Timed Alternating-Time Temporal Logic*

Thomas A. Henzinger[1] and Vinayak S. Prabhu[2]

[1]Department of Computer and Communication Sciences, EPFL
tah@epfl.ch
[2]Department of Electrical Engineering and Computer Sciences, UC Berkeley
vinayak@eecs.berkeley.edu

Abstract. We add freeze quantifiers to the game logic ATL in order to specify real-time objectives for games played on timed structures. We define the semantics of the resulting logic TATL by restricting the players to physically meaningful strategies, which do not prevent time from diverging. We show that TATL can be model checked over timed automaton games. We also specify timed optimization problems for physically meaningful strategies, and we show that for timed automaton games, the optimal answers can be approximated to within any degree of precision.

1 Introduction

Timed games are a formal model for the synthesis of real-time systems [22, 20]. While much research effort has been directed at algorithms for solving timed games [14, 9, 7, 16, 15, 8, 11], we find it useful to revisit the topic for two reasons. First, we wish to study a perfectly symmetric setup of the model, where all players (whether they represent a plant, a controller, a scheduler, etc.) are given equally powerful options for updating the state of the game, advancing time, or blocking time. Second, we wish to restrict all players to physically meaningful strategies, which do not allow a player to prevent time from diverging in order to achieve an objective. This restriction is often ensured by syntactic conditions on the cycles of timed automaton games [7, 16, 8, 21] or by semantic conditions that discretize time. We find such conditions unsatisfactory and unnecessary: unsatisfactory, because they rule out perfectly meaningful strategies that suggest an arbitrary but finite number of transitions in a finite interval of time; unnecessary, because timed automaton games can be solved without such conditions.

We do not present a new model for timed games, but review the model of [13], which is symmetric for all players and handles the divergence of dense time without constraining the players. We consider the two-player case. Previous work on the existence of controllers [14, 9, 20, 11] has in general required that time divergence be ensured by the controller —an unfair view in settings where the plant player can also block time. In our model, both players may block time, however, for a player to win for an objective, she must not be *responsible* for preventing

* This research was supported in part by the NSF grants CCR-0208875, CCR-0225610, and CCR-0234690.

time from diverging. To achieve this, we distinguish between *objectives* and *winning conditions*. An objective for a player is a set Φ of desired outcomes of the game. The winning condition WC maps the objective to another set of outcomes so that the player wins for $\mathsf{WC}(\Phi)$ using any strategy if and only if she wins for the original objective Φ using a physically meaningful strategy.

Let us be more precise. A timed game proceeds in an infinite sequence of turns. At each turn, both players propose a move: each move specifies an amount of time that the player is willing to let pass without action, possibly followed by an action that causes a discontinuous jump to a different state. The move with the *shorter* proposed time delay determines the next state of the game (if both players propose the same delays, then one of the corresponding actions is chosen nondeterministically). An outcome of the game is an infinite trajectory of continuous state segments (during which time passes) and discontinuous jumps. Let $\mathsf{Timediv}$ denote the outcomes for which time diverges (the other trajectories are often called "zeno" behaviors). Let $\mathsf{Blameless}_i$ denote the outcomes in which player $i \in \{1, 2\}$ proposes the shorter delay only finitely often. Clearly, player i is not responsible if time converges for an outcome in $\mathsf{Blameless}_i$. We therefore use the winning condition [13]

$$\mathsf{WC}_i(\Phi) \;=\; (\mathsf{Timediv} \cap \Phi) \;\cup\; (\mathsf{Blameless}_i \setminus \mathsf{Timediv}).$$

Informally, this condition states that if an outcome is time divergent, then it is a valid outcome, and hence must satisfy the objective Φ; and if it is not time divergent, then player i must not be responsible for the zeno behaviour. The winning conditions for both players are perfectly symmetric: since $\mathsf{WC}_1(\Phi) \cap \mathsf{WC}_2(\neg\Phi) = \emptyset$, at most one player can win.

In [6], several alternating-time temporal logics were introduced to specify properties of game structures, including the CTL-like logic ATL, and the CTL*-like logic ATL*. For example, the ATL formula $\langle\langle i \rangle\rangle \Diamond p$ is true at a state s iff player i can force the game from s into a state that satisfies the proposition p. We interpret these logics over *timed* game structures, and enrich them by adding *freeze* quantifiers [5] for specifying timing constraints. The resulting logics are called TATL and TATL*. The new logic TATL subsumes both the untimed game logic ATL, and the timed non-game logic TCTL [3]. For example, the TATL formula $\langle\langle i \rangle\rangle \Diamond_{\leq d}\, p$ is true at a state s iff player i can force the game from s into a p state in at most d time units. A version of TATL has recently been studied on durational concurrent structures [19].

The model checking of these logics requires the solution of timed games. Timed game structures are infinite-state. In order to consider algorithmic solutions, we restrict our attention to timed game structures that are generated by a finite syntax borrowed from timed automata [4]. By restricting the strategies of TATL games to physically meaningful strategies using WC, we obtain TATL* games. However, solving TATL* games is undecidable, because TATL* subsumes the linear-time logic TPTL [5], whose dense-time satisfiability problem is undecidable. We nontheless establish the decidability of TATL model checking, by carefully analyzing the fragment of TATL* we obtain through the WC

translation. We show that TATL model checking over timed automaton games is complete for EXPTIME; that is, no harder than the solution of timed automaton games with reachability objectives.

In timed games, as in optimal control, it is natural to study not only the decision problem if a player can force the game into a target state within d time units, but also the corresponding optimization problem: determine the *minimal* time d so that a player can force the game into a target state. Again we insist on the use of physically meaningful strategies. We show that for timed automaton games, the optimal answer can be computed to within any desired degree of precision. The exact optimization problem is still open; only special cases have been solved, such as the special case where every cycle of the timed automaton ensures syntactically that a positive amount of time passes [7], and the special case where the game is restricted to a finite number of moves [2]. The general case for *weighted* timed automaton games is known to be undecidable [10]. Average reward games in the framework of [13] are considered in [1], but with the durations of time moves restricted to either 0 or 1.

2 Timed Games

2.1 Timed Game Structures

We borrow our formalism from [13]. A *timed game structure* is a tuple $\mathcal{G} = \langle S, \Sigma, \sigma, A_1, A_2, \Gamma_1, \Gamma_2, \delta \rangle$ with the following components:

- S is a set of states.
- Σ is a finite set of propositions.
- $\sigma : S \mapsto 2^{\Sigma}$ is the observation map, which assigns to every state the set of propositions that are true in that state.
- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively. We assume that $\perp \notin A_i$, and write $A_i^{\perp}$ for $A_i \cup \{\perp\}$. The set of *moves* for player i is $M_i = \mathbb{R}_{\geq 0} \times A_i^{\perp}$. Intuitively, a move $\langle \Delta, a_i \rangle$ by player i indicates a waiting period of Δ time units followed by a discrete transition labeled with action a_i.
- $\Gamma_i : S \mapsto 2^{M_i} \setminus \emptyset$ are two move assignments. At every state s, the set $\Gamma_i(s)$ contains the moves that are available to player i. We require that $\langle 0, \perp \rangle \in \Gamma_i(s)$ for all states $s \in S$ and $i \in \{1, 2\}$. Intuitively, $\langle 0, \perp \rangle$ is a time-blocking stutter move.
- $\delta : S \times (M_1 \cup M_2) \mapsto S$ is the transition function. We require that for all time delays $\Delta, \Delta' \in \mathbb{R}_{\geq 0}$ with $\Delta' \leq \Delta$, and all actions $a_i \in A_i^{\perp}$, we have (1) $\langle \Delta, a_i \rangle \in \Gamma_i(s)$ iff both $\langle \Delta', \perp \rangle \in \Gamma_i(s)$ and $\langle \Delta - \Delta', a_i \rangle \in \Gamma_i(\delta(s, \langle \Delta', \perp \rangle))$; and (2) if $\delta(s, \langle \Delta', \perp \rangle) = s'$ and $\delta(s', \langle \Delta - \Delta', a_i \rangle) = s''$, then $\delta(s, \langle \Delta, a_i \rangle) = s''$.

