Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 340-351

# Average-Time Games*

**Marcin Jurdziński[1] and Ashutosh Trivedi[2]**

[1] Department of Computer Science, University of Warwick, UK
mju@dcs.warwick.ac.uk

[2] Computing Laboratory, University of Oxford, UK
trivedi@comlab.ox.ac.uk

ABSTRACT. An average-time game is played on the infinite graph of configurations of a finite timed automaton. The two players, Min and Max, construct an infinite run of the automaton by taking turns to perform a timed transition. Player Min wants to minimize the average time per transition and player Max wants to maximize it. A solution of average-time games is presented using a reduction to average-price game on a finite graph. A direct consequence is an elementary proof of determinacy for average-time games. This complements our results for reachability-time games and partially solves a problem posed by Bouyer et al., to design an algorithm for solving average-price games on priced timed automata. The paper also establishes the exact computational complexity of solving average-time games: the problem is EXPTIME-complete for timed automata with at least two clocks.

## 1 Introduction

*Real-time open systems* are computational systems that interact with environment and whose correctness depends critically on the time at which they perform some of their actions. The problem of design and verification of such systems can be formulated as *two-player zero-sum games*. A heart pacemaker is an example of a real-time open system as it interacts with the environment (heart, body movements, and breathing) and its correctness depends critically on the time at which it performs some of its actions (sending pace signals to the heart in real time). Other examples of safety-critical real-time open systems include nuclear reactor protective systems, industrial process controllers, aircraft-landing scheduling systems, satellite-launching systems, etc. Designing correct real-time systems is of paramount importance. Timed automata [2] are a popular and well-established formalism for modeling real-time systems, and games on timed automata can be used to model real-time open systems. In this paper, we introduce *average-time games* which model the interaction between the real-time open system and the environment; and we are interested in finding a strategy of the system which results in minimum average-time per transition, assuming adversarial environment.

**Related Work.** Games with quantitative payoffs can be studied as a model for optimal-controller synthesis [3, 1, 6]. Among various quantitative payoffs the average-price payoff [9, 8] is the most well-studied in game theory, Markov decision processes, and planning literature [8, 14], and it has numerous appealing interpretations in applications. Most algorithms for solving Markov decision processes [14] or games with average-price payoff

*This work was partially supported by the EPSRC grants EP/E022030/1 and EP/F001096/1.

work for finite graphs only [15, 8]. Asarin and Maler [3] presented the first algorithm for games on timed automata (timed games) with a quantitative payoff: reachability-time payoff. Their work was later generalized by Alur et al. [1] and Bouyer et al. [6] to give partial decidability results for reachability-price games on linearly-priced timed automata. The exact computational complexity of deciding the value in timed games with reachability-time payoff was shown to be EXPTIME in [11, 7]. Bouyer et al. [5] also studied the more difficult average-price payoffs, but only in the context of scheduling, which in game-theoretic terminology corresponds to 1-player games. They left open the problem of proving decidability of 2-player average-reward games on linearly-priced timed automata. We have recently extended the results of Bouyer et al. to solve 1-player games on more general concavely-priced timed automata [12]. In this paper we address the important and non-trivial special case of average-time games (i.e., all locations have unit costs), which was also left open by Bouyer et al.

**Our Contributions.** Average-time games on timed automata are introduced. This paper gives an elementary proof of determinacy for these games. A new type of region [2] based abstraction—boundary region graph—is defined, which generalizes the corner-point abstraction of Bouyer et al. [5]. Our solution allows computing the value of average-time games for an arbitrary starting state (i.e., including non-corner states). Finally, we establish the exact complexity of solving average-time games: the problem is EXPTIME-complete for timed automata with at least two clocks.

**Organization of the Paper.** In Section 2 we discuss average-price games (also known as mean-payoff games) on finite graphs and cite some important results for these games. In Section 3 we introduce average-time games on timed automata. In Section 4 we introduce some region-based abstractions of timed automata, including the closed region graph, and its subgraphs: the boundary region graph, and the region graph. While the region graph is semantically equivalent to the corresponding timed automaton, the boundary region graph has the property that for every starting state, the reachable state space is finite. We introduce average-time games on these graphs and show that if we have the solution of the average-time game for any of these graphs, then we get the solution of the average-time game for the corresponding timed automaton. In Section 5 we discuss the computational complexity of solving average-time games.

**Notations.** We assume that, wherever appropriate, sets $\mathbb{Z}$ of integers, $\mathbb{N}$ of non-negative integers and $\mathbb{R}$ of reals contain a maximum element $\infty$, and we write $\mathbb{N}_+$ for the set of positive integers and $\mathbb{R}_\oplus$ for the set of non-negative reals. For $n \in \mathbb{N}$, we write $(\!|n|\!)_\mathbb{N}$ for the set $\{0, 1, \ldots, n\}$, and $(\!|n|\!)_\mathbb{R}$ for the set $\{r \in \mathbb{R} \,:\, 0 \leq r \leq n\}$ of non-negative reals bounded by $n$. For a real number $r \in \mathbb{R}$, we write $|r|$ for its absolute value, we write $\lfloor r \rfloor$ for its integer part, i.e., the largest integer $n \in \mathbb{N}$, such that $n \leq r$, and we write $\wr r \int$ for its fractional part, i.e., we have $\wr r \int = r - \lfloor r \rfloor$.

## 2 Average-Price Games

A (perfect-information) two-player *average-price game* [15, 8] $\Gamma = (V, E, V_{\text{Max}}, V_{\text{Min}}, p)$ consists of a finite directed graph $(V, E)$, a partition $V = V_{\text{Max}} \cup V_{\text{Min}}$ of vertices, and a *price function* $\pi : E \to \mathbb{Z}$. A play starts at a vertex $v_0 \in V$. If $v_0 \in V_p$, for $p \in \{\, \text{Max}, \text{Min} \,\}$, then player $p$ chooses a successor of the current vertex $v_0$, i.e., a vertex $v_1$, such that $(v_0, v_1) \in E$, and $v_1$ becomes the new current vertex. When this happens then we say that player $p$ has made a move from the current vertex. Players keep making moves in this way indefinitely, thus forming an infinite path $r = (v_0, v_1, v_2, \dots)$ in the game graph. The goal of player Min is to minimize $\mathcal{A}_{\text{Min}}(r) = \limsup_{n\to\infty}(1/n) \cdot \sum_{i=1}^{n} \pi(v_{i-1}, v_i)$ and the goal of player Max is to maximize $\mathcal{A}_{\text{Max}}(r) = \liminf_{n\to\infty}(1/n) \cdot \sum_{i=1}^{n} \pi(v_{i-1}, v_i)$.

Strategies for players are defined as usual [15, 8]. We write $\Sigma_{\text{Min}}$ ($\Sigma_{\text{Max}}$) for the set of strategies of player Min (Max) and $\Pi_{\text{Min}}$ ($\Pi_{\text{Max}}$) for the set of positional strategies of player Min (Max). For strategies $\mu \in \Sigma_{\text{Min}}$ and $\chi \in \Sigma_{\text{Max}}$, and for an initial vertex $v \in V$, we write $\mathsf{run}(v, \mu, \chi)$ for the unique path formed if players start in the vertex $v$ and then they follow strategies $\mu$ and $\chi$, respectively. For brevity, we write $\mathcal{A}_{\text{Min}}(v, \mu, \chi)$ for $\mathcal{A}_{\text{Min}}(\mathsf{run}(v, \mu, \chi))$ and we write $\mathcal{A}_{\text{Max}}(v, \mu, \chi)$ for $\mathcal{A}_{\text{Max}}(\mathsf{run}(v, \mu, \chi))$.

