based** if for any location $\ell \in L$ the set of transitions from ℓ is entirely contained either in T_{Min} or T_{Max}. In this case we may partition $L \setminus G$ into L_{Min} and L_{Max}, the sets of locations from which every transition belongs to **Min** and **Max**, respectively. Abusing notation, we may then equivalently refer to $\mathcal{G}$ as the tuple $(L_{\text{Min}}, L_{\text{Max}}, G, \mathcal{X}, T, w)$. Finally, a WTG which is not turn-based is called **concurrent**. In the remainder of this paper, we shall exclusively work with turn-based weighted timed games.

Let $\mathcal{G} = (L, G, \mathcal{X}, T_{\text{Min}}, T_{\text{Max}}, w)$ be a weighted timed game. A **configuration** over $\mathcal{G}$ is a pair (ℓ, ν), where $\ell \in L$ and ν is a valuation on $\mathcal{X}$. We write $\mathscr{C}_{\mathcal{G}} = L \times \mathbb{R}^{\mathcal{X}}_{\geq 0}$ to denote the set of configurations over $\mathcal{G}$. Let $d \in \mathbb{R}_{\geq 0}$ be a **delay** and $t = \ell \xrightarrow{C,X} \ell' \in T$ be a discrete transition. One has a **delayed transition** (or simply a **transition** if the context is clear) $(\ell, \nu) \xrightarrow{d,t} (\ell', \nu')$ provided that $\nu + d \models C$ and $\nu' = (\nu + d)[X := 0]$. Intuitively, control remains in location ℓ for d time units, after which it transitions to location ℓ', resetting all the clocks in X to zero in the process. The **weight** of such a delayed transition is $d \cdot w(\ell) + w(t)$, taking into account both the time spent in ℓ as well as the weight of the discrete transition t.

As noted in [9], without loss of generality one can assume that no configuration (other than those associated with goal locations) is deadlocked; in other words, for any location $\ell \in L \setminus G$ and valuation $\nu \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$, there exists $d \in \mathbb{R}_{\geq 0}$ and $t \in T$ such that $(\ell, \nu) \xrightarrow{d,t} (\ell', \nu')$.[1]

[1]In our setting, this can be achieved by adding unguarded transitions to a sink location for all locations controlled by **Min** and unguarded transitions to a goal location for the ones controlled by **Max** (noting that in all our constructions, **Max**-controlled locations always have weight 0). Nevertheless, in the pictorial representations of timed-game fragments that appear in this paper, in the interest of clarity we omit such extraneous transitions and locations; we merely assume instead that neither player allows themself to end up in a deadlocked situation, unless a goal location has been reached.

Let $k \in \mathbb{N}$. A **run** ρ of length $|\rho| = k$ over $\mathcal{G}$ from a given configuration (ℓ_0, v_0) is a sequence of matching delayed transitions, as follows:

$$\rho = (\ell_0, v_0) \xrightarrow{d_0,t_0} (\ell_1, v_1) \xrightarrow{d_1,t_1} \cdots \xrightarrow{d_{k-1},t_{k-1}} (\ell_k, v_k)\,.$$

We denote by $\rho_i^{\mathscr{C}} = (\ell_i, v_i)$ the i-th configuration reached along the run ρ and $\rho_i^T = (d_i, t_i)$ the delayed transition picked from state configuration $\rho_i^{\mathscr{C}}$ in ρ. We also write $\rho_{|n}$, for $n \le |\rho|$, to denote the truncated run ending in configuration $\rho_n^{\mathscr{C}}$. The **weight** of ρ is the cumulative weight of the underlying delayed transitions:

$$\mathsf{weight}(\rho) = \sum_{i=0}^{|\rho|-1} (d_i \cdot w(\ell_i) + w(t_i))\,.$$

This definition can be extended to infinite runs; however, since no goal location is ever reached, the weight of an infinite run ρ is defined to be infinite: $\mathsf{weight}(\rho) = +\infty$.

A run is **maximal** if it is either infinite or cannot be extended further. Thanks to our deadlock-freedom assumption, finite maximal runs must end in a goal location. We refer to maximal runs as **plays**.

2.2 Graphical notation for WTG

In this paper, WTG are represented pictorially. We follow the conventions below:

- Blue circles represent locations controlled by **Min**.
- Red squares represent locations controlled by **Max**.
- Green circles are goal locations.
- Arrows represent transitions.
- Guards and resets are written next to the transition they are associated with.
- Resetting a clock x is denoted $x := 0$.
- Grey rectangles are modules, or in other words subgames: a transition entering a module transfers control to the starting location of the module. Modules can have outgoing edges.
- Numbers attached to locations denote their respective weight.
- Numbers in grey boxes attached to arrows denote the weight of the corresponding transition. Transitions without such boxes have weight 0.
- Green boxes attached to transitions are comments (or assertions) on the values of the clocks. Such assertions hold upon taking the transition. They are occasionally complemented by orange boxes which are assertions on the corresponding cost incurred.
- Some locations are decorated with a numbered flag for ease of reference in proofs.

2.3 Strategies and value of a game

We now define the notion of **strategy**. Recall that transitions of $\mathcal{G}$ are partitioned into sets $T_{\mathbf{Min}}$ and $T_{\mathbf{Max}}$, belonging respectively to Players **Min** and **Max**. Let Player $\mathsf{P} \in \{\mathbf{Min}, \mathbf{Max}\}$, and write $\mathcal{FR}_{\mathcal{G}}^{\mathsf{P}}$ to denote the collection of all non-maximal finite runs of $\mathcal{G}$ ending in a location belonging to Player P. A **strategy** for Player P is a mapping $\sigma_{\mathsf{P}} : \mathcal{FR}_{\mathcal{G}}^{\mathsf{P}} \to \mathbb{R}_{\ge 0} \times T$ such that for all finite runs $\rho \in \mathcal{FR}_{\mathcal{G}}^{\mathsf{P}}$ ending in configuration (ℓ, v) with $\ell \in L_{\mathsf{P}}$, if $\sigma_{\mathsf{P}}(\rho) = (d, t)$, then the delayed transition $(\ell, v) \xrightarrow{d,t} (\ell', v')$ is valid, where $\sigma_{\mathsf{P}}(\rho) = (d, t)$ and (ℓ', v') is some configuration (uniquely determined by $\sigma_{\mathsf{P}}(\rho)$ and v).

Let us fix a starting configuration (ℓ_0, v_0), and let $\sigma_{\mathbf{Min}}$ and $\sigma_{\mathbf{Max}}$ be strategies for Players **Min** and **Max** respectively (one speaks of a *strategy profile*). We write $\mathsf{play}_{\mathcal{G}}((\ell_0, v_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})$ to denote the unique maximal run starting from configuration (ℓ_0, v_0) and unfolding according to the strategy profile $(\sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})$: in other words, for every strict finite prefix ρ of $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})$ in $\mathcal{FR}^{\mathsf{P}}_{\mathcal{G}}$, the delayed transition immediately following ρ in $\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})$ is labelled with $\sigma_{\mathsf{P}}(\rho)$.

Recall that the objective of Player **Min** is to reach a goal location through a play whose weight is as small as possible. Player **Max** has an opposite objective, trying to avoid goal locations, and, if not possible, to maximise the cumulative weight of any attendant play. This gives rise to the following symmetrical definitions:

$$\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \inf_{\sigma_{\mathbf{Min}}} \left\{ \sup_{\sigma_{\mathbf{Max}}} \left\{ \mathsf{weight}(\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})) \right\} \right\} \text{ and}$$
$$\underline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \sup_{\sigma_{\mathbf{Max}}} \left\{ \inf_{\sigma_{\mathbf{Min}}} \left\{ \mathsf{weight}(\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})) \right\} \right\} .$$

$\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$ represents the smallest possible weight that Player **Min** can possibly achieve, starting from configuration (ℓ_0, ν_0), against best play from Player **Max**, and conversely for $\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$: the latter represents the largest possible weight that Player **Max** can enforce, against best play from Player **Min**.[2] As noted in [9], turned-based weighted timed games are *determined*, and therefore $\overline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0) = \underline{\mathsf{Val}}_{\mathcal{G}}(\ell_0, \nu_0)$ for any starting configuration (ℓ_0, ν_0); we denote this common value by $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0)$.

Remark 2.3. Note that $\mathsf{Val}_{\mathcal{G}}(\ell_0, \nu_0)$ can take on real numbers, or either of the values $-\infty$ and $+\infty$.

We can now state:

Definition 2.4 (Value Problem). Given a WTG $\mathcal{G}$ with starting location ℓ_0 and a threshold $c \in \mathbb{Q}$, the **Value Problem** asks whether $\mathsf{Val}_{\mathcal{G}}(\ell_0, \mathbf{0}) \leq c$.