The game proceeds as follows. If the current state of the game is s, then both players simultaneously propose moves $\langle \Delta_1, a_1 \rangle \in \Gamma_1(s)$ and $\langle \Delta_2, a_2 \rangle \in \Gamma_2(s)$. The move with the shorter duration "wins" in determining the next state of the game. If both moves have the same duration, then one of the two moves is chosen nondeterministically. Formally, we define the *joint destination function* $\delta_{\mathsf{jd}} : S \times M_1 \times M_2 \mapsto 2^S$ by

$$\delta_{\mathsf{jd}}(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle) = \begin{cases} \{\delta(s, \langle \Delta_1, a_1 \rangle)\} & \text{if } \Delta_1 < \Delta_2; \\ \{\delta(s, \langle \Delta_2, a_2 \rangle)\} & \text{if } \Delta_2 < \Delta_1; \\ \{\delta(s, \langle \Delta_1, a_1 \rangle), \delta(s, \langle \Delta_2, a_2 \rangle)\} & \text{if } \Delta_1 = \Delta_2. \end{cases}$$

The time elapsed when the moves $m_1 = \langle \Delta_1, a_1 \rangle$ and $m_2 = \langle \Delta_2, a_2 \rangle$ are proposed is given by $\mathsf{delay}(m_1, m_2) = \min(\Delta_1, \Delta_2)$. The boolean predicate $\mathsf{blame}_i(s, m_1, m_2, s')$ indicates whether player i is "responsible" for the state change from s to s' when the moves m_1 and m_2 are proposed. Denoting the opponent of player $i \in \{1, 2\}$ by $\sim i = 3 - i$, we define

$$\mathsf{blame}_i(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle, s') \;=\; \big(\Delta_i \leq \Delta_{\sim i} \;\wedge\; \delta(s, \langle \Delta_i, a_i \rangle) = s'\big).$$

A *run* of the timed game structure $\mathcal{G}$ is an infinite sequence $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ such that $s_k \in S$ and $m_i^k \in \Gamma_i(s_k)$ and $s_{k+1} \in \delta_{\mathsf{jd}}(s_k, m_1^k, m_2^k)$ for all $k \geq 0$ and $i \in 1, 2$. For $k \geq 0$, let $\mathsf{time}(r, k)$ denote the "time" at position k of the run, namely, $\mathsf{time}(r, k) = \sum_{j=0}^{k-1} \mathsf{delay}(m_1^j, m_2^j)$ (we let $\mathsf{time}(r, 0) = 0$). By $r[k]$ we denote the $(k+1)$-th state s_k of r. The run prefix $r[0..k]$ is the finite prefix of the run r that ends in the state s_k; we write $\mathsf{last}(r[0..k])$ for the ending state s_k of the run prefix. Let Runs be the set of all runs of $\mathcal{G}$, and let $\mathsf{FinRuns}$ be the set of run prefixes.

A *strategy* π_i for player $i \in \{1, 2\}$ is a function $\pi_i : \mathsf{FinRuns} \mapsto M_i$ that assigns to every run prefix $r[0..k]$ a move to be proposed by player i at the state $\mathsf{last}(r[0..k])$ if the history of the game is $r[0..k]$. We require that $\pi_i(r[0..k]) \in \Gamma_i(\mathsf{last}(r[0..k]))$ for every run prefix $r[0..k]$, so that strategies propose only available moves. The results of this paper are equally valid if strategies do not depend on past moves chosen by the players, but only on the past sequence of states and time delays [13]. For $i \in \{1, 2\}$, let Π_i be the set of player-i strategies. Given two strategies $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$, the set of possible *outcomes* of the game starting from a state $s \in S$ is denoted $\mathsf{Outcomes}(s, \pi_1, \pi_2)$: it contains all runs $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ such that $s_0 = s$ and for all $k \geq 0$ and $i \in \{1, 2\}$, we have $\pi_i(r[0..k]) = m_i^k$.

2.2 Timed Winning Conditions

An *objective* for the timed game structure $\mathcal{G}$ is a set $\Phi \subseteq \mathsf{Runs}$ of runs. The objective Φ is *untimed ω-regular* if there exists an ω-regular set $\Psi \subseteq (2^{\Sigma})^{\omega}$ of infinite sequences of sets of propositions such that a run $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ is in Φ iff the projection $\sigma(r) = \sigma(s_0), \sigma(s_1), \sigma(s_2), \ldots$ is in Ψ.

To win an objective Φ, a player must ensure that the possible outcomes of the game satisfy the *winning condition* $\mathsf{WC}(\Phi)$, a different subset of Runs. We distinguish between objectives and winning conditions, because players must win their objectives using only physically meaningful strategies; for example, a player should not satisfy the objective of staying in a safe set by blocking time forever. Formally, player $i \in \{1, 2\}$ *wins* for the objective Φ at a state $s \in S$ if there is a player-i strategy π_i such that for all opposing strategies $\pi_{\sim i}$, we have $\mathsf{Outcomes}(s, \pi_1, \pi_2) \subseteq \mathsf{WC}_i(\Phi)$. In this case, we say that player i has the *winning strategy* π_i. The winning condition is formally defined as

$$\mathsf{WC}_i(\Phi) \;=\; (\mathsf{Timediv} \cap \Phi) \;\cup\; (\mathsf{Blameless}_i \setminus \mathsf{Timediv}),$$

which uses the following two sets of runs:

- $\mathsf{Timediv} \subseteq \mathsf{Runs}$ is the set of all time-divergent runs. A run r is *time-divergent* if $\lim_{k \to \infty} \mathsf{time}(r, k) = \infty$.
- $\mathsf{Blameless}_i \subseteq \mathsf{Runs}$ is the set of runs in which player i is responsible only for finitely many transitions. A run $s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ belongs to the set $\mathsf{Blameless}_i$, for $i = \{1, 2\}$, if there exists a $k \geq 0$ such that for all $j \geq k$, we have $\neg\, \mathsf{blame}_i(s_j, m_1^j, m_2^j, s_{j+1})$.

Thus a run r belongs to $\mathsf{WC}_i(\Phi)$ if and only if the following conditions hold:

- if $r \in \mathsf{Timediv}$, then $r \in \Phi$;
- if $r \notin \mathsf{Timediv}$, then $r \in \mathsf{Blameless}_i$.

Informally, if time diverges, then the outcome of the game is valid and the objective must be met, and if time does not diverge, then only the opponent should be responsible for preventing time from diverging.

A state $s \in S$ in a timed game structure $\mathcal{G}$ is *well-formed* if both players can win at s for the trivial objective Runs. The timed game structure $\mathcal{G}$ is *well-formed* if all states of $\mathcal{G}$ are well-formed. Structures that are not well-formed are not physically meaningful. We retrict out attention to well-formed timed game structures.

A strategy π_i for player $i \in \{1, 2\}$ is *reasonable* if for all opposing strategies $\pi_{\sim i}$, all states $s \in S$, and all runs $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$, either $r \in \mathsf{Timediv}$ or $r \in \mathsf{Blameless}_i$. Thus, no what matter what the opponent does, a reasonable player-i strategy should not be responsible for blocking time. Strategies that are not reasonable are not physically meaningful. A timed game structure is thus well-formed iff both players have reasonable strategies. We now show that we can restrict our attention to games which allow only reasonable strategies.

Proposition 1. *Let $s \in S$ be a state of a well-formed time game structure $\mathcal{G}$, and let $\Phi \subseteq \mathsf{Runs}$ be an objective.*

1. *Player 1 wins for the objective Φ at the state s iff there is a* reasonable *player-1 winning strategy π_1^*, that is, for all player-2 strategies π_2, we have $\mathsf{Outcomes}(s, \pi_1^*, \pi_2) \subseteq \mathsf{WC}(\Phi)$.*
2. *Player 1 does not win for the objective Φ at s using only reasonable strategies iff there is a reasonable player-2 spoiling strategy π_2^*. Formally, for every reasonable player-1 strategy π_1^*, there is a player-2 strategy π_2 such that $\mathsf{Outcomes}(s, \pi_1^*, \pi_2) \not\subseteq \mathsf{WC}(\Phi)$ iff there is a* reasonable *player-2 strategy π_2^* such that $\mathsf{Outcomes}(s, \pi_1^*, \pi_2^*) \not\subseteq \mathsf{WC}(\Phi)$.*

The symmetric claims with players 1 and 2 interchanged also hold.

