Deciding Safety and Liveness in TPTL

David Basin[a], Carlos Cotrini Jiménez[a,*], Felix Klaedtke[b,1], Eugen Zălinescu[a]

[a] *Institute of Information Security, ETH Zurich, Switzerland*
[b] *NEC Europe Ltd., Heidelberg, Germany*

Abstract

We show that deciding whether a TPTL formula describes a safety property is EXPSPACE-complete. Moreover, deciding whether a TPTL formula describes a liveness property is in 2-EXPSPACE. Our algorithms for deciding these problems extend those presented by Sistla [1] to decide the corresponding problems for LTL.

Keywords: temporal logic, safety and liveness, verification, complexity

1. Introduction

Safety and liveness [2, 3] are two important classes of system properties. A safety property claims that something "bad" never happens and a liveness property claims that something "good" can eventually happen. Identifying a system property as a safety or liveness property helps in finding a suitable method for its verification. For example, model checking can be improved when the system specification is known to be a safety property [4]. Also, when a property is safety, runtime-verification techniques are applicable [5].

Propositional linear-time temporal logic (LTL) [6] is one of the most popular logics used to specify properties of concurrent programs, but it has a limitation: its models abstract away from the actual times when the system events occur, retaining only their temporal order. To overcome this limitation, there have been different approaches extending LTL with explicit time (see [7] for a survey) for reasoning about hard real-time requirements like "every request must be processed within 5 time units." Among them, timed propositional temporal logic (TPTL) [8] in discrete-timed models achieves a good balance between decidability and expressiveness.

Sistla [1] proved that deciding whether an LTL formula describes a safety property is PSPACE-complete and that for liveness properties the problem is in EXPSPACE. However, analogous results for TPTL have not, until now, been given. In this article, we build upon Sistla's ideas to decide the corresponding problems for TPTL. We prove that deciding whether a TPTL formula describes a safety property is EXPSPACE-complete and that for liveness properties the problem is in 2-EXPSPACE. To the best of our knowledge, establishing tight lower bounds for deciding liveness in TPTL and LTL are open problems.

The remainder of this article is organized as follows. In Section 2, we give background and, in particular, we recall TPTL's syntax and semantics. In Section 3, we introduce quasimodels and quasicounterexamples for TPTL. These notions, suitably adapted from [8, 9], facilitate the proof of correctness of our decision algorithms. In Sections 4 and 5, we prove our complexity results and in Section 6 we draw conclusions.

2. Preliminaries

An *infinite sequence* over a set S is a function from $\mathbb{N}$ to S and a *finite sequence* over S of length ℓ is a function from $\{0, 1, \ldots, \ell - 1\}$ to S. For a finite sequence α and a sequence β, let $\alpha\beta$ denote their concatenation and let $|\alpha|$ denote α's length. The prefix of length $i \in \mathbb{N}$ of a sequence α is the sequence $\alpha^{<i} := \alpha(0)\alpha(1)\ldots\alpha(i-1)$, where we assume that $|\alpha| > i$. The sequences $\alpha^{\leq i}$, $\alpha^{>i}$, and $\alpha^{\geq i}$ are defined similarly. For a finite nonempty sequence α, let α^ω be the infinite sequence $\alpha\alpha\ldots$. For a sequence α over $\mathbb{N}$, let $\bar{\alpha}$ be the sequence defined by $\bar{\alpha}(i) := \sum_{0 \leq k \leq i} \alpha(k)$ and $\bar{\alpha}(i, j) := \sum_{i < k \leq j} \alpha(k)$, for $i, j \in \mathbb{N}$ with $i \leq j$.

2.1. TPTL

Syntax. Let P be a finite set of atomic propositions and V a countable set of variables, with $V \cap P = \emptyset$. The terms π and formulas φ of TPTL are defined by the grammar

$$\begin{array}{rcl} \pi & ::= & x + c \mid c \\ \varphi & ::= & \mathsf{false} \mid p \mid \pi_1 \leq \pi_2 \mid \pi_1 \equiv_m \pi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \\ & & \bigcirc \varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \mid x.\varphi\,, \end{array}$$

where x, c, p, and m range over V, $\mathbb{N}$, P, and $\mathbb{N} \setminus \{0\}$, respectively. We abbreviate $x + 0$ by x. For a formula φ, we write $\neg\varphi$ for $\varphi \rightarrow \mathsf{false}$ and true for $\neg\mathsf{false}$. The syntactic sugar for the Boolean connectives $\wedge$ and $\vee$ is as expected. We let $\Diamond\,\psi := \mathsf{true}\,\mathsf{U}\,\psi$ and $\Box\,\psi := \neg\Diamond\neg\psi$. All occurrences of a variable x in a formula of the form $x.\psi$ are said to be *bound* by $x.\psi$. An occurrence of x in φ that is not bound by any subformula $x.\psi$ of φ is *free*. We denote with

*Corresponding author.
[1] This work was partly done when the author was at ETH Zurich.

$\varphi[x \mapsto z]$ the formula obtained from φ by replacing all free occurrences of $x \in V$ with $z \in V$. Finally, we write $\pi_1 \sim \pi_2$ to denote any formula of the form $\pi_1 \leq \pi_2$ or $\pi_1 \equiv_m \pi_2$, with π_1 and π_2 terms and $m \geq 1$. We call $\pi_1 \sim \pi_2$ a *time constraint*.

Let n_φ be the number of connectives in φ. Also, let

$$k_\varphi := 2 \cdot \big(\textstyle\prod_c (1+c)\big) \cdot \big(\textstyle\prod_m m\big),$$

where c ranges over the constants occurring in formulas of the form $\pi_1 \leq \pi_2$ in φ, with π_1 and π_2 terms, and m ranges over the constants such that $\equiv_m$ occurs in φ. When there are no constants in φ, we define $k_\varphi := 2$. We define the *length* of a formula as the number of symbols needed to write the formula, assuming that a binary encoding is used to represent constants and to enumerate variables. Note that the length of a formula φ is linear in $n_\varphi \log n_\varphi + \log k_\varphi$.

Semantics. Let $\Sigma = 2^P \times \mathbb{N}$ and let Σ^* and Σ^ω be the sets of all finite and infinite sequences over Σ respectively. We usually write a sequence

$$\big(\sigma(0), \delta(0)\big)\ \big(\sigma(1), \delta(1)\big)\ \ldots \in \Sigma^* \cup \Sigma^\omega$$

as $\sigma \otimes \delta$, where σ and δ are sequences over 2^P and $\mathbb{N}$, respectively. TPTL formulas are interpreted over *timed words*. A timed word is an infinite sequence $\sigma \otimes \delta \in \Sigma^\omega$ such that $\delta(i) > 0$, for infinitely many i. A timed word $\sigma \otimes \delta$ is *k-bounded*, for $k \in \mathbb{N}$, if $\delta(i) \leq k$, for all $i \in \mathbb{N}$.

Note that Alur and Henzinger [8] define timed words differently. There, a timed word is an infinite sequence $\sigma \otimes \tau \in \Sigma^\omega$ such that τ is non-decreasing and for all $i \in \mathbb{N}$, there is $j > i$ such that $\tau(j) > \tau(i)$. However, the sets of timed words of both definitions are essentially the same: We can map a timed word $\sigma \otimes \delta$ under our definition to the timed word $\sigma \otimes \bar{\delta}$ under their definition. Intuitively, for $i > 0$, $\delta(i)$ indicates the time elapsed between the events $\sigma(i-1)$ and $\sigma(i)$ and $\bar{\delta}(i)$ indicates the time when the event $\sigma(i)$ takes place.

A *valuation* is a mapping from V to $\mathbb{N}$. We extend valuations to terms in the usual way. For a timed word $\sigma \otimes \delta$, a formula ψ, a valuation v, and $i \in \mathbb{N}$, we define satisfaction, written $\sigma \otimes \delta, v, i \vDash \psi$, by induction on the structure of ψ.

$\sigma \otimes \delta, v, i \nvDash \mathsf{false}$	
$\sigma \otimes \delta, v, i \vDash p$	iff $p \in \sigma(i)$
$\sigma \otimes \delta, v, i \vDash \pi_1 \leq \pi_2$	iff $v(\pi_1) \leq v(\pi_2)$
$\sigma \otimes \delta, v, i \vDash \pi_1 \equiv_m \pi_2$	iff $v(\pi_1) \equiv_m v(\pi_2)$
$\sigma \otimes \delta, v, i \vDash \psi_1 \to \psi_2$	iff $\sigma \otimes \delta, v, i \nvDash \psi_1$ or $\sigma \otimes \delta, v, i \vDash \psi_2$
$\sigma \otimes \delta, v, i \vDash \bigcirc \psi$	iff $\sigma \otimes \delta, v, i+1 \vDash \psi$
$\sigma \otimes \delta, v, i \vDash \psi_1 \mathsf{U} \psi_2$	iff there is $j \geq i$ with $\sigma \otimes \delta, v, j \vDash \psi_2$ and $\sigma \otimes \delta, v, k \vDash \psi_1$, for all k with $i \leq k < j$
$\sigma \otimes \delta, v, i \vDash x.\psi$	iff $\sigma \otimes \delta, v[x \mapsto \bar{\delta}(i)], i \vDash \psi$

Here $v[x \mapsto \bar{\delta}(i)]$ is the valuation obtained from v by setting $v(x)$ to $\bar{\delta}(i)$. We say that $\sigma \otimes \delta$ satisfies a sentence φ (i.e. a formula without free variables) if $\sigma \otimes \delta, v, 0 \vDash \varphi$, for any valuation v.