For $v \in V$, we define the *upper value* $\overline{\mathsf{val}}(v) = \inf_{\mu\in\Sigma_{\text{Min}}} \sup_{\chi\in\Sigma_{\text{Max}}} \mathcal{A}_{\text{Min}}(v, \mu, \chi)$, and the *lower value* $\underline{\mathsf{val}}(v) = \sup_{\chi\in\Sigma_{\text{Max}}} \inf_{\mu\in\Sigma_{\text{Min}}} \mathcal{A}_{\text{Max}}(v, \mu, \chi)$. Note that the inequality $\underline{\mathsf{val}}(v) \le \overline{\mathsf{val}}(v)$ always holds. A game is determined if for every $v \in V$, we have $\underline{\mathsf{val}}(v) = \overline{\mathsf{val}}(v)$. We then write $\mathsf{val}(v)$ for this number and we call it the *value* of the average-price game at the vertex $v$.

We say that the strategies $\mu^* \in \Sigma_{\text{Min}}$ and $\chi^* \in \Sigma_{\text{Max}}$ are *optimal* for the respective players, if for every vertex $v \in V$, we have that $\sup_{\chi\in\Sigma_{\text{Max}}} \mathcal{A}_{\text{Min}}(v, \mu^*, \chi) = \overline{\mathsf{val}}(v)$ and $\inf_{\mu\in\Sigma_{\text{Min}}} \mathcal{A}_{\text{Min}}(v, \mu^*, \chi) = \underline{\mathsf{val}}(v)$. Liggett and Lippman [13] show that all perfect-information (stochastic) average-price games are positionally determined.

**THEOREM 1.** *[13] Every average-price game is determined, and optimal positional strategies exist for both players, i.e., for all $v \in V$, we have:*

$$\inf_{\mu\in\Pi_{Min}} \sup_{\chi\in\Sigma_{Max}} \mathcal{A}_{Min}(v, \mu, \chi) = \sup_{\chi\in\Pi_{Max}} \inf_{\mu\in\Sigma_{Min}} \mathcal{A}_{Max}(v, \mu, \chi).$$

The decision problem for average-price games is in NP $\cap$ co-NP; no polynomial-time algorithm is currently known for the problem.

# 3 Average-Time Games

## 3.1 Timed Automata

Before we present the syntax of the timed automata, we need to introduce some concepts. Fix a constant $k \in \mathbb{N}$ for the rest of this paper. Let $C$ be a finite set of *clocks*. Clocks in timed automata are usually allowed to take arbitrary non-negative real values. For the sake of simplicity and w.l.o.g [4], we restrict them to be bounded by some constant $k$, i.e., we consider only *bounded* timed automata models. A ($k$-bounded) *clock valuation* is a function $\nu : C \to (\!|k|\!)_{\mathbb{R}}$; we write $\mathcal{V}$ for the set $[C \to (\!|k|\!)_{\mathbb{R}}]$ of clock valuations. If $\nu \in \mathcal{V}$ and $t \in \mathbb{R}_{\oplus}$ then we write $\nu + t$ for the clock valuation defined by $(\nu + t)(c) = \nu(c) + t$, for all $c \in C$. For a set $C' \subseteq C$ of clocks and a clock valuation $\nu : C \to \mathbb{R}_{\oplus}$, we define $\mathsf{reset}(\nu, C')(c) = 0$ if $c \in C'$, and $\mathsf{reset}(\nu, C')(c) = \nu(c)$ if $c \notin C'$. A *corner* is an integer clock valuation, i.e., $\alpha$ is a corner if $\alpha(c) \in (\!|k|\!)_{\mathbb{N}}$, for every clock $c \in C$.

The set of *clock constraints* over the set of clocks $C$ is the set of conjunctions of *simple clock constraints*, which are constraints of the form $c \bowtie i$ or $c - c' \bowtie i$, where $c, c' \in C$, $i \in (\!|k|\!)_{\mathbb{N}}$, and $\bowtie \in \{ <, >, =, \leq, \geq \}$. There are finitely many simple clock constraints. For every clock valuation $\nu \in \mathcal{V}$, let $\mathsf{SCC}(\nu)$ be the set of simple clock constraints which hold in $\nu \in \mathcal{V}$. A *clock region* is a maximal set $P \subseteq \mathcal{V}$, such that for all $\nu, \nu' \in P$, $\mathsf{SCC}(\nu) = \mathsf{SCC}(\nu')$. In other words, every clock region is an equivalence class of the indistinguishability-by-clock-constraints relation, and vice versa. Note that $\nu$ and $\nu'$ are in the same clock region iff all clocks have the same integer parts in $\nu$ and $\nu'$, and if the partial orders of the clocks, determined by their fractional parts in $\nu$ and $\nu'$, are the same. For all $\nu \in \mathcal{V}$, we write $[\nu]$ for the clock region of $\nu$. A *clock zone* is a convex set of clock valuations, which is a union of a set of clock regions. Note that a set of clock valuations is a zone iff it is definable by a clock constraint. For $W \subseteq \mathcal{V}$, we write $\mathsf{clos}(W)$ for the smallest closed set in $\mathcal{V}$ which contains $W$. Observe that for every clock zone $W$, the set $\mathsf{clos}(W)$ is also a clock zone.

Let $L$ be a finite set of *locations*. A *configuration* is a pair $(\ell, \nu)$, where $\ell \in L$ is a location and $\nu \in \mathcal{V}$ is a clock valuation; we write $Q$ for the set of configurations. If $s = (\ell, \nu) \in Q$ and $c \in C$, then we write $s(c)$ for $\nu(c)$. A *region* is a pair $(\ell, P)$, where $\ell$ is a location and $P$ is a clock region. If $s = (\ell, \nu)$ is a configuration then we write $[s]$ for the region $(\ell, [\nu])$. We write $\mathcal{R}$ for the set of regions. A set $Z \subseteq Q$ is a *zone* if for every $\ell \in L$, there is a clock zone $W_\ell$ (possibly empty), such that $Z = \{(\ell, \nu) \,:\, \ell \in L \text{ and } \nu \in W_\ell\}$. For a region $R = (\ell, P) \in \mathcal{R}$, we write $\mathsf{clos}(R)$ for the zone $\{(\ell, \nu) \,:\, \nu \in \mathsf{clos}(P)\}$.

A *timed automaton* $\mathcal{T} = (L, C, S, A, E, \delta, \varrho)$ consists of a finite set of locations $L$, a finite set of clocks $C$, a set of *states* $S \subseteq Q$, a finite set of *actions* $A$, an *action enabledness function* $E : A \to 2^S$, a *transition function* $\delta : L \times A \to L$, and a *clock reset function* $\varrho : A \to 2^C$. We require that $S$, and $E(a)$ for all $a \in A$, are zones.