Proof. (1) Let π_1 be the winning strategy for player 1 for objective Φ at state s. Let π_1 be not reasonable. Then, by definition, there exists an opposing strategy π_2 such that for some run $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$, we have both $r \notin \mathsf{Timediv}$ and $r \notin \mathsf{Blameless}_1$. This contradicts the fact that π_1 was a winning strategy.
(2) Let π_1^* be any player-1 reasonable strategy. Player 1 loses for the objective Φ from state s, thus there exists a player 2 spoiling strategy π_2 such that $\mathsf{Outcomes}(s, \pi_1^*, \pi_2) \nsubseteq \mathsf{WC}(\Phi)$. This requires that for some run $r \in \mathsf{Outcomes}(s, \pi_1^*, \pi_2)$, we have either 1) $(r \in \mathsf{Timediv}) \wedge (r \notin \Phi)$ or 2) $(r \notin \mathsf{Timediv}) \wedge (r \notin \mathsf{Blameless}_1)$. We cannot have the second case, for π_1^* is a reasonable strategy, thus, the first case must hold. By definition, for every state s' in a well-formed time game structure, there exists a player-2 reasonable strategy $\pi_2^{s'}$. Now, let π_2^* be such that its acts like π_2 on the particular run r, and is like π_2^s otherwise, that is $\pi_2^*(r_f) = \pi_2(r_f)$, for all run prefixes r_f of r, and $\pi_2^*(r_f) = \pi_2^s(r_f)$ otherwise. The strategy π_2^* is reasonable, as for all strategies π_1', and for every run $r' \in \mathsf{Outcomes}(s, \pi_1', \pi_2^*)$, we have $(r' \in \mathsf{Timediv}) \vee (r' \in \mathsf{Blameless}_2)$. Since π_2^* acts like π_2 on the particular run r, it is also spoiling for the player-1 strategy π_1^*. □

Corollary 1. *For $i = \{1, 2\}$, let $\mathsf{Win}_i(\Phi)$ be the states of a well-formed timed game structure $\mathcal{G}$ at which player i can win for the objective Φ. Let $\mathsf{Win}_i^*(\Phi)$ be the states at which player i can win for the objective Φ when both players are restricted to use reasonable strategies. Then, $\mathsf{Win}_i(\Phi) = \mathsf{Win}_i^*(\Phi)$.*

Note that if π_1^* and π_2^* are player-1 and player-2 reasonable strategies, then for every state s and every run $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$, the run r is non-zeno. Thus, if we restrict our attention to plays in which both players use only reasonable strategies, then for every objective Φ, player i wins for the winning condition $\mathsf{WC}(\Phi)$ if and only if she wins for the winning condition Φ. We can hence talk *semantically* about games restricted to reasonable player strategies in well-formed timed game structures, without differentiating between objectives and winning conditions. From a *computational* perspective, we allow all strategies, taking care to distinguish between objectives and winning conditions. Proposition 1 indicates both approaches to be equivalent.

2.3 Timed Automaton Games

Timed automata [4] suggest a finite syntax for specifying infinite-state timed game structures. A *timed automaton game* is a tuple $\mathcal{T} = \langle L, \Sigma, \sigma, C, A_1, A_2, E, \gamma \rangle$ with the following components:

- L is a finite set of locations.
- Σ is a finite set of propositions.
- $\sigma : L \mapsto 2^{\Sigma}$ assigns to every location a set of propositions.
- C is a finite set of clocks. We assume that $z \in C$ for the unresettable clock z, which is used to measure the time elapsed since the start of the game.

- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively.
- $E \subseteq L \times (A_1 \cup A_2) \times \mathsf{Constr}(C) \times L \times 2^{C \backslash \{z\}}$ is the edge relation, where the set $\mathsf{Constr}(C)$ of *clock constraints* is generated by the grammar

  $$\theta ::= x \leq d \mid d \leq x \mid \neg\theta \mid \theta_1 \wedge \theta_2$$

  for clock variables $x \in C$ and nonnegative integer constants d. For an edge $e = \langle l, a_i, \theta, l', \lambda \rangle$, the clock constraint θ acts as a guard on the clock values which specifies when the edge e can be taken, and by taking the edge e, the clocks in the set $\lambda \subseteq C \backslash \{z\}$ are reset to 0. We require that for all edges $\langle l, a_i, \theta', l', \lambda' \rangle, \langle l, a_i, \theta'', l'', \lambda'' \rangle \in E$ with $l' \neq l''$, the conjunction $\theta' \wedge \theta''$ is unsatisfiable. This requirement ensures that a state and a move together uniquely determine a successor state.
- $\gamma : L \mapsto \mathsf{Constr}(C)$ is a function that assigns to every location an invariant for both players. All clocks increase uniformly at the same rate. When at location l, each player i must propose a move out of l before the invariant $\gamma(l)$ expires. Thus, the game can stay at a location only as long as the invariant is satisfied by the clock values.

A *clock valuation* is a function $\kappa : C \mapsto \mathbb{R}_{\geq 0}$ that maps every clock to a nonnegative real. The set of all clock valuations for C is denoted by $K(C)$. Given a clock valuation $\kappa \in K(C)$ and a time delay $\Delta \in \mathbb{R}_{\geq 0}$, we write $\kappa + \Delta$ for the clock valuation in $K(C)$ defined by $(\kappa + \Delta)(x) = \kappa(x) + \Delta$ for all clocks $x \in C$. For a subset $\lambda \subseteq C$ of the clocks, we write $\kappa[\lambda := 0]$ for the clock valuation in $K(C)$ defined by $(\kappa[\lambda := 0])(x) = 0$ if $x \in \lambda$, and $(\kappa[\lambda := 0])(x) = \kappa(x)$ if $x \notin \lambda$. A clock valuation $\kappa \in K(C)$ *satisfies* the clock constraint $\theta \in \mathsf{Constr}(C)$, written $\kappa \models \theta$, if the condition θ holds when all clocks in C take on the values specified by κ.

A *state* $s = \langle l, \kappa \rangle$ of the timed automaton game $\mathcal{T}$ is a location $l \in L$ together with a clock valuation $\kappa \in K(C)$ such that the invariant at the location is satisfied, that is, $\kappa \models \gamma(l)$. Let S be the set of all states of $\mathcal{T}$. In a state, each player i proposes a time delay allowed by the invariant map γ, together either with the action $\perp$, or with an action $a_i \in A_i$ such that an edge labeled a_i is enabled after the proposed time delay. We require that for $i \in \{1, 2\}$ and for all states $s = \langle l, \kappa \rangle$, if $\kappa \models \gamma(l)$, either $\kappa + \Delta \models \gamma(l)$ for all $\Delta \in \mathbb{R}_{\geq 0}$, or there exist a time delay $\Delta \in \mathbb{R}_{\geq 0}$ and an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that (1) $a_i \in A_i$ and (2) $\kappa + \Delta \models \theta$ and for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$, and (3) $(\kappa + \Delta)[\lambda := 0] \models \gamma(l')$. This requirement is necessary (but not sufficient) for well-formedness of the game.

The timed automaton game $\mathcal{T}$ defines the following timed game structure $[\![\mathcal{T}]\!] = \langle S, \Sigma, \sigma^*, A_1, A_2, \Gamma_1, \Gamma_2, \delta \rangle$:

- S is defined above.
- $\sigma^*(\langle l, \kappa \rangle) = \sigma(l)$.
- For $i \in \{1, 2\}$, the set $\Gamma_i(\langle l, \kappa \rangle)$ contains the following elements:
  1. $\langle \Delta, \perp \rangle$ if for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$.
  2. $\langle \Delta, a_i \rangle$ if for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$, and $a_i \in A_i$, and there exists an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that $\kappa + \Delta \models \theta$.

– $\delta(\langle l, \kappa\rangle, \langle \Delta, \bot\rangle) = \langle l, \kappa + \Delta\rangle$, and $\delta(\langle l, \kappa\rangle, \langle \Delta, a_i\rangle) = \langle l', (\kappa + \Delta)[\lambda := 0]\rangle$ for the unique edge $\langle l, a_i, \theta, l', \lambda\rangle \in E$ with $\kappa + \Delta \models \theta$.

The timed game structure $[\![\mathcal{T}]\!]$ is not necessarily well-formed, because it may contain cycles along which time cannot diverge. We will see below how we can check well-formedness for timed automaton games.

2.4 Clock Regions

Timed automaton games can be solved using a region construction from the theory of timed automata [4]. For a real $t \geq 0$, let $\mathsf{frac}(t) = t - \lfloor t \rfloor$ denote the fractional part of t. Given a timed automaton game $\mathcal{T}$, for each clock $x \in C$, let c_x denote the largest integer constant that appears in any clock constaint involving x in $\mathcal{T}$ (let $c_x = 0$ if there is no clock constraint invloving x). Two clock valuations $\kappa_1, \kappa_2 \in K(C)$ are *clock-region equivalent*, denoted $\kappa_1 \cong \kappa_2$, if the following three conditions hold:

1. For all $x \in C$, either $\lfloor \kappa_1(x) \rfloor = \lfloor \kappa_2(x) \rfloor$, or both $\lfloor \kappa_1(x) \rfloor > c_x$, $\lfloor \kappa_2(x) \rfloor > c_x$.
2. For all $x, y \in C$ with $\kappa_1(x) \leq c_x$ and $\kappa_1(y) \leq c_y$, we have $\mathsf{frac}(\kappa_1(x)) \leq \mathsf{frac}(\kappa_1(y))$ iff $\mathsf{frac}(\kappa_2(x)) \leq \mathsf{frac}(\kappa_2(y))$.
3. For all $x \in C$ with $\kappa_1(x) \leq c_x$, we have $\mathsf{frac}(\kappa_1(x)) = 0$ iff $\mathsf{frac}(\kappa_2(x)) = 0$.