The Value Problem differs subtly but importantly from the *Existence Problem*:

Definition 2.5 (Existence Problem). Given a WTG $\mathcal{G}$ with starting location ℓ_0 and a threshold $c \in \mathbb{Q}$, the **Existence Problem** asks whether **Min** has a strategy $\sigma_{\mathbf{Min}}$ such that

$$\sup_{\sigma_{\mathbf{Max}}} \left\{ \mathsf{weight}(\mathsf{play}_{\mathcal{G}}((\ell_0, \nu_0), \sigma_{\mathbf{Min}}, \sigma_{\mathbf{Max}})) \right\} \leq c .$$

3 Undecidability

To establish our undecidability result, we reduce the Halting Problem for two-counter machines to the Value Problem. A two-counter machine is a tuple $\mathcal{M} = (Q, q_i, q_h, T)$ where Q is a finite set of states, $q_i, q_h \in Q$ are the initial and final state and $T \subseteq (Q \times \{c, d\} \times Q) \cup (Q \times \{c, d\} \times Q \times Q)$ is a set of transitions. A two-counter machine is deterministic if for any state q there is at most one transition $t \in T$ which has q as first component. As its name suggests, a two-counter machine comes equipped with two counters, c and d, which are variables with values in $\mathbb{N}$. The semantics is as follows: a transition (q, e, q') increases the value of counter $e \in \{c, d\}$ by 1 and moves to state q'. A transition (q, e, q', q'') moves to q' if $e = 0$ and to q'' otherwise. In the latter case, it also decreases the value of e by 1. The Halting Problem for (deterministic) two-counter machines is known to be undecidable (see [14, Thm. 14-1]).

[2]Technically speaking, these values may not be literally achievable; however given any $\varepsilon > 0$, both players are guaranteed to have strategies that can take them to within ε of the optimal value.

3.1 Overview of the reduction

Let $\mathcal{M}$ be a two-counter machine with counters c and d. We consider a WTG $\mathcal{G}_\mathcal{M}$ between players **Min** and **Max** with two clocks x and y. **Min** is in charge of simulating the two-counter machine $\mathcal{M}$ while **Max** has the opportunity to punish **Min**'s errors. In encoding $\mathcal{M}$ as through $\mathcal{G}_\mathcal{M}$, some of the locations of $\mathcal{G}_\mathcal{M}$ will represent states of $\mathcal{M}$. Upon entering a location of $\mathcal{G}_\mathcal{M}$, a faithful encoding of counters c and d is represented through a clock valuation in which $y = 0$ and

$$x = 1 - \frac{1}{2^c 3^d 5^n},$$

where n is the number of steps of the execution of $\mathcal{M}$ that have been simulated so far. Furthermore, in order to simulate the next step faithfully,

- If the current state requires c to be incremented, **Min** should wait in this location until x reaches value $1 - \frac{1}{2^{c+1}3^d 5^{n+1}}$; this means that **Min** should wait $\frac{9}{10}(1-x)$ time units.
- If the current state requires d to be incremented, **Min** should wait in this location until x reaches value $1 - \frac{1}{2^{c}3^{d+1} 5^{n+1}}$; this means that **Min** should wait $\frac{14}{15}(1-x)$ time units.
- If the current state requires c to be decremented, **Min** should wait in this location until x reaches value $1 - \frac{1}{2^{c-1}3^d 5^{n+1}}$; this means that **Min** should wait $\frac{3}{5}(1-x)$ time units.
- If the current state requires d to be decremented, **Min** should wait in this location until x reaches value $1 - \frac{1}{2^{c}3^{d-1} 5^{n+1}}$; this means that **Min** should wait $\frac{2}{5}(1-x)$ time units.
- If the current state is a succesful zero-test for c or d, **Min** should wait in this location until x reaches value $1 - \frac{1}{2^{c}3^d 5^{n+1}}$; this means that **Min** should wait $\frac{4}{5}(1-x)$ time units.

After **Min** has chosen the waiting time, **Max** is given the opportunity to end the game with an increased cost if **Min** "cheated". In the case of a zero-test, if **Min** takes a transition towards the wrong state, **Max** is also given the opportunity to end the game asking for a proof that the corresponding counter is indeed 0 (respectively not 0). Finally, before simulating another step, **Min** is also given the opportunity to exit the game. Note also that since the halting state of $\mathcal{M}$ has no outgoing transitions, **Min** will be forced to end the game if she reaches the location representing this state.

3.2 Controlling the evolution of counters

In the overview, we have seen that there are two distinct instances in which **Max** is able to punish **Min**: either following an incorrect update of clock x, or an incorrect transition on a zero-test. In this subsection, we introduce a gadget for the first instance: the CEC (Counter Evolution Control) module. The module $\mathrm{CEC}^{M,N}_{\alpha,\beta}(x,y)$, depicted in Fig. 2, requires that $0 \leq y \leq x < 1$. We will denote $a+b$ and b the initial values of x and y, respectively. In a faithful run we expect a to have value $1 - \dfrac{1}{2^c 3^d 5^n}$ and b to be the time needed so that $a+b$ corresponds to the correct numerical encoding for the updated values of the counters. More precisely, depending on the cases described in the previous subsection, we need the CEC to enforce

$$b = \gamma(1-a) \text{ with } \gamma \in \left\{\frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{9}{10}, \frac{14}{15}\right\}.$$

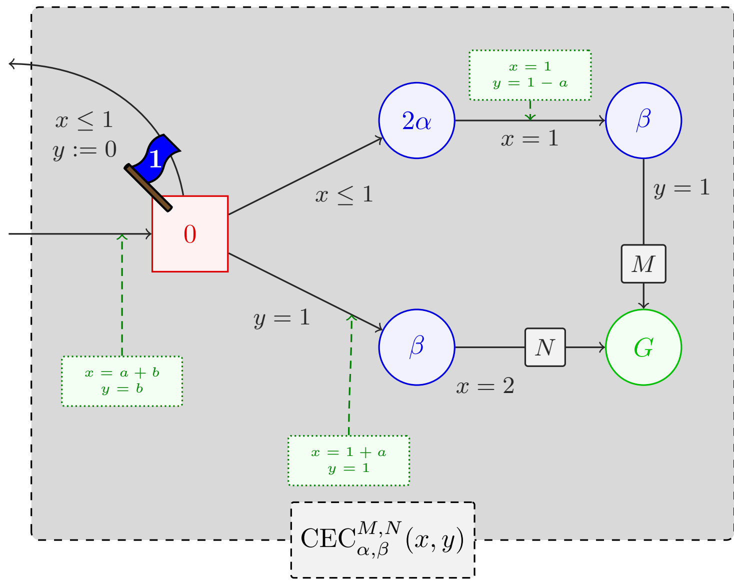
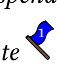


Fig. 2. The CEC (Counter Evolution Control) module

Lemma 3.1. *Let t be the time* **Max** *spends in state*  *of* $\mathrm{CEC}^{M,N}_{\alpha,\beta}(x, y)$*. Provided that the initial values of x and y upon entering the state*  *in* $\mathrm{CEC}^{M,N}_{\alpha,\beta}(x, y)$ *are in* $[0, 1)$ *with* $x \geq y$*, denoting by b the initial value of y and by $a + b$ the initial value of x, and assuming that the overall cost accumulated so far is $\alpha(a + b) + E$,* **Max** *can choose between the following three cases:*

(1) *The game continues with $x = a + b + t$, $y = 0$ and the accumulated cost is $\alpha(a + b + t) + (E - \alpha t)$.*
(2) *The game stops with cost*

$$(\alpha - \beta)(1 - a) - \alpha b - 2\alpha t + \alpha + \beta + M + E = (\beta - \alpha)a - \alpha b - 2\alpha t + 2\alpha + M + E\,.$$

(3) *The game stops with cost*

$$\alpha b - (\alpha - \beta)(1 - a) + \alpha + N + E = \alpha b - (\beta - \alpha)a + \beta + N + E\,.$$

Proof. The first case corresponds to the cost when **Max** decides to exit the module, the second when he goes through the upper path and the third case when he goes through the lower path. □

Corollary 3.2. *Let $0 \leq \beta < \alpha$. Let $\varepsilon > 0$. Provided that the initial values of x and y upon entering the state*  *in* $\mathrm{CEC}^{M,M+\beta}_{\alpha,\beta}(x, y)$ *are in* $[0, 1)$ *with* $x \geq y$*, denoting by b the initial value of y and by $a + b$ the initial value of x, and assuming that the overall cost accumulated so far is $\alpha(a + b) + E$,* **Max** *can either end the game with a final cost of*

$$\alpha\left(1 + \left|b - \left(1 - \frac{\beta}{\alpha}\right)(1 - a)\right|\right) + E + M + \beta$$

or exit the module with $x = a + b + t$, $y = 0$ and accumulated cost $\alpha(a + b + t) + (E - \alpha t)$.