Clock zones, from which zones $S$, and $E(a)$, for all $a \in A$, are built, are typically specified by clock constraints. Therefore, when we consider a timed automaton as an input of an algorithm, its size should be understood as the sum of sizes of encodings of $L$, $C$, $A$, $\delta$, and $\varrho$, and the sizes of encodings of clock constraints defining zones $S$, and $E(a)$, for all $a \in A$. Our definition of a timed automaton may appear to differ from the usual ones [2, 4], but the differences are superficial.

For a configuration $s = (\ell, \nu) \in Q$ and $t \in \mathbb{R}_\oplus$, we define $s + t$ to be the configuration $s' = (\ell, \nu + t)$ if $\nu + t \in \mathcal{V}$, and we then write $s \rightharpoonup_t s'$. We write $s \to_t s'$ if $s \rightharpoonup_t s'$ and for all $t' \in [0, t]$, we have $(\ell, \nu + t') \in S$. For an action $a \in A$, we define $\mathsf{succ}(s, a)$ to be the configuration $s' = (\ell', \nu')$, where $\ell' = \delta(\ell, a)$ and $\nu' = \mathsf{reset}(\nu, \varrho(a))$, and we then write $s \xrightharpoonup{a} s'$. We write $s \xrightarrow{a} s'$ if $s \xrightharpoonup{a} s'$; $s, s' \in S$; and $s \in E(a)$. For technical convenience, and without loss of generality, we will assume throughout that for every $s \in S$, there exists $a \in A$, such that $s \xrightarrow{a} s'$. For $s, s' \in S$, we say that $s'$ is in the future of $s$, or equivalently, that $s$ is in the past of $s'$, if there is $t \in \mathbb{R}_\oplus$, such that $s \to_t s'$; we then write $s \to_* s'$.

For $R, R' \in \mathcal{R}$, we say that $R'$ is in the future of $R$, or that $R$ is in the past of $R'$, if for all $s \in R$, there is $s' \in R'$, such that $s'$ is in the future of $s$; we then write $R \to_* R'$. Similarly, for $R, R' \in \mathcal{R}$, we write $R \xrightarrow{a} R'$ if there is $s \in R$, and there is $s' \in R'$, such that $s \xrightarrow{a} s'$.

A *timed action* is a pair $\tau = (t, a) \in \mathbb{R}_\oplus \times A$. For $s \in Q$, we define $\mathsf{succ}(s, \tau) = \mathsf{succ}(s, (t, a))$ to be the configuration $s' = \mathsf{succ}(s + t, a)$, i.e., such that $s \rightharpoonup_t s'' \xrightharpoonup{a} s'$, and we then write $s \xrightarrow{a}_t s'$. We write $s \xrightarrow{a}_t s'$ if $s \rightarrow_t s'' \xrightarrow{a} s'$, and we then say that $(s,(t,a),s')$ is a *transition* of the timed automaton. If $\tau = (t,a)$ then we write $s \xrightarrow{\tau} s'$ instead of $s \xrightarrow{a}_t s'$, and $s \xrightarrow{\tau} s'$ instead of $s \xrightarrow{a}_t s'$.

An infinite run of a timed automaton is a sequence $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle$, such that for all $i \geq 1$, we have $s_{i-1} \xrightarrow{\tau_i} s_i$. A finite run of a timed automaton is a finite sequence $\langle s_0, \tau_1, s_1, \tau_2, \ldots, \tau_n, s_n \rangle \in S \times ((A \times \mathbb{R}_\oplus) \times S)^*$, such that for all $i$, $1 \leq i \leq n$, we have $s_{i-1} \xrightarrow{\tau_i} s_i$. For a finite run $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots, \tau_n, s_n \rangle$, we define $\mathsf{length}(r) = n$, and we define $\mathsf{last}(r) = s_n$ to be the state in which the run ends. For a finite run $r = \langle s_0, \tau_1, s_1, \tau_2, \ldots, s_n \rangle$, we define time of the run as $\mathsf{time}(r) = \sum_{i=1}^{n} t_i$. We write $\text{Runs}_{\text{fin}}$ for the set of finite runs.

## 3.2 Strategies

An average-time game $\Gamma$ is a triple $(\mathcal{T}, L_{\text{Min}}, L_{\text{Max}})$, where $\mathcal{T} = (L, C, S, A, E, \delta, \varrho)$ is a timed automaton and $(L_{\text{Min}}, L_{\text{Max}})$ is a partition of $L$. We define $Q_{\text{Min}} = \{(\ell, \nu) \in Q : \ell \in L_{\text{Min}}\}$, $Q_{\text{Max}} = Q \setminus Q_{\text{Min}}$, $S_{\text{Min}} = S \cap Q_{\text{Min}}$, $S_{\text{Max}} = S \setminus S_{\text{Min}}$, $\mathcal{R}_{\text{Min}} = \{[s] : s \in Q_{\text{Min}}\}$, and $\mathcal{R}_{\text{Max}} = \mathcal{R} \setminus \mathcal{R}_{\text{Min}}$.

A *strategy* for Min is a function $\mu : \text{Runs}_{\text{fin}} \to A \times \mathbb{R}_\oplus$, such that if $\mathsf{last}(r) = s \in S_{\text{Min}}$ and $\mu(r) = \tau$ then $s \xrightarrow{\tau} s'$, where $s' = \mathsf{succ}(s, \tau)$. Similarly, a strategy for player Max is a function $\chi : \text{Runs}_{\text{fin}} \to A \times \mathbb{R}_\oplus$, such that if $\mathsf{last}(r) = s \in S_{\text{Max}}$ and $\chi(r) = \tau$ then $s \xrightarrow{\tau} s'$, where $s' = \mathsf{succ}(s, \tau)$. We write $\Sigma_{\text{Min}}$ for the set of strategies for player Min, and we write $\Sigma_{\text{Max}}$ for the set of strategies for player Max. If players Min and Max use strategies $\mu$ and $\chi$, resp., then the $(\mu, \chi)$-run from a state $s$ is the unique run $\mathsf{run}(s, \mu, \chi) = \langle s_0, \tau_1, s_1, \tau_2, \ldots \rangle$, such that $s_0 = s$, and for every $i \geq 1$, if $s_i \in S_{\text{Min}}$, or $s_i \in S_{\text{Max}}$, then $\mu(\mathsf{run}_i(s, \mu, \chi)) = \tau_{i+1}$, or $\chi(\mathsf{run}_i(s, \mu, \chi)) = \tau_{i+1}$, resp., where $\mathsf{run}_i(s, \mu, \chi) = \langle s_0, \tau_1, s_1, \ldots, s_{i-1}, \tau_i, s_i \rangle$.

We say that a strategy $\mu$ for Min is *positional* if for all finite runs $r, r' \in \text{Runs}_{\text{fin}}$, we have that $\mathsf{last}(r) = \mathsf{last}(r')$ implies $\mu(r) = \mu(r')$. A positional strategy for player Min can be then represented as a function $\mu : S_{\text{Min}} \to A \times \mathbb{R}_\oplus$, which uniquely determines the strategy $\mu^\infty \in \Sigma_{\text{Min}}$ as follows: $\mu^\infty(r) = \mu(\mathsf{last}(r))$, for all finite runs $r \in \text{Runs}_{\text{fin}}$. Positional strategies for player Max are defined and represented in the analogous way. We write $\Pi_{\text{Min}}$ and $\Pi_{\text{Max}}$ for the sets of positional strategies for player Min and for player Max, respectively.

