Acta Informatica manuscript No.
(will be inserted by the editor)

A Tool for Deciding the Satisfiability of Continuous-time Metric Temporal Logic

**Marcello M. Bersani · Matteo Rossi ·
Pierluigi San Pietro**

Received: date / Revised version: date

Abstract Constraint LTL over clocks is a variant of CLTL, an extension of linear-time temporal logic allowing atomic assertions in a concrete constraint system. Satisfiability of CLTL over clocks is here shown to be decidable by means of a reduction to a decidable SMT (Satisfiability Modulo Theories) problem. The result is a complete Bounded Satisfiability Checking procedure, which has been implemented by using standard SMT solvers. The importance of this technique derives from the possibility of translating various continuous-time metric temporal logics, such as MITL and QTL, into CLTL over clocks itself. Although standard decision procedures of these logics do exist, they have never been realized in practice. Suitable translations into CLTL over clocks have instead allowed us the development of the first prototype tool for deciding MITL and QTL. The paper also reports preliminary, but encouraging, experiments on some significant examples of MITL and QTL formulae.

1 Introduction

Constraint Linear Temporal Logic [19] (CLTL, for short) is an extension of linear-time temporal logic allowing atomic assertions in a concrete constraint system. By carefully choosing the constraint system, CLTL may be decidable, as well as expressive and well-suited to define infinite-state systems and their properties.

In this paper, we define a variant of CLTL, called CLTL over clocks (CLTLoc, for short), where arithmetic variables occurring in atomic assertions behave as *clocks*. At every (discrete) position in time, a clock measures the real time elapsed since the last position when the clock itself was "reset" (i.e., the variable was equal to 0); clocks can also be compared against an integer constant. By definition, in CLTLoc each position $i \in \mathbb{N}$ is associated with a real value (a "delay") corresponding to the "time elapsing" between i and the next position $i + 1$. This

This research was supported by the Programme IDEAS-ERC, Project 227977-SMScom and by PRIN Project 2010LYA9RH.

Politecnico di Milano – DEIB, Piazza Leonardo da Vinci, 32 – 20133 Milano, Italy

allows mixing of discrete events with continuous-time, a typical situation arising in many computer-controlled applications.

Satisfiability of CLTLoc is here shown to be decidable by means of a reduction to a decidable Satisfiability Modulo Theories (SMT, for short) problem, resulting in a complete Bounded Satisfiability Checking procedure. Although other automata-based decision procedures are also suitable to show decidability of CLTLoc (e.g., [19]), the novelty of our reduction is that it can easily be implemented by using standard SMT solvers, such as [28]. In fact, the paper also reports on a new, publicly available, software tool to verify CLTLoc, allowing the application of CLTLoc to the specification and the verification of timed systems. However, a further advantage of our approach is that various continuous-time metric temporal logics, such as Metric Interval Temporal Logic (MITL, for short) [4] and Quantified Temporal Logic (QTL, for short) [24], may be translated into CLTLoc itself. These translations have allowed us the development of the first available tool for deciding both MITL and QTL. In this paper we report encouraging experiments on some significant verification examples, such as the timed lamp and its properties, in CLTLoc, MITL and QTL. Further evidence of the generality and effectiveness of our approach is provided by our translation of the extension of QTL with so-called Pnueli and counting modalities [34] into CLTLoc, thus providing its first concrete decision procedure.

In general, the existing level of support for verification of continuous-time temporal logics is not as well developed as for discrete-time models. Uppaal [7] is the de-facto standard tool for verification of timed automata, but it does not support continuous-time temporal logics: not only satisfiability checking is not available in Uppaal, but even the formalization of system properties in temporal logic is not allowed, aside from rather simple invariants and reachability properties. Satisfiability Modulo Theories is a promising but well-consolidated field, supported by efficient solvers that are able to decide problems of many disciplines. In particular, decidable SMT problems have been already considered in the recent past, for instance to solve reachability [30] and the bounded version of language inclusion [6] for timed automata. The idea is to give a direct representation of bounded runs of timed automata through an SMT formula, capturing a bounded unrolling of the transition relation. Similarly, also Bounded Model-Checking (BMC, for short) of Linear Temporal Logic on timed automata [5] can be tackled by reducing the problem to an instance of an SMT problem, by using a technique extending the traditional BMC procedure for LTL over finite automata [18], but by restricting the set of valid runs to those that are periodic in the values of the clocks. Finite or periodic runs of timed automata can then be encoded in SMT formulae with explicit arithmetic. Nonetheless, also this approach has so far failed to produce a concrete decision procedure for logics such as MITL and QTL. This difficulty is caused by the gap of translating formulae into timed automata, a step which is avoided by our approach. Standard decision procedures of MITL and QTL logics were already defined some time ago (e.g., [4,27,35]), typically based on timed automata [3], but, to the best of our knowledge, they have never been realized in practice. This may suggest that these procedures are not easily implementable.