Proof. This is a direct application of Lem. 3.1 with $N = M + \beta$. □

Thus, if we take $\alpha = 30$ and $\beta \in \{18, 12, 6, 3, 2\}$, the value of b minimising the cost of **Max** reaching the goal location is indeed

$$b = \gamma(1 - a) \text{ for } \gamma \in \left\{\frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{9}{10}, \frac{14}{15}\right\} .$$

3.3 Controlling whether a counter is zero

We now handle the case in which **Min** has chosen the wrong state to move to when simulating a zero-test. To do so, depending on the case, **Max** has access to either a control-if-zero (ZC) or control-if-not-zero (NZC) module that checks whether a counter is indeed zero or not through a series of multiplications by suitable coefficients to reach 1. We first present the module that controls the multiplication (MC). It is depicted in Fig. 3.

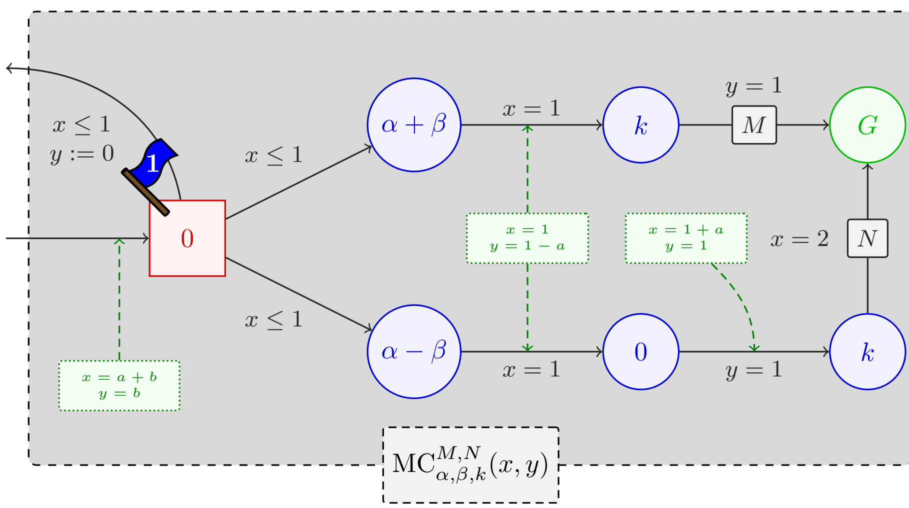


Fig. 3. The MC (Multiplication Control) module

$\mathrm{MC}^{M,N}_{\alpha,\beta,k}(x, y)$ requires that $0 \le y \le x < 1$. We will denote by $a+b$ and b the initial values of x and y, respectively. As such, this module behaves similarly to the CEC module. However, whereas the CEC module checks that, for a of the form $1 - \frac{1}{2^c 3^d 5^n}$, $a + b$ is of the same form, here $\mathrm{MC}^{M,N}_{\alpha,\beta,k}(x, y)$, for $k \in \{2, 3, 5\}$, checks that, for a of the form $\frac{1}{2^c 3^d 5^n}$, $a+b = ka \in \{\frac{1}{2^{c-1}3^d 5^n}, \frac{1}{2^c 3^{d-1} 5^n}, \frac{1}{2^c 3^d 5^{n-1}}\}$.

Lemma 3.3. *Let t be the time* **Max** *spends in state of $\mathrm{MC}^{M,N}_{\alpha,\beta,k}(x, y)$. Provided that the initial values of x and y upon entering the state in $\mathrm{MC}^{M,N}_{\alpha,\beta,k}(x, y)$ are in $[0, 1)$ with $x \ge y$, denoting by b the initial value of y and by $a + b$ the initial value of x, and assuming that the overall cost accumulated so far is $\alpha(a + b) + E$,* **Max** *can choose between the three following cases:*

(1) *The game continues with $x = a + b + t$, $y = 0$ and the accumulated cost is $\alpha(a + b + t) + (E - \alpha t)$.*
(2) *The game stops with cost $\alpha + \beta + M + E - (\alpha + \beta)t - \beta b + (k - \beta)a$.*
(3) *The game stops with cost $\alpha - \beta + N + E - (\alpha - \beta)t + \beta b + k - (k - \beta)a$.*

Proof. The first case corresponds to the cost when **Max** decides to exit the module, the second when he goes through the upper path and the third case when he goes through the lower path. □

Corollary 3.4. *Let $0 \leq \beta \leq \alpha$. Provided that the initial values of x and y upon entering the state in $\mathrm{MC}^{M,M+2\beta}_{\alpha,\beta,k\beta}(x,y)$ are in $[0,1)$ with $x \geq y$, denoting by b the initial value of y and by $a+b$ the initial value of x, and assuming that the overall cost accumulated so far is $\alpha(a+b)+E$,* **Max** *can either end the game with final cost*

$$\alpha + \beta + M + E + \beta\,|b - (k-1)a|$$

or exit the module with $x = a+b+t$, $y = 0$ and accumulated cost $\alpha(a+b+t) + (E - \alpha t)$.

Here we see clearly that the MC module checks for multiplication. The cost of **Max** ending the game is minimised for $b = (k-1)a$, hence $x = a + b = ka$.

We now introduce modules to control whether a counter is indeed 0 or not.

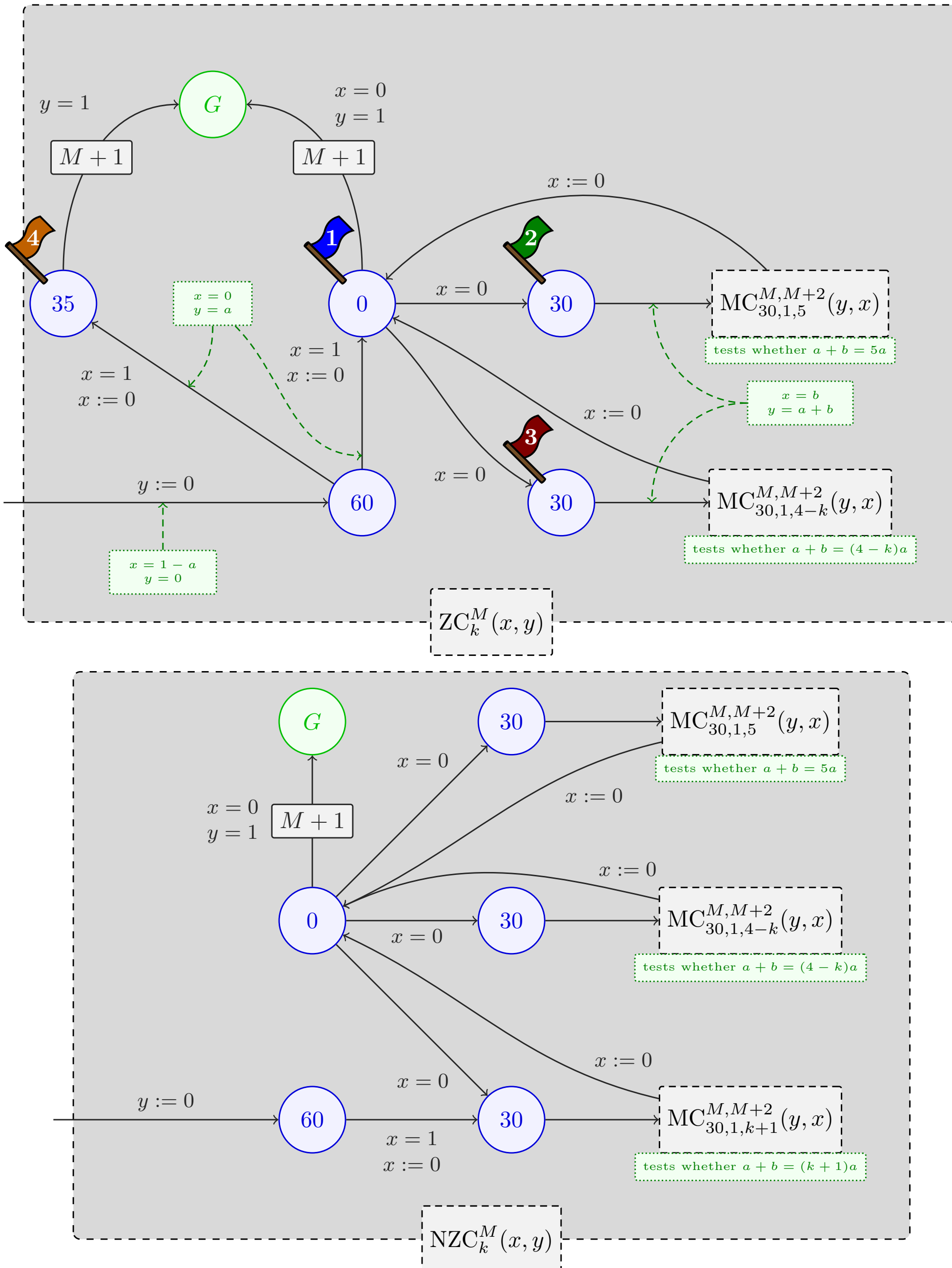


Fig. 4. The Zero-Control and Non-Zero-Control modules.