Two states $\langle l_1, \kappa_1\rangle, \langle l_2, \kappa_2\rangle \in S$ are *clock-region equivalent*, denoted $\langle l_1, \kappa_1\rangle \cong \langle l_2, \kappa_2\rangle$, iff $l_1 = l_2$ and $\kappa_1 \cong \kappa_2$. It is not difficult to see that $\cong$ is an equivalence relation on S. A *clock region* is an equivalence class with respect to $\cong$. There are finitely many clock regions; more precisely, the number of clock regions is bounded by $|L| \cdot \prod_{x \in C}(c_x + 1) \cdot |C|! \cdot 2^{|C|}$. For a state $s \in S$, we write $[s] \subseteq S$ for the clock region containing s.

Timed automaton games can be solved for untimed ω-regular objectives by considering the finite quotient game structure obtained from the clock-region equivalence [13]. There, also a variation of timed games is considered, where each move is either a time delay or an action (rather than a pair of both). The results of this paper carry over also to that model (which is strictly weaker, in that a player may be able to achieve fewer objectives).

3 TATL

The alternating-time temporal logics ATL and ATL* were introduced in [6] for specifying properties of untimed game structures. These logics are natural specification languages for multi-component systems, where properties need to be guarenteed by subsets of the components irrespective of the behavior of the other components. Each component represents a player in the game, and sets of players may form teams. We restrict our attention here to the two-player case (e.g., system vs. environment; or plant vs. controller), but all results can be extended to the multi-player case. For example, letting the system be player 1, and the environment player 2, the ATL formula $\langle\!\langle 1 \rangle\!\rangle \Box p$ specifies the property that the system will always remain in a safe set of p states, no matter what the environment does.

For timed systems, we need the players to use only reasonable strategies when achieving their objectives. We show that this requirement can be encoded within ATL* (but not within ATL). We also permit timing constraints within objectives. For example, the timed formula $\langle\langle 1\rangle\rangle\Diamond_{\leq d}\, p$ says that player 1 can reach a target set of p states within d time units, no matter what player 2 does (and again, player 1 must use only reasonable strategies to attain this goal). The resulting timed logics are called TATL and TATL*. While the model-checking problem of TATL* is undecidable over timed automaton games, we show that it is decidable for the fragment of TATL* that is obtained by adding to TATL the restriction to reasonable strategies.

3.1 Syntax and Semantics

Consider a fixed timed game structure $\mathcal{G} = \langle S, \Sigma, \sigma, A_1, A_2, \Gamma_1, \Gamma_2, \delta\rangle$. The temporal logic TATL (Timed Alternating-Time Temporal Logic) is interpreted over the states of $\mathcal{G}$. We use the syntax of *freeze quantification* [5] for specifying timing constraints within the logic. The freeze quantifier "$x\cdot$" binds the value of the clock variable x in a formula $\varphi(x)$ to the current time $t \in \mathbb{R}_{\geq 0}$; that is, the constraint $x\cdot\varphi(x)$ holds at time t iff $\varphi(t)$ does. For example, the property that "every p state is followed by a q state within d time units" can be written as: $\forall\Box x\cdot(p \rightarrow \Diamond y\cdot(q \wedge y \leq x + d))$. This formula says that "in every state with time x, if p holds, then there is a later state with time y such that both q and $y \leq x + d$ hold." Formally, given a set D of clock variables, a TATL formula is one of the following:

- $\textsc{true} \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$, where $p \in \Sigma$ is a proposition, and φ_1, φ_2 are TATL formulae.
- $x + d_1 \leq y + d_2 \mid x\cdot\varphi$, where $x, y \in D$ are clock variables and d_1, d_2 are nonnegative integer constants, and φ is a TATL formula. We refer to the clocks in D as *formula clocks.*
- $\langle\langle\mathfrak{P}\rangle\rangle\Box\varphi \mid \langle\langle\mathfrak{P}\rangle\rangle\varphi_1\,\mathcal{U}\varphi_2$, where $\mathfrak{P} \subseteq \{1, 2\}$ is a set of players, and $\varphi, \varphi_1, \varphi_2$ are TATL formulae.

We omit the next operator of ATL, which has no meaning in timed systems. The freeze quantifier $x\cdot\varphi$ binds all free occurrences of the formula clock variable x in the formula φ. A TATL formula is *closed* if it contains no free occurrences of formula clock variables. Without loss of generality, we assume that for every quantified formula $x\cdot\varphi$, if $y\cdot\varphi'$ is a subformula of φ, then x and y are different; that is, there is no nested reuse of formula clocks. When interpreted over the states of a timed automaton game $\mathcal{T}$, a TATL formula may also contain free (unquantified) occurrences of clock variables from $\mathcal{T}$.

There are four possible sets of players (so-called *teams*), which may collaborate to achieve a common goal: we write $\langle\langle\,\rangle\rangle$ for $\langle\langle\emptyset\rangle\rangle$; we write $\langle\langle i\rangle\rangle$ for $\langle\langle\{i\}\rangle\rangle$ with $i \in \{1, 2\}$; and we write $\langle\langle 1, 2\rangle\rangle$ for $\langle\langle\{1, 2\}\rangle\rangle$. Roughly speaking, a state s

satisfies the TATL formula $\langle\langle i\rangle\rangle\varphi$ iff player i can win the game at s for an objective derived from φ. The state s satisfies the formula $\langle\langle\,\rangle\rangle\varphi$ (resp., $\langle\langle 1,2\rangle\rangle\varphi$) iff every run (resp., some run) from s is contained in the objective derived from φ. Thus, the team $\emptyset$ corresponds to both players playing adversially, and the team $\{1,2\}$ corresponds to both players collaborating to achieve a goal. We therefore write $\forall$ short for $\langle\langle\,\rangle\rangle$, and $\exists$ short for $\langle\langle 1,2\rangle\rangle$, as in ATL.

We assign the responsibilities for time divergence to teams as follows: let $\mathsf{Blameless}_\emptyset = \mathsf{Runs}$, let $\mathsf{Blameless}_{\{1,2\}} = \emptyset$, and let $\mathsf{Blameless}_{\{i\}} = \mathsf{Blameless}_i$ for $i \in \{1,2\}$. A strategy $\pi_{\mathfrak{P}}$ for the team $\mathfrak{P}$ consists of a strategy for each player in $\mathfrak{P}$. We denote the "opposing" team by $\sim\!\mathfrak{P} = \{1,2\} \setminus \mathfrak{P}$. Given a state $s \in S$, a team-$\mathfrak{P}$ strategy $\pi_{\mathfrak{P}}$, and a team-$\sim\!\mathfrak{P}$ strategy $\pi_{\sim\mathfrak{P}}$, we define $\mathsf{Outcomes}(s, \pi_{\mathfrak{P}} \cup \pi_{\sim\mathfrak{P}}) = \mathsf{Outcomes}(s, \pi_1, \pi_2)$ for the player-1 strategy π_1 and the player-2 strategy π_2 in the set $\pi_{\mathfrak{P}} \cup \pi_{\sim\mathfrak{P}}$ of strategies. Given a team-$\mathfrak{P}$ strategy $\pi_{\mathfrak{P}}$, we define the set of possible outcomes from state s by $\mathsf{Outcomes}(s, \pi_{\mathfrak{P}}) = \cup_{\pi_{\sim\mathfrak{P}}} \mathsf{Outcomes}(s, \pi_{\mathfrak{P}} \cup \pi_{\sim\mathfrak{P}})$, where the union is taken over all team-$\sim\!\mathfrak{P}$ strategies $\pi_{\sim\mathfrak{P}}$.

