Permissive multi-strategies in timed games

Erwin Fang

March 2014

Research report LSV-14-04 (Version 1)

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Permissive multi-strategies in timed games*

Erwin Fang[a,b]

[a]RWTH Aachen University, Germany
[b]LSV, CNRS & École Normale Supérieure de Cachan, France

August 25, 2013

Under the guidance of

Patricia Bouyer-Decitre and Nicolas Markey
LSV, CNRS & École Normale Supérieure de Cachan, France

Abstract

This report deals with non-deterministic timed strategies, also called timed multi-strategies, for two-player games played between a controller and the environment on the structure of a timed automaton. The motivation of the concept of multi-strategies in timed systems is to have a certain flexibility of which action to be executed. We introduce a penalty for the controller as a quantitative measure for permissiveness of such strategies. Hereby, the aim is to prevent punctual timed requirements that are not realistic for practical controller-synthesis since real-time systems are subject to perturbation and imprecision of time. We present a game on two slightly different settings with a reachability objective. The aim of the controller is to keep the penalty as low as possible. We propose a fixed point computation for computing the optimal penalty for the controller and show that the most permissive timed multi-strategy for timed automata with one clock is computable in polynomial time. For the general case, we show that the most permissive strategy exists and that memoryless strategies are sufficient.

Acknowledgements I would like to express my deepest gratitude to my supervisors Patricia Bouyer-Decitre and Nicolas Markey. They managed to inspire me to the research topic during the class about foundations of real-time systems verification and throughout the internship. I am very thankful for all the fruitful discussions with Patricia and Nicolas. Especially, I want to thank Nicolas for proofreading the report and his generous support throughout the whole internship. Moreover, I am very grateful to have got the opportunity to take part in several scientific meetings and events through the Casting project and LSV. This internship was my greatest research experience so far. Thank you very much Patricia and Nicolas!

---

*Ce rapport est rédigé en anglais parce que la connaissance du français de l'auteur n'est pas suffisante.

Contents

1 Introduction

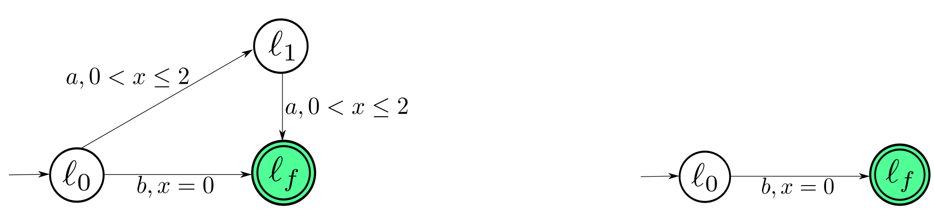
Background Demands and diversity of computerized systems have increased dramatically in the last decade. In particular, software and hardware systems have become one of the most complex artifacts built by humans. Reliability and correctness with respect to some properties or safety requirements are essential, especially in safety critical systems like embedded systems in airplanes, cars or nuclear power plants. Consequently, it is of high importance to *verify* whether the system is correctly implemented. However, this is a very difficult task to do manually which is the reason why *automatic verification* has to be considered. Therefore, it is necessary to define the system and the specifications formally. Using *formal methods*, the aim is to prove whether a system satisfies certain requirements or specifications. A popular and widely used approach for verification is *Model Checking*, which considers a model (abstraction) of a system and a specified property as an input and outputs whether or to which extents the property holds in that model. Commonly, such a model can be a finite state automaton or an extension of it, whereas a property is expressed by a logical formula.

In this work, we consider real-timed systems and use timed automata as an abstraction model. A timed automaton is a finite-state automaton extended with a finite number of clocks [1, 11]. Using these clocks, we are able to express timing constraints on the transitions as well as to reset a clock to zero.

One important key issue in investigating finite or infinite state systems is *reachability* that asks whether there exists an admissible run between two given states. In timed automata, reachability has been proven to be PSpace-complete for any finite number of clocks greater than two [8, 12] and NLogSpace-complete for one clock [1].

The problem Plain reachability is not a realistic approach for real-time systems in practice. On the one hand, real-time systems are usually subject to perturbation and disruption, while on the other hand, discretization of time may not be a sufficient abstract representation of the original system. This is why imprecision of time should be taken into account. Further, timed automata may describe unrealistic strategies that ask the controller to act faster and faster, for example, to execute infinitely many actions within a finite amount of time.

The concept and our approach In this work, a way to describe "permissive reachability" in timed transition systems will be introduced, i.e., the existence of a play to a target location that



(a) Player 1 has a winning strategy with a penalty of 2.

(b) Player 1 has to pay infinite penalty and loses.

Figure 1: Two intuitive examples of the reachability timed game with penalty.

is robust with respect to perturbation or imprecision of clocks. We model the scenario as a two-player reachability game played on a structure of a timed transition system. This kind of game is a convenient modeling paradigm of a system where a controller (Player 1) interacts with its environment (Player 2). The interesting question is whether there exists a *winning strategy* for Player 1, i.e., a way how the controller reacts to any possible action from the environment in order to reach some target state. Our goal is to measure the *permissiveness* of a strategy with respect to reachability for the controller independently of how the environment "behaves".

The concept of permissive strategies in games [3, 4] indicates whether or not more than one behavior as a move is allowed. We call a strategy multi-strategy when at each move in a play a set of actions instead of one precise action will be proposed. Multi-strategies can be seen as a generalization of strategies and can furthermore easily cope with errors caused by the environment. We introduce a penalty for Player 1 as a measure of permissiveness of a multi-strategy. The lower the penalty, the more permissive is the multi-strategy. In our case of timed games, the penalty can be assigned according to Player 1's requirement of time delay. Our main goal is to *avoid punctual timed requirements* since these are not realistic requirements in practice. Those shall be therefore "punished" as hard as possible by assigning an infinitely high penalty such that Player 1 loses. In the game, Player 1's objective is to reach a target location with as less penalty as possible. Player 2 has the contrary objective to prevent Player 1 reaching any target location and also to maximize his penalty. With this notion of permissiveness in timed multi-strategy, Player 1 loses when he chooses one discrete time point in a move. In Figure 1 we give two examples in order to give some basic intuition about this game. The goal in both structures is to reach ℓ_f. Figure 1a shows an example where Player 1 can win with a penalty of 2 by playing action a with the delay interval $(0, 1]$ in location ℓ_0 and ℓ_1. Figure 1b shows an example where Player 1 has no strategy to reach ℓ_f with a penalty less than $+\infty$. Player 1 has no other choice than choosing a punctual requirement in order to reach ℓ_f and gets an infinite penalty assigned, otherwise he stays in ℓ_0 and never reaches ℓ_f.

Related work The problem of perturbation in time has been intensively investigated within the community of real-time systems verification. A similar notion of our approach is the "*robustness analysis*" of timed automata [10, 15, 19] that decides whether a given timed automaton tolerates imprecision of time and eventually computes the maximum of tolerated imprecision. Further, robust implementation was investigated in [18] where the requirement is the possibility to automatically convert a given timed automaton to a timed automaton that is untimed language equivalent and tolerates perturbation of time.

In the domain of permissiveness of strategies in untimed games, some notions have been introduced for finite reachability games [4] and infinite mean-penalty parity games on weighted graphs [5]. In the finite game, one considers a multi-strategy that allows several outgoing moves at each vertex. The controller chooses a set of allowed edges and therefore blocks the remaining ones. The penalty will be assigned based on the number and weights of blocked edges. In this setting, optimal reachability has been shown to be PTime-complete [4]. Moreover, memoryless strategies are sufficient for the reachability objective. Regarding infinite game settings, the authors in [5] considered mean-penalty parity games on weighted graphs and proved that deciding a most permissive winning strategy is in NP $\cap$ Co-NP.

The goal of this work We introduce and formally define the notion of permissiveness in timed game. The goal is to determine the decidability and complexity of reachability problems. Further we propose a way how infinite games with penalty can be played and how the permissiveness can be measured. A classical question is whether winning strategies does not require memory, finite memory is sufficient or whether infinite memory is required. We will elaborate on the decidability on one clock and the memoryless property for the general case.

Structure The rest of this report is structured as follows: In Section 2, we introduce some arithmetic preliminaries that are necessary for understanding the contents and the basics of timed automaton. In Section 3, we define our game, the timed games with penalties. Further, we elaborate on pure reachability and turn-based reachability games, which includes decidability of the game played with one clock and that memoryless multi-strategies are sufficient for reachability games.

2 Preliminaries

In this section, we briefly introduce some arithmetic definitions on intervals, which we will use to describe our computation later, and the model of timed automaton.

2.1 Arithmetic preliminaries

In the beginning, we will introduce some basic notions and definitions that are essential for this work. We denote the set of non-negative integers by $\mathbb{N}$, non-negative rationals by $\mathbb{Q}_{\geq 0}$ and non-negative real numbers by $\mathbb{R}_{\geq 0}$.

- *Intervals over non-negative reals*
  The set of intervals over $\mathbb{R}_{\geq 0}$ will be denoted by

$$\mathcal{I}(\mathbb{R}_{\geq 0}) = \{I = \langle \alpha, \beta \rangle \mid \alpha \leq \beta, \alpha, \beta \in \mathbb{R}_{\geq 0} \cup \{+\infty\}\},$$

  where $\langle$ means either left-closed, [, or left-open, (, and $\rangle$ means either right-closed,], or right-open,). The length of an interval $I = \langle \alpha, \beta \rangle$ will be denoted as $|I| = \beta - \alpha$.

- *More-dimensional intervals over reals*
  The set of intervals over $\mathbb{R}_{\geq 0}$ of dimension n are defined as

$$\mathcal{I}(\mathbb{R}_{\geq 0})^n = \{(I_1, \ldots, I_n) \mid I_i \in \mathcal{I}(\mathbb{R})\}.$$

The definition of (more-dimensional) intervals over $\mathbb{N}$ and $\mathbb{Q}$ are analogue to the ones over reals as stated above. For our purpose, we define operations on (more-dimensional) intervals in the following way:

- *Addition of a vector and an interval*
  Given a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n_{\geq 0}$ and an interval $I = \langle \alpha, \beta \rangle \in \mathcal{I}(\mathbb{R}_{\geq 0})$, the addition is defined as

$$v + I = (\langle v_1 + \alpha, v_1 + \beta \rangle, \ldots, \langle v_n + \alpha, v_n + \beta \rangle) \quad \in \mathcal{I}(\mathbb{R}_{\geq 0})^n.$$

- *Addition of a vector and a multi-dimensional interval*
  Given a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n_{\geq 0}$ and a multi-dimensional interval $\mathbf{I} = (\langle \alpha_1, \beta_1 \rangle, \langle \alpha_2, \beta_2 \rangle, \ldots, \langle \alpha_n, \beta_n \rangle) \in \mathcal{I}(\mathbb{R}_{\geq 0})^n$, the addition is defined as

$$v + \mathbf{I} = (\langle v_1 + \alpha_1, v_1 + \beta_1 \rangle, \ldots, \langle v_n + \alpha_n, v_n + \beta_n \rangle) \quad \in \mathcal{I}(\mathbb{R}_{\geq 0})^n.$$

2.2 Timed automaton

2.2.1 The model

We use $\mathbb{R}_{\geq 0}$ as time domain. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a finite set of clocks. For any $x \in \mathcal{X}$, we define its valuation as a mapping $v : \mathcal{X} \to \mathbb{R}_{\geq 0}$. We denote the set of guards (clock constraints) by $\mathcal{C}(\mathcal{X})$ that is generated by the following grammar:

$$g ::= v(x) \sim c \mid g \wedge g \mid \neg g \mid true, \qquad \text{where } x \in C,\ c \in \mathbb{N},\ \sim \in \{<, \leq, =, \geq, >\}.$$

For the sake of simplicity, we write v for a valuation $v(x_1, \ldots, x_n) = (v(x_1), \ldots, v(x_n)) \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ of all clocks. For a clock $x \in \mathcal{X}$ and a constraint $x \sim c$, we say that v satisfies $x \sim c$ if, and only if, $v(x) \sim c$. For a clock valuation $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and a guard/constraint $g \in \mathcal{C}(\mathcal{X})$, we write $v \models g$ to indicate that v satisfies g. Given some $t \in \mathbb{R}_{\geq 0}$, $v+t$ is the valuation defined by $(v+t)(x) = v(x)+t$ for all $x \in \mathcal{X}$. For a subset of clocks $R \subseteq \mathcal{X}$, $v[R \leftarrow 0] =: v'$ is a valuation defined by $v'(x) = 0$ for all $x \in R$ and $v'(x) = v(x)$ for any $x \in \mathcal{X} \setminus R$.

A timed automaton is a five-tuple $\mathcal{A} = (L, \ell_0, \Sigma, \mathcal{X}, \Delta)$, where L is a finite set of locations with $\ell_0 \in L$ as the initial location and $L_F \subseteq L$ as the set of target locations, Σ a finite alphabet, $\mathcal{X}$ a finite set of clocks and $\Delta \subseteq L \times \mathcal{C}(\mathcal{X}) \times \Sigma \times 2^{\mathcal{X}} \times L$ a (non-deterministic) transition relation. We usually denote a transition between two locations ℓ and ℓ' by $\ell \xrightarrow{a,g,R} \ell'$ with $a \in \Sigma, g \in \mathcal{C}(\mathcal{X})$ and $R \subseteq \mathcal{X}$. A path in $\mathcal{A}$ is a sequence of the form

$$P = \ell_0 \xrightarrow{a_1,g_1,R_1} \ell_1 \xrightarrow{a_2,g_2,R_2} \ldots,$$

where $\ell_{i-1} \xrightarrow{a_i,g_i,R_i} \ell_i \in \Delta$ for every $i \geq 1$. The set of states will be denoted by S and a state $s \in S$ is a tuple consisting of a location and a clock valuation, i.e., $S = \{(\ell, v) \mid \ell \in L, v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\}$.