In these two modules, observe that MC is always invoked as $\mathrm{MC}(y, x)$, i.e., by swapping the roles of the two clocks. This is because it is easier to translate the encoding $(1 - a, 0)$ into $(0, a)$ rather than $(a, 0)$. This translation is the first step of both modules.

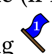
Proposition 3.5. *Let $k \in \{1, 2\}$. Let $1 - a \in [0, 1)$ be the initial value of x upon entering the module $\mathrm{ZC}_k^M(x, y)$. Let $30(1 - a) + E$ be the overall cost accumulated to date upon entering the module. Let*

$$\mu = \min \left\{ \left| \frac{1}{(4-k)^d 5^n} - a \right| : d, n \in \mathbb{N} \right\} .$$

Then

- **Min** *has a strategy that ensures a final cost of at most* $61 + M + E + 5\mu$.
- **Max** *has a strategy that ensures a final cost of at least* $61 + M + E + \mu$.

Proof. Let $p \in \mathbb{N} \setminus \{0\}$. We introduce the following quantities:

- a_p is the value of clock y upon entering  for the p-th time (if it exists).
- C_p is the accumulated cost so far upon entering  for the p-th time (if it exists).
- t_p is the time spent by **Min** in one of the states  or  upon leaving  for the p-th time (if it exists).
- ε_p is the time **Max** waits in one of the locations of MC controlled by him for the p-th time (if it exists).
- $E_p = C_p - 30(1 + a_p)$ is the difference between the expected and actual costs.

We have the following initial conditions:

$$a_1 = a \qquad C_1 = 30(1 + a) + E_1 \qquad \text{and} \qquad E_1 = E\,.$$

When everything is well defined we have the following recurrence relations:

$$a_{p+1} = a_p + t_p + \varepsilon_p \qquad C_{p+1} = C_p + 30t_p \qquad \text{and} \qquad E_{p+1} = E_p - 30\varepsilon_p\,.$$

Overall we have

$$a_p = a + \sum_{q=0}^{p-1}(t_q + \varepsilon_q) \qquad E_p = E - 30\left(\sum_{q=0}^{p-1}\varepsilon_q\right) \qquad \text{and} \qquad C_p = 30(1 + a_p) + E_p\,.$$

- Let us prove the first assertion of the proposition. Let d_p and n_p be integers that minimise $\mu_p \overset{\text{def}}{=} |\delta_p|$ for $\delta_p \overset{\text{def}}{=} \frac{1}{(4-k)^{d_p} 5^{n_p}} - a_p$. Now consider the following strategy for **Min**: If $0 < 1 - a \leq \frac{3-k}{8-2k}$ then **Min** takes the transition to State . Otherwise **Min** moves to State . If **Min** is in  for the p-th time then:

(1) If $d_p \geq 1$ then go to State  and wait $t_p = (3 - k)a_p + (4 - k)\delta_p$. Doing so, y has value $\frac{1}{(4-k)^{d_p - 1} 5^{n_p}}$ upon entering the MC module. Note that $t_p \geq 0$, since otherwise

$$\frac{1}{(4-k)^{d_p} 5^{n_p}} < \frac{1}{(4-k)^{d_p - 1} 5^{n_p}} < a_p$$

and thus

$$\left| \frac{1}{(4-k)^{d_p - 1} 5^{n_p}} - a_p \right| < \mu_p$$

which is a contradiction.

(2) If $d_p = 0$ and $n_p \geq 1$ then go to state  and wait $t_p = 4a_p + 5\delta_p$. Doing so, y has value $\frac{1}{5^{n_p - 1}}$ upon entering the MC module. Note that for the same reason as earlier, $t_p \geq 0$.

(3) If $d_p = n_p = 0$ and $a_p < 1$, then go to state  and wait $1 - a_p$.
(4) If $a_p = 1$, then go to the goal state.

Note that following this strategy, **Min** always selects t_p in such a way that $a_p + t_p$ is of the form $\frac{1}{(4-k)^d 5^n}$. Hence if $p > 1$ then

$$\varepsilon_{p-1} \in \left\{ \left| \frac{1}{(4-k)^d 5^n} - a_p \right| : d, n \in \mathbb{N} \right\} .$$

Therefore $\mu_p \leq \varepsilon_{p-1}$. We then have the following alternatives:

- **Min** goes to state . In this case, the final cost is

$$30(1-a) + E + 60a + 35(1-a) + M + 1 = 61 + M + E + 5(1-a) .$$

  Also here we have $a \geq \frac{5-k}{8-2k} = \frac{1}{2}\left(1 + \frac{1}{4-k}\right)$, which entails that $\mu = 1 - a$.
- **Max** decides to end the game after Case 1 of the above strategy.
  If $p = 1$, using Cor. 3.4, with optimal play from **Max**, the final cost is exactly

$$61 + M + E + (4-k)\mu_1 \leq 61 + M + E + 5\mu .$$

  If $p > 1$, using Cor. 3.4, with optimal play from **Max**, the final cost is exactly

$$\begin{aligned} 61 + M + E_p + (4-k)\mu_p &= 61 + M + E - 30\sum_{q=0}^{p-1}\varepsilon_q + (4-k)\mu_p \\ &\leq 61 + M + E - 30\sum_{q=0}^{p-2}\varepsilon_q - (26+k)\varepsilon_{p-1} \text{ since } \mu_p \leq \varepsilon_{p-1} \\ &\leq 61 + M + E + 5\mu . \end{aligned}$$

- **Max** decides to end the game after Case 2 of the above strategy. If $p = 1$, using Cor. 3.4, with optimal play from **Max**, the final cost is exactly

$$61 + M + E + 5\mu .$$

  If $p > 1$, using Cor. 3.4, with optimal play from **Max**, the final cost is exactly

$$\begin{aligned} 61 + M + E_p + 5\mu_p &= 61 + M + E - 30\sum_{q=0}^{p-1}\varepsilon_q + 5\mu_p \\ &\leq 61 + M + E - 30\sum_{q=0}^{p-2}\varepsilon_q - 25\varepsilon_{p-1} \text{ since } \mu_p \leq \varepsilon_{p-1} \\ &\leq 61 + M + E + 5\mu . \end{aligned}$$

- **Max** decides to end the game after Case 3 of the above strategy. First, observe that in this case $p > 1$. Indeed, in Case 3, $d_p = n_p = 0$, which in turn entails that $\frac{5-k}{8-2k} \leq a_p < 1$, i.e., a_p is closer to 1 than to $\frac{1}{4-k}$, which (for $p = 1$) mandates moving to state  instead of state .

Using Cor. 3.4, with optimal play from **Max**, the final cost is exactly

$$61 + M + E_p + |1 - (4-k)a_p| = 61 + M + E - 30\sum_{q=0}^{p-1}\varepsilon_q + |1 - (4-k)a_p|\,.$$

Following **Min**'s strategy, we know that $a_{p-1} + t_{p-1}$ is of the form $\dfrac{1}{(4-k)^d 5^n}$ with either $d > 0$ or $n > 0$. Therefore $a_{p-1} + t_{p-1} \leq \dfrac{1}{4-k}$ and $a_p \geq \dfrac{5-k}{8-2k}$, hence $\varepsilon_{p-1} \geq \dfrac{3-k}{8-2k}$. Therefore $|1-(4-k)a_p| \leq 3-k \leq 30\varepsilon_{p-1}$, and

$$61 + M + E - 30\sum_{q=0}^{p-1}\varepsilon_q + |1-(4-k)a_p| \leq 61 + M + E - 30\sum_{q=0}^{p-2}\varepsilon_q \leq 61 + M + E + 5\mu\,.$$

– Finally if the game ends via Case 4 of the above strategy then the final cost is exactly

$$61 + M + E_p = 61 + M + E - 30\sum_{q=0}^{p-1}\varepsilon_q \leq 61 + M + E + 5\mu\,.$$

In all the above cases, **Min** ensures a cost of at most $61 + M + E + 5\mu$.