2.2. Safety and liveness

A timed word τ *refutes the safety* of a sentence φ if τ does not satisfy φ and for every $i \in \mathbb{N}$, there is a sequence $\tau' \in \Sigma^\omega$ such that $\tau^{<i}\tau'$ satisfies φ. The sentence φ is *safe*—or *describes a safety property*—if there is no timed word refuting φ's safety [2, 3].

A sequence τ in Σ^* is a *good prefix for* φ if there is $\tau' \in \Sigma^\omega$ such that $\tau\tau'$ is a timed word that satisfies φ. The sentence φ *describes a liveness property* if every sequence in Σ^* is a good prefix for φ [2, 3].

2.3. Additional notions and machinery

Time-constraint normal form. Following [8], we show that we can restrict our attention to sentences of a certain form. Let φ be a sentence and z a variable not occurring in φ. The sentence $\tilde{\varphi}$ is obtained from φ by replacing every variable-free term c with $z + c$ and then performing the necessary arithmetic manipulations to leave any time constraint in the form $x + c \sim y$ or $x \sim y + c$ with $x, y \in V$ and $c \in \mathbb{N}$.

The following lemma follows from the observation that a timed word $\sigma \otimes \delta$ satisfies φ iff $\emptyset\sigma \otimes 0\delta$ satisfies $z. \bigcirc \tilde{\varphi}$.

Lemma 1. A sentence φ describes a safety property iff $z. \bigcirc \tilde{\varphi}$ does and φ describes a liveness property iff $z. \bigcirc \tilde{\varphi}$ does.

For the rest of the article, we assume without loss of generality that $z.\varphi$ is a sentence where every time constraint in φ is of the form $x + c \sim y$ or $x \sim y + c$, with $x, y \in V$ and $c \in \mathbb{N}$.

Updating time constraints. A key observation underlying the algorithm for deciding satisfiability in TPTL presented by Alur and Henzinger [8] is that every formula can be split into a present and a future condition. Note that $\sigma \otimes \delta, v, i \vDash \Diamond q$ iff $\sigma \otimes \delta, v, i \vDash q$ or $\sigma \otimes \delta, v, i+1 \vDash \Diamond q$. One must be careful when time constraints occur in the formula. For example, consider the expression $\sigma \otimes \delta, v, i \vDash z. \Diamond y. (y \leq z + 5 \wedge q)$. Note that z refers to the current time. This expression can be satisfied by having $\sigma \otimes \delta, v, i \vDash z. (z \leq z + 5 \wedge q)$ in the current state or $\sigma \otimes \delta, v, i+1 \vDash z. \Diamond y. (y \leq (z - \delta(i+1)) + 5 \wedge q)$ in the next state. Note that we updated the time constraint as the current time has changed by $\delta(i+1)$.

We recall some notation from [8] for updating time constraints. For a formula of the form $z.\psi$ and $d \in \mathbb{N}$, let $z.\psi^d$ be the formula obtained by replacing every occurrence of z in ψ with $z - d$. Formally, $z.\psi^d$ is defined inductively as follows.

- $z.\psi^0$ is $z.\psi$.
- $z.\psi^{d+1}$ results from $z.\psi^d$ by replacing every term of the form $z + (c+1)$ with $z + c$, and every subformula of the form $z \leq y + c$, $y + c \leq z$, and $z \equiv_m y + c$ with true, false, and $z \equiv_m y + ((c+1) \bmod m)$, respectively.

For example, let $z.\varphi = z.\Diamond y.(y \leq z + 5 \wedge q)$. Then $z.\varphi^2$, $z.\varphi^5$ and $z.\varphi^6$ are $z.\Diamond y.(y \leq z + 3 \wedge q)$, $z.\Diamond y.(y \leq z \wedge q)$, and $z.\Diamond y.(\mathsf{false} \wedge q)$, respectively.

The next lemma, from [8], shows that $z.\psi^d$ correctly denotes the formula $z.\psi$ after replacing every free occurrence of z with $z - d$. It is proved by induction on ψ's structure.

Lemma 2. For every formula $z.\psi$ and every $d \leq \bar{\delta}(i)$, we have $\sigma \otimes \delta, v, i \vDash z.\psi^d$ iff $\sigma \otimes \delta, v[z \mapsto \bar{\delta}(i) - d], i \vDash \psi$.

The closure of a formula. The algorithm of Alur and Henzinger follows the tableau method. A tableau for a formula $z.\varphi$ is built from a set $Cl(z.\varphi)$ of sentences called the *closure* of $z.\varphi$ [8]. The closure of $z.\varphi$ is the smallest set that contains $z.\varphi$ and is closed under the operation Sub, which is defined as:

- $Sub(z.\psi) := \{z.\psi\}$, if ψ is an atomic formula,
- $Sub(z.(\psi_1 \to \psi_2)) := \{z.\psi_1, z.\psi_2\}$,
- $Sub(z.\bigcirc\psi) := \{z.\psi^d \mid d \in \mathbb{N}\}$,
- $Sub(z.(\psi_1 \mathsf{U} \psi_2)) := \{z.\psi_1, z.\psi_2, z.\bigcirc(\psi_1 \mathsf{U} \psi_2)\}$, and
- $Sub(z.x.\psi) := \{z.\psi[x \mapsto z]\}$.

For example, for $z.\varphi = z.(p \mathsf{U} y.(y \leq z + 5))$, $Cl(z.\varphi)$ contains: $z.\mathsf{false}$, $z.y.\mathsf{false}$, $z.(p \mathsf{U} y.\mathsf{false})$, $z.\bigcirc(p \mathsf{U} y.\mathsf{false})$, $z.p$, $z.(z \leq z + i)$, $z.y.(y \leq z + i)$, $z.(p \mathsf{U} y.(y \leq z + i))$, and $z.\bigcirc(p \mathsf{U} y.(y \leq z + i))$, for $i \leq 5$.

Note that for any $z.\varphi$, $Cl(z.\varphi)$ only contains sentences. In particular, z is the only variable that occurs in any formula of the form $z.(\pi_1 \sim \pi_2) \in Cl(z.\varphi)$.

Avoiding valuations. The following lemma shows that when evaluating a sentence $z.\psi$ in $Cl(z.\varphi)$ at a position in a timed word, one need not consider valuations.

Lemma 3. Let $z.\psi$ be a sentence in $Cl(z.\varphi)$. For a timed word $\sigma \otimes \delta$, valuation v, and $i \in \mathbb{N}$, we have the following according to the form of $z.\psi$:

1. $\sigma \otimes \delta, v, i \nvDash z.\mathsf{false}$,
2. $\sigma \otimes \delta, v, i \vDash z.p$ iff $p \in \sigma(i)$, for $p \in P$,
3. $\sigma \otimes \delta, v, i \vDash z.z \sim z + c$ iff $0 \sim c$ and $\sigma \otimes \delta, v, i \vDash z.z + c \sim z$ iff $c \sim 0$, for $c \in \mathbb{N}$,
4. $\sigma \otimes \delta, v, i \vDash z.(\psi_1 \to \psi_2)$ iff $\sigma \otimes \delta, v, i \nvDash z.\psi_1$ or $\sigma \otimes \delta, v, i \vDash z.\psi_2$,
5. $\sigma \otimes \delta, v, i \vDash z.\bigcirc\psi$ iff $\sigma \otimes \delta, v, i + 1 \vDash z.\psi^{\delta(i+1)}$,
6. $\sigma \otimes \delta, v, i \vDash z.(\psi_1 \mathsf{U} \psi_2)$ iff (a) $\sigma \otimes \delta, v, i \vDash z.\psi_2$ or (b) $\sigma \otimes \delta, v, i \vDash z.\psi_1$ and $\sigma \otimes \delta, v, i \vDash z.\bigcirc(\psi_1 \mathsf{U} \psi_2)$, and
7. $\sigma \otimes \delta, v, i \vDash z.x.\psi$ iff $\sigma \otimes \delta, v, i \vDash z.\psi[x \mapsto z]$.

Proof. Use the following well-founded induction schema. First, prove the claim for all sentences in $Cl(z.\varphi)$ of the form $z.z \leq z + c$, $z.z + c \leq z$, $z.\mathsf{false}$, and $z.p$, for $p \in P$ and $c \in \mathbb{N}$. Then prove the claim for a sentence $z.\psi \in Cl(z.\varphi)$, assuming it holds for all the formulas in $Sub(z.\psi)$. For item 5, use Lemma 2. □

From Lemma 3 we immediately obtain the following.