## 3.3 Value of Average-Time Game

If player Min uses the strategy $\mu \in \Sigma_{\text{Min}}$ and player Max uses the strategy $\chi \in \Sigma_{\text{Max}}$ then player Min loses the value $\mathcal{A}_{\text{Min}}(s, \mu, \chi) = \limsup_{n \to \infty} (1/n) \cdot \mathsf{time}(\mathsf{run}_n(s, \mu, \chi))$, and player Max wins the value $\mathcal{A}_{\text{Max}}(s, \mu, \chi) = \liminf_{n \to \infty} (1/n) \cdot \mathsf{time}(\mathsf{run}_n(s, \mu, \chi))$. In an average-time game player Min is interested in minimizing the value she loses and player Max is interested in maximizing the value he wins. For every state $s \in S$ of a timed automaton, we define its *upper value* by $\overline{\mathsf{val}}^{\mathcal{T}}(s) = \inf_{\mu \in \Sigma_{\text{Min}}} \sup_{\chi \in \Sigma_{\text{Max}}} \mathcal{A}_{\text{Min}}(s, \mu, \chi)$, and its lower value $\underline{\mathsf{val}}^{\mathcal{T}}(s) = \sup_{\chi \in \Sigma_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{A}_{\text{Max}}(s, \mu, \chi)$.

The inequality $\underline{\mathsf{val}}^{\mathcal{T}}(s) \leq \overline{\mathsf{val}}^{\mathcal{T}}(s)$ always holds. An average-time game is *determined* if for every state $s \in S$, its lower and upper values are equal to each other; then we say that the *value* $\mathsf{val}^{\mathcal{T}}(s)$ exists and $\mathsf{val}^{\mathcal{T}}(s) = \underline{\mathsf{val}}^{\mathcal{T}}(s) = \overline{\mathsf{val}}^{\mathcal{T}}(s)$. For strategies $\mu \in \Sigma_{\text{Min}}$ and

$\chi \in \Sigma_{\text{Max}}$, we define $\text{val}^{\mu}(s) = \sup_{\chi \in \Sigma_{\text{Min}}} \mathcal{A}_{\text{Min}}(s, \mu, \chi)$, and $\text{val}^{\chi}(s) = \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{A}_{\text{Max}}(s, \mu, \chi)$. For an $\varepsilon > 0$, we say that a strategy $\mu \in \Sigma_{\text{Min}}$ or $\chi \in \Sigma_{\text{Max}}$ is *$\varepsilon$-optimal* if for every $s \in S$ we have that $\text{val}^{\mu}(s) \leq \text{val}^{\mathcal{T}}(s) + \varepsilon$ or $\text{val}^{\chi}(s) \geq \text{val}^{\mathcal{T}}(s) - \varepsilon$, respectively. Note that if a game is determined then for every $\varepsilon > 0$, both players have $\varepsilon$-optimal strategies.

We say that a strategy $\chi \in \Sigma_{\text{Max}}$ of player Max is a best response to a strategy $\mu \in \Sigma_{\text{Min}}$ of player Min if for all $s \in S$ we have that $\mathcal{A}_{\text{Min}}(s, \mu, \chi) = \sup_{\chi' \in \Sigma_{\text{Max}}} \mathcal{A}_{\text{Min}}(s, \mu, \chi')$. Similarly we say that a strategy $\mu \in \Sigma_{\text{Min}}$ of player Min is a best response to a strategy $\chi \in \Sigma_{\text{Max}}$ of player Max if for all $s \in S$ we have that $\mathcal{A}_{\text{Max}}(s, \mu, \chi) = \inf_{\mu' \in \Sigma_{\text{Min}}} \mathcal{A}_{\text{Max}}(s, \mu', \chi)$.

# 4 Region Abstractions

## 4.1 Region Graphs

The region automaton, originally proposed by Alur and Dill [2], is a useful abstraction of a timed automaton as it preserves the validity of qualitative reachability, safety, and $\omega$-regular properties. The *region automaton* [2] $\text{RA}(\mathcal{T}) = (\mathcal{R}, \mathcal{M})$ of a timed automaton $\mathcal{T}$ consists of:

- the set $\mathcal{R}$ of regions of $\mathcal{T}$, and
- $\mathcal{M} \subseteq \mathcal{R} \times (\mathcal{R} \times A) \times \mathcal{R}$, such that for all $a \in A$, and for all $R, R', R'' \in \mathcal{R}$, we have that $(R, R'', a, R') \in \mathcal{M}$ iff $R \to_* R'' \xrightarrow{a} R'$.

The region automaton, however, is not sufficient for solving average-time games as it abstract away the timing information. Corner-point abstraction, introduced by Bouyer et al. [5], is a refinement of region automaton which preserves some timing information. Formally, the corner-point abstraction $\text{CP}(\mathcal{T})$ of a timed automaton $\mathcal{T}$ is a finite graph $(V, E)$ such that:

- $V \subseteq Q \times \mathcal{R}$ such that $(s, R) \in V$ iff $s = (\ell, \nu) \in \text{clos}(R)$ and $\nu$ is a corner. Since timed automata we consider are bounded, there are finitely many regions, and every region has a finite number of corners. Hence the set of vertices finite.
- $E \subseteq V \times (\mathbb{R}_{\oplus} \times \mathcal{R} \times A) \times V$ such that for $(s, R), (s', R') \in V$ and $(t, R'', a) \in \mathbb{R}_{\oplus} \times \mathcal{R} \times A$, we have $((s, R), (t, R'', a), (s', R')) \in E$ iff $R \to_* R'' \xrightarrow{a} R'$ and $(s + t) \xrightarrow{a} s'$. Notice that such a $t$ is always a natural number.

Bouyer et al. [5] showed that the corner-point abstraction is sufficient for deciding one-player average-price problem if the initial state is a corner-state, i.e., a state whose clock valuation is a corner. It follows from our results that the corner-point abstraction can be used to solve average-time games on timed automata if the initial state is a corner state.

We introduce the *boundary region graph*, which is a generalization of the corner-point abstraction. We prove that the value of the average-time game on a timed automaton is equal to the value of the average-time game on the corresponding boundary region graph, for all starting states, not just for corner states. In the process, we introduce two other refinements of the region automaton, which we call the *closed region graph* and the *region graph*. The analysis of average-time games on those objects allows us to establish equivalence of average-time games on the original timed automaton and the boundary region graph.