To define the semantics of TATL, we need to distinguish between *physical time* and *game time*. We allow moves with zero time delay, thus a physical time $t \in \mathbb{R}_{\geq 0}$ may correspond to several linearly ordered states, to which we assign the game times $\langle t, 0\rangle, \langle t, 1\rangle, \langle t, 2\rangle, \ldots$ For a run $r \in \mathsf{Runs}$, we define the set of game times as

$$\mathsf{GameTimes}(r) \;=\; \begin{array}{l}\{\langle t, k\rangle \in \mathbb{R}_{\geq 0} \times \mathbb{N} \mid 0 \leq k < |\{j \geq 0 \mid \mathsf{time}(r,j) = t\}|\} \;\cup \\ \{\langle t, 0\rangle \mid \mathsf{time}(r,j) \geq t \text{ for some } j \geq 0\}.\end{array}$$

The state of the run r at a game time $\langle t, k\rangle \in \mathsf{GameTimes}(r)$ is defined as

$$\mathsf{state}(r, \langle t, k\rangle) \;=\; \begin{cases} r[j+k] \;\text{ if } \mathsf{time}(r,j) = t \text{ and for all } j' < j, \mathsf{time}(r,j') < t; \\ \delta(r[j], \langle t - \mathsf{time}(r,j), \bot\rangle) \;\text{ if } \mathsf{time}(r,j) < t < \mathsf{time}(r,j+1). \end{cases}$$

Note that if r is a run of the timed game structure $\mathcal{G}$, and $\mathsf{time}(r,j) < t < \mathsf{time}(r,j+1)$, then $\delta(r[j], \langle t - \mathsf{time}(r,j), \bot\rangle)$ is a state in S, namely, the state that results from $r[j]$ by letting time $t - \mathsf{time}(r,j)$ pass. We say that the run r *visits* a proposition $p \in \Sigma$ if there is a $\tau \in \mathsf{GameTimes}(r)$ such that $p \in \sigma(\mathsf{state}(r,\tau))$. We order the game times of a run lexicographically: for all $\langle t, k\rangle, \langle t', k'\rangle \in \mathsf{GameTimes}(r)$, we have $\langle t, k\rangle < \langle t', k'\rangle$ iff either $t < t'$, or $t = t'$ and $k < k'$. For two game times τ and τ', we write $\tau \leq \tau'$ iff either $\tau = \tau'$ or $\tau < \tau'$.

An *environment* $\mathcal{E} : D \mapsto \mathbb{R}_{\geq 0}$ maps every formula clock in D to a nonnegative real. Let $\mathcal{E}[x := t]$ be the environment such that $(\mathcal{E}[x := t])(y) = \mathcal{E}(y)$ if $y \neq x$, and $(\mathcal{E}[x := t])(y) = t$ if $y = x$. For a state $s \in S$, a time $t \in \mathbb{R}_{\geq 0}$, an environment $\mathcal{E}$, and a TATL formula φ, the satisfaction relation $(s, t, \mathcal{E}) \models_{\mathrm{td}} \varphi$ is defined inductively as follows (the subscript td indicates that players may win in only a physically meaningful way):

- $(s, t, \mathcal{E}) \models_{\mathrm{td}}$ TRUE.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} p$, for a proposition p, iff $p \in \sigma(s)$.

- $(s, t, \mathcal{E}) \models_{\mathrm{td}} \neg\varphi$ iff $(s, t, \mathcal{E}) \not\models_{\mathrm{td}} \varphi$.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} \varphi_1 \wedge \varphi_2$ iff $(s, t, \mathcal{E}) \models_{\mathrm{td}} \varphi_1$ and $(s, t, \mathcal{E}) \models_{\mathrm{td}} \varphi_2$.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} x + d_1 \leq y + d_2$ iff $\mathcal{E}(x) + d_1 \leq \mathcal{E}(y) + d_2$.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} x \cdot \varphi$ iff $(s, t, \mathcal{E}[x := t]) \models_{\mathrm{td}} \varphi$.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} \langle\langle \mathfrak{P} \rangle\rangle \Box \varphi$ iff there is a team-$\mathfrak{P}$ strategy $\pi_{\mathfrak{P}}$ such that for all runs $r \in \mathsf{Outcomes}(s, \pi_{\mathfrak{P}})$, the following conditions hold:

  If $r \in \mathsf{Timediv}$, then for all $\langle u, k \rangle \in \mathsf{GameTimes}(r)$, we have $(\mathsf{state}(r, \langle u, k \rangle), t + u, \mathcal{E}) \models_{\mathrm{td}} \varphi$. If $r \notin \mathsf{Timediv}$, then $r \in \mathsf{Blameless}_{\mathfrak{P}}$.
- $(s, t, \mathcal{E}) \models_{\mathrm{td}} \langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ iff there is a team-$\mathfrak{P}$ strategy $\pi_{\mathfrak{P}}$ such that for all runs $r \in \mathsf{Outcomes}(s, \pi_{\mathfrak{P}})$, the following conditions hold:

  If $r \in \mathsf{Timediv}$, then there is a $\langle u, k \rangle \in \mathsf{GameTimes}(r)$ such that $(\mathsf{state}(r, \langle u, k \rangle), t + u, \mathcal{E}) \models_{\mathrm{td}} \varphi_2$, and for all $\langle u', k' \rangle \in \mathsf{GameTimes}(r)$ with $\langle u', k' \rangle < \langle u, k \rangle$, we have $(\mathsf{state}(r, \langle u', k' \rangle), t + u', \mathcal{E}) \models_{\mathrm{td}} \varphi_1$. If $r \notin \mathsf{Timediv}$, then $r \in \mathsf{Blameless}_{\mathfrak{P}}$.

Note that for an $\exists$ formula to hold, we require time divergence (as $\mathsf{Blameless}_{\{1,2\}} = \emptyset$). Also note that for a closed formula, the value of the environment is irrelevant in the satisfaction relation. A state s of the timed game structure $\mathcal{G}$ *satisfies* a closed formula φ of TATL, denoted $s \models_{\mathrm{td}} \varphi$, if $(s, 0, \mathcal{E}) \models_{\mathrm{td}} \varphi$ for any environment $\mathcal{E}$.

We use the following abbreviations. We write $\langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U}_{\sim d} \varphi_2$ for $x \cdot \langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U}\, y \cdot (\varphi_2 \wedge y \sim x + d)$, where $\sim$ is one of $<$, $\leq$, $=$, $\geq$, or $>$. Interval constraints can also be encoded in TATL; for example, $\langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U}_{(d_1, d_2]} \varphi_2$ stands for $x \cdot \langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U}\, y \cdot (\varphi_2 \wedge y > x + d_1 \wedge y \leq x + d_2)$. We write $\Diamond \varphi$ for $\textsc{true}\, \mathcal{U} \varphi$ as usual, and therefore $\langle\langle \mathfrak{P} \rangle\rangle \Diamond_{\sim d} \varphi$ stands for $x \cdot \langle\langle \mathfrak{P} \rangle\rangle \Diamond y \cdot (\varphi \wedge y \sim x + d)$.

3.2 TATL*

TATL is a fragment of the more expressive logic called TATL*. There are two types of formulae in TATL*: *state formulae*, whose satisfaction is related to a particular state, and *path formulae*, whose satisfaction is related to a specific run. Formally, a TATL* state formula is one of the following:

(S1) TRUE or p for propositions $p \in \Sigma$.
(S2) $\neg\varphi$ or $\varphi_1 \wedge \varphi_2$ for TATL* state formulae φ, φ_1, and φ_2.
(S3) $x + d_1 \leq y + d_2$ for clocks $x, y \in D$ and nonnegative integer constants d_1, d_2.
(S4) $\langle\langle \mathfrak{P} \rangle\rangle \psi$ for $\mathfrak{P} \subseteq \{1, 2\}$ and TATL* path formulae ψ.

A TATL* path formula is one of the following:

(P1) A TATL* state formula.
(P2) $\neg\psi$ or $\psi_1 \wedge \psi_2$ for TATL* path formulae ψ, ψ_1, and ψ_2.
(P3) $x \cdot \psi$ for formula clocks $x \in D$ and TATL* path formulae ψ.
(P4) $\psi_1 \mathcal{U} \psi_2$ for TATL* path formulae ψ_1, ψ_2.

The logic TATL* consists of the formulae generated by the rules S1–S4. As in TATL, we assume that there is no nested reuse of formula clocks. Additional temporal operators are defined as usual; for example, $\Diamond\varphi$ stands for $\text{TRUE}\,\mathcal{U}\varphi$, and $\Box\varphi$ stands for $\neg\Diamond\neg\varphi$. The logic TATL can be viewed as a fragment of TATL* consisting of formuale in which every $\mathcal{U}$ operator is immediately preceeded by a $\langle\langle\mathfrak{P}\rangle\rangle$ operator, possibly with an intermittent negation symbol [6].