Lemma 4. For a timed word $\sigma \otimes \delta$, $i \in \mathbb{N}$ and two valuations v_1 and v_2, we have that $\sigma \otimes \delta, v_1, i \vDash z.\varphi$ iff $\sigma \otimes \delta, v_2, i \vDash z.\varphi$.

Valuations are therefore no longer necessary. Consider, for example, the formula $z.\varphi = z.\Diamond y.(y \leq z + 5 \wedge q)$ and the timed word $\sigma \otimes \delta$ with $\sigma = (\emptyset, \emptyset, \{q\}, \ldots)$ and $\delta = (0, 2, 7, \ldots)$. Using Lemma 3, checking whether $\sigma \otimes \delta, v, 0 \vDash z.\varphi$ reduces to checking if any of the following holds:

1. $\sigma \otimes \delta, v, 0 \vDash z.(z \leq z + 5)$ and $\sigma \otimes \delta, v, 0 \vDash q$,
2. $\sigma \otimes \delta, v, 1 \vDash z.(z \leq z + 3)$ and $\sigma \otimes \delta, v, 1 \vDash q$, or
3. $\sigma \otimes \delta, v, i \vDash z.\mathsf{false}$ and $\sigma \otimes \delta, v, i \vDash q$, for any $i \geq 2$.

Note that we do not need to store any time differences in v. We can therefore drop v and, from now on, we write $\sigma \otimes \delta, i \vDash z.\psi$ instead of $\sigma \otimes \delta, v, i \vDash z.\psi$.

Finite character of time. In the remainder of this section, we recall some other results from [8]. TPTL cannot distinguish between too large changes in time: for $d \geq k_\varphi$, we have $z.\varphi^d = z.\varphi^{d'}$, for some $d' < k_\varphi$, where k_φ is the value defined in Section 2.1. This observation is used to prove Lemma 7, which states that if $z.\varphi$ is satisfiable, then there is a k_φ-bounded timed word satisfying $z.\varphi$.

Let c_φ be 1 plus the largest constant that occurs in a formula of the form $\pi_1 \leq \pi_2$ in $z.\varphi$, with π_1 and π_2 terms, and let m_φ be the least common multiple of all constants m such that $\equiv_m$ appears in $z.\varphi$. When there are no such constants in $z.\varphi$, we let $c_\varphi := 1$ and $m_\varphi := 1$, respectively. For $d \in \mathbb{N}$, we define

$$\hat{d} := \begin{cases} c_\varphi \cdot m_\varphi + (d \bmod m_\varphi) & \text{if } d \geq c_\varphi \cdot m_\varphi, \\ d & \text{otherwise.} \end{cases}$$

Note that $\hat{d} < k_\varphi$, for any $d \in \mathbb{N}$.

Lemma 5. For a subformula $z.\psi \in Cl(z.\varphi)$ and $d \in \mathbb{N}$, we have $z.\psi^d = z.\psi^{\hat{d}}$.

Proof. The claim obviously holds if $d < c_\varphi \cdot m_\varphi$. Suppose that $d \geq c_\varphi \cdot m_\varphi$. By the definition of $z.\psi^d$, the only parts of $z.\psi$ affected are the subformulas of the form $\pi_1 \sim \pi_2$, with π_1, π_2 terms, and z occurring in the subformula. We distinguish the following cases based on the form of these subformulas, where $y \in V$ and $c \in \mathbb{N}$.

- $z + c \leq y$ or $z \leq y + c$. Both formulas become true in $z.\psi^e$, for any $e \geq c+1$ for the first one, and for any $e \geq 1$ for the second one. Note that $d > c$ and $\hat{d} > c$.

– $y + c \leq z$ or $y \leq z + c$. Here both formulas become false.

– $z + c \equiv_m y$ or $z \equiv_m y + c$. The two cases are similar, so we consider only the second one. Here ψ^d equals

$$z \equiv_m y + ((d + c) \bmod m)$$

and $\psi^{\hat{d}}$ is

$$z \equiv_m y + [(c_\varphi \cdot m_\varphi + (d \bmod m_\varphi) + c) \bmod m] .$$

If we simplify the last expression, we obtain

$$\begin{aligned}(c_\varphi \cdot m_\varphi &+ (d \bmod m_\varphi) + c) \bmod m \\ &= ((d \bmod m_\varphi) + c) \bmod m = (d + c) \bmod m,\end{aligned}$$

where the last equality follows from $(d \bmod m_\varphi) \bmod m = d \bmod m$. □

Lemma 6. Let $d_i \in \mathbb{N}$, for $1 \leq i \leq k$. Let Δ_k and $\hat{\Delta}_k$ be $d_1 + d_2 + \ldots + d_k$ and $\hat{d}_1 + \hat{d}_2 + \ldots + \hat{d}_k$, respectively. Then $z.\psi^{\Delta_k} = z.\psi^{\hat{\Delta}_k}$, for any subformula $z.\psi \in Cl\,(z.\varphi)$.

Proof. By induction on k. Note that $z.\psi^{\hat{\Delta}_k + \hat{d}_{k+1}}$ can be obtained by first computing $z.\psi^{\hat{\Delta}_k}$ and then computing from that $\left(z.\psi^{\hat{\Delta}_k}\right)^{\hat{d}_{k+1}}$. By the induction hypothesis, $\left(z.\psi^{\hat{\Delta}_k}\right)^{\hat{d}_{k+1}} = \left(z.\psi^{\Delta_k}\right)^{\hat{d}_{k+1}}$ and by Lemma 5, $\left(z.\psi^{\Delta_k}\right)^{\hat{d}_{k+1}} = \left(z.\psi^{\Delta_k}\right)^{d_{k+1}}$. Finally, $\left(z.\psi^{\Delta_k}\right)^{d_{k+1}} = z.\psi^{\Delta_k + d_{k+1}}$. Therefore, $z.\psi^{\hat{\Delta}_k + \hat{d}_k} = z.\psi^{\Delta_k + d_k}$. □

Lemma 7. Let $\sigma \otimes \delta$ be a timed word and let $\hat{\delta}$ be the sequence defined by $\hat{\delta}(i) := \widehat{\delta(i)}$, for $i \in \mathbb{N}$. Then $\sigma \otimes \hat{\delta}$ is a k_φ-bounded timed word that satisfies $z.\varphi$ iff $\sigma \otimes \delta$ satisfies $z.\varphi$.

Proof. Prove that $\sigma \otimes \hat{\delta}, 0 \vDash z.\psi$ iff $\sigma \otimes \delta, 0 \vDash z.\psi$, for all $z.\psi \in Cl(z.\varphi)$. For this, use the well-founded induction schema presented in the proof of Lemma 3. □

3. Quasimodels and quasicounterexamples

Our algorithm for deciding whether a TPTL sentence is safe is inspired by the algorithm presented in [1] for LTL, which, in turn, is based on an algorithm for deciding satisfiability in LTL [10]. We recall briefly how they work.

LTL models are infinite sequences over the alphabet 2^P. A model is *regular* if it has the form $\alpha\beta^\omega$, for some finite nonempty sequences α and β over 2^P. An LTL formula ψ is satisfiable iff there is a regular model that satisfies ψ. To decide whether an LTL formula ψ is satisfiable, the algorithm non-deterministically guesses two finite sequences f_1 and f_2 of sets of subformulas of ψ. The formula ψ is satisfiable iff there is a regular model $\alpha\beta^\omega$ such that the sequences f_1 and f_2 satisfy the following: for $i < |f_1|$, the set $f_1(i)$ contains exactly all the subformulas of ψ satisfied by $\alpha^{\geq i}\beta^\omega$ and for $j < |f_2|$, the set $f_2(j)$ contains exactly all the subformulas of ψ satisfied by $\beta^{\geq j}\beta^\omega$. In particular, $f_1(i)$ contains $\alpha(i)$ and $f_2(j)$ contains $\beta(j)$, for all $i < |f_1|$ and $j < |f_2|$. The sequence $f_1 f_2^\omega$ provides all the information needed to build α and β. Moreover, it contains evidence that $\alpha\beta^\omega$ satisfies ψ. The sequence $f_1 f_2^\omega$ is called a *quasimodel* for ψ. In general, a quasimodel for an LTL formula ψ is a sequence f of sets of subformulas of ψ for which there is a model γ that satisfies ψ and such that $f(i)$ contains all the subformulas of ψ satisfied by $\gamma^{\geq i}$. The elements of a quasimodel are called *quasistates* for ψ, which are maximal consistent sets of subformulas of ψ.

The algorithm for checking whether an LTL formula describes a safety property is similar but more involved. It non-deterministically guesses a representation of a *quasi-counterexample*, which consist of quasimodels $f, g_0, g_1, \ldots$, witnessing that the formula φ is not safe. In particular, f is a quasimodel for $\neg\varphi$ and $f^{<i}g_i$ is a quasimodel for φ, for every $i \in \mathbb{N}$.

These observations carry over from LTL to TPTL, with some modifications. The algorithms for satisfiability and safety work in the same way and analogous regularity properties hold for TPTL. We adapt the notions of quasistate, quasimodel, and quasicounterexample for TPTL in the Sections 3.1, 3.2, and 3.3, respectively. Quasistates and quasimodels were already adapted to TPTL in [8]—with different names though—and we recall them for the sake of completeness. Note that these notions are implicit in [10, 1] for LTL. Quasistates and quasimodels were introduced in [9] to simplify the correctness proofs for decision algorithms of some fragments of first-order temporal logic.