2.2.2 The semantics

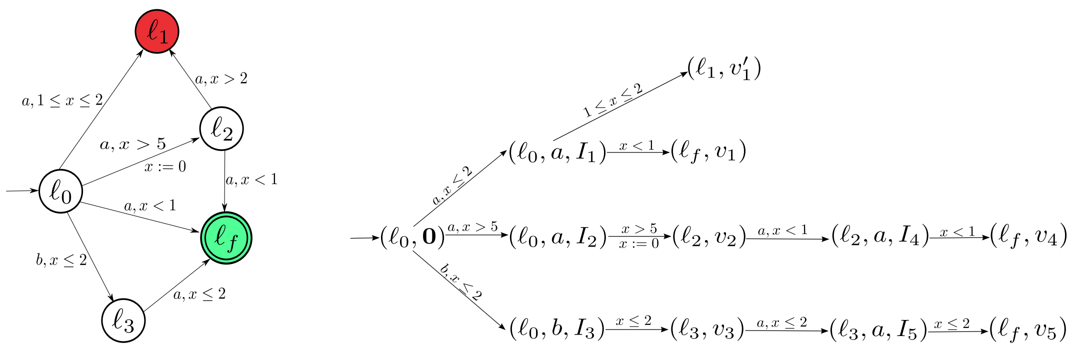
Given a timed automaton $\mathcal{A} = (L, \ell_0, \Sigma, \mathcal{X}, \Delta)$, its operational semantics is defined on a timed transition system $\mathcal{T}_{\mathcal{A}} = (S, s_0, \to)$ over the alphabet Σ. $S = \{(\ell, v) \mid \ell \in L, v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\}$ is the set of states with $s_0 = (\ell_0, \mathbf{0})$ as the initial state and $\to \subseteq S \times (\mathbb{R} \cup \Sigma) \times S$ the set of moves defined as follows:

- For a state $(\ell, v) \in S$ and a delay $d \in \mathbb{R}$, $(\ell, v) \xrightarrow{d} (\ell, v + d)$ is a *delay move*.
- For $\ell \xrightarrow{a,g,R} \ell' \in \Delta$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, $(\ell, v) \xrightarrow{a} (\ell', v[R \leftarrow 0])$ is an *action move* if $v \models g$.

A run in a timed automaton can be thus defined as a sequence of states together with delays and admissible actions that occur in-between. Formally,

$$\rho = (\ell_0, v_0) \xrightarrow{d_1} (\ell_0, v_1 = v_0 + d_1) \xrightarrow{a_1} (\ell_1, v_1) \xrightarrow{d_2} (\ell_1, v_2 = v_1 + d_2) \xrightarrow{a_2} (\ell_2, v_2) \xrightarrow{d_3} \ldots$$

is a run if and only if for every a_i, $\ell_{i-1} \xrightarrow{a_i,g_i,R_i} \ell_i \in \Delta$ and $v_i \models g_i$. For any finite run γ, we write $\mathsf{last}(\gamma)$ for the last state in γ and $\mathsf{last\_trans}(\gamma)$ for the last transition in γ.



(a) $\mathcal{G}$ (b) $[\![\mathcal{G}]\!]$

Figure 2: An example of a timed game $\mathcal{G}$ and its semantics $[\![\mathcal{G}]\!]$.
The double-circled location is the accepting/winning location.

3 Timed games with penalties

In this section we introduce the notion of timed games with penalties and formally define it with different semantics and two different winning objectives: state reachability and Büchi. Most of the formal syntax is inherited from “timed automaton games” [9]. The game semantics are modified and extended with the quantitative aspect of the penalty for Player 1. First, we define the game and its necessary game-theoretic concepts in general before we present three concrete game semantics.

3.1 Basic definitions

3.1.1 The structure of the game

A timed game with penalty is defined as a timed automaton with a set of target locations $L_F \subseteq L$, i.e., $\mathcal{G} = (L, \ell_0, \Sigma, \mathcal{X}, \Delta, L_F)$ where $(L, \ell_0, \Sigma, \mathcal{X}, \Delta)$ is a timed automaton. Player 1 (the controller) plays against Player 2 (the environment).

The semantics of a timed game with penalty $\mathcal{G}$ is given as a turn-based game structure

$$[\![\mathcal{G}]\!] = (S_1, S_2, (\ell_0, \mathbf{0}), \Sigma, \mathcal{X}, \delta, S_F)\,,$$

where $S_1 \subseteq L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, consisting of a location and a clock valuation, are the states controlled by Player 1, $S_2 \subseteq L \times \Sigma \times \mathcal{I}(\mathbb{R})$, consisting of a location with an action and an interval chosen/played by Player 1, are the states controlled by Player 2 and $S_F = \{(\ell_F, v) \mid \ell_F \in L_F, v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\}$ is the set of final states. δ is a transition function

$$\delta : (S_1 \times \Sigma \times \mathcal{C}(\mathcal{X})) \cup (S_2 \times \mathcal{C}(\mathcal{X}) \times 2^{\mathcal{X}}) \to S_2 \cup S_1$$

with the properties that a state $s \in S_1$ together with an action $a \in \Sigma$ and a guard $g \in \mathcal{C}(\mathcal{X})$ will only be mapped to some state $s' \in S_2$, i.e., $\delta(s, a, g) = s'$, and a state $s \in S_2$ together with a guard $g \in \mathcal{C}(\mathcal{X})$ and a set of clock to be reset $R \subseteq \mathcal{X}$ will be mapped to some state $s' \in S_1$, i.e., $\delta(s, g, R) = s'$.

δ is defined by the following rule:

$$\frac{(\ell, v) \xrightarrow{a, g_1 \vee \ldots \vee g_n} (\ell, a, v+I) \qquad \forall 1 \leq i \leq n: \; (\ell, a, v+I) \xrightarrow{g_i, R_i} (\ell_i, v+t)}{\forall 1 \leq i \leq n: \; \ell \xrightarrow{a, g_i, R_i} \ell_i \in \Delta}$$

for all $\ell, \ell_1, \ldots, \ell_n \in L$, $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, any interval $I \in \mathcal{I}(\mathbb{R})$ and any delay $t \in I$. We write $|\delta|$ for the number of single transitions/edges in $[\![\mathcal{G}]\!]$.

3.1.2 Moves, plays, strategies and penalties

Both players play different kind of moves, Player 1 makes multi-moves whereas Player 2 can only play delay moves.

Definition 3.1 (Multi-move)**.** *A multi-move or Player-1-move, or non-deterministic move, is a tuple $\overline{m} = (a, I) \in \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})$, where I can be seen as a delay-interval.*

The actual step will be determined after a corresponding delay move made by Player 2.

Definition 3.2 (Delay move)**.** *Given a multi-move $\overline{m} = (a, I)$, a delay move or restricted simple-move of Player-2 is a value $t \in \mathbb{R}_{\geq 0}$ such that $t \in I$.*

Definition 3.3 (A play in $\mathcal{G}$)**.** *A play in $\mathcal{G}$ is a sequence*

$$(\ell_0, v_0) \xrightarrow{a_1, I_1} (\ell_0, a_1, I_1) \xrightarrow{t_1} (\ell_1, v_1) \xrightarrow{a_2, I_2} (\ell_1, a_2, I_2) \xrightarrow{t_2} (\ell_2, v_2) \xrightarrow{a_3, I_3} \ldots,$$

where $\ell_{i-1} \xrightarrow{a_i, g_i, R_i} \ell_i \in \Delta$ and $v_{i-1} + t_i \models g_i$ has to hold for every i. Further, $(\ell_i, v_i) \in S_1$ and $(\ell_{i-1}, a_i, I_i) \in S_2$ for every i. For our convenience, we denote Player 1 states by s_i^1 and Player 2 states by s_i^2. The equivalent run as indicated above can be written as

$$s_0^1 \xrightarrow{a_1, I_1} s_0^2 \xrightarrow{t_1} s_1^1 \xrightarrow{a_2, I_2} s_1^2 \xrightarrow{t_2} s_2^1 \xrightarrow{a_3, I_3} \ldots,$$

where $s_i^1 \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and $s_i^2 \in L \times \Sigma \times \mathcal{I}(\mathbb{R})$.

For the empty string, we write ε. We use this notation for empty paths, runs or plays. Further for our convenience, we may abbreviate one step in a play $(\ell_{i-1}, v_{i-1}) \xrightarrow{a_i, I_i} (\ell_{i-1}, a_i, I_i) \xrightarrow{t_i} (\ell_i, v_i)$ by μ_i that consists of a multi-move by Player 1 and a restricted simple-move by Player 2. We write $\mu_1 \mu_2 \ldots$ for a play $(\ell_0, v_0) \xrightarrow{a_1, I_1} (\ell_0, a_1, I_1) \xrightarrow{t_1} (\ell_1, v_1) \xrightarrow{a_2, I_2} (\ell_1, a_2, I_2) \xrightarrow{t_2} (\ell_2, v_2) \xrightarrow{a_3, I_3} \ldots$.

The number of steps in a play ρ can be also called length of ρ, $\mathsf{length}(\rho)$.

We write $\delta(s)$ for the set of reachable states with one move, formally:

$$\delta(s) = \left\{ s' \in S \;\middle|\; \begin{array}{lll} \text{for } s \in S_1, & s \xrightarrow{a, I} s' & \forall a \in \Sigma, I \in \mathcal{I}(\mathbb{R}_{\geq 0}) \\ \text{for } s = (\ell, a, I) \in S_2, & s \xrightarrow{t} s' & \forall t \in I \end{array} \right\}$$

Moreover, we write Γ for the set of all runs that start in the initial state $(\ell_0, \mathbf{0})$ and Γ_{fin} for the set of finite runs starting in $(\ell_0, \mathbf{0})$.

Definition 3.4 (Sub-play and prefix order)**.** *Let $\rho = \mu_1 \mu_2 \ldots \mu_n$ be a play in a game $\mathcal{G}$. We say, ρ' is a **sub-play** of ρ, $\rho' \sqsubseteq \rho$, if there exists a set of indices $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ such that $\rho' = \mu_{i_1} \mu_{i_2} \ldots \mu_{i_k}$. We say, ρ' is a **prefix** of ρ, $\rho' \preceq \rho$, if $\rho' = \mu_1 \mu_2 \ldots \mu_\ell$, for some $\ell \leq n$.*

The notions of strict sub-play $\sqsubset$ and strict prefix $\prec$ relations are defined analogues.

As already mentioned, our aim is to measure the permissiveness of Player 1's strategy by assigning a penalty according to his multi-moves. We first give the definitions of how penalties will be charged.

Definition 3.5 (Penalty of a multi-move). *Given a multi-move $\overline{m} = (a, I) \in \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})$, the penalty of $\overline{m}$ is*

$$\mathsf{penalty}(\overline{m}) = \mathsf{penalty}(a, I) = \begin{cases} \frac{1}{|I|} & \text{if } |I| > 0 \\ +\infty & \text{otherwise.} \end{cases}$$

Given a step in a game $\mu_i = (\ell_{i-1}, v_{i-1}) \xrightarrow{a_i, I_i} (\ell_{i-1}, a_i, I_i) \xrightarrow{t_i} (\ell_i, v_i)$, we write $\mathsf{penalty}(\mu_i)$ for the penalty of Player 1 when playing (a_i, I_i), i.e., $\mathsf{penalty}(\mu_i) = \mathsf{penalty}(a_i, I_i)$.

Definition 3.6 (Multi-strategy for Player 1). *A multi-strategy for Player 1 is a function*

$$\sigma : \Gamma_{\mathit{fin}} \to \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})$$

that maps each history of a play (finite run) to an action and a delay interval.

Definition 3.7 (Strategy for Player 2). *A strategy for Player 2 is a function*

$$\tau : \Gamma_{\mathit{fin}} \times \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0}) \to \mathbb{R}_{\geq 0}.$$

Given a history of a play h (finite run) and knowing the move $\overline{m} = (a, I)$ of Player 1, $\tau(h, a, I) = d$ with the restriction that $d \in I$.

Further, we write $\mathsf{Out}(\sigma)$ for the set of runs that follow a (multi-)strategy σ. We consider the best possible responses (strategy) of Player 2, or the worst case for Player 1 respectively. The penalty for Player 1 indicates a quantity no matter what Player 2 does. In this paper, we consider two different penalty semantics for Player 1, namely additive penalty and taking the maximum penalty of a move within a run.

Definition 3.8 (Penalty of a play with *additive* semantics). *Given a finite run $\pi = s_0^1 \xrightarrow{a_1, I_1} s_0^2 \xrightarrow{t_1} s_1^1 \xrightarrow{a_2, I_2} s_1^2 \xrightarrow{t_2} s_2^1 \xrightarrow{a_3, I_3} \dots \xrightarrow{t_{n-1}} s_{n-1}^1 \xrightarrow{a_n, I_n} s_{n-1}^2 \xrightarrow{t_n} s_n^1$ in a game $\mathcal{G}$,*

$$\mathsf{penalty}_\pi = \sum_{i=1}^{n} \mathsf{penalty}(a_i, I_i) = \sum_{i=1}^{n} \left(\frac{1}{|I_i|} \right).$$

Definition 3.9 (Penalty of a play with *maximal* semantics). *Given a (possibly infinite) run $\pi = s_0^1 \xrightarrow{a_1, I_1} s_0^2 \xrightarrow{t_1} s_1^1 \xrightarrow{a_2, I_2} s_1^2 \xrightarrow{t_2} s_2^1 \xrightarrow{a_3, I_3} \dots$ in a game $\mathcal{G}$,*

$$\mathsf{penalty}_\pi = \sup_i \left(\mathsf{penalty}(a_i, I_i) \right) = \sup_i \left(\frac{1}{|I_i|} \right).$$

Lemma 3.1 (Penalty of a sub-play). *Given a finite play ρ and let ρ' be a sub-play of ρ, i.e., $\rho' \sqsubseteq \rho$. Then $\mathsf{penalty}_{\rho'} \leq \mathsf{penalty}_\rho$.*

In the game both players have contrary objectives. While Player 1 aims to reach a target state with the lowest possible penalty, Player 2 tries to prevent reaching any target state and moreover tries to let Player 1 pay as much penalty as possible.

Definition 3.10 (Penalty of a strategy)**.** *For a multi-strategy σ of Player 1, the (general) penalty of σ is*

$$\mathit{penalty}(\sigma) = \sup_{\pi \in \mathit{Out}(\sigma)} (\mathit{penalty}_\pi),$$

i.e., the penalty of the run generated by σ where Player 2 plays with a strategy that maximizes Player 1's penalty.