Temporal Logics such as Timed Propositional Temporal Logic (TPTL, for short), Metric Temporal Logic (MTL, for short), MITL and QTL, and operational models such as timed automata as well, may be interpreted over dense time domains in two ways: the "pointwise" semantics and the "continuous (or "interval-based") semantics [17]. In the pointwise semantics, an atomic formula is interpreted as an instantaneous event with a timestamp. A behavior (or run) of the system is described by a timed word, which is a sequence $(a_0, t_0)(a_1, t_1)\ldots$, where each a_i is a symbol of the alphabet and each t_i is a real-valued timestamp. A timed word must be strictly increasing ($t_i < t_{i+1}$) and must verify the non-Zeno condition, i.e., it is finite or it diverges to infinity. The pointwise semantics is very natural when considering specifications of timed automata, with atomic formulae interpreted as *state transitions*. In the continuous semantics, atomic formulae are instead interpreted as *state predicates*, i.e., continuous flows or *signals*. A signal (also called a timed state sequence) is a mapping associating values in $\mathbb{R}_+$ with states. A finite variability condition (strictly related to the non-Zeno condition) is always assumed. There are various results of expressiveness and decidability concerning MTL over the two semantics. First, it is obvious that a MTL (and MITL as well) formula interpreted over the pointwise semantics can always be translated into an equivalent MTL (or MITL) formula in the continuous semantics. However, other results are less immediate. For instance, MTL is undecidable in the continuous semantics [4] (unless time-singular intervals are ruled out, thus obtaining the logic MITL), but it is decidable, although not primitive recursive, in the pointwise version over finite models [32]–paving the way to showing that MTL in the continuous semantics is strictly more expressive than MTL in the pointwise semantics [20, 17] (over both finite and infinite models). No similar expressiveness result is known for MITL.

CLTLoc is naturally defined over timed words and it is as expressive as timed automata [14], i.e., it can define the same class of languages (the timed ω-regular). In this paper, we prove that CLTLoc is decidable, with an SMT-based procedure which has been implemented. We applied the resulting tool to a few examples of CLTLoc specifications, showing that their verification is feasible.

However, CLTLoc can also be used as a tool to interpret and verify other metric temporal logics. For instance, in [15] we provide a complete translation of MITL formulae over the continuous semantics into equisatisfiable CLTLoc formulae, thus allowing their verification with our CLTLoc tool. We have implemented this translation and we report on various experimental results on MITL specifications.

We also consider the case of MITL over the pointwise semantics, first by studying its expressiveness compared to CLTLoc. In this case, MITL is less expressive than CLTLoc, but we prove that CLTLoc is equally expressive with *projection-closed* MITL (pMITL, for short), an extension of MITL allowing existential propositional quantifiers [23]. Clearly, MITL formulae on timed words may be verified by a simple conversion into equivalent MITL formulae on signals, which can then be translated into CLTLoc. However, the translation from pMITL to CLTLoc, defined in the equivalence proof, is much more compact than the one defined for the continuous case, since in general signals may be more complex than timed words. Therefore, it may be more efficient to apply the new translation to convert MITL pointwise formulae directly into CLTLoc.

The paper is organized as follows. The first part is devoted to the main definitions and to the proofs of decidability of CLTLoc and of its practical implementation. Sect. 2 defines CLTLoc, and illustrates it by means of a running example (a timed lamp), while Sect. 3 proves that CLTLoc is decidable; Sect. 4 outlines the implemented SMT-based decision procedure of CLTloc. The remainder of the paper is devoted to illustrate the relations between MITL and CLTLoc, both in theory and in practice. Sect. 5 briefly recalls the definition of MITL and some of its variants, both in the pointwise and in the continuous cases; Sect. 6 shows that pMITL and CLTLoc are equally expressive over the pointwise semantics. Sect. 7 recalls the general idea of [12] behind the translation of MITL over the continuous semantics into CLTLoc, while Sect. 8 illustrates the tool for the verification of MITL, showing also experimental results for both CLTLoc and MITL. Sect. 9 concludes.