3.1. Quasistates

Definition 1. A *quasistate* for $z.\varphi$ is a pair (Φ, d), where $d \in \mathbb{N}$ and Φ is a maximally consistent subset of $Cl(z.\varphi)$, that is, Φ must satisfy the following conditions.

– $z.\mathsf{false} \notin \Phi$.

– $z.\,(z \sim z + c) \in \Phi$ iff $0 \sim c$, for every $z.\,(z \sim z + c) \in Cl(z.\varphi)$, and $z.\,(z + c \sim z) \in \Phi$ iff $c \sim 0$, for every $z.\,(z + c \sim z) \in Cl(z.\varphi)$.

– $z.(\psi_1 \to \psi_2) \in \Phi$ iff $z.\psi_1 \notin \Phi$ or $z.\psi_2 \in \Phi$, for every $z.(\psi_1 \to \psi_2) \in Cl(z.\varphi)$.

– $z.(\psi_1 \mathsf{U} \psi_2) \in \Phi$ iff (i) $z.\psi_2 \in \Phi$ or (ii) $z.\psi_1 \in \Phi$ and $z.\bigcirc(\psi_1 \mathsf{U} \psi_2) \in \Phi$, for every $z.(\psi_1 \mathsf{U} \psi_2) \in Cl(z.\varphi)$.

– $z.x.\psi \in \Phi$ iff $z.\psi[x \mapsto z] \in \Phi$, for every $z.x.\psi \in Cl(z.\varphi)$.

For $k \in \mathbb{N}$, we say a quasistate (Φ, d) is *k-bounded* if $d \leq k$ and we denote with $\sharp(z.\varphi)$ the number of k_φ-bounded quasistates for $z.\varphi$. □

By Lemma 5, the set $Sub(z.\bigcirc\psi)$ is finite, which implies that $Cl(z.\varphi)$ is finite. In particular, the size of $Cl(z.\varphi)$ is at most $n_\varphi k_\varphi$ [8]. Hence we have that

$$\sharp(z.\varphi) \leq 2^{n_\varphi k_\varphi} \cdot k_\varphi < 2^{(n_\varphi + 1)k_\varphi}.$$

In the following, we abuse notation and write $\psi \in (\Phi, d)$ to indicate that $\psi \in \Phi$, for a quasistate (Φ, d).

3.2. Quasimodels

Let f be a sequence of quasistates for $z.\varphi$ with $f(i) = (\Phi_i, d_i)$, for $i \in \mathbb{N}$, and let δ be the sequence defined by $\delta(i) := d_i$, for $i \in \mathbb{N}$. Recall that $\bar{\delta}(i,j) := \sum_{i<k\leq j} \delta(k)$. Suppose that $z.(\psi_1 \mathsf{U} \psi_2)$ occurs in Φ_i. Then we say that f *realizes the occurrence of* $z.(\psi_1 \mathsf{U} \psi_2)$ *in* Φ_i if there is $j \geq i$ such that $z.\psi_2^{\bar{\delta}(i,j)} \in \Phi_j$. When the set Φ_i is clear from the context, we say instead that f *realizes* $z.(\psi_1 \mathsf{U} \psi_2)$.

For (Φ, d) and (Φ', d') two quasistates for $z.\varphi$, we say that (Φ', d') is a *successor of* (Φ, d) if, for any $z.\bigcirc\psi \in Cl(z.\varphi)$, it holds that $z.\bigcirc\psi \in \Phi$ iff $z.\psi^{d'} \in \Phi'$.

Definition 2. A *quasimodel* for $z.\varphi$ is an infinite sequence f of quasistates for $z.\varphi$ with $f(i) = (\Phi_i, d_i)$ such that:

(QM-1) $d_i > 0$, for infinitely many i,

(QM-2) $z.\varphi \in f(0)$,

(QM-3) $f(i+1)$ is a successor of $f(i)$, for all $i \in \mathbb{N}$, and

(QM-4) any occurrence of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in f is realized by f.

The quasimodel is *k-bounded* if $d_i \leq k$, for all $i \in \mathbb{N}$. □

The proofs of the following two results are simple extensions of those presented in [11, 8]. They show a one-to-one correspondence between timed words satisfying $z.\varphi$ and quasimodels for $z.\varphi$.

Theorem 1.

1. Let $\sigma \otimes \delta$ be a timed word that satisfies $z.\varphi$ and let $f_{\sigma\otimes\delta}$ be the sequence defined by $f_{\sigma\otimes\delta}(i) := (\Phi_i, \delta(i))$ with
$$\Phi_i := \{z.\psi \in Cl(z.\varphi) \mid \sigma \otimes \delta, i \vDash z.\psi\}.$$
The sequence $f_{\sigma\otimes\delta}$ is a quasimodel for $z.\varphi$.

2. Let f be a quasimodel for $z.\varphi$ with $f(i) = (\Phi_i, d_i)$ and let $\sigma_f \otimes \delta_f$ be the pair of sequences defined by
$$\sigma_f(i) := \{p \in P \mid z.p \in \Phi_i\}$$
and $\delta_f(i) := d_i$, for any $i \in \mathbb{N}$. The pair $\sigma_f \otimes \delta_f$ is a timed word that satisfies $z.\varphi$.

Recall that for $d \in \mathbb{N}$, $\hat{d}$ is defined as $c_\varphi \cdot m_\varphi + (d \bmod m_\varphi)$ if $d \geq c_\varphi \cdot m_\varphi$ and $\hat{d} = d$, otherwise.

Theorem 2. Let f be an infinite sequence of quasistates for $z.\varphi$ with $f(i) = (\Phi_i, d_i)$ and let $\hat{f}$ be the infinite sequence defined as $\hat{f}(i) = (\Phi_i, \hat{d}_i)$. Then f is a quasimodel for $z.\varphi$ iff $\hat{f}$ is a k_φ-bounded quasimodel for $z.\varphi$.

Proof. We prove just the "only if" direction. The "if" direction is proved similarly. Requirements (QM-1) and (QM-2) are clear. For (QM-3) and (QM-4), use Lemmas 5 and 6. Finally, recall that $\hat{d}_i < k_\varphi$. Hence $\hat{f}$ is a k_φ-bounded quasimodel for $z.\varphi$. □

Lemma 8. Let f be a quasimodel for $z.\varphi$. If there are $i, j \in \mathbb{N}$ such that $i \leq j$ and $f(i) = f(j)$, then $f' = f^{\leq i} f^{>j}$ is also a quasimodel for $z.\varphi$.

Proof. We adapt the proof in [9] to TPTL. Let $f(i) = (\Phi_i, d_i)$, for $i \in \mathbb{N}$ and let δ be the sequence $\delta(i) := d_i$, for $i \in \mathbb{N}$. (QM-1) and (QM-2) clearly hold for f'. To check (QM-3), note that $z.\bigcirc\psi \in f(i)$ iff $z.\bigcirc\psi \in f(j)$ iff $z.\psi^{\delta_{j+1}} \in f(j+1)$. We check (QM-4) as follows. Let $z.(\psi_1 \mathsf{U} \psi_2) \in f(m)$ for some m. If $m > j$ then clearly $z.(\psi_1 \mathsf{U} \psi_2)$ is realized by f'. Suppose then $m \leq i$. If $z.\psi_2^{\bar{\delta}(m,\ell)} \in f(\ell)$ for some $\ell \leq i$, then we are done; otherwise, $z.(\psi_1 \mathsf{U} \psi_2)^{\bar{\delta}(m,i)}$ must occur in $f(i)$. It follows that $z.(\psi_1 \mathsf{U} \psi_2)^{\bar{\delta}(m,i)} \in f(j)$, and since f is a quasimodel for $z.\varphi$, the occurrence of $z.(\psi_1 \mathsf{U} \psi_2)^{\bar{\delta}(m,i)}$ in $f(j)$ is realized by $f^{\geq j}$. Hence $z.(\psi_1 \mathsf{U} \psi_2)$ is realized by f'. □

To decide whether there is a quasimodel for $z.\varphi$, the following lemma from [8] shows that we only need to find two particular finite sequences of quasistates.

Lemma 9. There is a quasimodel for $z.\varphi$ iff there are sequences f_1 and f_2 of k_φ-bounded quasistates for $z.\varphi$ such that:

1. $|f_1| \leq \sharp(z.\varphi)$ and $|f_2| \leq (|Cl(z.\varphi) + 2|) \cdot \sharp(z.\varphi)$,
2. $z.\varphi \in f_1(0)$,
3. $d > 0$ for some (Φ, d) in f_2,
4. $f_j(i+1)$ is a successor of $f_j(i)$ for $i < |f_j| - 1$ and $j \in \{1, 2\}$,
5. $f_2(0)$ is a successor of the last quasistates of f_1 and f_2, and
6. every occurrence of a formula of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f_2(0)$ is realized by f_2.