Player 1 wins if a target location is reached with finite penalty in the finite reachability game. In the infinite Büchi game, Player 1 wins if a target location will be reached infinitely often with finite penalty. Formally, the two winning conditions are:

Definition 3.11 (Reachability winning condition)**.** *Player 1 wins on a play π if some target location $\ell_f \in L_F$ is being reached with a finite penalty, i.e., there exists $s_f \in \{(\ell_f, v) \mid v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\} \subseteq S_F$ such that $\pi = s_0 \rightsquigarrow s_f$ and $\mathit{penalty}_\pi \leq +\infty$.*

Definition 3.12 (Büchi winning condition)**.** *Player 1 wins on a play π if some target location $\ell_f \in L_F$ is being reached infinitely often with a finite penalty, i.e., there exists $s_{f_1}, s_{f_2}, s_{f_3}, \ldots \in \{(\ell_f, v) \mid v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}\} \subseteq S_F$ such that $\pi = s_0 \rightsquigarrow s_{f_1} \rightsquigarrow s_{f_2} \rightsquigarrow \ldots$ and $\mathit{penalty}_\pi \leq +\infty$.*

For reachability, we apply Definition 3.8 with the additive semantics and for the Büchi winning condition, penalty will be assigned according to Definition 3.9 with the maximum semantics. In the rest of the paper, we will focus on the additive semantics with the reachability winning condition.

3.1.3 Permissiveness of timed multi-strategies

First of all, the aim is to decide whether a permissive timed multi-strategy for Player 1 exists, i.e., whether Player 1 has a strategy to win the game with finite penalty. The second intention is minimize the penalty. That means, we want to compute the optimal (lowest possible) penalty. For a timed game $\mathcal{G}$ and two players that play according to the rules defined in the previous paragraphs, we can state the following optimization problem:

Definition 3.13 (Most permissive strategy (optimization))**.** *Find a winning timed multi-strategy σ for Player 1 that minimizes the penalty, i.e., find σ such that for any winning timed multi-strategy σ', $\mathit{penalty}(\sigma) \leq \mathit{penalty}(\sigma')$.*

The equivalent decision problem can be defined as follows:

Definition 3.14 (Most permissive strategy (decision))**.** *Given some $\lambda \in \mathbb{Q}$, does there exist a winning timed multi-strategy σ for Player 1 such that $\mathit{penalty}(\sigma) \leq \lambda$?*

Notation 1. *Given a clock valuation $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and a guard $g \in \mathcal{C}(\mathcal{X})$, we write $\mathcal{I}(v, g) = \{I \in \mathcal{I}(\mathbb{R}_{\geq 0}) \mid v + t \models g \ \forall t \in I\}$ for the set of admissible delay intervals, and $\tau(v, g) = \{t \in \mathbb{R}_{\geq 0} \mid v + t \models g\}$ for the set of admissible delays.*

Given a state $(\ell, v) \in L \times \mathbb{R}_{\geq 0}^{\mathcal{X}}$, $\Sigma(\ell, v) = \{a \in \Sigma \mid \exists\, \ell \xrightarrow{a,g,R} \ell' \in \Delta \ \text{ s.t. } \ v \models g\}$ is the set of all possible actions at (ℓ, v).

For a location ℓ and an action $a \in \Sigma$,

$$g(\ell, a) = \bigvee_{\substack{g_i : \\ \ell \xrightarrow{a, g_i, R_i} \ell_i}} g_i$$

is the clock constraint in which playing a is allowed.

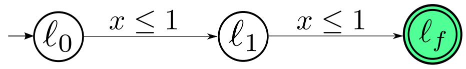


Figure 3: Region equivalence does not hold for permissiveness of timed games with penalty. In this example, the best strategy for Player 1 is to play $[0, 0.5]$ in ℓ_0 as well as in ℓ_1.

One difficulty in deciding the existence of winning strategies is that we cannot rely on regions. The notion of region equivalence class over clock valuations is a powerful tool to decide pure reachability problems in timed automata since it is possible to divide infinitely many states into finitely many equivalence classes or regions [1].

Definition 3.15 (Region equivalence)**.** *On a timed automaton $\mathcal{A} = (L, \ell_0, \Sigma, \mathcal{X}, \Delta)$ and its corresponding infinite state transition system $\mathcal{T}_{\mathcal{A}} = (S, s_0, \rightarrow)$, the region equivalence is a time-abstract bi-simulation that is defined by*

$$s = (\ell, v) \cong s' = (\ell', v') \iff \ell = \ell' \text{ and } v \equiv_{\mathcal{R}} v',$$

where $\equiv_{\mathcal{R}}$ is defined as follows: For each clock $x \in \mathcal{X}$, let c_x be the largest value that occurs on a clock constraint of any transition in Δ. For two valuations v and v', $v \equiv_{\mathcal{R}} v'$ if, and only if, the following three conditions hold:

- *For all $x \in \mathcal{X}$, either $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$[1], or $v(x) > c_x$ and $v'(x) > c$.*
- *For all $x, y \in \mathcal{X}$ with $v(x), v(y) \leq c_x$, $\{v(x)\} \leq \{v(y)\}$ if, and only if, $\{v'(x)\} \leq \{v'(y)\}$[2].*
- *For all $x \in \mathcal{X}$ where $v(x) \leq c_x$, $\{v(x)\} = 0$ if, and only if, $\{v'(x)\} = 0$[2].*

The example in Figure 3 shows that region equivalence does not help us to find the most permissive strategy. The states (ℓ_0, v) for all $v \in [0, 1]$ form one region, the same holds for (ℓ_1, v) and for all $v \in [0, 1]$. However, the best strategy for Player 1 in ℓ_0 as well as in ℓ_1 is to play the delay interval $[0, 0.5]$.

3.2 Reachability timed games with penalty

In this section, we will introduce timed games with penalty where each step in a play is *determined by Player* 1 in two different semantics.

3.2.1 Transition semantics

In this setting, we consider timed automata to be action-deterministic, i.e., given two edges $q \xrightarrow{a_1, g_1, R_1} q'$ and $q \xrightarrow{a_2, g_2, R_2} q''$, $a_1 = a_2$ if, and only if, $q' \neq q''$. We can define an inductive computation for the penalty from each location and for each clock valuation in the following way:

[1] $\lfloor \cdot \rfloor$ represents the integral part.
[2] $\{\cdot\}$ represents the fractional part.

Definition 3.16 (Inductive computation of the penalty)**.**
Base case: *For $i = 0$ and for all $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$,*

$$O^{(0)}(\ell, v) = \begin{cases} 0 & \text{for } \ell \in L_F \\ +\infty & \text{for } \ell \in L \setminus L_F. \end{cases} \tag{1}$$

***Step** $(i \to i+1)$**:** For all $i \geq 0$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$,*

$$O^{(i+1)}(\ell, v) = \inf_{\substack{a \in \Sigma(\ell, v), \\ \ell \xrightarrow{a,g,R} \ell'}} \inf_{I \subseteq \mathcal{I}(v,g)} \left(\frac{1}{|I|} + \sup_{t \in I} \left(O^{(i)}(\ell', v+t)[R := 0] \right) \right), \tag{2}$$

where $v + t \models g$.

Further, for the sake of simplicity, let

$$O(\ell, v) = \lim_{i \to +\infty} O^{(i)}(\ell, v)$$

for all $(\ell, v) \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$.

3.2.2 Pure interval semantics

In the pure interval semantics of quantitative timed games, the difference to the previous simple transition semantics is that the timed automaton is not necessarily action-deterministic but deterministic in the combination of action and clock valuation, meaning that for any two outgoing edges from a state q, $q \xrightarrow{a_1,g_1,R_1} q'$ and $q \xrightarrow{a_2,g_2,R_2} q''$, either $a_1 \neq a_2$ or $a_1 = a_2$ and for any valuation v, $v \not\models g_1 \wedge g_2$.

The tricky part in this game semantics lies in handling multi-moves that overlap with several guards of outgoing transitions for one action. For this case, we propose an inductive computation on $[\![\mathcal{G}]\!]$ on the turn-based game structure.

Definition 3.17 (Inductive computation of the penalty)**.**
Base case: *For $i = 0$ and for all $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$,*

$$O^{(0)}(\ell, v) = \begin{cases} 0 & \text{for } \ell \in L_F \\ +\infty & \text{for } \ell \in L \setminus L_F. \end{cases} \tag{3}$$

***Step** $(i \to i+1)$**:** For all $i \geq 0$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$,*

$$O^{(i+1)}(\ell, v) = \inf_{a \in \Sigma(\ell, v)} \inf_{I \subseteq \mathcal{I}(v, g(\ell, a))} \left(\frac{1}{|I|} + \sup_{\substack{\ell_i \in \ell E \text{ s.t.} \\ \ell \xrightarrow{a, g_i, R_i} \ell_i}} \sup_{t \in I} \left(O^{(i)}(\ell_i, v + t[R := 0]) \right) \right), \tag{4}$$

where $v + t \models g_i$.

We write $O(\ell, v)$ for the limit of the fixed point computation.

3.2.3 Correctness of the computations

In this section we want to show the correctness of the given computations in the sections 3.2.1 and 3.2.2. In particular we show that the function O computes the penalty of the best possible strategy under the worst case of the environment, i.e., assuming that Player 2 plays correctly with respect to his goals. First, we will indicate some structural properties:

Lemma 3.2 ($v \mapsto O^{(i)}(\ell, v)$ is non-decreasing.). *For every $i \in \mathbb{N}$, $\ell \in L$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, $v \mapsto O^{(i)}(\ell, v)$ is non-decreasing.*

Lemma 3.3 ($i \mapsto O^{(i)}(\ell, v)$ is non-increasing.). *For every $i \in \mathbb{N}$, $\ell \in L$ and $v \in \mathbb{R}_{\geq 0}^{|C|}$, $i \mapsto O^{(i)}(\ell, v)$ is non-increasing.*

Notation 2. *For a state $s \in L \times \mathcal{C}(\mathcal{X})$ and a winning timed multi-strategy σ, $\mathsf{penalty}_s(\sigma)$ is the penalty for a play π starting in s that is consistent with σ.*

Lemma 3.4 (The function O is almost-optimal.). *For every computation step $i \in \mathbb{N}$, for every state $(\ell, v) \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, there exists a timed multi-strategy σ for Player 1 that wins in maximum i steps starting at (ℓ, v) such that*

$$\mathsf{penalty}_{(\ell,v)}(\sigma) \leq O^{(i)}(\ell, v) + \epsilon \qquad \forall \epsilon > 0.$$

Lemma 3.5 (O is optimal under one clock.). *$O^{(i)}(\ell, v)$ is the optimal penalty for any timed multi-strategy that wins in not more than i steps starting in ℓ (as the initial state), i.e., for any timed multi-strategy σ that wins in maximum i steps starting in (ℓ, v),*

$$O^{(i)}(\ell, v) \leq \mathsf{penalty}(\sigma).$$

According to Lemma 3.5, $O^{(i+1)}$ correctly computes the optimal penalty for the restricted case of one clock, that is the least possible penalty Player 1 has to pay independently of what Player 2 plays (worst case).

Lemma 3.4 tells us that in the general case of any finite number of clocks, the computation is correct up to a small constant ϵ.

3.2.4 Decidability on one clock

The most permissive strategy for both of the games *with one clock* defined in this section is computable in polynomial time. We will state some structural properties that are used to prove termination of the fixed point equations stated in the Definitions 3.16 and 3.17.

Notation 3. *We denote the set of all transitions/edges on which x gets reseted ($x := 0$) by E_{reset}.*

For a subset $E_r \subseteq E_{reset}$, let $\mathsf{Out}_{E_r}(\sigma) = \{\rho \in \mathsf{Out}(\sigma) \mid \mathsf{last\_trans}(\rho) \in E_r\}$.

For a timed multi-strategy σ, a prefix of a play h and a subset $L' \subseteq L$ of locations, we define $\Gamma^h_{L'}(\sigma) = \{\rho \mid h \cdot \rho \in \mathsf{Out}(\sigma), |\rho|_{E_{reset}} = 0, \mathsf{last}(\rho) \in L'\}$.

Lemma 3.6. *Let σ be a winning timed multi-strategy. For all subsets of reset edges $E_r \subseteq E_{reset}$, all runs $\rho \in \mathsf{Out}(\sigma)$ and any prefix h_0 of ρ ending with an edge in E_r, there is a maximum extension that also ends with an edge in E_r that is a prefix of ρ. Formally,*

$$\forall E_r \subseteq E_{reset} \quad \forall h_0 \in \mathsf{Out}_{E_r}(\sigma) \quad \exists h_{max} \succeq h_0 \quad \forall h' \succ h_{max} : \qquad h' \notin \mathsf{Out}_{E_r}(\sigma).$$

Corollary 3.7 (Corollary of Lemma 3.6)**.** *$\mathit{Out}_{E_r}(\sigma)$ is finite and there exist maximal elements with respect to the partial order $\prec$.*

Lemma 3.8. *Let σ be a winning timed multi-strategy. Before the first occurrence, between two occurrences and after the last occurrence of any reset edges any location will be visited finitely often and consequently there exists a maximal element in $\Gamma_{L'}^{\sigma,h}$, i.e.,*

$$\forall L' \subseteq L \quad \forall h \in \mathit{Out}_{E_{reset}}(\sigma) \quad \forall \rho_0 \in \Gamma_{L'}^{\sigma,h} \quad \exists \rho_{max} \succeq \rho_0 \quad \forall \rho' \succ \rho_{max} : \qquad \rho' \notin \Gamma_{L'}^{\sigma,h}.$$

Theorem 3.9. *Any winning timed multi-strategy for Player 1 can be converted to a winning strategy with less or the same penalty by using each reset edge maximum once.*

In other words: Let σ be a winning timed multi-strategy for Player 1. There exists a winning timed multi-strategy σ' with $\mathit{penalty}(\sigma') \leq \mathit{penalty}(\sigma)$ such that for any reset edge $e \in E_{reset}$ and any run $\rho \in \mathit{Out}(\sigma')$, ρ takes e at most once.

Proof. We show this property by induction over all subsets $E_i \subseteq E_{reset}$ with $|E_i| = i$. Given a winning timed multi-strategy of Player 1, we construct a winning timed multi-strategy σ_{E_i} for each E_i that satisfies the following properties:

(1.) $\forall \rho \in \mathsf{Out}(\sigma_{E_i}), \forall e \in E_i : \quad \rho$ visits e maximum once,

(2.) $\mathsf{penalty}(\sigma_{E_i}) \leq \mathsf{penalty}(\sigma)$.

Base case: $\qquad E_0 = \emptyset$

Condition (1.) holds trivially for any multi-strategy. Let $\sigma_{E_0} = \sigma$, then Condition (2.) obviously holds as well.