**Closed Region Graph.** A *closed region graph* $\overline{\mathcal{T}} = (\overline{Q}, \overline{E})$ of a timed automaton $\mathcal{T}$ is a refinement of its region automaton, where $\overline{Q} = \{(s, R) \ : \ s \in \text{clos}(R) \text{ and } R \in \mathcal{R}\}$ and $\overline{E} \subseteq \overline{Q} \times (\mathbb{R}_{\oplus} \times \mathcal{R} \times A) \times \overline{Q}$, such that for all $(s, R), (s', R') \in \overline{Q}$ and $(t, R'', a) \in \mathbb{R}_{\oplus} \times \mathcal{R} \times A$,

we have $((s,R),(t,R'',a),(s',R')) \in \overline{E}$ iff $s' = \mathsf{succ}(s,t,a)$, $(R,R'',a,R') \in \mathcal{M}$, and $s+t \in \mathsf{clos}(R'')$. For a region $R \in \mathcal{R}$ we define the set $\overline{Q}(R) \subseteq \overline{Q}$ to be $\{(s,R) : (s,R) \in \overline{Q}\}$.
**Boundary Region Graph.** For a timed automaton $\mathcal{T}$, its *boundary region graph* $\widehat{\mathcal{T}} = (\widehat{Q}, \widehat{E})$ is a sub-graph of its closed region graph $\overline{\mathcal{T}} = (\overline{Q}, \overline{E})$ with $\widehat{Q} = \overline{Q}$ and $\widehat{E} \subseteq \overline{E}$, such that for all $(s,R),(s',R') \in \widehat{Q}$ and $(t,R'',a) \in \mathbb{R}_{\oplus} \times \mathcal{R} \times A$, we have $((s,R),(t,R'',a),(s',R')) \in \widehat{E}$ if: either $R \in \mathcal{R}_{\mathrm{Min}}$ and $t = \inf\{t : s+t \in \mathsf{clos}(R'')\}$, or $R \in \mathcal{R}_{\mathrm{Max}}$ and $t = \sup\{t : s+t \in \mathsf{clos}(R'')\}$. Boundary region graphs have the following property.

**PROPOSITION 2.** *For every configuration in a boundary region graph the set of reachable configurations is finite.*

We say that a configuration $q = (s = (\ell, \nu), R)$ is *corner configuration* if $\nu$ is a corner.

**PROPOSITION 3.** *The reachable sub-graph of the a boundary region graph $\widehat{\mathcal{T}}$ from a corner configuration is same as the corner-point abstraction $\mathsf{CP}(\mathcal{T})$.*

**Region Graph.** The *region graph* $\widetilde{\mathcal{T}} = (\widetilde{Q}, \widetilde{E})$ of a timed automaton $\mathcal{T}$ is a sub-graph of its closed region graph $\overline{\mathcal{T}} = (\overline{Q}, \overline{E})$ with $\widetilde{Q} = \overline{Q}$ and $\widetilde{E} \subseteq \overline{E}$, such that $((s,R),(t,R'',a),(s',R')) \in \widetilde{E}$ if $s+t \in R''$. The timed automaton $\mathcal{T}$ and the corresponding region graph $\widetilde{\mathcal{T}}$ are equivalent in the following sense.

**PROPOSITION 4.** *Let $\mathcal{T}$ be a timed automaton and $\widetilde{\mathcal{T}} = (\widetilde{Q}, \widetilde{E})$ be its region graph. For every $s, s' \in S$ and $(t,a) \in \mathbb{R}_{\oplus} \times A$, we have $s \xrightarrow{a}_t s'$ if and only if $((s,[s]),(t,[s+t],a),(s',[s'])) \in \widetilde{E}$.*

**Runs of Region Graphs.** An infinite run of the closed region graph $\overline{\mathcal{T}}$ is an infinite sequence $\langle q_0, \tau_1, q_1, \tau_1, \ldots \rangle$, such that for all $i \geq 1$, we have $(q_{i-1}, \tau_i, q_i) \in \overline{E}$. A finite run of the closed region graph $\overline{\mathcal{T}}$ is a finite sequence $\langle q_0, \tau_1, q_1, \tau_1, \ldots, q_n \rangle \in \overline{Q} \times ((\mathbb{R}_{\oplus} \times \mathcal{R} \times A) \times \overline{Q})^*$, such that for all $1 \leq i \leq n$, we have $(q_{i-1}, \tau_i, q_i) \in \overline{E}$. Runs of the boundary region graph and the region graph are defined analogously. For a graph $\mathcal{G} \in \{\overline{\mathcal{T}}, \widehat{\mathcal{T}}, \widetilde{\mathcal{T}}\}$, we write $\mathrm{Runs}^{\mathcal{G}}_{\mathrm{fin}}$ for the set of its finite runs and $\mathrm{Runs}^{\mathcal{G}}_{\mathrm{fin}}(q)$ for the set of its finite runs from a configuration $q \in \overline{Q}$. Notice that for all $q \in \overline{Q}$ we have that $\mathrm{Runs}^{\widehat{\mathcal{T}}}(q) \subseteq \mathrm{Runs}^{\overline{\mathcal{T}}}(q)$ and $\mathrm{Runs}^{\widetilde{\mathcal{T}}}(q) \subseteq \mathrm{Runs}^{\overline{\mathcal{T}}}(q)$. For a finite run $r = \langle q_0, (t_1, R_1, a_1), q_1, (t_2, R_2, a_2), \ldots, q_n \rangle$ we define $\mathsf{time}(r) = \sum_{i=1}^{n} t_i$, and we denote the last configuration of the run by $\mathsf{last}(r) = q_n$.
**Run Types of Region Graphs.** Type of a finite run $\langle (s_0, R_0), (t_1, R'_1, a_1), (s_1, R_1), \ldots, (s_n, R_n) \rangle$ is the finite sequence $\langle R_0, (R'_1, a_1), R_1, (R'_2, a_2), \ldots, R_n \rangle$. The type of an infinite run is defined analogously. For a (finite or infinite) run $r$, we write $[\![r]\!]_{\mathcal{R}}$ for its type. We write $\mathrm{Types}_{\mathrm{fin}}$ and Types for the set of types of finite runs and the set of types of infinite runs, respectively.

## 4.2 Simple Functions and Boundary Timed Actions

A function $F : \overline{Q} \to \mathbb{R}$ is *simple* [3, 11] if either: there is $e \in \mathbb{Z}$, such that for every $(s,R) \in \overline{Q}$, we have $F(s,R) = e$; or there are $e \in \mathbb{Z}$ and $c \in C$, such that for every $(s,R) \in \overline{Q}$ we have $F(s,R) = e - s(c)$. We say that a function $F : \overline{Q} \to \mathbb{R}$ is *regionally simple* or *regionally constant*, respectively, if for every region $R \in \mathcal{R}$ the function $F$, over domain $\overline{Q}(R)$, is simple or constant, respectively.

Define the finite set of *boundary timed actions* $\mathbb{A} = (\!|k|\!)_{\mathbb{N}} \times C \times A \times \mathcal{R}$. For $q = (s,R) \in \overline{Q}$ and $\alpha = (b,c,a,R'') \in \mathbb{A}$, we define $t(s,\alpha) = b - s(c)$. If $s + t(s,\alpha) \in \mathsf{clos}(R'')$ then the function $\mathsf{succ}(q, \alpha)$ is defined and we have $q' = (\mathsf{succ}(s, \tau(\alpha)), R')$, where $\tau(\alpha) = (t(s, \alpha), a)$ and $R'' \xrightarrow{a} R'$. We sometimes write $q \xrightarrow{\alpha} q'$ if $q' = \mathsf{succ}(q, \alpha)$.

### 4.3 Strategies

Let $\Gamma = (\mathcal{T}, L_{\mathrm{Min}}, L_{\mathrm{Max}})$ be an average-time game. The partition $(L_{\mathrm{Min}}, L_{\mathrm{Max}})$ naturally gives rise to average-time games on the closed region graph $\overline{\Gamma} = (\overline{\mathcal{T}}, \overline{Q}_{\mathrm{Min}}, \overline{Q}_{\mathrm{Max}})$, the boundary region graph $\widehat{\Gamma} = (\widehat{\mathcal{T}}, \widehat{Q}_{\mathrm{Min}}, \widehat{Q}_{\mathrm{Max}})$, and the region graph $\widetilde{\Gamma} = (\widetilde{\mathcal{T}}, \widetilde{Q}_{\mathrm{Min}}, \widetilde{Q}_{\mathrm{Max}})$.