Proof. The proof of an analogous lemma in [9] applies here as well. For the "if" direction, note that $f_1 f_2^\omega$ is a quasimodel for $z.\varphi$. We prove the "only if" direction, where we assume that f is a quasimodel for $z.\varphi$ with $f(i) = (\Phi_i, d_i)$. By Theorem 2, we assume f is k_φ-bounded.

Take s such that $f(s) = f(i)$, for infinitely many $i > s$. Apply Lemma 8 whenever $i_1 < i_2 < s$ and $f(i_1) = f(i_2)$. This yields a quasimodel $f_1 f^{\geq s}$ with $|f_1| \leq \sharp(z.\varphi)$.

We now explain how to get f_2. Suppose there is a formula of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f^{\geq s}(0)$. Take $k \geq 0$ such that $z.\psi_2^{\bar{\delta}(s,s+k)} \in f^{\geq s}(k)$, where δ is the sequence defined by $\delta(i) := d_i$, for $i \in \mathbb{N}$. Apply Lemma 8 whenever $i_1 < i_2 \leq k$ and $f^{\geq s}(i_1) = f^{\geq s}(i_2)$. This yields the quasimodel $f_1 f^{\geq s}(0) f' f^{>s'}$, where $s' := s + k$. Note that $f^{\geq s}(0) f'$ has length at most $\sharp(z.\varphi)$ and realizes the occurrence of $z.(\psi_1 \mathsf{U} \psi_2) \in f^{\geq s}(0)$. Suppose there is another formula in $f^{\geq s}(0)$ of the form $z.(\psi_1' \mathsf{U} \psi_2')$. If f' realizes $z.(\psi_1' \mathsf{U} \psi_2')$ then do nothing; otherwise, take k' such that

$z.\psi_2'^{\Delta} \in f^{>s'}(k')$. Here Δ is the sum of all values of d such that (Φ, d) is a quasistate in $f'\, f^{>s'}(0)\, f^{>s'}(1) \ldots f^{>s'}(k')$. Then apply Lemma 8 to remove repeated quasistates in $f^{>s'}(0)\, f^{>s'}(1) \ldots f^{>s'}(k')$. As a result, we get a quasimodel $f_1\, f^{\geq s}(0)\, f''\, f^{>s''}$, where $s'' := s' + k'$. Note that $f^{\geq s}(0)\, f''$ realizes both occurrences of $z.(\psi_1 \mathsf{U} \psi_2)$ and $z.(\psi_1' \mathsf{U} \psi_2')$ in $f^{\geq s}(0)$ and has length at most $2 \cdot \sharp(z.\varphi)$. Continue in this way for any other formula in $f^{\geq s}(0)$ of the form $z.(\psi_1'' \mathsf{U} \psi_2'')$. After this, we obtain a quasimodel $f_1\, f^{\geq s}(0)\, f^{\circ}\, f^{>k^{\circ}}$, where $f^{\geq s}(0) f^{\circ}$ realizes any occurrence in $f^{\geq s}(0)$ of the form $z.(\psi_1 \mathsf{U} \psi_2)$ and has length at most $|Cl(z.\varphi)| \cdot \sharp(z.\varphi)$.

Now, take t such that $f^{>k^{\circ}}(t)$ has the form (Φ, d) with $d > 0$ and use Lemma 8 to remove repeated quasistates in $f^{>k^{\circ}}(0) f^{>k^{\circ}}(1) \ldots f^{>k^{\circ}}(t)$. This yields a quasimodel $f_1\, f^{\geq s}(0)\, f^{\circ\circ}\, f^{>k^{\circ\circ}}$ where $k^{\circ\circ} := k^{\circ} + t$ and $f^{\geq s}(0)\, f^{\circ\circ}$ not only realizes all occurrences of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f^{\geq s}(0)$, but also contains a quasistate (Φ, d) with $d > 0$. Finally, take t' such that $f^{>k^{\circ\circ}}(t') = f^{\geq s}(0)$. Such a quasistate $f^{>k^{\circ\circ}}(t')$ exists because we chose $f^{\geq s}(0) = f(s)$ as a quasistate that occurs infinitely often in f. Use Lemma 8 to obtain a quasimodel $f_1\, f^{\geq s}(0)\, f^{\circ\circ\circ}\, f^{\geq k^{\circ\circ\circ}}$, where $k^{\circ\circ\circ} := k^{\circ\circ} + t'$. We define f_2 as the sequence $f^{\geq s}(0) f^{\circ\circ\circ}$. It is easy to check that f_1 and f_2 meet all the requirements. □

3.3. Quasicounterexamples

We now define quasicounterexamples for TPTL. Suppose $z.\varphi$ is not safe. Then there is a timed word τ that does not satisfy $z.\varphi$ and is such that for every $i \in \mathbb{N}$ the sequence $\tau^{<i}$ can be extended to a timed word τ_i that satisfies $z.\varphi$. We can see the family τ, τ_0, $\tau_1, \ldots$ as a tree. The main branch is τ and for every $i \geq 1$, the sequence $\tau_i(i)\, \tau_i(i+1) \ldots$ branches from $\tau(i-1)$. A quasicounterexample for $z.\varphi$ contains all the necessary information to build such a family of timed words when $z.\varphi$ is not safe.

Two quasistates (Φ_1, d_1) and (Φ_2, d_2) are *compatible* if $d_1 = d_2$ and $z.p \in \Phi_1$ iff $z.p \in \Phi_2$ for every $p \in P$. Two sequences of quasistates are *compatible* if they are element-wise compatible.

We now give the definition of quasicounterexample.

Definition 3. Let f be an infinite sequence of quasistates and g a mapping from $\mathbb{N} \times \mathbb{N}$ to quasistates. The pair (f, g) is a *quasicounterexample* for $z.\varphi$ if

- f is a quasimodel for $z.\neg\varphi$,
- $g{\restriction}i$ is a quasimodel for $z.\varphi$, for all $i \in \mathbb{N}$, where $g{\restriction}i$ is the sequence $g(0,0)g(1,0)\ldots g(i-1,0)g(i,1)g(i,2)\ldots$, and
- f and $g(0,0)g(1,0)\ldots$ are compatible.

A quasicounterexample is k-bounded if f and $g{\restriction}i$, for all $i \in \mathbb{N}$, are all k-bounded. □

Lemma 10. The sentence $z.\varphi$ is not safe iff it has a k_φ-bounded quasicounterexample.

Proof. ($\Leftarrow$) Let (f, g) be a k_φ-bounded quasicounterexample for $z.\varphi$. According to Theorem 1, let $\sigma_f \otimes \delta_f$ and $\sigma_{g\restriction j} \otimes \delta_{g\restriction j}$ be the timed words defined by f and $g{\restriction}j$, for each $j \in \mathbb{N}$, respectively. Note that $\sigma_f \otimes \delta_f$ satisfies $z.\neg\varphi$ and $\sigma_{g\restriction j} \otimes \delta_{g\restriction j}$ satisfies $z.\varphi$ for all j. Now, since f and $g(0,0)g(1,0)\ldots$ are compatible, the finite prefix of $\sigma_f \otimes \delta_f$ of length $\ell \in \mathbb{N}$ is the same finite prefix of $\sigma_{g\restriction\ell} \otimes \delta_{g\restriction\ell}$ of length ℓ. Hence, every finite prefix of $\sigma_f \otimes \delta_f$ can be extended to a timed word that satisfies $z.\varphi$. Therefore, $z.\varphi$ is not safe.

($\Rightarrow$) Suppose $z.\varphi$ is not safe. Then there is a timed word τ satisfying $z.\neg\varphi$ such that for any $j \in \mathbb{N}$ the prefix $\tau^{<j}$ can be extended to a timed word τ_j satisfying $z.\varphi$. For $j \in \mathbb{N}$, let f and h_j be the quasimodels for $z.\neg\varphi$ and $z.\varphi$ defined by τ and τ_j according to Theorem 1, respectively. By Theorem 2, assume f and h_j are k_φ-bounded, for $j \in \mathbb{N}$.

Let $H = \{h_0, h_1, \ldots\}$. We may regard H as a set of infinite words from the alphabet consisting of all k_φ-bounded quasistates for $z.\varphi$, which is finite. We build inductively a sequence α of quasistates as follows. Let $\alpha(0)$ be the k_φ-bounded quasistate such that $\alpha(0) = h(0)$ for infinitely many $h \in H$. For the inductive step, suppose we have already built the first $i+1$ quasistates $\alpha^{\leq i} = \alpha(0)\, \alpha(1) \ldots \alpha(i)$, and that $\alpha^{\leq i} = h^{\leq i}$ for infinitely many $h \in H$. Let $\alpha(i+1)$ be a k_φ-bounded quasistate such that $\alpha^{\leq i}\alpha(i+1) = h^{\leq i}h(i+1)$ for infinitely many $h \in H$. Such a quasistate exists because $h(i+1)$ can take at most $\sharp(z.\varphi)$ possible values and there are infinitely many $h \in H$ with $\alpha^{\leq i} = h^{\leq i}$. By construction, the sequence α has the following property: for each $i \geq 0$, there are infinitely many $h \in H$ such that $h^{<i}$ is a prefix of α.