Step:

We randomly pick some $e \in E_{reset} \setminus E_i$ and let $E_{i+1} = E_i \cup \{e\}$. We consider all runs or histories in $h \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ that end with the transition e. By Lemma 3.6 there exists a maximal extension $h_{max} \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ for each h. Further, we consider all histories γ that are prefixes of any outcome of σ_{E_i}. We distinguish two possible cases: Either the edge e is not contained in γ at all ($|\gamma|_e = 0$) or γ is of the form $h \cdot m$ with $h \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ and its finite, possibly empty, extension m. We construct $\sigma_{E_{i+1}}$ as follows:

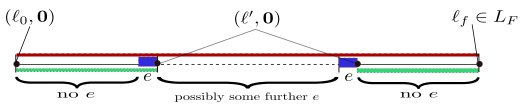
$$\sigma_{E_{i+1}}(\gamma) = \begin{cases} \sigma_{E_i}(\gamma) & \text{if } |\gamma|_e = 0 \text{ or } |\gamma|_e > 1 \\ \sigma_{E_i}(h_{max} \cdot m) & \text{otherwise.} \end{cases}$$

The intuition of this construction is illustrated in Figure 1a.

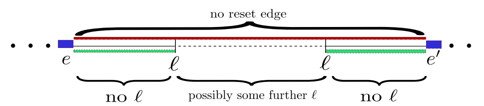
Claim: The conditions (1.) and (2.) hold for $\sigma_{E_{i+1}}$.

To show that condition (1.) holds, we assume by contradiction that after the first occurrence of e in any run $\rho \in \mathsf{Out}(\sigma_{E_{i+1}})$, e occurs again afterwards. However, by construction of $\sigma_{E_{i+1}}$, any finite history with one occurrence of e will be replaced by the strategy that is used after the last e-occurrence and the same extension after e. In other words, $h \cdot m$ will be replaced by $h_{max} \cdot m$ where $h \in \mathit{Out}_{\{e\}}(\sigma_{E_i})$. Hence, there has to be a further e after the last occurrence of e in h_{max}, which is a contradiction.

In order to show (2.), we consider every play in $\mathsf{Out}(\sigma_{E_i})$. Due to the construction, no run in $\mathsf{Out}(\sigma_{E_{i+1}})$ takes e more than once and for every $\rho \in \mathsf{Out}(\sigma_{E_i})$, $\sigma_{E_{i+1}}$ generates a sub-path ρ' that



(a) Edge elimination according to the proof of Theorem 3.9.



(b) Short cut between two clock-reset edges as in the proof of Theorem 3.10.

Figure 4: Illustration of the proof ideas of the Theorems 3.9 and 3.10. On the left, one complete winning play from the initial location to a target location, and on the right, sub-play between two reseting transitions. On both constructions, one induction step cuts off the dashed part of the play.

leads to an accepting location, i.e., for every $\rho \in \mathsf{Out}(\sigma_{E_i})$ there is *no* $\overline{\rho} \in \mathsf{Out}(\sigma_{E_{i+1}})$ such that $\rho \sqsubset \overline{\rho}$, and there exists a play ρ' such that $\rho' \sqsubseteq \rho$. Exploiting the property of Lemma 3.1 that for playsρ, ρ' with $\rho' \sqsubseteq \rho$, $\mathsf{penalty}_{\rho'} \leq \mathsf{penalty}_{\rho}$ holds, we can conclude $\mathsf{penalty}(\sigma_{E_i}) \leq \mathsf{penalty}(\sigma)$. ☐

Theorem 3.10. *Any winning timed multi-strategy for Player 1 can be converted to a winning strategy with less or the same penalty by visiting each location at most once before the first occurrence, between two successive occurrences and after the last occurrence of any clock-reset transition.*

In other words: Let σ be a winning timed multi-strategy. There exists a winning timed multi-strategy σ' such that along any outcome $\rho \in \mathsf{Out}(\sigma')$ any location ℓ will be visited at most once before the first occurrence, between two successive occurrences and after the last occurrence of any clock-reset transition.

Proof. We show this property by induction on all subsets $L_i \subseteq L$ with $|L_i| = i$. Given a winning timed multi-strategy σ of Player 1, we construct a winning timed multi-strategy σ_{L_i} such that the following properties hold:

(1.) $\forall \ell \in L_i \quad \forall e \in E_{reset} \quad \forall h \in \mathsf{Out}_{\{e\}}(\sigma_{L_i}) \cup \{\varepsilon\} \quad \forall \rho \in \Gamma^h_{L_i}(\sigma_{L_i}) :$ $\qquad \rho$ visits ℓ at most once,

(2.) $\mathsf{penalty}(\sigma_{L_i}) \leq \mathsf{penalty}(\sigma)$.

Base case: $\qquad L_0 = \emptyset$

(1.) holds because $\Gamma^h_{L_i}(\sigma_{L_i}) = \emptyset$. Let $\sigma_{L_0} = \sigma$, then (2.) holds trivially.

Step:

We pick any $\ell \in L \setminus L_i$ and let $L_{i+1} = L_i \cup \{\ell\}$. By Lemma 3.8 there exists a maximal continuation $\rho_{max} \in \Gamma^h_{\{\ell\}}(\sigma_{L_i})$ for any outcome $h \cdot \rho$ of σ_{L_i} with $h \in \mathsf{Out}_{E_{reset}}(\sigma_{L_i})$ and $|\rho|_{E_{reset}} = 0$. Either $\gamma \in \mathsf{Out}(\sigma_{L_i})$ is of the form $h \cdot \rho \cdot m$ with $h \in \mathsf{Out}_{E_{reset}}(\sigma_{L_i}) \cup \{\varepsilon\}$, $\rho \in \Gamma^h_{\{\ell\}}(\sigma_{E_i})$ and a suitable continuation m, or $|\gamma|_\ell = 0$. We construct $\sigma_{L_{i+1}}$ as follows:

$$\sigma_{L_{i+1}}(\gamma) = \begin{cases} \sigma_{L_i}(h \cdot \rho_{max} \cdot m) & \text{if } |\rho|_\ell = 1, \\ \sigma_{L_i}(\gamma) & \text{otherwise.} \end{cases}$$

The intuition of this construction is illustrated in Figure 1b.

Claim: The conditions (1.) and (2.) hold for $\sigma_{L_{i+1}}$.

To show that condition (1.) holds, let us assume the contrary. Hence, for some location $\ell \in L_{i+1}$, some clock-reset transition e and the following holds:

$$\exists h \in Out_{\{e\}}(\sigma_{L_{i+1}}) \cup \{\varepsilon\} \quad \exists \rho \in \Gamma^h_{L_{i+1}}(\sigma_{L_{i+1}}): \quad \rho \text{ visits } \ell \text{ more than once.}$$

Since it does not hold for L_i (induction hypothesis), only $\ell \in L_{i+1} \setminus L_i$ can be visited more than once. However, according to the construction of $\sigma_{L_{i+1}}$, Player 1 plays $\sigma_{L_i}(h \cdot \rho_{max})$ after the history $h \cdot \rho_0$, i.e., Player 1 plays the move at the first occurrence of ℓ before the clock-reset transition e that is played in σ_{L_i} after the last occurrence before e. If ℓ occurs in ρ once again afterwards, there must be one more occurrence of ℓ after the last occurrence of ℓ before e, i.e., there exists $\rho' \succ \rho_{max}$ such that $\rho' \in \Gamma^h_{\{\ell\}}(\sigma_{L_i})$. This is a contradiction to Lemma 3.8.

Condition (2.) follows from Lemma 3.1, since $\sigma_{L_{i+1}}$ either uses the strategy of σ_{L_i} or produces sub-plays of a run in $\mathsf{Out}(\sigma_{L_i})$. The idea is similar as in the proof of Theorem 3.9. ☐

Corollary 3.11. *For every winning timed multi-strategy σ, there exists a winning timed multi-strategy σ' such that the length of any outcome $\rho \in \mathsf{Out}(\sigma')$ is bounded by $|Q| \cdot (|\delta|_{reset} + 1)$.*

Corollary 3.12. *For every winning timed multi-strategy σ, there is a winning timed multi-strategy σ' such that*

$$O^{(N)}(\ell_0, \mathbf{0}) \leq \mathsf{penalty}(\sigma') \leq \mathsf{penalty}(\sigma),$$

where $N \leq |Q| \cdot (|\delta| + 1)$.

Proof. Given a winning timed multi-strategy σ, we can construct a winning timed multi-strategy σ' that is bounded by $|Q| \cdot (|\delta|_{reset} + 1) \leq |Q| \cdot (|\delta| + 1)$. Based on the procedures in Theorem 3.9 and 3.10, each outcome $\rho \in Out(\sigma)$ produces a corresponding $\rho' \in Out(\sigma')$ with $\rho' \sqsubseteq \rho$. By applying Lemma 3.1, we get $\mathsf{penalty}(\sigma') \leq \mathsf{penalty}(\sigma)$.

Since the length of the play with the lowest possible penalty is at most $|Q| \cdot (|\delta| + 1)$, an optimal strategy requires not more than $|Q| \cdot (|\delta| + 1)$ steps and hence, not more than $N \leq |Q| \cdot (|\delta| + 1)$ computation steps on the O function are required. Applying Lemma 3.5, we get $O^{(N)}(\ell_0, \mathbf{0}) \leq \mathsf{penalty}(\sigma')$. ☐

Corollary 3.13. *The sequence $\left(O^{(i)}(\ell_0, \mathbf{0})\right)_{i \in \mathbb{N}}$ stabilizes in $N \leq |Q| \cdot (|\delta| + 1)$ computation steps, i.e., $O^{(N+k)}(\ell_0, \mathbf{0}) = O^{(N)}(\ell_0, \mathbf{0})$ for any $k > 0$.*

Proof. For the sake of contradiction, assume that $O^{(N+k)}(\ell_0, \mathbf{0}) \neq O^{(N)}(\ell_0, \mathbf{0})$. Then, by applying Lemma 3.3, $O^{(N+k)}(\ell_0, \mathbf{0}) < O^{(N)}(\ell_0, \mathbf{0})$ has to hold.

Let us consider a timed multi-strategy $\overline{\sigma}$ that witnesses $O^{(N+k)}(\ell_0, \mathbf{0})$, meaning that $\mathsf{penalty}(\overline{\sigma}) = O^{(N+k)}(\ell_0, \mathbf{0})$. Applying Corollary 3.12, we get $O^{(N)}(\ell_0, \mathbf{0}) \leq \mathsf{penalty}(\overline{\sigma}) = O^{(N+k)}(\ell_0, \mathbf{0})$, which is a contradiction to our assumption. ☐

Corollary 3.14. *The worst case runtime for computing O given in Definition 3.16 is $\mathcal{O}(|L| \cdot |\delta|^2)$.*

Considering the maximal possible length of $\mathcal{O}(|\delta| \cdot |L|)$ and that at every state in a run, there may be up to $|\delta|$ many transition possibilities, we can conclude a worst-case runtime of $O(|L| \cdot |\delta|^2)$. Hence, the decision problem indicated in Definition 3.14 is solvable in PTIME.

However, the question of the precise complexity class of the decision problem is still open. What we can say is that this decision problem is harder than the shortest path problem on directed weighted graphs that asks whether a path of weight below a certain threshold C exists between two given nodes s and t in the graph. This shortest path problem is known to be in NC (Nick's class) [7, 17]. We can easily simulate the shortest path problem on a timed game structure as follows: We do not modify the structure of the given graph and we actually only need to encode the weights in clock constraints and resets. We put a clock reset on every edge and put $0 \le x \le {}^{1}\!/\!{}_{w}$ as clock constraint, where w is the given weight of the edge. Further, we assign an action to every edge such that no two outgoing edges from one node/location has the same action and we also use s and t as the initial and final states, respectively. Finally we also use C, $\lambda = C$, as the threshold in the decision problem of Definition 3.14. Clearly, there exists a path from s to t with weight bounded by C if, and only if, Player 1 has a winning timed multi-strategy with a penalty bounded by $\lambda = C$.

To summarize, the decision problem for this game setting is solvable in PTime and is harder than the shortest path problem for weighted directed graphs. It would be interesting to do further investigations in order to determine the precise complexity class.

3.3 Turn-based reachability timed game with penalty

We will increase the "expressive power" of the game by giving Player 2 more opportunities such as the possibility to let time elapse in some locations. We therefore introduce a new kind of location that is *fully* controlled by Player 2. In order to avoid confusions with Player 2 states in the semantical turn-based game structure, we indicate locations for Player 1 and Player 2 by L_I and L_{II} respectively. Further, we introduce "urgent locations" where any outgoing transition will be taken without delay. Each urgent location is owned by either Player 1 or Player 2. Only an action can be chosen in those locations which is executed immediately without any delay.

3.3.1 Semantics

The turn-based timed game is played on a structure

$$\mathcal{G}_{tb} = (L_I, L_{II}, L_I^u, L_{II}^u, \ell_0, \Sigma, \mathcal{X}, \Delta, L_F),$$

where $\ell_0 \in L_I \cup L_{II}$ and $L_F \subseteq L_I \cup L_{II} \cup L_I^u \cup L_{II}^u$. $(L_I \cup L_{II} \cup L_I^u \cup L_{II}^u, \ell_0, \Sigma, \mathcal{X}, \Delta)$ is a timed automaton and the semantics of L_I is equivalent to the "ordinary" timed game with penalty as defined in Section 3.1.1. L_I^u and L_{II}^u are the set of urgent locations of Player 1 and Player 2 respectively. Moreover, we consider action-timed determinism as in the pure interval semantics, which means that for any two outgoing edges from a location ℓ, $\ell \xrightarrow{a_1, g_1, R_1} \ell'$ and $\ell \xrightarrow{a_2, g_2, R_2} \ell''$, either $a_1 \neq a_2$ or $a_1 = a_2$ and for any valuation v, $v \not\models g_1 \wedge g_2$. The way how Player 2 can behave in a L_{II} location is defined as a delay move.

The reason for introducing urgent locations is to simplify the modeling of atomic sequences of locations in a game. It is semantically equivalent to add an extra clock that is reset on every incoming transition and having the invariant of $x = 0$ on the location itself. Hence, time does not pass in any urgent location [2]. The nice side effect is that urgent locations also reduce the state space.