2 Constraint LTL over clocks

Constraint LTL (CLTL [19,10]) is an extension of LTL allowing atomic formulae over a *constraint system* $\mathcal{D} = (D, \mathcal{R})$, where D is a specific domain of interpretation for a finite set of variables V and for constants, and $\mathcal{R}$ is a finite family of relations on D (of various arities). CLTLoc is a special case of CLTL, where the domain D is $\mathbb{R}_+$ (the set of nonnegative reals), the set $\mathcal{R}$ of relations is $\{<, =\}$ and the variables in V are interpreted as *clocks*.

Let AP be a finite set of atomic propositions. A *temporal term* α is a constant $c \in \mathbb{N}$ or a clock $x \in V$. A *constraint* is a formula of the form $\alpha \sim \beta$, where $\sim$ is in $\{<, =\}$ and α, β are temporal terms. Well-formed CLTLoc formulae are defined as follows:

$$\phi := p \mid \alpha \sim \alpha \mid \phi \wedge \phi \mid \neg\phi \mid \mathbf{X}(\phi) \mid \mathbf{Y}(\phi) \mid \phi\mathbf{U}\phi \mid \phi\mathbf{S}\phi$$

where $p \in AP$, symbol $\sim$ stands for $<$ or $=$, and α is a temporal term. **X**, **Y**, **U** and **S** are the usual "next", "previous", "until" and "since" operators of LTL, with the same meaning. Boolean operators $\vee, \top, \bot, \Rightarrow$ can be introduced as usual; the "globally" **G**, "eventually" **F**, "release" **R**, and "trigger" **T** operators may be defined as in LTL, i.e., $\phi\mathbf{R}\psi$ is $\neg(\neg\phi\mathbf{U}\neg\psi)$, $\phi\mathbf{T}\psi$ is $\neg(\neg\phi\mathbf{S}\neg\psi)$, $\mathbf{G}\phi$ is $\bot\mathbf{R}\phi$ and $\mathbf{F}\phi$ is $\top\mathbf{U}\phi$.

The semantics of CLTLoc is defined with respect to the constraint system $(\mathbb{R}, <, =)$ and the strict linear order $(\mathbb{N}, <)$ representing *positions* in time. The values of clocks are defined by a mapping $\sigma : \mathbb{N} \times V \to \mathbb{R}_+$, assigning, for every position $i \in \mathbb{N}$, a real value $\sigma(i, x)$ to each clock $x \in V$. Intuitively, a clock x measures the time elapsed since the last time when $x = 0$, i.e., the last "reset" of x. To ensure that time progresses at the same rate for every clock, σ must satisfy the following condition: for every position $i \in \mathbb{N}$, there exists a "time delay" $\delta_i > 0$ such that for every clock $x \in V$:

$$\sigma(i+1, x) = \begin{cases} \sigma(i, x) + \delta_i, & \text{time progress} \\ 0 & \text{reset } x. \end{cases}$$

In this case, σ is called a *clock assignment*.

An *interpretation* of CLTLoc is a pair (π, σ), where σ is a clock assignment and $\pi : \mathbb{N} \to \wp(AP)$ is a mapping associating a set of propositions with each position in $\mathbb{N}$. The semantics of CLTLoc at a position $i \in \mathbb{N}$ over an interpretation (π, σ) is defined in Table 1, where we assume that $\sigma(i, c) = c$ whenever c is a constant.

A formula $\phi \in$ CLTLoc is *satisfiable* if there exists an interpretation (π, σ) such that $(\pi, \sigma), 0 \models \phi$. In this case, we say that (π, σ) is a *model* of ϕ and we write simply $(\pi, \sigma) \models \phi$.