In a closed region graph, a strategy of player Min $\mu$ is a (partial) function $\mu : \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}} \to \mathbb{R}_{\oplus} \times \mathcal{R} \times A$, such that for a run $r \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$, if $\mathsf{last}(r) = (s, R) \in Q_{\mathrm{Min}}$ then $\mu(r) = (t, R', a)$ is defined, and it is such that $(s + t) \in \mathsf{clos}(R')$ and $(R, (R', a), R'') \in \mathcal{M}$, for some $R'' \in \mathcal{R}$. Strategies of player Max is defined analogously. We write $\overline{\Sigma}_{\mathrm{Min}}$ and $\overline{\Sigma}_{\mathrm{Max}}$ for the set of strategies of player Min and player Max, respectively. We say that a strategy $\sigma$ is positional if for all runs $r_1, r_2 \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$, $\mathsf{last}(r_1) = \mathsf{last}(r_2)$ implies $\mu(r_1) = \mu(r_2)$. We define the run starting from configuration $q \in \overline{Q}$ and following strategies $\mu$ and $\chi$, of player Max and player Min, respectively, in a straightfoward manner and we write $\mathsf{run}(q, \mu, \chi)$ to denote this run. For every $n \geq 1$, we write $\mathsf{run}_n(q, \mu, \chi)$ for the prefix of the run $\mathsf{run}(q, \mu, \chi)$ of length $n$.

We say that a strategy $\sigma$ is an *admissible strategy* if for all finite runs $r \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$, we have $\sigma(r) = (t, R', a)$ such that $s + t \in R'$, where $(s, R) = \mathsf{last}(r)$. Note that both players have only admissible strategies on the region graph. We write $\widetilde{\Sigma}_{\mathrm{Min}}$ and $\widetilde{\Sigma}_{\mathrm{Max}}$ for the set of admissible strategies of player Min and player Max, respectively.

We say that a strategy $\mu$ of player Min is a *boundary strategy* if for all finite runs $r \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$, we have $\mu(r) = (t, R', a)$, such that $t = \inf\{t \, : \, s + t \in \mathsf{clos}(R')\}$, where $(s, R) = \mathsf{last}(r)$. We say that a strategy $\chi$ of player Max is a *boundary strategy* if for all finite runs $r \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$, we have $\chi(r) = (t, R', a)$, such that $t = \sup\{t \, : \, s + t \in \mathsf{clos}(R')\}$, where $(s, R) = \mathsf{last}(r)$. Both players have only boundary strategies in the boundary region graph. We write $\widehat{\Sigma}_{\mathrm{Min}}$ and $\widehat{\Sigma}_{\mathrm{Max}}$ for the set of boundary strategies of player Min and player Max, respectively.

**PROPOSITION 5.** *For every boundary strategy $\sigma$ and for every run $r$, if $\sigma(r) = (t, R', a)$ then there exists a boundary timed action $\alpha = (b, c, a, R') \in \mathbb{A}$ such that $t(s, \alpha) = t$, where $(s, R) = \mathsf{last}(r)$.*

By Proposition 5 a run of the closed region graph in which both players use boundary strategies, can be represented as a sequence $\langle q_0, \alpha_1, q_1, \alpha_2, \ldots \rangle$. Such a run is called a *boundary run*. For a boundary strategy $\sigma$, we define the function $\widehat{\sigma} : \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}} \to \mathbb{A}$ as follows: if for a run $r$ we have $\sigma(r) = (t, R', a)$, then $\widehat{\sigma}(r) = (b, c, a, R')$, such that $b - s(c) = t$, where $(s, R) = \mathsf{last}(r)$.

**Type-Preserving Boundary Strategies.** We say that a boundary strategy $\sigma$ is *type-preserving*, if for all finite runs $r_1, r_2 \in \mathrm{Runs}_{\mathrm{fin}}^{\overline{\mathcal{T}}}$ such that $[\![r_1]\!]_{\mathcal{R}} = [\![r_2]\!]_{\mathcal{R}}$, we have that $\widehat{\sigma}(r_1) = \widehat{\sigma}(r_2)$. We write $\Xi_{\mathrm{Min}}$ and $\Xi_{\mathrm{Max}}$ for the sets of type-preserving boundary strategies of players Min and Max, respectively. Notice that for type-preserving boundary strategies $\mu \in \Xi_{\mathrm{Min}}$ and $\chi \in \Xi_{\mathrm{Max}}$, for every region $R \in \mathcal{R}$ and for all configurations $q, q' \in \overline{Q}(R)$, we have that $[\![\mathsf{run}(q, \mu, \chi)]\!]_{\mathcal{R}} = [\![\mathsf{run}(q', \mu, \chi)]\!]_{\mathcal{R}}$.

Note that the following inclusions hold.

$$\Xi_{\text{Min}} \subseteq \widehat{\Sigma}_{\text{Min}} \subseteq \overline{\Sigma}_{\text{Min}} \quad \text{and} \quad \widetilde{\Sigma}_{\text{Min}} \subseteq \overline{\Sigma}_{\text{Min}}, \quad \text{and}$$
$$\Xi_{\text{Max}} \subseteq \widehat{\Sigma}_{\text{Max}} \subseteq \overline{\Sigma}_{\text{Max}} \quad \text{and} \quad \widetilde{\Sigma}_{\text{Max}} \subseteq \overline{\Sigma}_{\text{Max}}$$

**PROPOSITION 6.** *For every $n \geq 1$, and for all type-preserving boundary strategies $\mu \in \Xi_{Min}$ and $\chi \in \Xi_{Max}$, the function $\mathsf{time}(\mathsf{run}_n(\cdot, \mu, \chi))$ is regionally simple.*

Given a type-preserving boundary strategy $\sigma$ and $\varepsilon > 0$, we define an admissible strategy $\sigma_\varepsilon$ as follows: for a finite run $r \in \text{Runs}^{\mathcal{T}}_{\text{fin}}$, if $\widehat{\sigma}(r) = (b, c, a, R')$ then $\sigma_\varepsilon(r) = (t, R', a)$ such that $b - s(c) - \varepsilon \leq t \leq b - s(c) + \varepsilon$, where $(s, R) = \mathsf{last}(r)$.

Given a boundary strategy $\sigma$ and a configuration $q \in \overline{Q}$, we define the type-preserving boundary strategy $\sigma[q]$, which agrees with the strategy $\sigma$ on all the runs starting from the configuration $q$. Formally, for a given $\sigma$ the type-preserving boundary strategy $\sigma[q]$ is such that for all runs $r \in \text{Runs}_{\text{fin}}(q)$, we have $\widehat{\sigma[q]}(r) = \widehat{\sigma}(r)$.