For each $i \geq 0$, let g_i be some $h \in H$ such that $h^{<i}$ is a prefix of α. Note that $g_{i_1}^{<i_1}$ is a prefix of $g_{i_2}^{<i_2}$, for $i_1 < i_2$. We define the mapping g by $g(i,0) = g_{i+1}(i)$ and $g(i,j) = g_i(i+j-1)$, for all $i \geq 0$ and $j \geq 1$. Note that $g{\restriction}i = g_i$, for any $i \in \mathbb{N}$, and $g(0,0)g(1,0)\ldots = \alpha$. It is easy to see that (f, g) is a quasicounterexample for $z.\varphi$. □

For a function g mapping $\mathbb{N} \times \mathbb{N}$ into quasistates, we define $g^{\leq i}$ as the restriction of g over $\{0, 1, \ldots, i\} \times \mathbb{N}$. Other functions such as $g^{<i}, g^{\geq i}, g^{>i}$ are defined analogously.

Suppose g_1 and g_2 are mappings from $\{0, 1, \ldots, k\} \times \mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$ into quasistates respectively. Let $g_1 g_2$ be the mapping obtained by concatenating both grids $\{0, 1, \ldots, k\} \times \mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$ along the first dimension.

Lemma 11. Let (f, g) be a quasicounterexample for $z.\varphi$ such that $f(i) = f(j)$ and $g(i,0) = g(j,0)$ for some $i < j$. Then $(f^{\leq i} f^{>j}, g^{\leq i} g^{>j})$ is a quasicounterexample for $z.\varphi$.

Proof. Let $g' = g^{\leq i} g^{>j}$. Note $f^{\leq i} f^{>j}$ is a quasimodel for $z.\varphi$ and each $g'{\restriction}i$ is a quasimodel for $z.\varphi$, by Lemma 8. Clearly, $f^{\leq i} f^{>j}$ and $g'(0,0)g'(1,0)\ldots$ are compatible. □

4. Deciding safety in TPTL

The following theorem gives a computable criterion for deciding whether a TPTL sentence is safe. This theorem

naturally extends the criterion for deciding whether an LTL formula is safe [1].

Theorem 3. The sentence $z.\varphi$ is not safe iff there are finite sequences f_1, f_2, h_1, h_2, h_3, h_4 meeting the following requirements.

1. $f_1 f_2^\omega$ is a quasimodel for $z.\neg\varphi$ and $h_1 h_2 h_3 h_4^\omega$ is a quasimodel for $z.\varphi$.
2. $|f_1| = |h_1| \leq \sharp(z.\varphi)^2$, $|f_2| = |h_2| \leq (|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)^2$, $|h_3| \leq \sharp(z.\varphi)$, and $|h_4| \leq (|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)$.
3. $f_1 f_2$ and $h_1 h_2$ are compatible.
4. The first quasistate of h_2 is a successor of the last quasistate of h_2.

Proof. ($\Rightarrow$) Suppose $z.\varphi$ is not safe. Then $z.\varphi$ has a k_φ-bounded quasicounterexample (f, g). Suppose $f(i) = (\Phi_i, d_i)$, for $i \in \mathbb{N}$ and let δ be the sequence defined by $\delta(i) := d_i$, for $i \in \mathbb{N}$. We construct f_1, f_2, h_1, h_2, h_3, and h_4 using ideas similar to those used in Lemma 9.

We start with f_1 and h_1. Take s such that $g(s, 0) = g(i, 0)$ and $f(s) = f(i)$, for infinitely many $i > s$. Apply Lemma 11 whenever $i_1 < i_2 < s$, $g(i_1, 0) = g(i_2, 0)$, and $f(i_1) = f(i_2)$. This yields the quasicounterexample

$$(f_1 f^{\geq s}, g_1 g^{\geq s})$$

with $|f_1| \leq \sharp(z.\varphi)^2$. Take h_1 as the sequence

$$g_1(0, 0) g_1(1, 0) \ldots g_1(|f_1| - 1, 0).$$

We now explain how to get f_2 and h_2. Suppose there is a formula in $f^{\geq s}(0)$ of the form $z.(\psi_1 \mathsf{U} \psi_2)$. Take any k such that $z.\psi_2^{\bar{\delta}(s,s+k)} \in f^{\geq s}(k)$. Apply Lemma 11 whenever $i_1 < i_2 \leq k$, $f^{\geq s}(i_1) = f^{\geq s}(i_2)$, and $g^{\geq s}(i_1, 0) = g^{\geq s}(i_2, 0)$. This yields the quasicounterexample

$$\big(f_1\, f^{\geq s}(0)\, f'\, f^{>s+k}, g_1\, g^{\geq s}(0, \cdot)\, g'\, g^{>s+k}\big),$$

where $g^{\geq s}(0, \cdot)$ is the restriction of $g^{\geq s}$ to $\{0\} \times \mathbb{N}$. Note that $f^{\geq s}(0) f'$ realizes the occurrence $z.(\psi_1 \mathsf{U} \psi_2) \in f^{\geq s}(0)$ and has length at most $\sharp(z.\varphi)^2$. Repeat this procedure for all other formulas of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f^{\geq s}(0)$. After this, we get a quasicounterexample

$$\big(f_1\, f^{\geq s}(0)\, f^\circ\, f^{>k^\circ}, g_1\, g^{\geq s}(0, \cdot)\, g^\circ\, g^{>k^\circ}\big),$$

where $f^{\geq s}(0) f^\circ$ realizes all formulas of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f^{\geq s}(0)$ and has length at most $|Cl(z.\varphi)| \cdot \sharp(z.\varphi)^2$. Following the ideas of Lemma 9, we can reshape this quasicounterexample into one of the form

$$\big(f_1\, f^{\geq s}(0)\, f^{\circ\circ}\, f^{\geq k^{\circ\circ}}, g_1\, g^{\geq s}(0, \cdot)\, g^{\circ\circ}\, g^{\geq k^{\circ\circ}}\big),$$

where $f^{\geq k^{\circ\circ}}(0) = f^{\geq s}(0)$, $g^{\geq k^{\circ\circ}}(0, 0) = g^{\geq s}(0, 0)$, and $f^{\geq s}(0) f^{\circ\circ}$ realizes all the formulas of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f^{\geq s}(0)$, has a quasistate (Φ, d) with $d > 0$, and has length $(|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)^2$. Finally, let $f_2 = f^{\geq s}(0) f^{\circ\circ}$ and let h_2 be the sequence of all quasistates in $g^{\geq s}(0, \cdot) g^{\circ\circ}$ whose second coordinate is 0. Note that:

1. f_1 and f_2 satisfy the requirements of Lemma 9, hence $f_1 f_2^\omega$ is a quasimodel for $z.\neg\varphi$,
2. $|f_2| = |h_2| \leq (|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)^2$,
3. $f_1 f_2$ and $h_1 h_2$ are compatible, and
4. the first quasistate of h_2 is a successor of the last quasistate of h_2.

It remains to build h_3 and h_4. Let

$$\gamma = g^{>k^{\circ\circ}}(0, 1)\, g^{>k^{\circ\circ}}(0, 2) \ldots$$

Note that $h_1 h_2 \gamma$ is a quasimodel for $z.\varphi$. We build h_3 and h_4 from γ such that $h_1 h_2 h_3 h_4^\omega$ is a quasimodel for $z.\varphi$ in a similar way as in the proof of Lemma 9.

($\Leftarrow$) First, we show that any finite prefix of $h_1 h_2^\omega$ can be extended to a quasimodel for $z.\varphi$. For this, it suffices to show that for any $i \geq 1$, the sequence $h_1 h_2^i h_3 h_4^\omega$ is a quasimodel for $z.\varphi$. Requirements (QM-1) and (QM-2) follow from $h_1 h_2 h_3 h_4^\omega$ being a quasimodel for $z.\varphi$. (QM-3) follows from condition 4 in the theorem. For (QM-4), let $z.\psi_1 \mathsf{U} \psi_2$ be a formula occurring somewhere in $h_1 h_2^i h_3 h_4^\omega$. If $z.(\psi_1 \mathsf{U} \psi_2)$ occurs in h_3 or h_4 then this is trivial. Suppose it occurs in the first quasistate of a copy of h_2. If $z.(\psi_1 \mathsf{U} \psi_2)$ is not realized by that copy h_2, then either $z.(\psi_1 \mathsf{U} \psi_2)^\Delta$ for suitable Δ occurs in the first quasistate of the next copy of h_2 or in the first quasistate of h_3. In the latter case, we are done; in the former, just repeat the argument until $z.(\psi_1 \mathsf{U} \psi_2)$ is realized or it occurs in the first quasistate of h_3. The case when $z.(\psi_1 \mathsf{U} \psi_2)$ occurs in h_1 is similar.