$$
\begin{aligned}
(\pi,\sigma), i \models p &\Leftrightarrow p \in \pi(i) \text{ for } p \in AP \\
(\pi,\sigma), i \models \alpha_1 \sim \alpha_2 &\Leftrightarrow (\sigma(i,\alpha_1) \sim \sigma(i,\alpha_2)) \\
(\pi,\sigma), i \models \neg\phi &\Leftrightarrow (\pi,\sigma), i \not\models \phi \\
(\pi,\sigma), i \models \phi \wedge \psi &\Leftrightarrow (\pi,\sigma), i \models \phi \,\text{and}\, (\pi,\sigma), i \models \psi \\
(\pi,\sigma), i \models \mathbf{X}\,(\phi) &\Leftrightarrow (\pi,\sigma), i+1 \models \phi \\
(\pi,\sigma), i \models \mathbf{Y}\,(\phi) &\Leftrightarrow (\pi,\sigma), i-1 \models \phi \wedge i > 0 \\
(\pi,\sigma), i \models \phi\mathbf{U}\psi &\Leftrightarrow \exists\, j \geqslant i : (\pi,\sigma), j \models \psi \;\; \wedge (\pi,\sigma), n \models \phi \;\forall\; i \leqslant n < j \\
(\pi,\sigma), i \models \phi\mathbf{S}\psi &\Leftrightarrow \exists\, 0 \leqslant j \leqslant i : (\pi,\sigma), j \models \psi \;\wedge (\pi,\sigma), n \models \phi \;\forall\; j < n \leqslant i
\end{aligned}
$$

Table 1 Semantics of CLTLoc.

By definition, the initial value of a clock, $\sigma(0, x)$, may be any non-negative value. If needed, one or more of the clocks may be initialized to 0 just by adding a constraint of the form $x = 0$. It is often convenient to assume that at every position there is at least one clock which is not reset. To ensure that this is the case, just add a new clock *Now*, which is never reset, except possibly at position 0. Hence, the time delay δ_i is uniquely defined in each position i as $\sigma(i+1, \mathit{Now}) - \sigma(i, \mathit{Now})$.

Before going further, to motivate our approach, we provide an example of a CLTLoc formula representing a simple yet realistic timed system.

Example 1

We consider the LTL specification of a timed lamp and its properties (studied in Sect. 8) from [33]. The lamp is controlled by two buttons, labeled ON and OFF respectively, which cannot be pressed simultaneously. The lamp itself can be either on or off. When ON is pressed the lamp is immediately turned on, regardless of its current state; similarly, if OFF is pushed then the lamp is immediately turned off, also regardless of its current state. After ON is pressed, the lamp will not stay on forever, but, if no more buttons are pressed, it will automatically turn off with a delay Δ, a positive real constant. By pressing the ON button before the timeout expiration then the timeout is extended by a new delay Δ.

Our CLTLoc formula makes use of atomic propositions *on*, *off* and l representing, respectively, events "push button ON" and "push button OFF" and the state "light is on". Clocks may be used to measure the exact time elapsed since the last *on*; clearly some clock must be "reset" (i.e., set to 0, in analogy to Timed Automata) whenever ON is pressed; when a clock is equal to Δ then the timeout *expires.* To simplify the introduction of clocks, we first define a few shorthands called *rst-c*, $test_{c=\Delta}$ and $test_{0<c\leqslant\Delta}$. They have the intuitive meaning (which will be formalized after the main specification) that they are true if, and only if, a clock c is reset or, respectively, $c = \Delta$, or $0 < c \leqslant \Delta$. The specification of the lamp, still lacking the precise clock specification, is defined by formula $\mathbf{G}\left(\bigwedge_{i=0}^{5}(\varphi_i)\right)$.

$$
\begin{aligned}
\varphi_1 :=&\quad \neg(on \wedge off) \\
\varphi_2 :=&\quad on \Leftrightarrow rst\text{-}c \\
\varphi_3 :=&\quad \mathbf{Y}(l) \Rightarrow test_{0<c\leqslant\Delta} \\
\varphi_4 :=&\quad turnoff \Leftrightarrow \mathbf{Y}(l) \wedge (off \vee test_{c=\Delta}) \\
\varphi_5 :=&\quad l \Leftrightarrow \neg turnoff\, \mathbf{S}\; on.
\end{aligned}
$$

Formula φ_1 ensures mutual exclusion; φ_2 states that the timeout must be (re)started whenever button ON is pressed; φ_3 constrains the time elapsed since the previous instant if the light was on at that moment (i.e., not more than $\varDelta$); φ_4 defines (for readability) an event *turnoff*, capturing the two cases when the lamp (supposed to be ON in the previous instant) must be turned off at the current instant (i.e., OFF being pressed or the timeout expiring); finally, φ_5 gives the specification of the light, as being on if, and only if, there was in the past an *on* event not followed by a *turnoff*. Initialization is implicit in the specification (at instant 0, the light is off).