- Let us prove the second assertion of the proposition. Assume that **Min** has gone to State  instead of . Let us write:

– $k_p = \begin{cases} 5 & \text{if } \textbf{Min} \text{ chooses State [2] at step } p \\ 4-k & \text{if } \textbf{Min} \text{ chooses State [3] at step } p\,. \end{cases}$ (State flags:  and )

– $\eta_p = \left|t_p - (k_p - 1)a_p\right|$.

Consider the following strategy for **Max**:

(1) Always choose $\varepsilon_p = 0$.
(2) If **Max** is in an $L_{\mathbf{Max}}$ location of a MC module for the p-th time and if **Min** has just left State  with $\eta_p \geq \mu$ then immediately end the game through the path maximising the cost.
(3) Otherwise, immediately accept and go back to state .

Using Cor. 3.4, if **Max** ends the game in Case 2, he secures a final cost of

$$61 + M + E_p + \eta_p = 61 + M + E + \eta_p \geq 61 + M + E + \mu\,.$$

Assume now that we always have $\eta_p < \mu$, i.e., **Max** never ends the game by himself. By contradiction, let us show that **Min** can never achieve the condition $y = 1$ needed to exit from State . Since $\varepsilon_p = 0$ for all p, and since **Max** always takes the transition back to State , we have $t_p = a_{p+1} - a_p$ for all p. Therefore $\eta_p = |a_{p+1} - k_p a_p|$.
If the game has an infinite number of transitions then **Max** wins hence the value of the run is infinite (and thus greater than $61 + M + E + \mu$). If the game ends at step P for some $P \in \mathbb{N}$, we must have $a_P = 1$. Let us consider the following property for $p \in \{1, \ldots, P-1\}$:

$$(\mathscr{P}_p): \qquad \left|\prod_{q=1}^{p}\frac{1}{k_{P-q}} - a_{P-p}\right| < \mu\sum_{m=1}^{p}\ \prod_{q=m}^{p}\frac{1}{k_{P-q}}\,.$$

We prove this property by induction on p.

– We have $|1 - k_{P-1}a_{P-1}| = |a_P - k_{P-1}a_{P-1}| = \eta_{P-1} < \mu$. This can be rewritten as

$$\left|\frac{1}{k_{P-1}} - a_{P-1}\right| < \frac{\mu}{k_{P-1}}\,,$$

which is exactly $(\mathscr{P}_1)$.

– Assume that $(\mathscr{P}_p)$ has been proven for some $p \in \{1, \dots, P-2\}$. We have

$$\begin{aligned}\left|\prod_{q=1}^{p}\frac{1}{k_{P-q}} - k_{P-(p+1)}a_{P-(p+1)}\right| &\leq \left|\prod_{q=1}^{p}\frac{1}{k_{P-q}} - a_{P-p}\right| + \left|a_{P-p} - k_{P-(p+1)}a_{P-(p+1)}\right| \\ &\leq \left|\prod_{q=1}^{p}\frac{1}{k_{P-q}} - a_{P-p}\right| + \eta_{P-p} \\ &< \mu\left(\sum_{m=1}^{p}\ \prod_{q=m}^{p}\frac{1}{k_{P-q}}\right) + \mu\,.\end{aligned}$$

Thus

$$\left|\prod_{q=1}^{p+1}\frac{1}{k_{P-q}} - a_{P-(p+1)}\right| < \mu\left(\sum_{m=1}^{p}\ \prod_{q=m}^{p+1}\frac{1}{k_{P-q}}\right) + \frac{\mu}{k_{P-(p+1)}} = \mu\sum_{m=1}^{p+1}\ \prod_{q=m}^{p+1}\frac{1}{k_{P-q}}\,.$$

This proves $(\mathscr{P}_{p+1})$, which concludes our induction.

We have

$$\left|\prod_{q=1}^{P-1}\frac{1}{k_{P-q}} - a_1\right| < \mu\sum_{m=1}^{P-1}\ \prod_{q=m}^{P-1}\underbrace{\frac{1}{k_{P-q}}}_{<\frac{1}{4-k}} \leq \mu\sum_{m=1}^{P-1}\frac{1}{(4-k)^{P-m}} = \mu\underbrace{\sum_{m=1}^{P-1}\frac{1}{(4-k)^{m}}}_{<\frac{1}{3-k}<1} < \mu\,.$$

But we notice that

$$\left|\prod_{q=1}^{P-1}\frac{1}{k_{P-q}} - a_1\right| = \left|\prod_{q=1}^{P-1}\frac{1}{k_{P-q}} - a\right| \in \left\{\left|\frac{1}{(4-k)^d 5^n} - a\right| : d, n \in \mathbb{N}\right\}\,.$$

Thus by definition of μ,

$$\mu \leq \left|\prod_{q=1}^{P-1}\frac{1}{k_{P-q}} - a_1\right| < \mu\,,$$

a contradiction. Intuitively, in order to exit from State , **Min** needs to compensate the gap μ, and cannot do so without triggering case (2) of **Max**'s strategy. Therefore we conclude that **Min** never transitions to the goal state from State . This means that the strategy for **Max** under consideration ensures a cost of at least $61 + M + E + \mu$, as required. □

PROPOSITION 3.6. *Let $k \in \{1, 2\}$. Let $1 - a \in [0, 1)$ be the initial value of x upon entering module $\mathrm{NZC}_k^M(x, y)$. Let $30(1 - a) + E$ be the overall cost accumulated thus far when entering the module. Let*

$$\mu = \min\left\{\left|\frac{1}{(k+1)^c(4-k)^d 5^n} - a\right| : c, d, n \in \mathbb{N} \quad c > 0\right\}\,.$$

Then

- **Min** *has a strategy that ensures a final cost of at most* $61 + M + E + 5\mu$.
- **Max** *has a strategy that ensures a final cost of at least* $61 + M + E + \mu$.

PROOF. The proof is almost identical to that of Prop. 3.5. The main difference between modules ZC_k^M and NZC_k^M is that NZC_k^M forces **Min** to do a multiplication by $k+1$ before multiplying by $k+1$, $4-k$, and 5 arbitrarily many times. □

3.4 Combining modules

Definition 3.7. Let $\mathcal{M} = (Q, q_i, q_h, T)$ be a deterministic two-counter machine. We assume without loss of generality that q_h has no outgoing transitions. Define the WTG $\mathcal{G}_{\mathcal{M}} = (L_{\mathbf{Min}}, L_{\mathbf{Max}}, G, \mathcal{X}, T', w)$ as follows:

- $L_{\mathbf{Min}}$ contains Q.
- For $q \in Q$, $w(q) = 30$.
- $\mathcal{X} = \{x, y\}$.
- For every $e = (q, k, q') \in T$, with $q, q' \in Q$ and $k \in \{1, 2\}$, we add the module T_e to $\mathcal{G}_{\mathcal{M}}$ defined as follows:

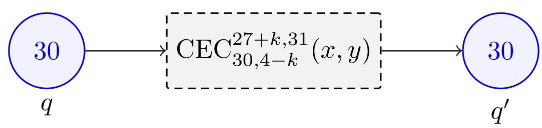


Fig. 5. Transition module for increments.

- For every $e = (q, k, q_z, q_{nz}) \in T$, with $q, q_z, q_{nz} \in Q$ and $k \in \{1, 2\}$, we add the module T_e to $\mathcal{G}$ defined as follows:

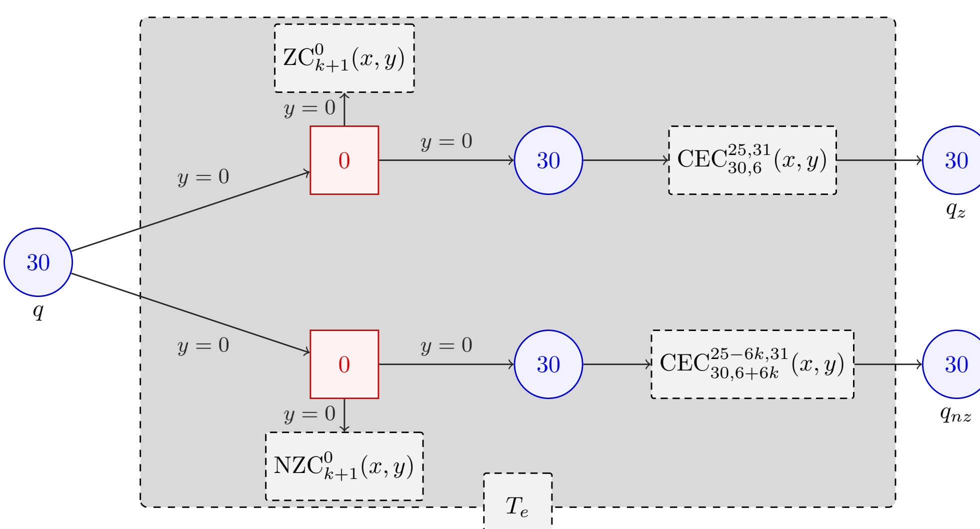


Fig. 6. Transition module for branches.