### 4.4 Value of Average-Time Game

For the strategies $\mu \in \overline{\Sigma}_{\text{Min}}$ and $\chi \in \overline{\Sigma}_{\text{Max}}$ of respective players and a configuration $q \in \overline{Q}$ we define $\mathcal{A}_{\text{Min}}(q, \mu, \chi) = \limsup_{n\to\infty}(1/n) \cdot \mathsf{time}(\mathsf{run}_n(q, \mu, \chi))$ and $\mathcal{A}_{\text{Max}}(q, \mu, \chi) = \liminf_{n\to\infty}(1/n) \cdot \mathsf{time}(\mathsf{run}_n(q, \mu, \chi))$. For average-time games on a graph $\mathcal{G} \in \{\overline{\mathcal{T}}, \widehat{\mathcal{T}}, \widetilde{\mathcal{T}}\}$ we define the lower-value $\underline{\mathsf{val}}^{\mathcal{G}}(q)$, the upper-value $\overline{\mathsf{val}}^{\mathcal{G}}(q)$ and the value $\mathsf{val}^{\mathcal{G}}(q)$ of a configuration $q \in \overline{Q}$ in a straightfoward manner.

### 4.5 Determinacy of Average-Time Games on the Boundary Region Graph

Positional determinacy of average-time games on the boundary region graph is immediate from Proposition 2 and Theorem 1.

**THEOREM 7.** *The average-time game on $\widehat{\mathcal{T}}$ is determined, and there are optimal positional strategies in $\widehat{\mathcal{T}}$, i.e., for every $q \in \overline{Q}$, we have:*

$$\mathsf{val}^{\widehat{\mathcal{T}}}(q) = \inf_{\mu \in \widehat{\Pi}_{Min}} \sup_{\chi \in \widehat{\Sigma}_{Max}} \mathcal{A}_{Min}(q, \mu, \chi) = \sup_{\chi \in \widehat{\Pi}_{Max}} \inf_{\mu \in \widehat{\Sigma}_{Min}} \mathcal{A}_{Max}(q, \mu, \chi).$$

**PROPOSITION 8.** *For all $\mu \in \Xi_{Min}$ and $\chi \in \Xi_{Max}$, the functions $\mathcal{A}_{Min}(\cdot, \mu, \chi)$ and $\mathcal{A}_{Max}(\cdot, \mu, \chi)$ are regionally constant.*

**LEMMA 9.** *In $\widehat{\mathcal{T}}$, if $\mu \in \widehat{\Sigma}_{Min}$ and $\chi \in \widehat{\Sigma}_{Max}$ are mutual best responses from $q \in \overline{Q}$, then $\mu[q] \in \Xi_{Min}$ and $\chi[q] \in \Xi_{Max}$ are mutual best responses from every $q' \in \overline{Q}([q])$.*

PROOF. We argue that $\chi[q]$ is a best response to $\mu[q]$ from $q' \in \overline{Q}([q])$ in $\widehat{\mathcal{T}}$; the other case is analogous. For all $\chi' \in \widehat{\Sigma}_{\text{Max}}$, we have the following:

$$\mathcal{A}_{\text{Min}}(q', \mu[q], \chi[q]) = \mathcal{A}_{\text{Min}}(q, \mu[q], \chi[q]) \geq \mathcal{A}_{\text{Min}}(q, \mu[q], \chi'[q']) =$$
$$\mathcal{A}_{\text{Min}}(q', \mu[q], \chi'[q']) = \mathcal{A}_{\text{Min}}(q', \mu[q], \chi').$$

The first equality follows from Proposition 8; the inequality follows because $\chi$ is a best response to $\mu$ from $q$; the second equality follows from Proposition 8 again; and the last equality is straightforward. ∎

**THEOREM 10.** *There are optimal type-preserving boundary strategies in $\widehat{\mathcal{T}}$, i.e., for every $q \in \overline{Q}$, we have:*

$$\mathsf{val}^{\widehat{\mathcal{T}}}(q) = \inf_{\mu \in \Xi_{Min}} \sup_{\chi \in \widehat{\Sigma}_{Max}} \mathcal{A}_{Min}(q, \mu, \chi) = \sup_{\chi \in \Xi_{Max}} \inf_{\mu \in \widehat{\Sigma}_{Min}} \mathcal{A}_{Max}(q, \mu, \chi).$$

PROOF. Let $\mu^* \in \Xi_{\text{Min}}$ and $\chi^* \in \Xi_{\text{Max}}$ be mutual best responses in $\widehat{\mathcal{T}}$; existence of such strategies follows from Lemma 9. Moreover, we can assume that the strategies $\mu^*$ and $\chi^*$ have finite memory; this can be achieved by taking positional strategies $\mu \in \widehat{\Sigma}_{\text{Min}}$ and $\chi \in \widehat{\Sigma}_{\text{Max}}$ in Lemma 9. We then have the following:

$$\inf_{\mu \in \Xi_{\text{Min}}} \sup_{\chi \in \widehat{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu, \chi) \leq \sup_{\chi \in \widehat{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu^*, \chi) = \mathcal{A}_{\text{Min}}(q, \mu^*, \chi^*) =$$
$$\mathcal{A}_{\text{Max}}(q, \mu^*, \chi^*) = \inf_{\mu \in \widehat{\Sigma}_{\text{Min}}} \mathcal{A}_{\text{Max}}(q, \mu, \chi^*) \leq \sup_{\chi \in \Xi_{\text{Max}}} \inf_{\mu \in \widehat{\Sigma}_{\text{Min}}} \mathcal{A}_{\text{Max}}(q, \mu, \chi).$$

The first and last inequalities are straightforward because $\mu^* \in \Xi_{\text{Min}}$ and $\chi^* \in \Xi_{\text{Max}}$. The first equality holds because $\chi^*$ is a best response to $\mu^*$ in $\widehat{\mathcal{T}}$, and the third equality holds because $\mu^*$ is a best response to $\chi^*$ in $\widehat{\mathcal{T}}$. Finally, the second equality holds because strategies $\mu^*$ and $\chi^*$ have finite memory. ∎

## 4.6 Determinacy of Average-Time Games on the Closed Region Graph

**LEMMA 11.** *In $\overline{\mathcal{T}}$, for every strategy in $\Xi_{Min}$ there is a best response in $\Xi_{Max}$, and for every strategy in $\Xi_{Max}$ there is a best response in $\Xi_{Min}$.*

**THEOREM 12.** *The average-time game on $\overline{\mathcal{T}}$ is determined, and there are optimal type-preserving boundary strategies in $\overline{\mathcal{T}}$, i.e., for every $q \in \overline{Q}$, we have:*

$$\mathsf{val}^{\overline{\mathcal{T}}}(q) = \inf_{\mu \in \Xi_{Min}} \sup_{\chi \in \overline{\Sigma}_{Max}} \mathcal{A}_{Min}(q, \mu, \chi) = \sup_{\chi \in \Xi_{Max}} \inf_{\mu \in \overline{\Sigma}_{Min}} \mathcal{A}_{Max}(q, \mu, \chi) = \mathsf{val}^{\widehat{\mathcal{T}}}(q).$$