To complete the specification, we must formalize also the behavior of clocks. In CLTLoc, "resetting a clock" c, e.g., following an *on* event, is as simple as stating that $on \Rightarrow c = 0$; testing a clock c against a constant $\varDelta$ and causing say, a *turnoff* is a simple as stating that $c = \varDelta \Rightarrow turnoff$. Unfortunately, the same clock cannot be tested and reset at the same time. When this is required, it is possible to introduce two clocks c_0 and c_1, rather than just one clock, so that they can be reset alternatively: only one of the two clocks is reset and a new reset of the same clock will eventually occur only after the occurrence of a reset of the other clock. The behavior of this clock pair is described by the axiom $\mathbf{G}\,(\varphi_6 \wedge \varphi_7)$, where formulae φ_6, φ_7 are:

$$\begin{aligned}\varphi_6 :=&\quad \bigwedge_{i\in\{0,1\}} \Big(c_i = 0 \Rightarrow \neg\mathbf{X}\Big((c_{\overline{(i+1)}_2} > 0\,\mathbf{U}\,c_i = 0)\Big)\Big)\\ \varphi_7 :=&\quad c_0 = 0 \Rightarrow \neg(c_1 = 0)\end{aligned}$$

and $\bar{\cdot}_2$ stands for the modulo 2 operator (i.e., $\overline{1}_2 = 1$, $\overline{2}_2 = 0$). Finally, the above clock shorthands *rst-c*, $test_{c=\varDelta}$ and $test_{0<c\leqslant\varDelta}$ are defined as follows:

$$\begin{aligned} rst\text{-}c &\Leftrightarrow c_0 = 0 \vee c_1 = 0\\ test_{0<c\leqslant\varDelta} &\Leftrightarrow \textstyle\bigvee_{i\in\{0,1\}} 0 < c_i \leqslant \varDelta\\ test_{c=\varDelta} &\Leftrightarrow \textstyle\bigvee_{i\in\{0,1\}} \Big(c_i = \varDelta \wedge (c_{\overline{(i+1)}_2} > \varDelta \vee c_{\overline{(i+1)}_2} = 0)\Big)\,. \end{aligned}$$

3 Decidability and Complexity of CLTLoc

In this section, we show that CLTLoc is decidable in PSPACE, by reducing its satisfiability problem to checking the emptiness of a generalized Büchi automaton: for every CLTLoc formula ϕ, one can build a generalized Büchi automaton $\mathcal{A}_\phi^{\mathcal{R}}$ which has an accepting run if, and only if, ϕ is satisfiable.

Automaton $\mathcal{A}_\phi^{\mathcal{R}}$ accepts words including both sequences of symbolic valuations [19,8] and constraints representing the clock regions induced by ϕ. $\mathcal{A}_\phi^{\mathcal{R}}$ is built using a slight variation of the construction of [19,8], where instead variables are not restricted to behave as clocks: it is defined as the product of the Büchi automaton $\mathcal{A}_\phi$ recognizing the symbolic models of ϕ [19] with the automaton $\mathcal{A}_{\mathcal{R}_\phi}$ recognizing the language of successive regions of $\mathcal{R}_\phi$.

Preliminaries

We recall some fundamental definitions (see e.g. [8] and [19] for further details).

A generalized Büchi automaton $\mathcal{A}$ is 5-tuple (Σ, S, I, η, F), where Σ is the input alphabet, S is the state space, $I \subseteq S$ is the set of initial states, $F \subseteq \wp(S)$ is the accepting condition (namely, a set of sets of states), and $\eta \subseteq S \times \Sigma \times S$ is the transition relation. Let $w = w_0 w_1 \ldots$, with each $w_i \in \Sigma$, be an ω-word. A run of $\mathcal{A}$ with label w is an infinite sequence $s_0 s_1 \ldots$ of states in S such that: $s_0 \in I$ and, for all $i \geqslant 0$, $(s_i, w_i, s_{i+1}) \in \eta$. The run is accepting if it visits at least one state of every set of the accepting condition infinitely often. A word w is recognized by $\mathcal{A}$ if, and only if, there exists an accepting run with label w.