The semantics of TATL* formulae are defined with respect to an environment $\mathcal{E} : D \mapsto \mathbb{R}_{\geq 0}$. We write $(s, t, \mathcal{E}) \models \varphi$ to indicate that the state s of the timed game structure $\mathcal{G}$ satisfies the TATL* state formula φ at time $t \in \mathbb{R}_{\geq 0}$; and $(r, \tau, t, \mathcal{E}) \models \psi$ to indicate that the suffix of the run r of $\mathcal{G}$ which starts from game time $\tau \in \mathsf{GameTimes}(r)$ satisfies the TATL* path formula ψ, provided the time at the initial state of r is t. Unlike TATL, we allow all strategies for both players (including unreasonable strategies), because we will see that the use of reasonable strategies can be enforced within TATL* by certain path formulae. Formally, the satisfaction relation $\models$ is defined inductively as follows. For state formulae φ,

- $(s, t, \mathcal{E}) \models \text{TRUE}$.
- $(s, t, \mathcal{E}) \models p$, for a proposition p, iff $p \in \sigma(s)$.
- $(s, t, \mathcal{E}) \models \neg\varphi$ iff $(s, t, \mathcal{E}) \not\models \varphi$.
- $(s, t, \mathcal{E}) \models \varphi_1 \wedge \varphi_2$ iff $(s, t, \mathcal{E}) \models \varphi_1$ and $(s, t, \mathcal{E}) \models \varphi_2$.
- $(s, t, \mathcal{E}) \models x + d_1 \leq y + d_2$ iff $\mathcal{E}(x) + d_1 \leq \mathcal{E}(y) + d_2$.
- $(s, t, \mathcal{E}) \models \langle\langle\mathfrak{P}\rangle\rangle\psi$ iff there is a team-$\mathfrak{P}$ strategy $\pi_{\mathfrak{P}}$ such that for all runs $r \in \mathsf{Outcomes}(s, \pi_{\mathfrak{P}})$, we have $(r, \langle 0, 0\rangle, t, \mathcal{E}) \models \psi$.

For path formulae ψ,

- $(r, \langle u, k\rangle, t, \mathcal{E}) \models \varphi$, for a state formula φ, iff $(\mathsf{state}(r, \langle u, k\rangle), t + u, \mathcal{E}) \models \varphi$.
- $(r, \tau, t, \mathcal{E}) \models \neg\psi$ iff $(r, \tau, t, \mathcal{E}) \not\models \psi$.
- $(r, \tau, t, \mathcal{E}) \models \psi_1 \wedge \psi_2$ iff $(r, \tau, t, \mathcal{E}) \models \psi_1$ and $(r, \tau, t, \mathcal{E}) \models \psi_2$.
- $(r, \langle u, k\rangle, t, \mathcal{E}) \models x{\cdot}\psi$ iff $(r, \langle u, k\rangle, t, \mathcal{E}[x := t + u]) \models \psi$.
- $(r, \tau, t, \mathcal{E}) \models \psi_1 \mathcal{U} \psi_2$ iff there is a $\tau' \in \mathsf{GameTimes}(r)$ such that $\tau \leq \tau'$ and $(r, \tau', t, \mathcal{E}) \models \psi_2$, and for all $\tau'' \in \mathsf{GameTimes}(r)$ with $\tau \leq \tau'' < \tau'$, we have $(r, \tau'', t, \mathcal{E}) \models \psi_1$.

A state s of the timed game structure $\mathcal{G}$ *satisfies* a closed formula φ of TATL*, denoted $s \models \varphi$, if $(s, 0, \mathcal{E}) \models \varphi$ for any environment $\mathcal{E}$.

3.3 Model Checking TATL

We restrict our attention to timed automaton games. Given a closed TATL (resp. TATL*) formula φ, a timed automaton game $\mathcal{T}$, and a state s of the timed game structure $[\![\mathcal{T}]\!]$, the *model-checking problem* is to determine whether $s \models_{\mathrm{td}} \varphi$ (resp., $s \models \varphi$). The alternating-time logic TATL* subsumes the linear-time logic TPTL [5]. Thus the model-checking problem for TATL* is undecidable. On the other hand, we now solve the model-checking problem for TATL by reducing it to a special kind of TATL* problem, which turns out to be decidable.

Given a TATL formula φ over the set D of formula clocks, and a timed automaton game $\mathcal{T}$, we look at the timed automaton game $\mathcal{T}_\varphi$ with the set $C_\varphi = C \uplus D$ of clocks (we assume $C \cap D = \emptyset$). Let c_x be the largest constant to which the formula variable x is compared in φ. We pick an invariant $\gamma(l)$ in $\mathcal{T}$ and modify it to $\gamma(l)' = \gamma(l) \wedge (x \leq c_x \vee x \geq c_x)$ in $\mathcal{T}_\varphi$ for every formula clock $x \in D$ (this is to inject the proper constants in the region equivalence relation). Thus, $\mathcal{T}_\varphi$ acts exactly like $\mathcal{T}$ except that it contains some extra clocks which are never used. As in [13], we represent the sets $\mathsf{Timediv}$ and $\mathsf{Blameless}_i$ using ω-regular conditions. We look at the enlarged automaton game structure $\widehat{[\![\mathcal{T}_\varphi]\!]}$ with the state space $\widehat{S} = S_\varphi \times \{T, F\}^3$, and an augmented transition relation $\widehat{\delta}_{\mathsf{jd}} : \widehat{S} \times M_1 \times M_2 \mapsto 2^{\widehat{S}}$. In an augmented state $\langle s, \mathit{tick}, \mathit{bl}_1, \mathit{bl}_2\rangle \in \widehat{S}$, the component $s \in S_\varphi$ is a state of the original game structure $[\![\mathcal{T}_\varphi]\!]$, tick is true if the global clock z has crossed an integer boundary in the last transition, and bl_i is true if player i is to blame for the last transition. Formally, $\langle\langle l', \kappa'\rangle, \mathit{tick}', \mathit{bl}_1', \mathit{bl}_2'\rangle \in \widehat{\delta}_{\mathsf{jd}}(\langle\langle l, \kappa\rangle, \mathit{tick}, \mathit{bl}_1, \mathit{bl}_2\rangle, m_1, m_2)$ iff (1) $\langle l', \kappa'\rangle \in \delta_{\mathsf{jd}}(\langle l, \kappa\rangle, m_1, m_2)$; (2) $\mathit{tick}' = \text{TRUE}$ if $\kappa'(z) - \kappa(z) \geq 1$, and FALSE otherwise; and (3) $\mathit{bl}_i' = \mathsf{blame}_i(\langle l, \kappa\rangle, m_1, m_2, \langle l', \kappa'\rangle)$. It can be seen that a run is in $\mathsf{Timediv}$ iff tick is true infinitely often, and that the set $\mathsf{Blameless}_i$ corresponds to runs along which bl_i is true only finitely often. We extend the clock equivalence relation to these expanded states: $\langle\langle l, \kappa\rangle, \mathit{tick}, \mathit{bl}_1, \mathit{bl}_2\rangle \cong \langle\langle l', \kappa'\rangle, \mathit{tick}', \mathit{bl}_1', \mathit{bl}_2'\rangle$ iff $l = l', \mathit{tick} = \mathit{tick}', \mathit{bl}_1 = \mathit{bl}_1', \mathit{bl}_2 = \mathit{bl}_2'$ and $\kappa \cong \kappa'$. Finally, we extend bl to teams: $\mathit{bl}_\emptyset = \text{FALSE}$, $\mathit{bl}_{\{1,2\}} = \text{TRUE}$, $\mathit{bl}_{\{i\}} = \mathit{bl}_i$.

We will use the algorithm of [13] which computes winning sets for timed automaton games with untimed ω-regular objectives. The algorithm uses the *controllable predecessor operator*, $\mathsf{CPre}_1 : 2^{\widehat{S}} \mapsto 2^{\widehat{S}}$ in its fixpoint computation, defined formally by $\mathsf{CPre}_1(\widehat{X}) \equiv \{\widehat{s} \mid \exists m_1 \in \Gamma_1(\widehat{s})\ \forall m_2 \in \Gamma_2(\widehat{s})(\widehat{\delta}_{\mathsf{jd}}(\widehat{s}, m_1, m_2) \subseteq \widehat{X}\})$. Intuitively, $\widehat{s} \in \mathsf{CPre}_1(\widehat{X})$ iff player 1 can force the augmented game into $\widehat{X}$ from $\widehat{s}$ in one move. The CPre_1 operator is invariant over states of a region, that is, for $\widehat{X}$ a union of regions, and $\widehat{s} \cong \widehat{s}'$, we have $\widehat{s} \in \mathsf{CPre}_1(\widehat{X})$ iff $\widehat{s}' \in \mathsf{CPre}_1(\widehat{X})$. This invariance follows from the fact that if player 1 can force the game into a region $\widehat{R}$ from $\widehat{s}$, then he can do so from any other state $\widehat{s}' \cong \widehat{s}$. The region invariance of CPre_1 allows the us to work on the region game graph. So long as we work with state sets that correspond to unions of regions, we get winning sets that are also unions of regions. We show that we can maintain this invariant when model checking TATL.