The semantics of a turn-based reachability timed game $\mathcal{G}$ is given as a turn-based timed game structure

$$[\![\mathcal{G}_{tb}]\!] = (S_I, S_{II}, S_I^u, S_{II}^u, (\ell_0, \mathbf{0}), \Sigma, \mathcal{X}, \delta_{tb}, S_F),$$

where $S_I = S_1 \cup S_2$ as in $[\![\mathcal{G}]\!]$ (see Section 3.1) and $S_{II} \subseteq L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$. $S_I^u \subseteq L_I^u \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and $S_{II}^u \subseteq L_{II}^u \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ are the urgent states for each player. The set $S_I \cup S_{II} \cup S_I^u \cup S_{II}^u$ will be abbreviated by S. δ_{tb} is a transition function

$$\delta_{tb} : (S_1 \times \Sigma \times \mathcal{C}(\mathcal{X})) \cup (S_2 \times \mathcal{C}(\mathcal{X}) \times 2^{\mathcal{X}}) \cup ((S_{II} \cup S_I^u \cup S_{II}^u) \times \Sigma \times \mathcal{C}(\mathcal{X}) \times 2^{\mathcal{X}}) \longrightarrow S,$$

where any state $s \in S_1$ together with an action a and guard g will be mapped to some $s' \in S_2$, any $s \in S_2$ with some guard g and a set of clocks R to be reset will be mapped to some $s' \in S_1$, and any state $s \in S_{II} \cup S_I^u \cup S_{II}^u$ with some action a, guard g and a set of clocks R to be reset will be mapped to some state $s' \in S$. In particular, δ is defined as follows:

- For any state $(\ell, v) \in S_I$:

$$\frac{(\ell, v) \xrightarrow{a, g_1 \vee \ldots \vee g_n} (\ell, a, v + I) \qquad \forall 1 \leq i \leq n : \ (\ell, a, v + I) \xrightarrow{g_i, R_i} (\ell_i, v + t)}{\forall 1 \leq i \leq n : \ \ell \xrightarrow{a, g_i, R_i} \ell_i \in \Delta}$$

  for all $\ell_1, \ldots, \ell_n \in L_I \cup L_{II} \cup L_I^u \cup L_{II}^u$, $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, any interval $I \in \mathcal{I}(\mathbb{R})$ and any delay $t \in I$. (Same definition as in the "ordinary timed game with penalty" given in Section 3.1.1.)

- For any state $(\ell, v) \in S_{II}$:

$$\frac{(\ell, v) \xrightarrow{a, g, R} (\ell', v + t[R := 0])}{\ell \xrightarrow{a, g, R} \ell' \in \Delta}$$

  for all $\ell' \in L_I \cup L_{II} \cup L_I^u \cup L_{II}^u$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$.

- For any state $(\ell, v) \in S_I^u$ or S_{II}^u:

$$\frac{(\ell, v) \xrightarrow{a, g, R} (\ell', v[R := 0])}{\ell \xrightarrow{a, g, R} \ell' \in \Delta}$$

  for all $\ell' \in L_I \cup L_{II} \cup L_I^u \cup L_{II}^u$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$.

The move in any location in L_I is equivalent to the moves in the non-turnbased game defined in Section 3.1, where Player 1 makes a multi-move and Player 2 a delay move. For the other locations, we will now define the allowed moves.

Definition 3.18 (Action-delay-move). *An action-delay-move is a tuple $m = (a, d)$ executed by Player 2 in a location in L_{II}, where $a \in \Sigma$ is an action and $d \in \mathbb{R}_{\geq 0}$ is a delay.*

Player 2 is allowed to chose any $\Sigma \times \mathbb{R}_{\geq 0}$ tuple. However, a player loses when the game ends in a non-target location where his move cannot be executed. When in a Player 2 location no transition is available for (a, d), i.e. $\neg \exists \ell \xrightarrow{a, g, R} \ell' \in \Delta$ such that $v + d \models g$. In other words, any player is always forced to play an admissible move.

Definition 3.19 (Action move). *An action move or urgent move consists of choosing one action $a \in \Sigma$ which can be play in urgent locations.*

When a play arrives at an urgent location $\ell \in L_I^u \cup L_{II}^u$, Player 1, or Player 2 respectively, chooses an action $a \in \Sigma$ which will be executed without any delay.

Definition 3.20 (Strategy for Player 1)**.** *A strategy of Player* 1 *can be formalized as a function*

$$\sigma : \Gamma_{fin} \longrightarrow (\Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})) \cup \Sigma,$$

such that for a prefix of a play (history) $h \in \Gamma_{fin}$*,*

- $\sigma(h) \in \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})$ *if* $\mathsf{last}(h) \in L_I$*,*
- $\sigma(h) \in \Sigma$ *if* $\mathsf{last}(h) \in L_I^u$ *(urgent location).*

Definition 3.21 (Strategy for Player 2)**.** *A strategy of Player* 2 *can be formalized as a function*

$$\tau : (\Gamma_{fin} \times \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})) \cup \Gamma_{fin} \rightarrow \mathbb{R}_{\geq 0} \cup (\Sigma \times \mathbb{R}_{\geq 0}) \cup \Sigma,$$

such that for a prefix of a play (history) $h \in \Gamma_{fin}$*,*

- $\tau(h, a, I) \in \mathbb{R}_{\geq 0}$ *if* $\mathsf{last}(h) \in L_I \times \Sigma \times \mathcal{I}(\mathbb{R}_{\geq 0})$*,*
- $\tau(h) \in \Sigma \times \mathbb{R}_{\geq 0}$ *if* $\mathsf{last}(h) \in L_{II}$*,*
- $\tau(h) \in \Sigma$ *if* $\mathsf{last}(h) \in L_{II}^u$ *(urgent location).*

The computation of the function O is done equivalently for locations in L_I as in the "ordinary" timed game with penalty (introduced in Section 3.2, Definition 3.17). An inductive computation for Player 1's penalty from locations in L_{II}, L_I^u and L_{II}^u is given below.

Definition 3.22 (Inductive computation of the penalty)**.**

Base case: *For* $i = 0$ *and for all* $v \in \mathbb{R}_{\geq 0}^{|C|}$*:*

$$O_q^{(0)}(v) = \begin{cases} 0 & \text{for } \ell \in L_F \\ +\infty & \text{for } \ell \notin L_F \} \end{cases} \tag{5}$$

Step $(i \rightarrow i+1)$***:*** *For all* $i \geq 0$ *and* $v \in \mathbb{R}_{\geq 0}^{|\mathcal{C}|}$*,*

- *for all* $\ell \in L_I$*:*

$$O^{(i+1)}(\ell, v) = \inf_{a \in \Sigma(\ell,v)} \inf_{I \subseteq \mathcal{I}(v, g(\ell,a))} \left(\frac{1}{|I|} + \sup_{\substack{\ell_i \in \ell E \text{ s.t.} \\ \ell \xrightarrow{a, g_i, R_i} \ell_i}} \sup_{t \in I} \Big(O^{(i)}(\ell_i, v + t[R_i := 0]) \Big) \right), \tag{6}$$

  where $v + t \models g_i$*,*

- *for all* $\ell \in L_{II}$*:*

$$O^{(i+1)}(\ell, v) = \sup_{\substack{\ell' \in \ell E \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \sup_{t \in \tau(v,g)} \Big(O^{(i)}(\ell', v + t[R := 0]) \Big) \tag{7}$$

- *for all* $\ell \in L_I^u$*:*

$$O^{(i+1)}(\ell, v) = \inf_{\substack{a \in \Sigma(\ell,v) \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \Big(O^{(i)}(\ell', v[R := 0]) \Big) \tag{8}$$

- *for all* $\ell \in L^u_{II}$*:*

$$O^{(i+1)}(\ell, v) = \sup_{\substack{a \in \Sigma(\ell,v) \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \Big(O^{(i)}(\ell', v[R := 0]) \Big) \tag{9}$$

We write $O(\ell, v)$ for the limit of the fixed point computation $\lim_{i \to +\infty} O^{(i)}(\ell, v)$.

Correctness

The correctness of the Equation (6) has been shown in Section 3.2.3, which is the essential case where Player 1 has the control and wants to decrease the penalty as much as possible. In all states in L_{II}, we have the contrary objective, namely to increase the penalty as much as possible. What remains to show is the correctness of the equations 7, 8 and 9. First, we state two assisting lemmata before we formally state the correctness of the equation system in Definition 3.22.

Lemma 3.15. *For all* $(\ell, v) \in L_{II} \times \mathbb{R}^{|\mathcal{X}|}_{\geq 0}$*,* $O(\ell, v)$ *computes the highest possible penalty for Player 2, i.e., in each state controlled by Player 2 the move that maximizes the penalty will be taken.*

Lemma 3.16. *The equations 8 and 9 correctly compute the optimal penalty for Player* 1 *with respect to the goal of Player* 1*, and Player* 2*, respectively. That means:*

- *For any* $\ell \in L^u_I$*,* $O^{(i+1)}(\ell, v)$ *computes the least possible penalty from* (ℓ, v) *where no time elapses until the next state* (ℓ', v)*.*
- *For any* $\ell \in L^u_{II}$*,* $O^{(i+1)}(\ell, v)$ *computes the highest possible penalty from* (ℓ, v) *without time delay to reach the next location.*

Proposition 3.17. O *correctly computes the almost-optimal (lowest possible) penalty for Player* 1*, i.e., for every computation step* i*, for every state* $(\ell, v) \in L \times \mathbb{R}^{|\mathcal{X}|}_{\geq 0}$ *there exists a timed multi-strategy* σ *that wins in at most* i *steps starting in* (ℓ, v) *such that*

$$\mathsf{penalty}(\sigma) \leq O^{(i)}(\ell, v) + \epsilon \qquad \forall \epsilon > 0.$$

3.3.2 Decidability on one clock

In this section, we show that the most permissive strategy problem on one clock is decidable also in this setting. Moreover it is PTime-complete. We will first show the PTime-membership before we discuss the hardness.

We note that the lemmata 3.6 and 3.8 also hold in this setting since their proofs show its properties for a more general class. Also the properties of Theorem 3.9 and Theorem 3.10 hold in this turn-based reachability timed game with penalty on one clock. Although the idea of the proofs is very similar to the "simple" (non-turn-based) timed game with penalty, it requires several additional technical constructions. We will therefore state the two theorems in a slightly different presentation adapted to this setting of turn-based reachability timed games with penalty again and will further discuss the additional cases for the proofs.

Theorem 3.18. *Let* σ *be a winning timed multi-strategy for Player* 1*. There exists a winning timed multi-strategy* σ' *with* $\mathsf{penalty}(\sigma') \leq \mathsf{penalty}(\sigma)$ *such that Player* 1 *forces any clock-reset transition to be taken at most once.*

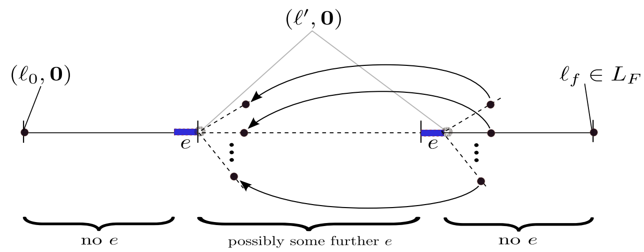


Figure 5: Elimination of multiple occurrences of a clock-reset edges whose target location is uncontrollable for Player 1 (shown as ∘).

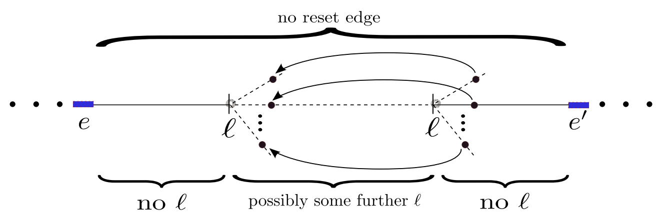


Figure 6: Elimination of multiple occurrences of a location between two clock-reset edges that is uncontrollable for Player 1 (shown as ∘).

Theorem 3.18 tells us that any clock-reset transition will occur at most once in any outcome $\rho \in \mathsf{Out}(\sigma')$. The tricky part in the construction of σ' lies in handling the case when the target location of a reset transition $\ell \xrightarrow{a,g,x:=0} \ell'$ is controlled by Player 2, i.e., $\ell' \in L_{II} \cup L^u_{II}$. In this case, the multi-strategy of Player 1 has to be adapted on the first following location that Player 1 controls which is necessarily before the next clock-reset transition. If there is no location controlled by Player 1 between two reset transitions in any outcome ρ of σ, σ cannot be winning anymore. The construction is illustrated in Figure 5. For details of the construction we refer to the proof in the appendix.

Theorem 3.19. *Let σ be a winning timed multi-strategy for Player 1. There exists a winning timed multi-strategy σ' with $\mathsf{penalty}(\sigma') \leq \mathsf{penalty}(\sigma)$ such that Player 1 forces any location to be visited at most once before the occurrence of any clock-reset transitions, between two successive reset transitions e, e', meaning that there is no reset transition e'' between e and e', and after the last reset transition.*

Theorem 3.19 tells us that we can construct/modify a strategy that visits any location ℓ between two reset transitions at most once no matter if ℓ belongs to Player 1 or 2. The idea is equivalent to the construction for Theorem 3.18 and the construction is illustrated in Figure 6. For details, we refer to the proof in the appendix.

The hardness can be shown by two different reductions. First, we show that reachability in alternating graphs can be simulated in LogSpace with our game, where we only use locations in L_I and L_{II} and without any urgent states. Then, we show a reduction from the "alternating monotone fanout 2 circuit value problem" problem by using only locations in L_I and L^u_{II} without any locations fully controlled by Player 2.

Hardness reduction - Alternative 1 We first give the definition of the PTIME-complete problem [16], we want to simulate.