Given a set of clocks V, a *valuation* is a mapping $v : V \to \mathbb{R}_+$, i.e., v associates a value in $\mathbb{R}_+$ with each clock of V (a clock assignment as introduced in Section 2 is a sequence of valuations). For convenience, valuations can be extended to take into account constants in $\mathbb{N}$ so that $v(c) = c$ for all $c \in \mathbb{N}$. A constraint $\alpha \sim \beta$ is *satisfied* by a valuation v, written $v \models_{\mathbb{R}} \alpha \sim \beta$, if $v(\alpha) \sim v(\beta)$. Let ϕ be a CLTLoc formula and $x, y \in V$. If $c(x)$ is the maximum constant in ϕ clock x is compared to, then

$$ac(x) := \{x = 0, 0 < x\} \cup \{x < c,\ c < x,\ x = c \mid \forall c \in \mathbb{N}^+, c \leqslant c(x)\}$$

is the set of all clock constraints of x and

$$ac(x, y) := \{x + c \sim y,\ y \sim x + c,\ y + d \sim x,\ x \sim y + d \mid \forall c, d \in \mathbb{N}, c < c(y), d < c(x)\},$$

where $\sim \in \{<, =\}$, is the set of all clock constraints comparing x and y. Finally, set:

$$ac(\phi) := \bigcup_{x,y \in V, x \neq y} ac(x) \cup ac(x, y)$$

is the set of all clock constraints induced by ϕ.

Regions

We shortly recall the definition of regions from [3], with a generalization to deal with constraints of the form $x \sim y$. The clock space is the set $\mathbb{R}_+^{|V|}$. Every valuation $v : V \to \mathbb{R}_+$ can be considered an element of the clock space. For every real z, let $\lfloor z \rfloor$, $\{z\}$ be the floor and the fractional part of z, respectively. Let $\mathcal{R}_\phi$ be the finite partition induced on $\mathbb{R}_+^{|V|}$ by the following equivalence relation. We say that two valuations $v, v' : V \to \mathbb{R}_+$ are Alur-Dill equivalent if

1. for all $x \in V$, $v(x) > c(x)$ iff $v'(x) > c(x)$;
2. for all $x \in V$, if $v(x) \leqslant c(x)$ then $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ and, moreover, $\{v(x)\} = 0$ iff $\{v'(x)\} = 0$;
3. for all $x, y \in V$, if $v(x) \leqslant c(x) \wedge v(y) \leqslant c(y)$ then $\{v(x)\} \leqslant \{v(y)\}$ iff $\{v'(x)\} \leqslant \{v'(y)\}$;
4. for all $x, y \in V$, $v(x) \leqslant v(y)$ iff $v'(x) \leqslant v'(y)$.

The elements of $\mathcal{R}_\phi$ are called the clock regions of ϕ (*regions* for short). By definition of $\mathcal{R}_\phi$, given a region R, if $v \in R$, $\chi \in ac(\phi)$ and $v \models_{\mathbb{R}} \chi$, then for every $v' \in R$ also v' satisfies χ, i.e., $v' \models_{\mathbb{R}} \chi$. Hence, a region R may be described as the set of constraints $\chi \in ac(\phi)$ such that $v \models_{\mathbb{R}} \chi$, for every $v \in R$; thus, with slight abuse of notation we also write $v \models_{\mathbb{R}} R$; in addition, if ξ is a set of constraints, we denote with $\xi \cup R$ the union of ξ and χ.

A region $R' \in \mathcal{R}_\phi$ is called the *time-successor* of region $R \in \mathcal{R}_\phi$ if for every $v \in R$ there exists a positive $t \in \mathbb{R}_+$ and a valuation $v' \in R'$ such that for every clock $x \in V$ either $v'(x) = 0$ or $v'(x) = v(x) + t$. Notice that the definition of time-successor also considers that some clocks can be reset in R'.

To enforce the time-successor relation, let $\mathcal{A}_{\mathcal{R}_\phi} = (\mathcal{R}_\phi, \delta)$ be the finite automaton where $\mathcal{R}_\phi$ is the set of states and $\delta \subseteq \mathcal{R}_\phi \times \mathcal{R}_\phi$ is the transition relation containing all pairs (R, R') such that $R' \in \mathcal{R}_\phi$ is a time-successor of $R \in \mathcal{R}_\phi$. Since in CLTLoc clocks do not necessarily start from zero, each region is considered as potentially initial.