We first consider the subset of TATL in which formulae are clock variable free. Using the encoding for time divergence and blame predicates, we can embed the notion of reasonable winning strategies into TATL* formulae.

Lemma 1. *A state s in a timed game structure $[\![\mathcal{T}_\varphi]\!]$ satisfies a formula clock variable free* TATL *formula φ in a meaningful way, denoted $s \models_{\text{td}} \varphi$, iff the state $\widehat{s} = \langle s, \text{FALSE}, \text{FALSE}, \text{FALSE}\rangle$ in the expanded game structure $\widehat{[\![\mathcal{T}_\varphi]\!]}$ satisfies the* TATL* *formula $\mathsf{atlstar}(\varphi)$, that is, iff $\widehat{s} \models \mathsf{atlstar}(\varphi)$ where $\mathsf{atlstar}$ is a partial mapping from* TATL *to* TATL**, defined inductively as follows:*

$\mathsf{atlstar}(\text{TRUE}) = \text{TRUE}$

$$\mathsf{atlstar}(p) = p$$
$$\mathsf{atlstar}(\neg\varphi) = \neg\,\mathsf{atlstar}(\varphi); \qquad \mathsf{atlstar}(\varphi_1 \wedge \varphi_2) = \mathsf{atlstar}(\varphi_1) \wedge \mathsf{atlstar}(\varphi_2)$$
$$\mathsf{atlstar}(\langle\langle \mathfrak{P} \rangle\rangle \Box\varphi) = \langle\langle \mathfrak{P} \rangle\rangle\,((\Box\Diamond\, \mathit{tick} \rightarrow \Box\,\mathsf{atlstar}(\varphi)) \wedge (\Diamond\Box\neg\, \mathit{tick} \rightarrow \Diamond\Box\neg\, \mathit{bl}_{\mathfrak{P}}))$$
$$\mathsf{atlstar}(\langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \,\mathcal{U} \varphi_2) = \langle\langle \mathfrak{P} \rangle\rangle \begin{pmatrix} (\Box\Diamond\, \mathit{tick} \rightarrow \mathsf{atlstar}(\varphi_1)\,\mathcal{U}\,\mathsf{atlstar}(\varphi_2)) \,\wedge \\ (\Diamond\Box\neg\, \mathit{tick} \rightarrow \Diamond\Box\neg\, \mathit{bl}_{\mathfrak{P}}) \end{pmatrix}$$

Now, for φ a clock variable free TATL formula, $\mathsf{atlstar}(\varphi)$ is actually an ATL* formula. Thus, the untimed ω-regular model checking algorithm of [13] can be used to (recursively) model check $\mathsf{atlstar}(\varphi)$. As we are working in the continuous domain, we need to ensure that for an until formula $\langle\langle \mathfrak{P} \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, team $\mathfrak{P}$ does not "jump" over a time at which $\neg(\varphi_1 \vee \varphi_2)$ holds. This can be handled by introducing another player in the opposing team $\sim\!\mathfrak{P}$, the *observer*, who can only take pure time moves. The observer entails the opposing team to observe *all* time points. The observer is necessary only when $\mathfrak{P} = \{1, 2\}$. We omit the details.

A naive extension of the above approach to full TATL does not immediately work, for then we get TATL* formulae which are not in ATL* (model checking for TATL* is undecidable). We do the following: for each formula clock constraint $x + d_1 \leq y + d_2$ appearing in the formula φ, let there be a new proposition p_α for $\alpha = x + d_1 \leq y + d_2$. We denote the set of all such formula clock constraint propositions by Λ. A state $\langle l, \kappa \rangle$ in the timed automaton game $\mathcal{T}_\varphi$ satisfies p_α for $\alpha = x + d_1 \leq y + d_2$ iff $\kappa(x) + d_1 \leq \kappa(y) + d_2$. The propositions p_α are invariant over regions, maintaining the region-invariance of sets in the fixpoint algorithm of [13], thus allowing us to work over the region game graph.

Lemma 2. *For a* TATL *formula φ, let φ^Λ be obtained from φ by replacing all formula variable constraints $x + d_1 \leq y + d_2$ with equivalent propositions $p_\alpha \in \Lambda$. Let $[\![\mathcal{T}_\varphi]\!]^\Lambda$ denote the timed game structure $[\![\mathcal{T}_\varphi]\!]$ together with the propositions from Λ. Then,*

1. *We have $s \models_{\mathrm{td}} \varphi$ for a state s in the timed game structure $[\![\mathcal{T}_\varphi]\!]$ iff the state $s \models_{\mathrm{td}} \varphi^\Lambda$ in $[\![\mathcal{T}_\varphi]\!]^\Lambda$.*
2. *Let $\varphi^\Lambda = w \cdot \psi^\Lambda$. Then, in the structure $[\![\mathcal{T}_\varphi]\!]^\Lambda$ the state $s \models_{\mathrm{td}} \varphi^\Lambda$ iff $s[w := 0] \models_{\mathrm{td}} \psi^\Lambda$.*
3. *Let $\varphi = \langle\langle \mathfrak{P} \rangle\rangle \Box p \mid \langle\langle \mathfrak{P} \rangle\rangle p_1 \mathcal{U} p_2$, where p, p_1, p_2 are propositions that are invariant over states of regions in $\mathcal{T}_\varphi$. Then for $s \cong s'$ in $\mathcal{T}_\varphi$, we have $s \models_{\mathrm{td}} \varphi$ iff $s' \models_{\mathrm{td}} \varphi$.*

Lemmas 1 and 2 together with the EXPTIME algorithm for timed automaton games with untimed ω-regular region objectives give us a recursive model-checking algorithm for TATL.

Theorem 1. *The model-checking problem for* TATL *(over timed automaton games) is* EXPTIME*-complete.*

EXPTIME-hardness follows from the EXPTIME-hardness of alternating reachability on timed automata [17].

Model checking of TATL allows us to check the well-formedness of a timed automaton game $\mathcal{T}$: a state s of the timed game structure $[\![\mathcal{T}]\!]$ is well-formed iff $s \models_{\mathrm{td}} (\langle\langle 1\rangle\rangle \Box \textsc{true}) \wedge (\langle\langle 2\rangle\rangle \Box \textsc{true})$ This well-formedness check is the generalization to the game setting of the non-zenoness check for timed automata, which computes the states s such that $s \models_{\mathrm{td}} \exists\Box\textsc{true}$ [18]. If not all states of $[\![\mathcal{T}]\!]$ are well-formed, then the location invariants of $\mathcal{T}$ can be strengthened to characterize well-formed states (note that the set of well-formed states consists of a union of regions).

4 The Minimum-Time Problem

The *minimum-time problem* is to determine the minimal time in which a player can force the game into a set of target states, using only reasonable strategies. This game problem is the generalisation of the non-game version of [12]. Formally, given a player $i \in \{1,2\}$, a timed game structure $\mathcal{G}$, a target proposition $p \in \Sigma$, and a run r of $\mathcal{G}$, let

$$T^i_{\mathrm{visit}}(r,p) = \begin{cases} \infty \;\; \text{if } r \notin \mathsf{Timediv} \text{ and } r \notin \mathsf{Blameless}_i; \\ \infty \;\; \text{if } r \in \mathsf{Timediv} \text{ and } r \text{ does not visit } p; \\ 0 \;\; \text{if } r \notin \mathsf{Timediv} \text{ and } r \in \mathsf{Blameless}_i; \\ \inf\{t \in \mathbb{R}_{\geq 0} \mid p \in \sigma(\mathsf{state}(r, \langle t,k\rangle)) \text{ for some } k\} \;\; \text{otherwise.} \end{cases}$$

Then, the *minimal time* for player 1 to force the game from a start state $s \in S$ to a p state is defined as

$$T^1_{\min}(s,p) = \inf_{\pi_1 \in \Pi_1} \sup_{\pi_2 \in \Pi_2} \{T^1_{\mathrm{visit}}(r,p) \mid r \in \mathsf{Outcomes}(s,\pi_1,\pi_2)\}.$$

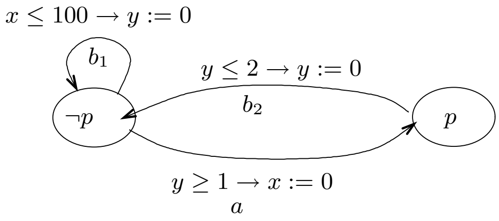
The minimal time $T^2_{\min}(s,p)$ for player 2 is defined symmetrically. As in Corollary 1, the minimal time remains the same when both players are restricted to use reasonable strategies.