To build a quasicounterexample for $z.\varphi$ use the facts that (i) any finite prefix of $h_1 h_2^\omega$ can be extended to a quasimodel for $z.\varphi$ and (ii) the sequences $f_1 f_2^\omega$ and $h_1 h_2^\omega$ are compatible. $\square$

The following example illustrates how the sequences f_1, f_2, h_1, h_2, h_3, and h_4 work together. Consider the formula $\varphi = p \wedge \Diamond(\neg p \wedge \bigcirc \Box p)$, which is not safe. The timed word $(\{p\}, 1)^\omega$ does not satisfy it, but any finite prefix $(\{p\}, 1)^i$, with $i \in \mathbb{N}$, can be extended to the timed word $(\{p\}, 1)^i\, (\emptyset, 1)\, (\{p\}, 1)^\omega$, which satisfies φ. This information is represented by letting f_1 and h_1 be the empty sequence, $f_2 = h_2 = (\{p\}, 1)$, $h_3 = (\emptyset, 1)$, and $h_4 = (\{p\}, 1)$. These sequences satisfy the requirements of Theorem 3 and encode the timed words $(\{p\}, 1)^\omega = f_1 f_2^\omega$ and $(\{p\}, 1)^i\, (\emptyset, 1)\, (\{p\}, 1)^\omega = h_1 h_2^i h_3 h_4^\omega$, for $i \in \mathbb{N}$.

Theorem 4. Deciding whether a TPTL sentence is safe is EXPSPACE-complete.

Proof. EXPSPACE-hardness follows from the fact that deciding whether a TPTL formula is valid is EXPSPACE-complete [8]. The sentence $z.\varphi$ is valid iff $z.\varphi \vee \Diamond q$ is safe, where q is an atomic proposition not occurring in $z.\varphi$.

We now present a non-deterministic algorithm that decides whether a TPTL sentence is safe by guessing finite

sequences $f_1, f_2, h_1, h_2, h_3, h_4$ of quasistates that satisfy the requirements of Theorem 3. This algorithm uses an amount of memory exponential in the length of $z.\varphi$. By Savitch's theorem, it follows that deciding whether a TPTL sentence is safe is in EXPSPACE.

First, guess a number $\ell_1 \leq \sharp(z.\varphi)^2$. Now guess two compatible k_φ-bounded quasistates (Φ_0, d_0) and (Ψ_0, e_0), with $z.\varphi \notin \Phi_0$ and $z.\varphi \in \Psi_0$. They are the first quasistates for f_1 and h_1 respectively. Next, for i from 1 to $\ell_1 - 1$, guess two compatible k_φ-bounded quasistates (Φ_i, d_i) and (Ψ_i, e_i) that are successors of (Φ_{i-1}, d_{i-1}) and (Ψ_{i-1}, e_{i-1}) respectively. This gives rise to the two sequences f_1 and h_1. Similarly, guess the sequences f_2 and h_2. Guess a number $\ell_2 \leq (|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)^2$ and guess two compatible k_φ-bounded quasistates (Φ'_0, d'_0) and (Ψ'_0, e'_0). These quasistates must be successors of $(\Phi_{\ell-1}, d_{\ell-1})$ and $(\Psi_{\ell-1}, e_{\ell-1})$. To check conditions 1 and 4 of Theorem 3, set a variable $b = 0$ and create a table T with all the formulas of the form $z.(\psi_1 \mathsf{U} \psi_2)$ that occur in Φ'_0. Next, guess the rest of the two sequences f_2 and h_2, checking that the new pair is a successor of the previous one. Every time the next quasistate (Φ'_i, d'_i) for f_2 is guessed, set $b = 1$ if $d'_i > 0$ and remove from T all the formulas that are realized by Φ'_i. That is, remove from T all occurrences of the form $z.(\psi_1 \mathsf{U} \psi_2)$ such that $z.\psi_2^{d'_1+\ldots+d'_i}$ appears in Φ'_i. After guessing f_2 and h_2, check that (i) $b = 1$, (ii) the first quasistate of h_2 is a successor of the last quasistate of h_2, and (iii) T is empty to ensure that f_2 realizes all occurrences of the form $z.(\psi_1 \mathsf{U} \psi_2) \in f_2(0)$. We guess h_3 and h_4 in a similar way. The space used for h_1 and h_2 can be reused for h_3 and h_4.

For this algorithm we need space for ℓ_1, ℓ_2, T, and four quasistates. Note that ℓ_i is at most

$$\big(|Cl(z.\varphi)| + 2\big) \cdot \sharp(z.\varphi)^2 \leq (n_\varphi k_\varphi + 2)\, 2^{2(n_\varphi+1)k_\varphi},$$

for $i \in \{1, 2\}$. Thus for each ℓ_i we need $\mathcal{O}(n_\varphi k_\varphi)$ space. For each quasistate we need $\mathcal{O}(|Cl(z.\varphi)|\, k_\varphi) = \mathcal{O}(n_\varphi k_\varphi^2)$ space, and for T we also need $\mathcal{O}(|Cl(z.\varphi)|)$ space. Since n_φ and k_φ are linear and exponential in the length of $z.\varphi$ respectively, the algorithm takes space exponential in the length of $z.\varphi$. □

5. Deciding liveness in TPTL

In this section, we extend the algorithm presented in [1] to decide whether a given TPTL sentence $z.\varphi$ describes a liveness property.

Recall that $\Sigma = 2^P \times \mathbb{N}$. Let $\hat{\Sigma}_\varphi$ be the restriction of Σ to those pairs $(a, d) \in 2^P \times \mathbb{N}$ with $d \leq k_\varphi$. Let $\tau = \sigma \otimes \delta$ be a sequence in Σ^* of length ℓ. The sequence τ is a *k-good prefix for $z.\varphi$* if $\delta(i) \leq k$, for every $i < \ell$, and there is a $\tau' \in \Sigma^\omega$ such that $\tau\tau'$ is a k-bounded timed word that satisfies $z.\varphi$.

Lemma 12. The sentence $z.\varphi$ describes a liveness property iff every $\sigma \in \hat{\Sigma}_\varphi^*$ is a k_φ-good prefix for $z.\varphi$.

Proof. ($\Rightarrow$) Let $\sigma = (a_0, d_0)(a_1, d_1)\ldots(a_k, d_k)$ be a sequence in $\hat{\Sigma}_\varphi^* \subseteq \Sigma^*$. By assumption, there is $\sigma' \in \Sigma^\omega$ of the form $\sigma' = (a_{k+1}, d_{k+1})(a_{k+2}, d_{k+2})\ldots$ such that $\sigma\sigma'$ is a timed word that satisfies $z.\varphi$. Let

$$\hat{\sigma'} := (a_{k+1}, \hat{d}_{k+1})(a_{k+2}, \hat{d}_{k+2})\ldots.$$

By Lemma 7, $\sigma\hat{\sigma'}$ satisfies $z.\varphi$. So σ is a k_φ-good prefix for $z.\varphi$.

($\Leftarrow$) For $\sigma = (a_0, d_0)(a_1, d_1)\ldots(a_k, d_k) \in \Sigma^*$, let $\hat{\sigma} := (a_0, \hat{d}_0)(a_1, \hat{d}_1)\ldots(a_k, \hat{d}_k)$, which is in $\hat{\Sigma}_\varphi^*$. By assumption, $\hat{\sigma}$ is a k_φ-good prefix for $z.\varphi$. So, there is $\sigma' \in \Sigma^\omega$ such that $\hat{\sigma}\sigma'$ is a k_φ-bounded timed word that satisfies $z.\varphi$. By Lemma 7, $\sigma\sigma'$ satisfies $z.\varphi$, so σ is a good prefix for $z.\varphi$. □

Definition 4. An infinite sequence of quasistates for a formula $z.\varphi$ is called a *fulfilling path for $z.\varphi$* if it meets the conditions (QM-1), (QM-3), and (QM-4) from Definition 2. □

The following lemma is proved similarly to Lemma 9.

Lemma 13. There is a fulfilling path for $z.\varphi$ iff there are sequences f_1 and f_2 of k_φ-bounded quasistates for $z.\varphi$ such that:

1. $|f_1| \leq \sharp(z.\varphi)$ and $|f_2| \leq (|Cl(z.\varphi) + 2|) \cdot \sharp(z.\varphi)$,
2. $d > 0$ for some (Φ, d) in f_2,
3. $f_j(i+1)$ is a successor of $f_j(i)$ for $i < |f_j| - 1$ and $j \in \{1, 2\}$,
4. $f_2(0)$ is a successor of the last quasistates of f_1 and f_2, and
5. every occurrence of the form $z.(\psi_1 \mathsf{U} \psi_2)$ in $f_2(0)$ is realized by f_2.

Lemma 14. There is an algorithm that, given a quasistate (Φ_0, d_0) for a formula $z.\varphi$, decides whether there is a fulfilling path f for $z.\varphi$ such that $f(0) = (\Phi_0, d_0)$. The algorithm uses space exponential in the length of $z.\varphi$.