Symbolic valuations and symbolic models

Let $const(\phi)$ be the set of constants occurring in ϕ, and let C be a set of constraints over $V \cup const(\phi)$. A set C is *satisfied* by a valuation v, written $v \models_{\mathbb{R}} C$, if $v \models_{\mathbb{R}} \xi$, for all $\xi \in C$. Given a valuation v, let C_v be the set of all constraints over $V \cup const(\phi)$ such that $v \models_{\mathbb{R}} C_v$.

Definition 1 A *symbolic valuation sv* for ϕ is a set of constraints over $V \cup const(\phi)$ for which there is a valuation v such that $sv = C_v$. $SV(\phi)$ denotes the set of symbolic valuations associated with ϕ.

For example, consider a formula ϕ over two clocks x and y with $const(\phi) = \{0, 1\}$. An example of a symbolic valuation for ϕ is $sv = \{x < y, x < 1, 0 < y, 1 < y, x = 0\}$. In fact, if v is a valuation such that $v(x) = 0$ and $v(y) > 1$ then sv is satisfied by v, and there is no other constraint over $V \cup const(\phi)$ which is both satisfied by v and not included in sv.

The satisfiability of a set of constraints, for the constraint system $(\mathbb{R}, <, =)$ considered in this work, is decidable [19]. Given a symbolic valuation sv and a constraint ξ over $V \cup const(\phi)$, we write $sv \overset{sym}{\models} \xi$ if for every valuation v' such that $v' \models_{\mathbb{R}} sv$ we have $v' \models_{\mathbb{R}} \xi$. The symbolic satisfaction relation $\overset{sym}{\models}$ can also be extended to infinite sequences $\nu \in SV(\phi)^\omega$ of symbolic valuations; it is defined as $\models$, but for the case of constraints. If ξ is a constraint then:

$$\nu, i \overset{sym}{\models} \xi \Leftrightarrow \nu(i) \overset{sym}{\models} \xi.$$

Then, given a CLTLoc formula ϕ, we say that the sequence ν is a *symbolic model* for ϕ when $\nu, 0 \overset{sym}{\models} \phi$. Notice that relation $\overset{sym}{\models}$ is defined for CLTLoc formulae that do not include propositional letters (i.e., for which $AP = \emptyset$); this is without loss of generality, as it will be shown below. It is natural to extend relation $\models_{\mathbb{R}}$ to sequences of valuations, i.e., clock assignments: a clock assignment $\sigma : \mathbb{N} \times V \to \mathbb{R}_+$ satisfies a sequence of symbolic valuations ν if, for each i, the valuation at position i satisfies $\nu(i)$, which we write as $\sigma, i \models_{\mathbb{R}} \nu(i)$. Then, we say that a symbolic model ν *admits an arithmetical model* if there exists a clock assignment σ that satisfies ν.

Definition and properties of $\mathcal{A}_\phi^{\mathcal{R}}$

The closure of ϕ, denoted $cl(\phi)$, is the smallest set containing all subformulae of ϕ and closed under negation. An *atom* $\Gamma \subseteq cl(\phi)$ is a maximally consistent set, i.e., such that for each subformula ψ and ζ of ϕ':

- $\psi \in \Gamma \Leftrightarrow \neg\psi \notin \Gamma$,
- $\psi \wedge \zeta \in \Gamma \Leftrightarrow \psi, \zeta \in \Gamma$,
- $\psi \vee \zeta \in \Gamma \Leftrightarrow \psi \in \Gamma$ or $\zeta \in \Gamma$.

Let Q be the set of all the atoms Γ. Automaton A_ϕ is a generalized Büchi automaton, defined in [19], of the form $(SV(\phi), Q \times SV(\phi), I, \eta, F)$, for suitable I, η, F. Automaton A_ϕ is a modified version of the Vardi-Wolper automaton for LTL formulae, recognizing models of ϕ, i.e., it accepts infinite sequences of symbolic valuations that admit an arithmetical model. The following lemma regarding CLTL is a direct consequence of Lemmata 4.3 and 5.3 of [19] and is key to prove the decidability of the satisfiability of CLTLoc formulae.

Lemma 1 ([19]) *Let ϕ be a CLTL formula over the constraint system $(\mathbb{R}, <, =)$ and let $\nu \in SV(\phi)^\omega$. Then, ν is recognized by $\mathcal{A}_\phi$ if, and only if, $\nu \overset{sym}{\models} \phi$ and ν admits an arithmetical model.*

Thanks to the following result, we can apply Lemma 1 also to formulae that include propositional letters, which can be implicitly transformed into equisatisfiable formulae that do not include propositional letters.