Proposition 2. *Given a state s, a proposition p of a well-formed timed game structure $\mathcal{G}$, and $i \in \{1,2\}$, let $T^{i,*}_{\min}(s,p)$ be the minimal time for player i to force the game from s to a p state with both players restricted to use only reasonable strategies. Then, $T^{i,*}_{\min}(s,p) = T^i_{\min}(s,p)$.*

As an example consider the timed automaton game in Figure 1 with initial state $s_0 = \langle \neg p, (x=0, y=0)\rangle$. The action a belongs to player 1 and $b_j, j \in \{1,2\}$ to player 2. Not all runs in the game graph are non-zeno, in particular player 2 can keep take b_1 and keep player 1 from reaching p from s_0. However, it can be easily seen that physically, player 1 will be able to have p satisfied by time 101.

To solve the minimum-time problem, we consider a well-formed timed automaton game $\mathcal{T} = \langle L, \Sigma, \sigma, C, A_1, A_2, E, \gamma\rangle$.

Lemma 3. *For a state s and a target proposition $p \in \Sigma$ of a well-formed timed game automaton $\mathcal{T}$,*



$x \leq 100 \rightarrow y := 0$
b_1
$y \leq 2 \rightarrow y := 0$
b_2
$\neg p$
p
$y \geq 1 \rightarrow x := 0$
a

Fig. 1. A timed automaton game.

1. *Let $s \models_{\mathrm{td}} \langle\langle i \rangle\rangle \diamond p$. Then there exists $d < \infty$ such that $s \models_{\mathrm{td}} \langle\langle i \rangle\rangle \diamond_{\leq d}\, p$.*
2. *Let $s \models_{\mathrm{td}} \langle\langle i \rangle\rangle \diamond_{\leq d}\, p$. Then the minimal time for player i to reach p from state s is less than or equal to d, that is, $T^i_{\min}(s, p) \leq d$.*
3. *Let $s \not\models_{\mathrm{td}} \langle\langle i \rangle\rangle \diamond_{\leq d}\, p$. Then the minimal time for player i to reach p from state s is not less than d, that is, $T^i_{\min}(s, p) \geq d$.*

Proof. We prove the first claim for $i = 1$. We have that $s \models \langle\langle 1 \rangle\rangle \diamond p$, thus there is a player-1 strategy π_1 such that for all opposing strategies π_2 of player 2, and for all runs $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$ we have that, 1) if time diverges in run r then r contains a state satisfying p, and 2) if time does not diverge in r, then player 1 is blameless. Suppose that for all $d > 0$ we have $s \not\models_{\mathrm{td}} \langle\langle 1 \rangle\rangle \diamond_{\leq d}\, p$. We have that player 1 cannot win for his objective of $\diamond_{\leq d}\, p$, in particular, π_1 is not a winning strategy for this new objective. Hence, there is a player-2 strategy π_2^d such that for some run $r_d \in \mathsf{Outcomes}(s, \pi_1, \pi_2^d)$ either 1) time converges and player 1 is to blame or 2) time diverges in run r_d and r_d contains a location satisfying p, but not before time d. Player 1 does not have anything to gain by blocking time, so assume time diverges in run r_d (or equivalently, assume π_1 to be a reasonable strategy by Proposition 1). The only way strategies π_2^d and runs r_d can exist for every $d > 0$ is if player 2 can *force* the game (while avoiding p) so that a portion of the run lies in a region cycle $R_{k_1}, \ldots R_{k_m}$, with *tick* being true in one of the regions of the cycle (note that a system may stay in a region for at most one time unit). Now, if a player can control the game from state s so that the next state lies in region R, then he can do the same from any state s' such that $s' \cong s$. Thus, it must be that player 2 has a strategy π_2^* such that a run in $\mathsf{Outcomes}(s, \pi_1, \pi_2^*)$ corresponds to the region sequence $R_0, \ldots, R_k, (R_{k_1}, \ldots R_{k_m})^\omega$, with none of the regions satisfying p. Time diverges in this run as *tick* is infinitely often true due to the repeating region cycle. This contradicts the fact the π_1 was a winning strategy for player 1 for $\langle\langle 1 \rangle\rangle \diamond p$.

□

Lemma 3 suggests the following algorithm for computing the minimal time to reach p to within a precision of one: first confirm that $\langle\langle 1 \rangle\rangle \diamond p$ holds, then iteratively check whether $\langle\langle 1 \rangle\rangle \diamond_{\leq k}\, p$ holds for $k = 0, 1, 2, \ldots$. Any desired degree of precision can be achieved by the standard trick of "blowing" up the timed automaton. The algorithm is exponential both in the size of the timed automaton game and in the number of bits used to encode the desired precision.

Theorem 2. *Given a timed automaton game $\mathcal{T}$, a state s, and a target proposition p, the minimal time $T^i_{\min}(s, p)$ for player $i \in \{1, 2\}$ to force the game from s to a p state can be computed to within any desired degree of precision.*

The dual *maximal-time problem* asks what is the maximal time for which a player can ensure that the system stays within p states. Corresponding results hold for the maximum-time problem: for timed automaton games, it can be computed it to within any desired degree of precision in exponential time.

The problems of computing the exact minimal and maximal times for timed automaton games are open.

References

1. B. Adler, L. de Alfaro, and M. Faella. Average reward timed games. In *FORMATS 05*, LNCS 3829, pages 65–80. Springer, 2005.
2. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *ICALP 04*, LNCS 3142, pages 122–133. Springer, 2004.
3. R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking in dense real-time. *Inf. and Comp.*, 104(1):2–34, 1993.
4. R. Alur and D.L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
5. R. Alur and T.A. Henzinger. A really temporal logic. *Journal of the ACM*, 41:181–204, 1994.
6. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.
7. E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *HSCC 99*, LNCS 1569, pages 19–30. Springer, 1999.
8. P. Bouyer, F. Cassez, E. Fleury, and K.G. Larsen. Optimal strategies in priced timed game automata. In *FSTTCS 04*, LNCS 3328, pages 148–160. Springer, 2004.
9. P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *CAV 03*, LNCS 2725, pages 180–192. Springer, 2003.
10. T. Brihaye, V. Bruyère, and J.F. Raskin. On optimal timed strategies. In *FORMATS 05*, LNCS 3829, pages 49–64. Springer, 2005.
11. F. Cassez, A. David, E. Fleury, K.G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR 05*, LNCS 3653, pages 66–80. Springer, 2005.
12. C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4):385–415, 1992.
13. L. de Alfaro, M. Faella, T.A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *CONCUR 03*, LNCS 2761, pages 144–158. Springer, 2003.
14. D. D'Souza and P. Madhusudan. Timed control synthesis for external specifications. In *STACS 02*, LNCS 2285, pages 571–582. Springer, 2002.
15. M. Faella, S. La Torre, and A. Murano. Automata-theoretic decision of timed games. In *VMCAI 02*, LNCS 2294, pages 94–108. Springer, 2002.
16. M. Faella, S. La Torre, and A. Murano. Dense real-time games. In *LICS 02*, pages 167–176. IEEE Computer Society, 2002.
17. T.A. Henzinger and P.W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221:369–392, 1999.

18. T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111:193–244, 1994.
19. F. Laroussinie, N. Markey, and G. Oreiby. Model checking timed ATL for durational concurrent game structures. In *FORMATS 06*, LNCS. Springer, 2006.
20. O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems (an extended abstract). In *STACS 95*, pages 229–242, 1995.
21. A. Pnueli, E. Asarin, O. Maler, and J. Sifakis. Controller synthesis for timed automata. In *Proc. System Structure and Control*. Elsevier, 1998.
22. H. Wong-Toi and G. Hoffmann. The control of dense real-time discrete event systems. In *Proc. of 30th Conf. Decision and Control*, pages 1527–1528, 1991.