Definition 3.23 (Reachability on alternating graphs)**.** *Given an alternating graph $G = (V, V_\exists, V_\forall, E)$ with $V = V_\exists \uplus V_\forall$ and two vertices $s, t \in V$. We say, t is reachable from s in G if and only if $P^G(s,t)$ which is defined as follows:*

- $P^G(x,x)$ *for any* $x \in V$
- *If* $x \in V_\exists$ *and* $P^G(z,y)$ *holds for some* $(x,z) \in E$*, then* $P^G(x,y)$ *holds as well.*
- *Let* $x \in V_\forall$ *and there exists some* $z \in V$ $(z \neq x)$ *such that* $(x,z) \in E$*. If* $P^G(z,y)$ *holds for all* z *with* $(x,z) \in E$*, then* $P^G(x,y)$ *holds as well.*

Proposition 3.20. *The most permissive strategy problem for turn-based reachability timed games with penalty without urgent locations is* PTIME*-hard.*

Proof. We simulate the reachability problem on an alternating graph G on the turn-based reachability timed game with penalty $\mathcal{G}$ without urgent locations,

$$\mathcal{G} = (L_I, L_{II}, \ell_0, \Sigma, \mathcal{X}, \Delta, L_F)$$

with $\mathcal{X} = \{x\}$ as follows: (We omit the urgent locations L_I^u and L_{II}^u for this reduction.)

- $L \leftarrow V$
- $L_I \leftarrow V_\exists$
- $L_{II} \leftarrow V_\forall$
- λ is the number of edges leaving L_I states
- $\ell_0 \leftarrow s$
- $L_F \leftarrow \{t\}$
- $|\Sigma|$ is the maximal outgoing node degree in G

Further, every edge in G will be a transition in $\mathcal{G}$ such that the two structures are isomorphic apart from the labels on $\mathcal{G}$. We randomly assign actions to each transition with the only constraint that no two outgoing edges from one location are labeled with the same action. The clock constraint on each transition is $0 \leq x \leq 1$. x will be reseted on every transition.

Claim: $P^G(s,t)$ holds if and only if there exists a winning timed multi-strategy σ for Player 1 with $\mathsf{penalty}(\sigma) \leq \lambda$.

The correctness of this construction, or the proof of this claim respectively is rather straightforward. For details, we refer to the appendix section A. □

Hardness reduction - Alternative 2 We will reduce our decision problem as stated in Definition 3.14 from the PTIME-complete problem "*alternating monotone fanout 2 circuit value problem*" [13, 14]:

Definition 3.24 (Alternating monotone fanout 2 CVP)**.** *Given a binary encoding of a Boolean circuit C, composed of `AND` and `OR` gates where any path in C alternates `AND` and `OR` gates, and inputs $x_1, \ldots, x_n$. Any input x_i is required to be connected to an `OR` gate and the output must directly come from an `OR` gate. Moreover, each input or internal gate has a fanout of exactly two.*

Question: Does C output 1 *on the input $x_1, \ldots, x_n$?*

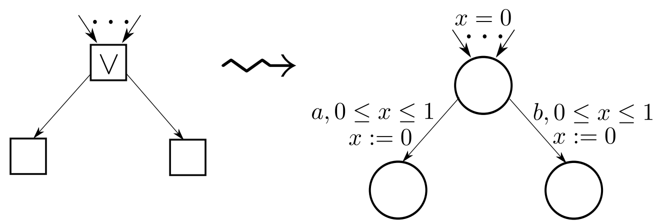


Figure 7: Simulating OR gates on a timed game structure. All locations on the right belong to Player 1.

Proposition 3.21. *The most permissive strategy problem for turn-based reachability timed games with penalty without Player* 2 *locations is* PTIME*-hard.*

Proof (Sketch). We simulate the alternating monotone fanout 2 CVP problem on the turn-based reachability timed game with penalty $\mathcal{G}$ without using Player 2 locations L_{II},

$$\mathcal{G} = (L_I, L^u_{II}, \ell_0, \Sigma, \mathcal{X}, \Delta, L_F).$$

Given a boolean circuit C with only `OR` and `AND` gates, where all paths start and end with an `OR` gate and alternate between `OR` and `AND` gates. Moreover each gate has two outgoing arcs. We transform C into a timed automaton as follows:

- An `OR` gate with two outgoing arcs to $\texttt{gate}_1$ and $\texttt{gate}_2$ will be simulated by three locations ℓ, ℓ_1 and ℓ_2 which are all Player 1 locations. We connect ℓ to ℓ_1 with action a, guard $0 \leq x \leq 1$ and reset x. We do it similarly for ℓ to ℓ_2 with action b, guard $0 \leq x \leq 1$ and $x := 0$. This transformation is graphically illustrated in Figure 7.
- An `AND` gate with two outgoing arcs to $\texttt{gate}_1$ and $\texttt{gate}_2$ will be simulated by four locations ℓ, ℓ^u, ℓ_1 and ℓ_2, where $\ell, \ell_1, \ell_2 \in L_I$ are Player 1 locations and $\ell^u \in L^u_{II}$ is an urgent location controlled by Player 2. The edge $\ell \xrightarrow{a, 0 \leq x \leq 2} \ell^u$ allows Player 1 to reach the urgent location with a penalty of $1/2$ and the delay choice of Player 2 already in ℓ determines whether the play continues to ℓ_1 or ℓ_2. Therefore we connect ℓ^u to ℓ_1 with action a and guard $0 \leq x \leq 1$, and ℓ^u to ℓ_2 with action a and guard $1 < x \leq 2$. Note that no time can elapse in ℓ^u. Figure 8 shows a graphical illustration.

We can assume that each non-urgent location will be arrived with $x = 0$ because (by construction) every incoming edge (to a non-urgent location) contains a clock reset.

Finally, C evaluates to `true` if, and only if, Player 1 has a multi-strategy σ of

$$\mathsf{penalty}(\sigma) \leq |\wedge|/2 + |\vee|,$$

where $|\wedge|$ is the number of `AND`-levels and $|\vee|$ the number of `AND`-levels in C. □

3.4 Properties of multi-strategies

In this section, we show that every winning timed multi-strategy σ for *turn-based reachability timed games with penalty* can be converted to a **memoryless** winning timed multi-strategy σ'. First of all we define what memoryless in a timed multi-strategy means.

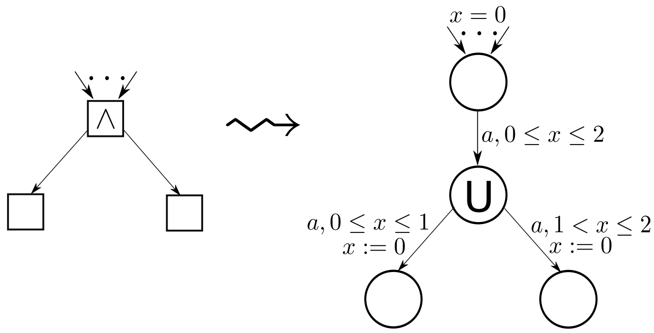


Figure 8: Simulating AND gates on a timed game structure. All plain locations (without any caption) on the right belong to Player 1, the location labeled with U is an urgent location where no time elapses.

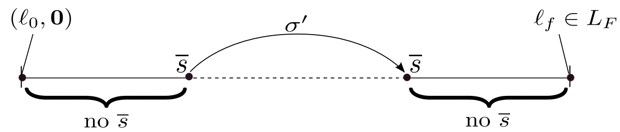


Figure 9: Constructing σ' based on σ such that the state $\overline{s}$ occurs only once. Intuitively, the dashed part of the play will be cut off.

Definition 3.25 (Memoryless timed multi-strategy)**.** *A timed multi-strategy σ is memoryless, or positional, if $\sigma(h) = \sigma(\mathsf{last}(h))$ for any history h of a play starting from $(\ell_0, \mathbf{0})$.*

In other words, a move made according to a memoryless strategies only depends on the current state. The history before reaching the current state is irrelevant for the decision of the move.

Lemma 3.22. *Given a winning timed multi-strategy σ, there exists a winning timed multi-strategy σ' that visits each state $s = (\ell, v) \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ at most once.*

Proof. First, we observe that any outcome $\rho \in \mathsf{Out}(\sigma)$ is finite. Let S_ρ be the (finite) set of states occurring in ρ.

Based on σ, a winning timed multi-strategy σ' that visits any state at most once can be defined as follows: For every play $\rho \in \mathsf{Out}(\sigma)$ and for every $s \in S_\rho$, let $h^s_{max} \preceq \rho$ be the longest prefix of ρ that ends in s. Then for any history (prefix of a play) h, σ' can be defined as

$$\sigma'(h) = \begin{cases} \sigma(h^s_{max}) & \text{if } \mathsf{last}(h) = s \text{ and } |h|_s = 1, \\ \sigma(h) & \text{otherwise.} \end{cases}$$

Every $\rho \in \mathsf{Out}(\sigma)$ The correctness of the construction of σ' is not hard to verify since σ is winning, h is a prefix of a winning play and $\sigma(h^s_{max})$ is a move within a winning play. The construction is visualized in Figure 9. □

Notation 4. *For a timed multi-strategy σ, we define S_σ as the set of states that are reachable by playing according to σ.*

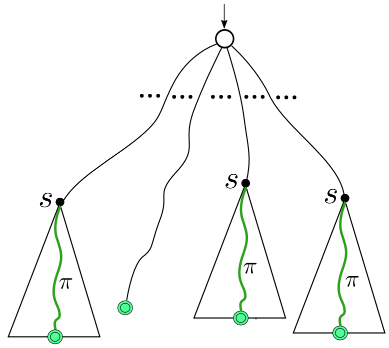


Figure 10: All plays $\rho \in Out(\sigma)$ for a winning timed multi-strategy σ shown as an infinitely branching tree with uncountably many branches at each state. For a reachable state $s \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, there exists a set of winning plays that is independently from any previous history/play until s (indicated as the triangle). π is a winning play with the least possible penalty.

Lemma 3.23. *S_σ is infinite and uncountable for every winning timed multi-strategy σ.*

Lemma 3.23 is straightforward since a winning σ never proposes discrete timed multi-moves. That means, every multi-move is an interval over $\mathbb{R}$ of length greater than zero that contains infinitely and uncountably many elements (possible delay moves).

Theorem 3.24. *For any winning timed multi-strategy σ, there exists a memoryless winning multi-strategy σ'.*

Proof. Given a winning timed multi-strategy σ, by Lemma 3.22 we know the existence of a winning multi-strategy σ' that visits any state $s \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ at most once.

Consider the set S_σ. We recall that from each reached state $s = (\ell, v)$, the goal is to reach a location $\ell_f \in L_F$ the with the least possible penalty.

We can define a strategy $\overline{\sigma}$ as follows: For every state (ℓ, v), the sets $\Sigma(\ell, v)$ and $\mathcal{I}(v, g(\ell, a))$ are independently from any history. Further, the set of winning plays

$$\Gamma_{win}^{(\ell,v)} = \{\pi \mid \pi = (\ell, v) \rightsquigarrow \ell_f \ \textit{ for any } \ell_f \in L_F\}$$

from (ℓ, v) is also independently from the history leading to (ℓ, v) and there exists some play $\pi \in \Gamma_{win}^{(\ell,v)}$ such that $\mathsf{penalty}_\pi \leq \mathsf{penalty}_{\pi'}$ for any $\pi' \in \Gamma_{win}^{(\ell,v)}$ (illustrated in Figure 10). Let (a, I) be the first move of Player 1 in π. Then, we assign

$$\overline{\sigma}(h) := \begin{cases} (a, I) & \text{if } \mathsf{last}(h) = (\ell, v) \\ \sigma(h) & \text{otherwise.} \end{cases}$$

Remark: There may be more than one winning play from a certain state with lowest penalty. In this case, one run π will be randomly chosen.

$\overline{\sigma}$ is memoryless due to the construction. For every state (ℓ, v) we assigned the first move of some winning play with lowest penalty to every history that ends in (ℓ, v) which implies the memoryless property. □

4 Conclusion and future directions

In this work, we have studied timed games with penalty and the permissiveness of multi-strategies. We introduced the penalty for the controller as a quantitative measure of permissiveness. The focus was on the reachability objective. We have shown that deciding a most permissive strategy on one clock is in PTime and in particular PTime-complete for turn-based reachability games. Moreover, memoryless strategies suffice for the reachability winning condition in the general case.

As a future work, we want to investigate the game played on two and more clocks on the one hand, and extend the timed game with penalty to a concurrent setting. Since deciding a winning strategy in classical timed games is already ExpTime-complete, we conjecture the complexity for timed game with penalty to be a very hard computational problem. It would be very interesting to determine the decidability and complexity classes.

References

[1] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, April 1994.

[2] Gerd Behrmann, Alexandre David, and Kim Guldstrand Larsen. A Tutorial on Uppaal. In Marco Bernardo and Flavio Corradini, editors, *Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM-RT 2004*, volume 3185 of *Lecture Notes in Computer Science*, pages 200–236, Bertinoro, Italy, September 2004. Springer.

[3] Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO - Theoretical Informatics and Applications*, 36:261–275, 7 2002.

[4] Patricia Bouyer, Marie Duflot, Nicolas Markey, and Gabriel Renault. Measuring permissivity in finite games. In *Proceedings of the 20th International Conference on Concurrency Theory (CONCUR'09)*, volume 5710 of *Lecture Notes in Computer Science*, pages 196–210, Bologna, Italy, September 2009. Springer.

[5] Patricia Bouyer, Nicolas Markey, Jörg Olschewski, and Michael Ummels. Measuring permissiveness in parity games: Mean-payoff parity games revisited. In Tevfik Bultan and Pao-Ann Hsiung, editors, *Proceedings of the 9th International Symposium on Automated Technology for Verification and Analysis (ATVA'11)*, volume 6996 of *Lecture Notes in Computer Science*, pages 135–149, Taipei, Taiwan, October 2011. Springer.

[6] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robust reachability in timed automata: A game-based approach. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP'12) – Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 128–140, July 2012.

[7] Stephen Arthur Cook. *Towards a Complexity Theory of Synchronous Parallel Computation*. Technical report (University of Toronto. Dept. of Computer Science). University of Toronto, Department of Computer Science, 1980.

[8] Costas Courcoubetis and Mihalis Yannakakis. Minimum and maximum delay problems in real-time systems. *Form. Methods Syst. Des.*, 1(4):385–415, December 1992.

[9] Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. The element of surprise in timed games. In *CONCUR*, pages 142–156, 2003.

[10] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robustness and implementability of timed automata. In *Proc. Joint Conf. Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System (FORMATS+FTRTFT04), volume 3253 of LNCS*, pages 118–133, 2004.

[11] David L. Dill. Timing assumptions and verification of finite-state concurrent systems. In *Automatic Verification Methods for Finite State Systems*, pages 197–212, 1989.

[12] John Fearnley and Marcin Jurdzinski. Reachability in two-clock timed automata is PSPACE-complete. In *ICALP (2)*, pages 212–223, 2013.