PROOF. We have the following:

$$\inf_{\mu \in \Xi_{\text{Min}}} \sup_{\chi \in \overline{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu, \chi) = \inf_{\mu \in \Xi_{\text{Min}}} \sup_{\chi \in \Xi_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu, \chi) =$$
$$\sup_{\chi \in \Xi_{\text{Max}}} \inf_{\mu \in \Xi_{\text{Min}}} \mathcal{A}_{\text{Max}}(q, \mu, \chi) = \sup_{\chi \in \Xi_{\text{Max}}} \inf_{\mu \in \overline{\Sigma}_{\text{Min}}} \mathcal{A}_{\text{Max}}(q, \mu, \chi),$$

where the first and last equalities follow from Lemma 11, and the second equality follows from Theorem 10. Now it is routine to show that $\underline{\mathsf{val}}^{\overline{\mathcal{T}}}(q) \geq \underline{\mathsf{val}}^{\widehat{\mathcal{T}}}(q)$ and $\overline{\mathsf{val}}^{\overline{\mathcal{T}}}(q) \leq \overline{\mathsf{val}}^{\widehat{\mathcal{T}}}(q)$. It concludes the proof that the average-time game on $\overline{\mathcal{T}}$ is determined, and there are optimal type-preserving boundary strategies in $\overline{\mathcal{T}}$. ∎

### 4.7 Determinacy of Average-Time Games on the Region Graph

**LEMMA 13.** *If the strategies $\mu^* \in \Xi_{Min}$ and $\chi^* \in \Xi_{Max}$ are optimal for respective players in $\overline{\mathcal{T}}$ then for every $\varepsilon > 0$, we have that*

$$\sup_{\chi \in \overline{\Sigma}_{Max}} \mathcal{A}_{Min}(q, \mu^*_\varepsilon, \chi) \leq \mathsf{val}^{\overline{\mathcal{T}}}(q) + \varepsilon \ \text{ and } \inf_{\mu \in \overline{\Sigma}_{Min}} \mathcal{A}_{Max}(q, \mu, \chi^*_\varepsilon) \geq \mathsf{val}^{\overline{\mathcal{T}}}(q) - \varepsilon.$$

**THEOREM 14.** *The average-time game on $\widetilde{\mathcal{T}}$ is determined, and for every $q \in \overline{Q}$, we have $\mathsf{val}^{\widetilde{\mathcal{T}}}(q) = \mathsf{val}^{\overline{\mathcal{T}}}(q)$.*

PROOF. Let $\mu^* \in \Xi_{\text{Min}}$ be an optimal strategy of player Min in $\overline{\mathcal{T}}$. Let us fix an $\varepsilon > 0$.

$$\overline{\mathsf{val}}^{\widetilde{\mathcal{T}}}(q) = \inf_{\mu \in \widetilde{\Sigma}_{\text{Min}}} \sup_{\chi \in \widetilde{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu, \chi) \leq \sup_{\chi \in \widetilde{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu^*_\varepsilon, \chi) \leq \sup_{\chi \in \overline{\Sigma}_{\text{Max}}} \mathcal{A}_{\text{Min}}(q, \mu^*_\varepsilon, \chi) \leq \mathsf{val}^{\overline{\mathcal{T}}}(q) + \varepsilon.$$

The second inequality follows because $\mu^*_\varepsilon \in \widetilde{\Sigma}_{\text{Min}}$ and the third inequality follows because $\widetilde{\Sigma}_{\text{Max}} \subseteq \overline{\Sigma}_{\text{Max}}$. The last inequality follows from Lemma 13 because $\mu^* \in \Xi_{\text{Min}}$ is an optimal strategy in $\overline{\mathcal{T}}$. Similarly we show that for every $\varepsilon > 0$ we have that $\underline{\mathsf{val}}^{\widetilde{\mathcal{T}}}(q) \geq \mathsf{val}^{\overline{\mathcal{T}}}(q) - \varepsilon$. Hence it follows that $\mathsf{val}^{\widetilde{\mathcal{T}}}(q)$ exists and its value is equal to $\mathsf{val}^{\overline{\mathcal{T}}}(q)$. ∎

### 4.8 Determinacy of Average-Time Games on Timed Automata

**THEOREM 15.** *The average-time game on $\mathcal{T}$ is determined, and for every $s \in S$, we have:*

$$\mathsf{val}^{\mathcal{T}}(s) = \mathsf{val}^{\widetilde{\mathcal{T}}}(s, [s]) = \mathsf{val}^{\overline{\mathcal{T}}}(s, [s]) = \mathsf{val}^{\widehat{\mathcal{T}}}(s, [s]).$$

## 5 Complexity

The main decision problem for average-time game is as follows: given an average-time game $\Gamma = (\mathcal{T}, L_{\text{Min}}, L_{\text{Max}})$, a state $s \in S$, and a number $B \in \mathbb{R}_\oplus$, decide whether $\mathsf{val}(s) \leq B$.

From Theorem 15 we know that in order to solve an average-time game starting from an initial state of a timed automaton, it is sufficient to solve the average-time game on the set of states of the boundary region graph of the automaton that are reachable from the initial state. Observe that every region, and hence also every configuration of the game, can be represented in space polynomial in the size of the encoding of the timed automaton and of the encoding of the initial state, and that every move of the game can be simulated in polynomial time. Therefore, the value of the game can be computed by a straightforward alternating PSPACE algorithm, and hence the problem is in EXPTIME because APSPACE = EXPTIME.

One can prove EXPTIME-hardness of average-time games on timed automata with at least two clocks by a reduction from countdown games [10], similar to the reduction from countdown games to reachability-time games on timed automata [11].

**THEOREM 16.** *Average-time games are EXPTIME-complete on timed automata with at least two clocks.*

## References

[1] R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *International Colloquium on Automata, Languages and Programming, ICALP 2004*, volume 3142 of *LNCS*, pages 122–133. Springer, 2004.

[2] R. Alur and D. Dill. A theory of timed automata. In *Theoretical Computer Science*, volume 126, pages 183–235, 1994.

[3] E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In F. W. Vaandrager and J. H. van Schuppen, editors, *HSCC 1999*, volume 1569 of *LNCS*, pages 19–30. Springer, 1999.

[4] P. Bouyer, T. Brihaye, V. Bruyère, and J. Raskin. On the optimal reachability problem on weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.

[5] P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *Hybrid Systems: Computation and Control, HSCC 2004*, volume 2993 of *LNCS*, pages 203–218. Springer, 2004.

[6] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal strategies in priced timed game automata. In *FSTTCS'04*, volume 3328 of *LNCS*, pages 148–160. Springer, 2004.

[7] T. Brihaye, T. A. Henzinger, V. S. Prabhu, and J. Raskin. Minimum-time reachability in timed games. In *ICALP 2007*, volume 4596 of *LNCS*, pages 825–837. Springer, 2007.

[8] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[9] D. Gillette. Stochastic games with zero stop probabilities. In *Contributions to the Theory of Games*, volume 39 of *Annals of Mathematics Studies*, pages 179–187. Princeton University Press, 1957.

[10] M. Jurdziński, J. Sproston, and F. Laroussinie. Model checking probabilistic timed automata with one or two clocks. *Logical Methods in Computer Science*, 4(3):12, 2008.

[11] M. Jurdziński and A. Trivedi. Reachability-time games on timed automata. In *ICALP 2007*, volume 4596 of *LNCS*, pages 838–849. Springer, 2007.

[12] M. Jurdziński and A. Trivedi. Concavely-priced timed automata. In *FORMATS 2008*, volume 5215 of *LNCS*, pages 48–62. Springer, 2008.

[13] T. Liggett and S. Lipman. Stochastic games with perfect information and time average payoff. *SIAM Review*, 11:604–607, 1969.

[14] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[15] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.