- Finally, for every $q \in Q$ (including q_h), we add the following exit module for **Min**:

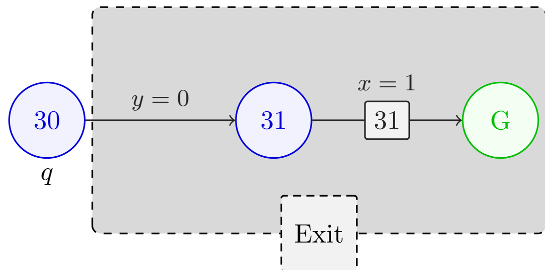


Fig. 7. **Min**'s exit module.

Proposition 3.8. *Let $\mathcal{M} = (Q, q_i, q_h, T)$ be a deterministic two-counter machine.*

- *If $\mathcal{M}$ does not halt then $\mathcal{G}_{\mathcal{M}}$, starting from configuration $(q_i, \mathbf{0})$, has value at most* 61.
- *If $\mathcal{M}$ halts in at most N steps then $\mathcal{G}_{\mathcal{M}}$, starting from configuration from $(q_i, \mathbf{0})$, has value at least* $61 + \frac{11}{12 \times 30^{5N}}$.

Proof. We first introduce some key quantities. Let $p \in \mathbb{N} \setminus \{0\}$. We let $(q_p, c_p, d_p)_p$ be the unique sequence of states and values of the two counters along the (deterministic) execution of $\mathcal{M}$. Here, for all p, $c_p + d_p \leq p$. Recall that the locations of $\mathcal{G}_{\mathcal{M}}$ comprise those contained in Q. We let $(\widehat{q}_p)$ be the sequence of states in Q visited along a given play of $\mathcal{G}_{\mathcal{M}}$. Let us write, upon entering a state $\widehat{q}_p$:

- a_p to denote the value of clock x.
- C_p for the accumulated cost so far.
- $E_p = C_p - 30a_p$ for the difference between the expected cost and the actual one.
- $\mu_p = \min \left\{ \left| 1 - \frac{1}{2^c 3^d 5^p} - a_p \right| : c, d \in \mathbb{N} \quad c + d \leq p \right\}$ to denote the minimal difference between a_p and a valid counter encoding.
- $\widehat{c}_p$ and $\widehat{d}_p$ to stand for natural numbers such that $\widehat{c}_p + \widehat{d}_p \leq p$ and
$$\left| 1 - \frac{1}{2^{\widehat{c}_p} 3^{\widehat{d}_p} 5^p} - a_p \right| = \mu_p \,.$$
- $\delta_p = 1 - \frac{1}{2^{\widehat{c}_p} 3^{\widehat{d}_p} 5^p} - a_p$.

We have the following initial conditions:

$$a_1 = 0 \qquad \mu_1 = \delta_1 = 0 \qquad E_1 = 0 \qquad \text{and} \qquad C_1 = 0 \,.$$

After entering a state $\widehat{q}_p$, and until visiting the next state $\widehat{q}_{p+1}$, **Min** can wait in a state of weight 30, immediately followed by a CEC where **Max** can also wait. We let t_p be the time spent by **Min** before the CEC module is entered, and ε_p be the time spent by **Max** in his state of the CEC module before $\widehat{q}_{p+1}$.

When everything is well defined we have the following recurrence relations:

$$a_{p+1} = a_p + t_p + \varepsilon_p \qquad C_{p+1} = C_p + 30t_p \qquad \text{and} \qquad E_{p+1} = E_p - 30\varepsilon_p \,.$$

Overall we have

$$a_p = \sum_{q=0}^{p-1} (t_q + \varepsilon_q) \qquad E_p = -30 \sum_{q=0}^{p} \varepsilon_q \qquad \text{and} \qquad C_p = 30a_p + E_p = 30 \sum_{q=10}^{p-1} t_q \,.$$

- Assume that $\mathcal{M}$ does not halt. Let $N \in \mathbb{N}$. Consider the following strategy for **Min**: upon entering q_p:

(1) If $1 - a_p < \dfrac{1}{30^N}$ then take the exit module.

(2) If $1 - \dfrac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} - a_p < 0$ then take the exit module.

(3) If $1 - \dfrac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} - a_p \geq 0$ then **Min** plays in order to reach q_{p+1}, and waits

$$t_p = 1 - \frac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} - a_p = \delta_p + \frac{1}{2^{c_p} 3^{d_p} 5^p} - \frac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} .$$

As long as the game continues, we have that for $p > 0$,

$$a_{p+1} = a_p + t_p + \varepsilon_p \qquad \text{and} \qquad a_p + t_p = 1 - \frac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} .$$

Therefore

$$\varepsilon_p \in \left\{ \left| 1 - \frac{1}{2^c 3^d 5^{p+1}} - a_{p+1} \right| : c, d \in \mathbb{N} \quad c + d \leq p \right\} ,$$

hence $\varepsilon_p \geq \mu_{p+1}$.

- If **Min** ends the game in Case 1, then the final cost is

$$C_p + 31(1 - a_p) + 31 = 61 + E_p + 1 - a_p < 61 + \frac{1}{30^N} .$$

- If **Max** ends the game in a ZC_k module, then Prop. 3.5 ensures that **Min** can secure a cost of at most

$$61 + E_p + 5 \min \left\{ \left| \frac{1}{(4-k)^d 5^n} - 1 + a_p \right| \,\middle|\, d, n \in \mathbb{N} \right\} \leq 61 + E_p + 5\mu_p .$$

Note that, if $p = 1$, then $a_p = 0$ and this quantity is actually $61 < 61 + \frac{1}{30^N}$. Otherwise, since $\varepsilon_{p-1} \geq \mu_p$, **Min** can secure a cost of at most

$$61 - 30 \sum_{q=0}^{p-2} \varepsilon_q - 30\varepsilon_{p-1} + 5\varepsilon_{p-1} \leq 61 < 61 + \frac{1}{30^N} .$$

- Similarly, if **Max** ends the game in a NZC_k module, then Prop. 3.6 ensures that **Min** and secure a cost of at most $61 < 61 + \frac{1}{30^N}$.
- If **Max** ends the game in a $\mathrm{CEC}_{30,\beta}^{31-\beta,31}$ module, Cor. 3.2 ensures that the final cost is

$$61 + E_p + 30 \left| t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p) \right| .$$

Since **Min** plays according to the execution of $\mathcal{M}$, the parameter β is actually such that

$$\left(1 - \frac{\beta}{30}\right) \frac{1}{2^{c_p} 3^{d_p} 5^p} = \frac{1}{2^{c_p} 3^{d_p} 5^p} - \frac{1}{2^{c_{p+1}} 3^{d_{p+1}} 5^{p+1}} .$$

Hence

$$
\begin{aligned}
t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p) &= 1 - \frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} - a_p - \left(1 - \frac{\beta}{30}\right)\left(\delta_p + \frac{1}{2^{c_p}3^{d_p}5^p}\right) \\
&= 1 - \frac{1}{2^{c_p}3^{d_p}5^p} - a_p - \left(1 - \frac{\beta}{30}\right)\delta_p \\
&= \delta_p - \left(1 - \frac{\beta}{30}\right)\delta_p \\
&= \frac{\beta}{30}\delta_p \,.
\end{aligned}
$$

We then conclude that the final cost is

$$61 + E_p + \beta|\delta_p| = 61 + E_p + \beta\mu_p \,.$$

Thus if $p = 1$ then the cost is 61, and if $p > 1$, since $\mu_p \leq \varepsilon_{p-1}$ and $\beta \in \{2, 3, 6, 12, 18\}$, we have $\beta < 30$ and the final cost is at most

$$61 - 30\sum_{q=0}^{p-2}\varepsilon_q - (30 - \beta)\varepsilon_{p-1} \leq 61 < 61 + \frac{1}{30^N} \,.$$

– Finally, if **Min** ends the game in Case 2 then the total cost is

$$61 + E_p + 1 - a_p = 61 - 30\sum_{q=0}^{p-2}\varepsilon_q - 30\varepsilon_{p-1} + 1 - a_p \,.$$

By assumption, in Case 2 we have

$$1 - a_p < \frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \,.$$

However, since

$$a_{p-1} + t_{p-1} = 1 - \frac{1}{2^{c_p}3^{d_p}5^p} \,,$$

we know that **Max** has made an important mistake to trigger Case 2:

$$
\begin{aligned}
\varepsilon_{p-1} = a_p - a_{p-1} - t_{p-1} &= a_p - 1 + \frac{1}{2^{c_p}3^{d_p}5^p} \\
&> \frac{1}{2^{c_p}3^{d_p}5^p} - \frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \,.
\end{aligned}
$$

Note also that there is $\gamma \in \left\{\frac{2}{5}, \frac{3}{5}, \frac{1}{5}, \frac{1}{10}\frac{1}{15}\right\}$ such that

$$\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} = \gamma\frac{1}{2^{c_p}3^{d_p}5^p} \,.$$

Thus

$$\frac{1}{2^{c_p}3^{d_p}5^p} \geq \frac{5}{3}\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \qquad \text{and} \qquad \varepsilon_{p-1} > \frac{2}{3}\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \,.$$

Combining all of this, the final cost after the exit module is at most

$$\begin{aligned} 61 - 30\sum_{q=0}^{p-2}\varepsilon_q - 30\varepsilon_{p-1} + 1 - a_p &< 61 - 30\sum_{q=0}^{p-2}\varepsilon_q - \frac{2}{3}30\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} + \frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \\ &< 61 - 30\sum_{q=0}^{p-2}\varepsilon_q - 19\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}} \\ &< 61 < 61 + \frac{1}{30^N}\,. \end{aligned}$$

Overall, for any $N \in \mathbb{N}$, **Min** can secure a cost of at most $61 + \frac{1}{30^N}$. Thus the value of the game $\mathcal{G}_{\mathcal{M}}$ is at most 61.