[13] Leslie M. Goldschlager, Ralph A. Shaw, and John Staples. The maximum flow problem is log space complete for p. *Theoretical Computer Science*, 21(1):105 – 111, 1982.

[14] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. A compendium of problems complete for p, 1991.

[15] Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In *In Proceedings of HART 97*, pages 331–345, 1997.

[16] Neil Immerman. *Descriptive complexity.* Graduate texts in computer science. Springer, 1999.

[17] Ketan Mulmuley and Pradyut Shah. A lower bound for the shortest path problem. *Journal of Computer and System Sciences*, 63(2):253 – 267, 2001.

[18] Ocan Sankur, Patricia Bouyer, and Nicolas Markey. Shrinking timed automata. In Supratik Chakraborty and Amit Kumar, editors, *Proceedings of the 31st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'11)*, volume 13 of *Leibniz International Proceedings in Informatics*, pages 90–102, Mumbai, India, December 2011. Leibniz-Zentrum für Informatik.

[19] Ocan Sankur, Patricia Bouyer, Nicolas Markey, and Pierre-Alain Reynier. Robust controller synthesis in timed automata. In Pedro R. D'Argenio and Hernán Melgratti, editors, *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *Lecture Notes in Computer Science*, pages 546–560, Buenos Aires, Argentina, August 2013.

A Remaining proofs of Section 3

Lemma 3.1

Proof of Lemma 3.1. (See page 7.) For any $\rho = \mu_1\mu_2\dots\mu_n$, if ρ' is a sub-play, then it is of the form $\mu_{i_1}\mu_{i_2}\dots\mu_{i_k}$ with $\{i_1,\dots,i_k\} \subseteq \{1,\dots,n\}$. Consequently, the following holds:

$$\mathsf{penalty}_{\rho'} = \sum_{i=i_1}^{i_k} \mathsf{penalty}(\mu_i) \leq \sum_{i=1}^{n} \mathsf{penalty}(\mu_i) = \mathsf{penalty}_{\rho},$$

which proves the statement. □

Lemma 3.2

Proof of Lemma 3.2. (See page 11.) For the sake of contradiction, let us assume that for some clock valuation $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and some iteration step i the function decreases, i.e., $O_\ell^{(i)}(v) > O_\ell^{(i)}(v+\varepsilon)$ for some $\varepsilon > 0$. Consequently at $v+\varepsilon$ there are some "transition opportunities" which were not available at v. However this contradicts with the definition of the timed automaton model defined in Section 2.2. □

Lemma 3.3

Proof of Lemma 3.3. (See page 11.) For any $\ell \in L_F$, the claim holds by definition.

For the sake of contradiction, assume that for some $\ell \in L \setminus L_F$, $i \in \mathbb{N}$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, $O_\ell^{(i)}(v) < O_\ell^{(i+1)}(v)$ holds. Then we have the following scenario:

- $\exists \ell'_1 :\ \ell \longrightarrow \ell'_1$ such that $O_{\ell'_1}^{(i)}(v'_1) > O_{\ell'_1}^{(i+1)}(v'_1)$ for some $v'_1 \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$
- $\exists \ell'_2 :\ \ell'_1 \longrightarrow \ell'_2$ such that $O_{\ell'_2}^{(i)}(v'_2) > O_{\ell'_2}^{(i+1)}(v'_2)$ for some $v'_2 \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$
- $\dots$
- $\exists \ell'_n :\ \ell'_{n-1} \longrightarrow \ell'_n$ such that $O_{\ell'_n}^{(i)}(v'_n) > O_{\ell'_n}^{(i+1)}(v'_n)$ for some $v'_n \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ and $\ell'_n \longrightarrow \ell_{goal}$

  $\Rightarrow \quad O_{\ell_{goal}}^{(i)}(v') > O_{\ell_{goal}}^{(i+1)}(v')$ for some $v' \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$

Contradiction! □

Lemma 3.4

Proof of Lemma 3.4. (See page 11.) We show this property constructively by induction on the number of computation steps i.

For each state (ℓ, v) and every i, we construct a strategy $\sigma^i_{(\ell,v)}$ that wins in maximum i steps *starting from* (ℓ, v) such that

$$\mathsf{penalty}_{(\ell,v)}(\sigma^i_{(\ell,v)}) \leq O^{(i)}(\ell, v) + \epsilon \qquad \forall \epsilon > 0 \tag{10}$$

Base case: $i = 0$
For any timed multi-strategy σ of Player 1, the above stated property trivially holds for 0

computation steps. For any $\ell \in L_F$, σ wins in 0 steps, hence $\mathsf{penalty}(\sigma) = O^{(0)}(\ell, v) = 0$ for any $v \in \mathcal{X}$. Conversely, for any $\ell \notin L_F$, $\mathsf{penalty}(\sigma) = O^{(0)}(\ell, v) = +\infty$ holds for any $v \in \mathcal{X}$.

Step: $i \mapsto i + 1$

Assume that the induction hypothesis holds for some i and according to Lemma 3.2 ($i \mapsto O^{(i)}(\ell, v)$ is non-decreasing), we consider two cases for any state $(\ell, v) \in L \times \mathcal{C}(\mathcal{X})$:

- $O^{(i+1)}(\ell, v) = O^{(i)}(\ell, v)$
  Let $\sigma^{i+1}_{(\ell,v)} := \sigma^{i}_{(\ell,v)}$. Then $\mathsf{penalty}(\sigma^{i+1}_{(\ell,v)}) = \mathsf{penalty}(\sigma^{i}_{(\ell,v)}) \le O^{(i)}(\ell, v) + \varepsilon = O^{(i+1)}(\ell, v) + \epsilon$.
- $O^{(i+1)}(\ell, v) < O^{(i)}(\ell, v)$
  The idea here is to play the action and interval that minimizes the penalty for the whole path according to the definition of O (as in Equation 4). By induction hypothesis, the almost optimal strategy $\sigma^{i}_{(\ell',v')}$ that wins in maximum i steps from some state ℓ exists. Now consider all locations ℓ' that have a transition to ℓ and assume that the run that witnesses $O^{(i+1)}(\ell', v')$ for some clock valuation v' goes through (ℓ, v) after the first step $(a, I, t) \in \Sigma \times \mathcal{I}(\mathbb{R}_{\ge 0}) \times \mathbb{R}_{\ge 0}$ in the play and further plays according to $\sigma^{i}_{(\ell',v')}$, i.e., $O^{(i+1)}(\ell', v') = \frac{1}{|I|} + O^{(i)}(\ell, v)$ and $(\ell', v) \xrightarrow{a,I} (\ell', a, v' + I) \xrightarrow{t} (\ell, v = v' + t) \to \ldots \to (\ell_f, v'')$ for some $\ell_f \in L_F$ and $v \in \mathbb{R}^{|\mathcal{X}|}_{\ge 0}$. Further, this means that $\mathsf{penalty}(a, I) + \mathsf{penalty}_{(\ell',v')}(\sigma^{i}_{(\ell',v')})$ is the lowest possible penalty from (ℓ', v'). Finally, we set $\sigma^{i+1}_{(\ell,v)}$ as follows:

$$\sigma^{i+1}_{(\ell,v)}(\rho) = \begin{cases} (a, I) & \text{if } |\rho| = 0, \\ \sigma^{i}_{(\ell',v')}(\rho) & \text{otherwise.} \end{cases}$$

  The following inequation shows that our construction is correct and preserves the property in (10):

$$\begin{aligned} \frac{1}{|I|} + \mathsf{penalty}_{(\ell,v)}(\sigma^{i}_{(\ell,v)}) &= \mathsf{penalty}_{(\ell',v')}(\sigma^{i+1}_{(\ell',v')}) \\ &\le \frac{1}{|I|} + O^{(i)}(\ell, v) + \epsilon \\ &= O^{(i+1)}(\ell', v') + \epsilon \qquad \forall \epsilon > 0 \end{aligned}$$

□

Lemma 3.5

Proof of Lemma 3.5. (See page 11.) We show $O^{(i)}(\ell, v) \le \mathsf{penalty}(\sigma)$ for any winning σ by a simple induction on the number of iterations i.

Base: For $i = 0$ and any clock valuation $v \in \mathbb{R}$, the property is trivial, since $O^{(0)}(\ell, v) = \mathsf{penalty}(\sigma) = +\infty$ for any $\ell \notin L_F$ and $O^{(0)}(\ell, v) = \mathsf{penalty}(\sigma) = 0$ for any $\ell \in L_F$.

Step: Assuming that the property holds for some $i > 0$, $\ell_i \in L$ and $v_i \in \mathbb{R}$, i.e., $O^{(i)}(\ell_i, v_i) \le \mathsf{penalty}(\sigma)$. Consider all locations ℓ_{i+1} that have a transition to ℓ_i, i.e., $\ell_{i+1} \xrightarrow{a,g,R} \ell_i$ for some $a \in \Sigma, g \in \mathcal{C}(\mathcal{X})$ and $R \subseteq \mathcal{X}$. According to the recursive definition of $O^{(i+1)}$ and for some $v_{i+1} \le v_i$, $O^{(i+1)}(\ell_{i+1}, v_{i+1})$ computes the minimum between the sum of penalty to get to ℓ_i and the penalty from ℓ_i to some target location, and the penalty from ℓ_i to some target location that takes not more than $i + 1$ steps, which is the lowest possible penalty for strategies that win in $i + 1$ steps from (ℓ_{i+1}, v_{i+1}). □

Lemma 3.6

Proof of Lemma 3.6. (See page 11.) For the sake of contradiction, we assume that for some $E_r \subseteq E_{reset}$ the following holds:

$$\exists h_0 \in \mathsf{Out}_{E_r}(\sigma) \quad \forall h \succeq h_0 \quad \exists h' \succ h : \qquad h' \in \mathsf{Out}_{E_r}(\sigma)$$

Then, for any $h \succeq h_0$, we get

- $\exists h_1 \succ h :\ h_1 \in \mathsf{Out}_{E_r}(\sigma)$
- $\exists h_2 \succ h_1 :\ h_2 \in \mathsf{Out}_{E_r}(\sigma)$ (by transitivity $h_2 \succ h$)
- ...
- $\forall n \in \mathbb{N},\ \exists h_{n+1} \succ h_n :\ h_{n+1} \in \mathsf{Out}_{E_r}(\sigma)$,
  (by transitivity $h_{n+1} \succ h$)

which leads to an infinite chain. Further, some edge $e \in E_r$ will be taken infinitely often, which means that some configuration $(\ell, v(x) = 0)$ will be visited infinitely in the run $\rho := (\lim_{i\to+\infty} h_i) \in \mathsf{Out}_{E_r}(\sigma) \subseteq \mathit{Out}(\sigma)$, where $\ell \in L \setminus L_F$. Hence, no $\ell \in L_F$ will be ever visited and hence, σ is not winning. Note that the game ends when any $\ell{\in}L_F$ is visited. Consequently σ is not a winning multi-strategy. Contradiction! □

Lemma 3.8

Proof of Lemma 3.8. (See page 12) We show it by contradiction, similar as in the proof of Lemma 3.6.

Let us assume that for some subset of locations $L' \subseteq L$, the following holds:

$$\exists h \in \mathsf{Out}_{E_{reset}}(\sigma) \quad \exists \rho_0 \in \Gamma^{\sigma,h}_{L'} \quad \forall \rho \succeq \rho_0 \quad \exists \rho' \succ \rho : \qquad \rho' \in \Gamma^{\sigma,h}_{L'}.$$

Then, for any $\rho \succeq \rho_0$ we get

- $\exists \rho_1 \succ \rho :\ \rho_1 \in \Gamma^{\sigma,h}_{L'}$
- $\exists \rho_2 \succ \rho :\ \rho_2 \in \Gamma^{\sigma,h}_{L'}$ (by transitivity $\rho_2 \succ \rho$)
- ...
- $\forall n \in \mathbb{N},\ \exists \rho_{n+1} \succ \rho_n :\ \rho_{n+1} \in \Gamma^{\sigma,h}_{L'}$,
  (by transitivity $\rho_{n+1} \succ \rho$)

which leads to an infinite chain. We note that any infinite path is not winning as the game is won by Player 1 when any target location is visited (the first time). Hence, $\lim_{n\to+\infty} \rho_n$ is not winning. Consequently, σ is not a winning timed multi-strategy for Player 1, which is a contradiction. □

Corollary 3.11

Proof of Corollary 3.11. (See page 14.) Given σ, we can construct a winning σ' that takes each clock-reset edge at most once according to Theorem 3.9. Further, using σ', we can construct a winning σ'' that visits each location at most once before the first occurrence of a clock-reset edge, after the last occurrence of any clock-reset edge and between two reset edges e, e' (where no other e'' is in-between e and e'). It is not hard to see that the result of the two procedures leads to a length bounded by $|Q| \cdot (|\delta|_{reset} + 1)$, i.e., for any $\rho \in \mathsf{Out}(\sigma'')$, $\mathsf{length}(\rho) \leq |Q| \cdot (|\delta|_{reset} + 1)$. □

Lemma 3.15

Proof of Lemma 3.15. (See page 18.)

We show that for all $(\ell, v) \in L_{II} \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$, $O^{(i)}(\ell, v)$, $O(\ell, v)$ maximizes the value/penalty, i.e., it computes the highest possible value by taking the Player 2's objective into account. For any (ℓ, v), we consider the number of moves, i, that is needed until the game ends, i.e., that either Player 1 has won or the play has reached some set from which Player 1 cannot win anymore.

We show it by induction on i. Let $i = 0$, then either Player 1 has already either won or definitely lost. Let $i := i + 1$ and assume $O^{(i)}(\ell', v')$ holds for all outgoing adjacent states (ℓ', v'). Since the game is clearly determined, either (1.) Player 1 has a winning strategy, or (2.) Player 2 has a winning strategy. In any case, Player 2 tries to maximize the penalty for Player 2. Let $(a, t) \in \Sigma \times \mathbb{R}_{\geq 0}$ be the move that leads to $(\ell', v' = v + t)$ and that maximizes the penalty from (ℓ, v). The penalty from (ℓ, v) is equal to the penalty from (ℓ', v'), i.e.,

$$O^{(i+1)}(\ell, v) = O^{(i)}(\ell', v' = v + t[R := 0]) = \sup_{\substack{\ell' \in \ell E \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \sup_{t \in \tau(v,g)} \left(O^{(i)}(\ell', v + t[R := 0] \right),$$

which is what we wanted to show. □

Lemma 3.16

Proof of Lemma 3.16. (See page 18.) Assume there exist n possible next locations $(\ell_1, v), \ldots, (\ell_n, v)$.