Proof. By Savitch's theorem, it suffices to give a nondeterministic algorithm that uses space exponential in n_φ. The algorithm guesses a fulfilling path of the form described in Lemma 13. First, guess the lengths of f_1 and f_2, namely ℓ_1 and ℓ_2 with $\ell_1 \leq \sharp(z.\varphi)$ and $\ell_2 \leq (|Cl(z.\varphi)| + 2) \cdot \sharp(z.\varphi)$. Then for i from 1 to $\ell_1 - 1$, guess a k_φ-bounded quasistate (Φ_i, d_i) that is a successor of (Φ_{i-1}, d_{i-1}). After guessing f_1, let $b = 0$ and let T be the set of all formulas of the form $z.(\psi_1 \mathsf{U} \psi_2) \in \Phi_{\ell_1-1}$. Guess f_2 as follows. First, guess a k_φ-bounded quasistate (Φ'_0, d'_0) that is a successor of $(\Phi_{\ell_1-1}, d_{\ell_1-1})$. Then for i from 1 to $\ell_2 - 1$, guess a k_φ-bounded quasistate (Φ'_i, d'_i) that is a successor of (Φ'_{i-1}, d'_{i-1}). Every time the next quasistate (Φ'_i, d'_i) for f_2 is guessed, set b to 1 if $d'_i > 0$ and remove from T all formulas $z.(\psi_1 \mathsf{U} \psi_2)$ such that $z.\psi_2^{d'_1+\ldots+d'_i} \in \Phi'_i$. After guessing $(\Phi'_{\ell_2-1}, d'_{\ell_2-1})$, check that $b = 1$, T is empty, and (Φ'_0, d'_0)

is a successor of $(\Phi'_{\ell_2-1}, d'_{\ell_2-1})$. If these checks succeed, then by Lemma 13 there is a fulfilling path for $z.\varphi$. The complexity result follows from the proof of Theorem 4. $\square$

Recall that an automaton is a tuple $\langle Q, \Gamma, \delta, q_0, F\rangle$, where Q is a finite nonempty set of *states*, Γ is *a finite nonempty alphabet*, $q_0 \in Q$ is the *initial state*, $\delta \subseteq Q \times \Gamma \times Q$ is the *transition relation*, and $F \subseteq Q$ is the set of *accepting states*. We can see δ as a set of directed edges between states that are labeled with elements of Γ.

Theorem 5. There is an algorithm that decides whether a formula $z.\varphi$ describes a liveness property. The algorithm uses space doubly exponential in the length of $z.\varphi$.

Proof. The algorithm has two parts. First, build an automaton $\mathcal{A}$ over the alphabet $\hat{\Sigma}_\varphi$ that accepts $\sigma \in \hat{\Sigma}^*_\varphi$ iff σ is a k_φ-good prefix for $z.\varphi$. Second, check if $\mathcal{A}$ accepts all the words in $\hat{\Sigma}^*_\varphi$. By Lemma 12, $\mathcal{A}$ accepts all the words in $\hat{\Sigma}^*_\varphi$ iff $z.\varphi$ describes a liveness property.

First, we define $\mathcal{A}$. $\mathcal{A}$'s set of states is the set of all k_φ-bounded quasistates for $z.\varphi$ together with a distinguished initial state called *init*. For two states s_1, s_2 and $(a, d) \in \hat{\Sigma}_\varphi$, there is an edge from s_1 to s_2 labeled with (a, d) iff

1. $s_2 = (\Phi, d')$ with $d' = d$,
2. $p \in a$ iff $p \in \Phi$ for all $p \in P$,
3. if $s_1 = init$ then $z.\varphi \in \Phi$, and
4. if $s_1 \neq init$ then s_2 is a successor of s_1.

Finally, for every state s different from *init*, apply Lemma 14 and make s accepting iff there is a fulfilling path f for $z.\varphi$ such that $f(0) = s$.

Next, we prove that $\mathcal{A}$ accepts $\sigma \in \hat{\Sigma}^*_\varphi$ iff σ is a k_φ-good prefix for $z.\varphi$. If σ is accepted by $\mathcal{A}$, then there is a path $init, s_0, s_1, \ldots, s_\ell$ in $\mathcal{A}$ such that s_ℓ is an accepting state and the concatenation of the labels of the edges in the path reads σ. Let f be a fulfilling path for $z.\varphi$ such that $f(0) = s_\ell$. It is easy to prove that $s_0 s_1 \ldots s_{\ell-1} f$ is a k_φ-bounded quasimodel for $z.\varphi$ and that the timed word defined by this quasimodel according to Theorem 1 is an extension of σ. Therefore, σ is a k_φ-good prefix for $z.\varphi$. Suppose now that $\sigma = (a_0, d_0)(a_1, d_1)\ldots(a_{\ell-1}, d_{\ell-1})$ is a k_φ-good prefix for $z.\varphi$. Let $\sigma' = (a_\ell, d_\ell)(a_{\ell+1}, d_{\ell+1})\ldots \in \hat{\Sigma}^\omega_\varphi$ be such that $\sigma\sigma'$ is a k_φ-bounded timed word that satisfies $z.\varphi$. Let $f_1 = (\Phi_0, d_0)(\Phi_1, d_1)\ldots(\Phi_{\ell-1}, d_{\ell-1})$ and $f_2 = (\Phi_\ell, d_\ell)(\Phi_{\ell+1}, d_{\ell+1})\ldots$ be the sequences of quasistates for $z.\varphi$, where $\Phi_i := \{z.\psi \in Cl(z.\varphi) \mid \sigma\sigma', i \vDash z.\psi\}$, for $i \in \mathbb{N}$. By Theorem 1, $f_1 f_2$ is a quasimodel for $z.\varphi$. It follows that f_2 is a fulfilling path for $z.\varphi$ and hence $f_2(0)$ is an accepting state in $\mathcal{A}$. Also, $init\, f_1 f_2(0)$ is a path in $\mathcal{A}$ such that the concatenation of the edges in the path reads σ. Therefore, $\mathcal{A}$ accepts σ.

We now analyze the complexity of building $\mathcal{A}$ and checking if $\mathcal{A}$ accepts all the words in $\hat{\Sigma}^*_\varphi$. The number of states of $\mathcal{A}$ is $1 + \sharp(z.\varphi) \leq 2^{(n_\varphi+1)k_\varphi} = 2^{\mathcal{O}(n_\varphi k_\varphi)}$. The size of the alphabet $\hat{\Sigma}_\varphi$ is $\left|2^P\right| \cdot k_\varphi = \mathcal{O}(k_\varphi)$. Therefore, building $\mathcal{A}$ takes $2^{\mathcal{O}(n_\varphi k_\varphi)}$ space. Checking if $\mathcal{A}$ accepts $\hat{\Sigma}^*_\varphi$ takes space polynomial in the number of $\mathcal{A}$'s states, which is $2^{\mathcal{O}(n_\varphi k_\varphi)}$ space. The values of n_φ and k_φ are respectively linear and exponential in the length of $z.\varphi$, and therefore our algorithm uses space doubly exponential in the length of $z.\varphi$. $\square$

6. Conclusion

Sistla [1] proved that deciding safety and liveness for LTL are PSPACE-complete and in EXPSPACE, respectively. We have carried over his proofs to TPTL and proved that the corresponding problems for TPTL are EXPSPACE-complete and in 2-EXPSPACE, respectively. Concerning liveness, we have the following lower bounds. Checking liveness is PSPACE-hard for LTL and EXPSPACE-hard for TPTL. This is because φ is satisfiable iff $\Diamond\varphi$ describes a liveness property. Tighter lower bounds for deciding liveness remain unknown for both LTL and TPTL. Note that we considered a discrete time domain for TPTL. In the case of dense time, satisfiability for TPTL is undecidable [8], and thus checking safety and liveness are both undecidable.

References

[1] A. P. Sistla, Safety, liveness and fairness in temporal logic, Formal Asp. Comput. 6 (5) (1994) 495–511.

[2] L. Lamport, Proving the correctness of multiprocess programs, IEEE Trans. Software Eng. 3 (2) (1977) 125–143.

[3] B. Alpern, F. B. Schneider, Defining liveness, Inform. Process. Lett. 21 (4) (1985) 181–185.

[4] O. Kupferman, M. Y. Vardi, Model checking of safety properties, Form. Method. Syst. Des. 19 (3) (2001) 291–314.

[5] M. Leucker, C. Schallhart, A brief account of runtime verification, J. Log. Algebr. Program. 78 (5) (2009) 293–303.

[6] A. Pnueli, The temporal logic of programs, in: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77), IEEE Computer Society, 1977, pp. 46–57.

[7] R. Alur, T. A. Henzinger, Logics and models of real time: A survey, in: Proceedings of the 1991 REX Workshop on Real-Time: Theory in Practice, Vol. 600 of Lect. Notes Comput. Sci., Springer, 1992, pp. 74–106.

[8] R. Alur, T. A. Henzinger, A really temporal logic, J. ACM 41 (1) (1994) 181–203.

[9] I. Hodkinson, F. Wolter, M. Zakharyaschev, Decidable fragments of first-order temporal logics, Ann. Pure Appl. Logic 106 (1) (2000) 85–134.

[10] A. P. Sistla, E. M. Clarke, The complexity of propositional linear temporal logics, J. ACM 32 (3) (1985) 733–749.

[11] R. Alur, T. A. Henzinger, Real-time logics: Complexity and expressiveness, in: Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science (LICS'90), IEEE Computer Society, 1990, pp. 390–401.