- Assume that $\mathcal{M}$ halts in N steps. Consider the following strategy for **Max**:
(1) Always choose $\varepsilon_p = 0$.
(2) Immediately punish a wrong transition going to ZC_k^0 or NZC_k^0 if **Min** unfaithfully simulates a zero-test.
(3) If **Max** is in his state of a $\mathrm{CEC}_{30,\beta}^{31-\beta,31}$ module for the p-th time then accept the transition if $\left|t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p)\right| < \frac{1}{30^{5N+1}}$ and punish otherwise.

Note that, in particular, $E_p = 0$ for all p. We show that as long as q_p exists (i.e., $p \leq N$) and that the game runs at least until q_p is reached, the following property holds:

$$(\mathscr{P}_p) : (\widehat{c_p}, \widehat{d_p}) = (c_p, d_p) \wedge \mu_p \leq \frac{1}{12 \times 30^{5N}}\,.$$

- $(\mathscr{P}_1)$ holds by definition.
- Assume that $(\mathscr{P}_p)$ holds and that state q_{p+1} exists. Supposing that we reach q_{p+1} entails that **Min** did choose the correct transition in case of zero-tests and that she did not exit via the exit module. The game also did go through a $\mathrm{CEC}_{30,\beta}^{31-\beta,31}$ module where β is such that

$$\frac{\beta}{30}\frac{1}{2^{c_p}3^{d_p}5^p} = \frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}\,.$$

Also, according **Max**'s strategy, we must have

$$\left|t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p)\right| < \frac{1}{30^{5N+1}}\,.$$

since otherwise **Max** would have ended the game. Thus

$$\begin{aligned}\left|1-\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}-a_{p+1}\right| &= \left|1-\frac{\beta}{30}\frac{1}{2^{c_p}3^{d_p}5^p}-a_p-t_p\right| \\ &= \left|1-a_p-\frac{\beta}{30}\left(1-a_p-\delta_p\right)-t_p\right| \\ &= \left|\left(1-\frac{\beta}{30}\right)(1-a_p)+\frac{\beta}{30}\delta_p-t_p\right| \\ &\leq \frac{\beta}{30}\mu_p+\left|t_p-\left(1-\frac{\beta}{30}\right)(1-a_p)\right| \\ &< \frac{3}{5}\mu_p+\frac{1}{30^{5N+1}} \\ &\leq \frac{3}{5}\frac{1}{12\times 30^{5N}}+\frac{1}{30^{5N+1}}=\frac{1}{12\times 30^{5N}}\,.\end{aligned}$$

Also, since $5^{2(p+1)} > 2^{c_{p+1}}3^{d_{p+1}}5^{p+1}$, we have

$$\begin{aligned}&\min\left\{\left|\frac{1}{2^c3^d5^n}-\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}\right| : \begin{array}{c} c,d,n\in\mathbb{N} \\ c+d\leq n \\ (c,d,n)\neq(c_{p+1},d_{p+1},p+1)\end{array}\right\} \\ &= \min\left\{\left|\frac{1}{2^c3^d5^n}-\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}\right| : \begin{array}{c} c,d,n\in\{0,\ldots,2(p+1)\} \\ c+d\leq n \\ (c,d,n)\neq(c_{p+1},d_{p+1},p+1)\end{array}\right\}\end{aligned}$$

Observe that any element of the above set is a multiple of $\frac{1}{30^{2(p+1)}}$ and cannot be 0. Therefore

$$\min\left\{\left|\frac{1}{2^c3^d5^n}-\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}\right| : \begin{array}{c} c,d,n\in\{0,\ldots,2(p+1)\} \\ c+d\leq n \\ (c,d,n)\neq(c_{p+1},d_{p+1},p+1)\end{array}\right\}\geq\frac{1}{30^{2(p+1)}}\geq\frac{1}{30^{2N}}$$

because, by assumption, q_{p+1} exists hence $p+1\leq N$. Hence, for any c,d,n such that $c+d\leq n$ and differ from c_{p+1}, d_{p+1}, and $p+1$, respectively, we have

$$\left|1-\frac{1}{2^c3^d5^n}-a_p\right|\geq\frac{1}{30^{2N}}-\frac{1}{12\times 30^{5N}}>\frac{1}{12\times 30^{5N}}\,.$$

Therefore $(\widehat{c}_{p+1},\widehat{d}_{p+1})=(c_{p+1},d_{p+1})$ and

$$\mu_{p+1}=\left|1-\frac{1}{2^{c_{p+1}}3^{d_{p+1}}5^{p+1}}-a_{p+1}\right|\leq\frac{1}{12\times 30^{5N}}\,.$$

By induction, we conclude that $(\mathscr{P}_p)$ holds for any $p\leq P$, where $P\leq N$ is the number of consecutive states of $\mathcal{M}$ that are correctly simulated by the play of our game $\mathcal{G}_{\mathcal{M}}$.

– Assume that **Min** ends the game at step $p\leq N$. The final cost is then

$$61+1-a_p=61+\delta_p+\frac{1}{2^{c_p}3^{d_p}5^p}\geq 61+\delta_p+\frac{1}{5^{2N}}\geq 61+\frac{1}{5^{2N}}-\frac{1}{12\times 30^{5N}}\geq 61+\frac{11}{12\times 30^{5N}}\,.$$

– Assume that **Max** ends the game with the ZC_1^0 module. Here we have $c_p \neq 0$. Then, using Prop. 3.5, **Max** can secure a cost of

$$\begin{aligned} & 61 + \min\left\{\left|\frac{1}{3^d 5^n} - 1 + a_p\right| : d, n \in \mathbb{N}\right\} \\ & = 61 + \min\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p} - \delta_p\right| : d, n \in \mathbb{N}\right\} \\ & \geq 61 + \min\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : d, n \in \mathbb{N}\right\} - \mu_p \\ & \geq 61 + \min\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : d, n \in \mathbb{N}\right\} - \frac{1}{12 \times 30^{5N}} . \end{aligned}$$

Since $c_p + d_p \leq p$, we have $3^{3p} > 5^{2p} > 2^{c_p} 3^{d_p} 5^p$ and hence

$$\min\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : d, n \in \mathbb{N}\right\} = \min\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : \begin{array}{l} d \in \{0, \ldots, 3p\} \\ n \in \{0, \ldots, 2p\} \end{array}\right\} .$$

Note that all the elements in $\left\{\left|\frac{1}{3^d 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : \begin{array}{l} d \in \{0, \ldots, 3p\} \\ n \in \{0, \ldots, 2p\} \end{array}\right\}$ are multiples of $\frac{1}{30^{3p}}$ and cannot be 0 since $c_p \neq 0$. Also, since we are simulating a transition, we must have $p \leq N$. We can then conclude that **Max** can secure a cost of at least

$$61 + \frac{1}{30^{3p}} - \frac{1}{12 \times 30^{5N}} \geq 61 + \frac{11}{12 \times 30^{5N}} .$$

Similar reasoning and calculations are used in the three following cases.