- Player 1 chooses some (ℓ_i, v) that has the lowest penalty from there to a winning location. Note that each ℓ_i is reachable by a different action. Let π_i be the sub-play that witnesses the optimal run from (ℓ_i, v) to a winning location. Then, $\mathsf{penalty}_{\pi_i} \leq \mathsf{penalty}_{\pi_j}$ for all $j \neq i$, which means the penalty from (ℓ, v) is

$$\inf_{\substack{a_i \in \Sigma(\ell, v) \\ \ell \xrightarrow{a_i, g_i, R_i} \ell_i}} (\mathsf{penalty}_{\pi_i}) = \inf_{\substack{a_i \in \Sigma(\ell, v) \\ \ell \xrightarrow{a_i, g_i, R_i} \ell_i}} (O^{(i)}(\ell_i, v)) = O^{(i+1)}(\ell, v)$$

  for some i.

- Player 2 chooses some (ℓ_i, v) that has the maximum penalty among all adjacent locations. Let π_i be the subrun that witnesses the optimal run from (ℓ_i, v) to a winning location. Then, $\mathsf{penalty}_{\pi_i} \geq \mathsf{penalty}_{\pi_j}$ for all $j \neq i$, which means the penalty from (ℓ, v) is

$$\sup_{\substack{a \in \Sigma(\ell, v) \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \left(\mathsf{penalty}_{\pi_i} \right) = \sup_{\substack{a \in \Sigma(\ell, v) \text{ s.t.} \\ \ell \xrightarrow{a,g,R} \ell'}} \left(O^{(i)}(\ell', v[R := 0]) \right) = O^{(i+1)}(\ell, v)$$

  for some i.

This shows that the equations 8 and 9 correctly compute the optimal penalty according to the objectives of Player 1 and Player 2 respectively. □

Theorem 3.18

Proof of Theorem 3.18. (See page 18.) We show this property by induction over all subsets $E_i \subseteq E_{reset}$ with $|E_i| = i$. Given a winning timed multi-strategy σ of Player 1, we construct a winning timed multi-strategy σ_{E_i} that satisfies the following properties:

(1.) $\forall \rho \in \mathsf{Out}(\sigma_{E_i}), \forall e \in E_i :\quad \rho$ visits e maximum once,

(2.) $\mathsf{penalty}(\sigma_{E_i}) \leq \mathsf{penalty}(\sigma)$.

Base case: $\quad E_0 = \emptyset$
Condition (1.) holds trivially for any multi-strategy. Let $\sigma_{E_0} = \sigma$, then (2.) obviously holds as well.

Step:
We randomly pick any reset transition/edge $e \in E_{reset} \setminus E_i$ and let $E_{i+1} = E_i \cup \{e\}$. We consider all runs/histories in $h \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ where the last transition is e. Applying Lemma 3.6, there exists a maximum extension $h_{max} \in Out_{\{e\}}(\sigma_{E_i})$ for any history h. Further, we consider any history γ with $\mathsf{last}(\gamma) \in L_I \cup L_I^u$ that is a prefix of any outcome $\rho \in \mathsf{Out}(\sigma_{E_i})$. Either e is not contained in γ, i.e., $|\gamma|_e = 0$, or γ is of the form $h \cdot m$ with $h \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ and a finite, possibly empty, extension m. We distinguish two cases, where the first case if equivalent to the construction for the non-turnbased game setting described in Section 3.2.4 and construct $\sigma_{E_{i+1}}$ as follows:

- Case 1: $\quad \mathsf{target}(e) \in L_I \cup L_I^u$

$$\sigma_{E_{i+1}}(\gamma) = \begin{cases} \sigma_{E_i}(\gamma) & \text{if } |\gamma|_e = 0 \text{ or } |\gamma|_e > 1 \\ \sigma_{E_i}(h_{max} \cdot m) & \text{otherwise } (|\gamma|_e = 1). \end{cases}$$

- Case 2: $\quad \mathsf{target}(e) \notin L_I \cup L_I^u$
  For each history $\gamma \in \{\gamma \prec \rho \mid \rho \in \mathsf{Out}(\sigma_{E_i}), \mathsf{last}(\gamma) \in L_I \cup L_I^u\}$, we consider the set of sub-plays of γ between $\mathsf{target}(e) =: \hat{\ell}$ and the first following occurrence of a location $\overline{\ell} \in L_I \cup L_I^u$, i.e.

$$M_\gamma^e = \left\{ m = (\hat{\ell}, \mathbf{0}) \to (\hat{\ell}_1, v_1) \to \ldots \to (\hat{\ell}_k, v_k) \to (\ell', v') \sqsubseteq \gamma \;\middle|\; \begin{array}{l} |m|_e = 0, \\ \ell' \in L_I \cup L_I^u, \\ \hat{\ell}_i \notin L_I \cup L_I^u \text{ for } 1 \leq i \leq k \end{array} \right\}.$$

  Note that M_γ^e is non-empty, that means, between two clock-reset transition e and e' there has to be controllable locations for Player 1. Otherwise σ is not winning anymore since between $\mathsf{source}(e)$ and $\mathsf{target}(e')$ there would be an accessible loop without Player 1 locations. Then Player 2 could play the strategy to loop forever.

  Consequently, γ is *either* of the form $h \cdot m_0 \cdot m$ with $h \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$, $m_0 \in M_\gamma^e$ and a finite, possibly empty, extension of the play m, *or* $|\gamma|_e = 0$. We construct $\sigma_{E_{i+1}}$ as follows:

$$\sigma_{E_{i+1}}(\gamma) = \begin{cases} \sigma_{E_i}(h_{max} \cdot m_0 \cdot m) & \text{if } |\gamma|_e = 1, \\ \sigma_{E_i}(\gamma) & \text{otherwise.} \end{cases}$$

Claim: The conditions (1.) and (2.) hold for $\sigma_{E_{i+1}}$

(1.) For the sake of contradiction, let us assume, there exist some run $\rho \in \mathsf{Out}(\sigma_{E_{i+1}})$ that visits $e \in E_{i+1} \setminus E_i$ more than once. Consider all histories $h \in \mathsf{Out}(\sigma_{E_{i+1}})$ that ends with the first occurrence of e. For the case when Player 1 controls the arrived location after taking e, the multi-strategy for Player 1 at this point is going to be replaced by the strategy σ_{E_i} after the maximum extension $h_{max} \in \mathsf{Out}_{\{e\}}(\sigma_{E_i})$ of h that ends with e. For the case when Player 1 does not control the arrived location after taking e, the multi-strategy for Player 1 after arriving the next controllable location *on all possible continuing plays* is going to be replaced by σ_{E_i} after the last possible occurrence of e in the continuing play and the corresponding extension until the next controllable location for Player 1 is reached.

In both cases, there must be another occurrence of e after the last occurrence of e such that e can occur more than once in any outcome of $\sigma_{E_{i+1}}$. A contradiction!

(2.) Analogue to the proof of Theorem 3.9, we consider every play $\rho \in \mathsf{Out}(\sigma_{E_i})$. According to the construction of $\sigma_{E_{i+1}}$, $\sigma_{E_{i+1}}$ generates a winning sub-play ρ' for every $\rho \in \mathsf{Out}(\sigma_{E_i})$, $\rho' \sqsubseteq \rho$ and there is no $\overline{\rho} \in \mathsf{Out}(\sigma_{E_{i+1}})$ such that $\rho \sqsubseteq \overline{\rho}$. Applying Lemma 3.1, we get $\mathsf{penalty}_{\rho'} \leq \mathsf{penalty}_{\rho}$ and consequently $\mathsf{penalty}(\sigma_{E_{i+1}}) \leq \mathsf{penalty}(\sigma_{E_i}) \leq \mathsf{penalty}(\sigma)$.

□

Theorem 3.19

Proof of Theorem 3.19. (See page 19.) We show this property by induction on all subsets $L_i \subseteq L$ with $|L_i| = i$. Given a winning timed multi-strategy σ, we construct a winning timed multi-strategy σ_{L_i} such that the following properties hold:

(1.) $\forall \ell \in L_i \quad \forall e \in E_{reset} \quad \forall h \in \mathsf{Out}_{\{e\}}(\sigma_{L_i}) \cup \{\varepsilon\} \quad \forall \rho \in \Gamma^h_{L_i}(\sigma_{L_i}) :$ $\qquad \rho$ visits ℓ at most once,

(2.) $\mathsf{penalty}(\sigma_{L_i}) \leq \mathsf{penalty}(\sigma)$.

Base case: $\quad L_0 = \emptyset$
(1.) holds because $\Gamma^h_{L_i}(\sigma_{L_i}) = \emptyset$. Let $\sigma_{L_0} = \sigma$, then (2.) holds trivially.
Step:
We pick any $\ell \in L \setminus L_i$ and let $L_{i+1} = L_i \cup \{\ell\}$. By Lemma 3.8 there exists a maximal continuation $\rho_{max} \in \Gamma^h_{\{\ell\}}(\sigma_{L_i})$ for any outcome $h \cdot \rho$ of σ_{L_i} such that $h \in \mathsf{Out}_{E_{reset}}(\sigma_{L_i})$ and $|\rho|_{E_{reset}} = 0$. We consider finite outcomes $\gamma \in \mathsf{Out}(\sigma_{E_i})$ that are either of the form $h \cdot \rho \cdot m$ where $h \in \mathsf{Out}_{E_{reset}} \cup \{\varepsilon\}$, $\rho \in \Gamma^h_{\{\ell\}}(\sigma_{E_i})$ and m is any continuation, or $|\gamma|_{E_{reset}} = 0$. Additionally, we have to distinguish the cases whether ℓ is controllable by Player 1 or not. We construct $\sigma_{L_{i+1}}$ as follows:

- Case 1: $\quad \ell \in L_I \cup L^u_I$

$$\sigma_{L_{i+1}}(\gamma) = \begin{cases} \sigma_{L_i}(h \cdot \rho_{max} \cdot m) & \text{if } |\rho|_\ell = 1 \\ \sigma_{L_i}(\gamma) & \text{otherwise.} \end{cases}$$

- Case 2: $\quad \ell \notin L_I \cup L^u_I$
  Here, we need to consider the set of sub-plays γ between ℓ and the first following location

that is controllable by Player 1, i.e.

$$N_\gamma^{(\ell,v)} = \left\{ n = (\ell, v) \to (\hat{\ell}_1, v_1) \to \ldots \to (\hat{\ell}_k, v_k) \to (\ell', v') \sqsubseteq \gamma \;\middle|\; \begin{array}{l} |n|_{E_{reset}} = 0, \\ \hat{\ell}_i \notin L_I \cup L_I^u \text{ for } 1 \le i \le k \end{array} \right\}$$

Note that $N_\gamma^{(\ell,v)}$ is non-empty, that means, between two clock-reset transition e and e', and between two occurrences of a location ℓ controlled by Player 2, there has to be locations controlled by Player 1. Otherwise σ is not winning anymore since there would be an accessible loop without Player 1 locations. Then Player 2 could play the strategy to loop forever.

Any outcome $\gamma \in \mathsf{Out}_{\{e\}}(\sigma_{L_i})$ is either of the form $h \cdot \rho \cdot n_0 \cdot m$ such that $h \in \mathsf{Out}_{E_{reset}}(\sigma_{L_i}) \cup \{\varepsilon\}$, $\rho \in \Gamma^h_{\{\ell\}}(\sigma_{L_i})$, $n_0 \in N_\gamma^{(\ell,v)}$ where ρ ends in (ℓ, v), and m is a finite, possibly empty, extension that ends in a location in L_I or L_I^u. We construct $\sigma_{L_{i+1}}$ in this case as follows:

$$\sigma_{L_{i+1}}(\gamma) = \begin{cases} \sigma_{L_i}(h \cdot \rho_{max} \cdot n_0 \cdot m) & \text{if } |\rho|_\ell = 1 \\ \sigma_{L_i}(\gamma) & \text{otherwise.} \end{cases}$$

Claim: The properties (1.) and (2.) hold for $\sigma_{L_{i+1}}$.

(1.) According to the construction of σ_{L_i}, any path that contains ℓ more than once will be replaced by a path that contains ℓ only once, since after the first occurrence of ℓ, $\sigma_{L_{i+1}}$ takes the strategy of σ_{L_i} after the last occurrence of ℓ. In case, ℓ still occurs more than once, there must be another occurrence of ℓ after the last occurrence of ℓ which is absurd.

(2.) Here we again observe that we create sub-plays and do not “stretch” any play in comparison to σ_{L_i}, which implies property (2.).

□

Proposition 3.20

Correctness of the claim in Proposition 3.20. (See page 20.)

- “⇒”
  Let us assume that $P^G(s,t)$ holds. If $s = t$, Player 1 trivially has a winning strategy without penalty. Otherwise, there exists a path from s to t without vertex repetitions where at each $V_\exists$ vertex there is an outgoing adjacent vertex from which t is “alternated reachable” and at each $V_\forall$ vertex not equal to t there is at least one outgoing edge and from every outgoing adjacent vertex t is reachable. Considering this path, one can easily construct a winning multi-strategy σ for Player 1 where the penalty is bounded by λ since the cost of every transition is 1 due to construction.

- “⇐”
  Let us assume, Player 1 has a winning strategy σ with $\mathsf{penalty}(\sigma) \le \lambda$. That means, there exists a path from s to t such that in each Player 2 location reaching t can not be prevented. Let $\rho = s \rightsquigarrow t \in \mathsf{Out}(\sigma)$ be the winning play with $\mathsf{penalty}(\sigma) \le \lambda$. W.l.o.g., we can assume that there is no state repetition in ρ due to the Theorems 3.9 and 3.10 and because every transition resets the clock. Further the cost of any transition for Player 1 is 1 according to

the construction. Consequently, we can conclude that every occurring Player 1 location in ρ has an outgoing transition to either another Player 1 location where Player 1 has a winning strategy or to a Player 2 location where Player 2 is not able to make a move such that t cannot be reached anymore. Since, Player 1 locations correspond to $V_\exists$ vertices and Player 2 locations to $V_\forall$ vertices and because both structures are isomorphic apart from the labels of $\mathcal{G}$, there is an $s \rightsquigarrow t$ path in the alternated graph G.

□