– Assume that **Max** ends the game with the ZC_2^0 module. Here, we have $d_p \neq 0$. Then using Prop. 3.5, **Max** can secure a cost of

$$61 + \min\left\{\left|\frac{1}{2^c 5^n} - \frac{1}{2^{c_p} 3^{d_p} 5^p}\right| : c, n \in \mathbb{N}\right\} - \frac{1}{12 \times 30^{5N}} .$$

Using the fact that $c_p + d_p \leq p$, $2^{5p} > 5^{2p} > 2^{c_p} 3^{d_p} 5^p$, we conclude that **Max** can secure a cost of at least

$$61 + \frac{1}{30^{5p}} - \frac{1}{12 \times 30^5 N} \geq 61 + \frac{11}{12 \times 30^{5N}} .$$

– Assume that **Max** ends the game with the NZC_1^0 module. Here, we have $c_p = 0$. Then using Prop. 3.6, **Max** can secure a cost of

$$61 + \min\left\{\left|\frac{1}{2^c 3^d 5^n} - \frac{1}{3^{d_p} 5^p}\right| : c, d, n \in \mathbb{N} \quad c > 0\right\} - \frac{1}{12 \times 30^{5N}} .$$

Using that $d_p \leq p$, $2^{5p} > 3^{3p} > 5^{2p} > 3^{d_p} 5^p$ and we conclude that **Max** can secure a cost of at least

$$61 + \frac{1}{30^{5p}} - \frac{1}{12 \times 30^{5N}} \geq 61 + \frac{11}{12 \times 30^{5N}} .$$

– Assume that **Max** ends the game with the NZC_2^0 module. We then just switch the roles of c and d in this previous case to get that **Max** can secure a cost of at least

$$61 + \frac{1}{30^{5p}} - \frac{1}{12 \times 30^{5N}} \geq 61 + \frac{11}{12 \times 30^{5N}} .$$

– If **Max** ends the game in a $\mathrm{CEC}_{30,\beta}^{31-\beta,31}$ module, Cor. 3.2 ensures that the final cost is

$$61 + E_p + 30\left|t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p)\right| .$$

By assumption, in this case we have $\left|t_p - \left(1 - \frac{\beta}{30}\right)(1 - a_p)\right| \geq \frac{1}{30^{5N+1}}$, leading to a cost of at least $61 + \frac{1}{30^{5N}} \geq 61 + \frac{11}{12\times 30^{5N}}$.

In all cases in which we reach the goal state, the cost is at least $61 + \frac{11}{12\times 30^{5N}}$, which concludes the proof. □

THEOREM 1.1. *The Value Problem for two-player, turn-based, time-bounded, two-clock, weighted timed games with non-negative integer weights is undecidable. The same holds for weighted timed games over unbounded time otherwise satisfying the same hypotheses.*

PROOF. Let $\mathcal{M}$ be a deterministic two-counter machine. Note that in $\mathcal{G}_{\mathcal{M}}$, clock x is never reset (except in some specific exiting modules ending the game) and is always upper-bounded by 1. Therefore any play has duration at most 1 time unit plus the total time spent in the control modules, which one easily verifies is at most 2 time units. In other words, by construction the WTG $\mathcal{G}_{\mathcal{M}}$ requires at most 3 time units for any execution. Prop. 3.8 asserts that the halting problem for a deterministic two-counter machine reduces to the Value problem for 2-clock weighted timed games, which concludes the proof. □

THEOREM 3.9. *The Existence Problem for two-player, turn-based, time-bounded, two-clock, weighted timed games with non-negative integer weights is undecidable. The same holds for weighted timed games over unbounded time otherwise satisfying the same hypotheses.*

PROOF SKETCH. Let $\mathcal{M}$ be a deterministic two-counter machine. We define $\mathcal{G}_{\mathcal{M}}^{E}$ similarly to $\mathcal{G}_{\mathcal{M}}$, except that we add a "soft-exit" module for **Min** to halting states of $\mathcal{M}$. The soft-exit module is simply the exit module where the location cost of 31 has been replaced by 30. Then **Min** has a strategy to enforce a cost of at most 61 iff $\mathcal{M}$ halts.

Indeed, if $\mathcal{M}$ halts, then **Min** can reach a halting state without cheating (i.e., she faithfully simulates $\mathcal{M}$), and exits at cost 61 through a soft-exit module. If **Max** chooses to leave through a CEC or MC module, this also yields a cost of at most 61. As seen in Prop. 3.8, every delay taken by **Max** in a CEC or MC module is a net negative for **Max**. On the other hand, if $\mathcal{M}$ does not halt, the only way for **Min** to exit the game through a soft-exit module is to cheat in order to reach a halting state. The normal exit module for **Min** yields cost strictly greater than 61. Consider the strategy where **Max** punishes any cheating by exiting the game, no matter how small. Then either **Min** never cheats, and the game never ends, thereby incurring cost $+\infty$, or **Min** cheats and is punished, in which case the game ends with cost strictly above 61. □

References

[1] Rajeev Alur, Mikhail Bernadsky, and P. Madhusudan. 2004. Optimal Reachability for Weighted Timed Games. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 122–133.

[2] Rajeev Alur and Thomas A. Henzinger. 1997. Modularity for Timed and Hybrid Systems. In *CONCUR (Lecture Notes in Computer Science, Vol. 1243)*. Springer, 74–88.

[3] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. 2005. Optimal Strategies in Priced Timed Game Automata. In *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 148–160.

[4] Patricia Bouyer, Samy Jaziri, and Nicolas Markey. 2015. On the Value Problem in Weighted Timed Games. In *Proceeding of the 26th International Conference on Concurrency Theory (CONCUR 2015) (Leibniz International Proceedings in*

Informatics (LIPIcs), Vol. 42), Luca Aceto and David de Frutos Escrig (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 311–324. doi:10.4230/LIPIcs.CONCUR.2015.311

[5] Patricia Bouyer, Kim Guldstrand Larsen, Nicolas Markey, and Jacob Illum Rasmussen. 2006. Almost Optimal Strategies in One Clock Priced Timed Games. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4337)*, S. Arun-Kumar and Naveen Garg (Eds.). Springer, 345–356. doi:10.1007/11944836_32

[6] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. 2011. On Reachability for Hybrid Automata over Bounded Time. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 6756)*, Luca Aceto, Monika Henzinger, and Jirí Sgall (Eds.). Springer, 416–427. doi:10.1007/978-3-642-22012-8_33

[7] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. 2013. Time-Bounded Reachability for Monotonic Hybrid Automata: Complexity and Fixed Points. In *Automated Technology for Verification and Analysis - 11th International Symposium, ATVA 2013, Hanoi, Vietnam, October 15-18, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8172)*, Dang Van Hung and Mizuhito Ogawa (Eds.). Springer, 55–70. doi:10.1007/978-3-319-02444-8_6

[8] Thomas Brihaye, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi. 2014. Adding Negative Prices to Priced Timed Games. In *CONCUR 2014 – Concurrency Theory*, Paolo Baldan and Daniele Gorla (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 560–575.

[9] Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. 2023. Optimal controller synthesis for timed systems. *Log. Methods Comput. Sci.* 19, 1 (2023).

[10] Quentin Guilmant and Joël Ouaknine. 2024. Inaproximability in Weighted Timed Games. In *35th International Conference on Concurrency Theory (CONCUR 2024) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 311)*, Rupak Majumdar and Alexandra Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 27:1–27:15. doi:10.4230/LIPIcs.CONCUR.2024.27

[11] Thomas A. Henzinger, Benjamin Horowitz, and Rupak Majumdar. 1999. Rectangular Hybrid Games. In *CONCUR (Lecture Notes in Computer Science, Vol. 1664)*. Springer, 320–335.

[12] Mark Jenkins, Joël Ouaknine, Alexander Rabinovich, and James Worrell. 2010. Alternating Timed Automata over Bounded Time. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*. IEEE Computer Society, 60–69. doi:10.1109/LICS.2010.45

[13] Oded Maler, Amir Pnueli, and Joseph Sifakis. 1995. On the Synthesis of Discrete Controllers for Timed Systems (An Extended Abstract). In *STACS (Lecture Notes in Computer Science, Vol. 900)*. Springer, 229–242.

[14] Marvin L. Minsky. 1967. *Computation: Finite and Infinite Machines*. Prentice-Hall. 255–258 pages.

[15] Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. 2022. Decidability of One-Clock Weighted Timed Games with Arbitrary Weights. In *33rd International Conference on Concurrency Theory (CONCUR 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 243)*, Bartek Klin, Sławomir Lasota, and Anca Muscholl (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 15:1–15:22. doi:10.4230/LIPIcs.CONCUR.2022.15

[16] Joël Ouaknine, Alexander Rabinovich, and James Worrell. 2009. Time-Bounded Verification. In *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings (Lecture Notes in Computer Science, Vol. 5710)*, Mario Bravetti and Gianluigi Zavattaro (Eds.). Springer, 496–510. doi:10.1007/978-3-642-04081-8_33

[17] Joël Ouaknine and James Worrell. 2010. Towards a Theory of Time-Bounded Verification. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 6199)*, Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis (Eds.). Springer, 22–37. doi:10.1007/978-3-642-14162-1_3