Lemma 2 *Every CLTLoc formula ϕ with set of propositional letters AP can be transformed into another CLTLoc formula ϕ' that only includes clocks (i.e., for which $AP' = \emptyset$), and such that ϕ is satisfiable if, and only if, ϕ' is satisfiable.*

Proof Formula ϕ' is simply obtained by replacing, in ϕ, each propositional letter p with constraint $c_p = 0$ on a fresh clock c_p; that is, $c_p \in V_{AP}$ such that $V_{AP} \cap V = \emptyset$, and the set of clocks of ϕ' is $V \cup V_{AP}$. It is easy to see by induction that $(\pi, \sigma) \models \phi$ if, and only if, $(\pi', \sigma') \models \phi'$, where $\pi' : \mathbb{N} \to \emptyset$, $\sigma' : \mathbb{N} \times (V \cup V_{AP}) \to \mathbb{R}_+$, for all $x \in V$ we have $\sigma'(i, x) = \sigma(i, x)$ and for all $c_p \in V_{AP}$ we have $\sigma'(i, c_p) = 0$ if, and only if, $p \in \pi(i)$. □

Now, we define $\mathcal{A}_\phi^\mathcal{R}$ as a modification of A_ϕ to be consistent with the regions in $\mathcal{R}_\phi$: $\mathcal{A}_\phi^\mathcal{R}$ is the generalized Büchi automaton $(SV(\phi) \times \mathcal{R}_\phi, Q \times SV(\phi) \times \mathcal{R}_\phi, I', \eta', F \times \mathcal{R}_\phi)$, defined as follows. Relation η' is defined as:
$\langle(\Gamma, sv, R), (sv, R), (\Gamma', sv', R')\rangle \in \eta'$ if, and only if,

1. $\langle(\Gamma, sv), sv, (\Gamma', sv')\rangle \in \eta$ and
2. $(R, R') \in \delta$ and
3. $sv' \cup R'$ is satisfiable.

Set $I' \subseteq I \times \mathcal{R}_\phi$ consists of initial states (atoms) of A_ϕ, that are consistent with the regions in $\mathcal{R}_\phi$, i.e., if $(\Gamma, sv) \in I$ and $R \in \mathcal{R}_\phi$, then $sv \cup R$ is satisfiable.

Satisfiability of quantifier-free formulae over $(\mathbb{R}, <, =)$ is well-known to be decidable [36, 22].

We have the following auxiliary result.

Lemma 3 *Let ϕ be a CLTLoc formula, with clocks V, regions $\mathcal{R}_\phi$, and symbolic valuations $SV(\phi)$. Then:*

1. *If v is a valuation for the clocks of V, then there is exactly one region $R \in \mathcal{R}_\phi$ such that $v \models_\mathbb{R} R$, and exactly one symbolic valuation $sv \in SV(\phi)$ such that $v \models_\mathbb{R} sv$.*
2. *If $R \in \mathcal{R}_\phi$, then there is exactly one symbolic valuation $sv \in SV(\phi)$ such that, for all valuations v such that $v \models_\mathbb{R} R$, it holds that $v \models_\mathbb{R} sv$.*

3. *If $R \in \mathcal{R}_\phi$, $sv \in SV(\phi)$ and $sv \cup R$ is satisfiable, for each valuation v such that $v \models_{\mathbb{R}} R$ it also holds that $v \models_{\mathbb{R}} sv$.*

Proof Property 1 derives from Lemma 3 of [8], and from the fact that the set of clock regions is a partition of the clock space, and a valuation assigns a value to each clock inside the clock space.

To show that property 2 holds, let us consider a region $R \in \mathcal{R}_\phi$, and two valuations v_1, v_2 such that $v_1 \models_{\mathbb{R}} R$ and $v_2 \models_{\mathbb{R}} R$. By property 1, there is exactly one $sv_1 \in SV(\phi)$ such that $v_1 \models_{\mathbb{R}} sv_1$, and one $sv_2 \in SV(\phi)$ such that $v_2 \models_{\mathbb{R}} sv_2$. We need to show that $sv_1 = sv_2$. First of all, recall that region R corresponds to the set of constraints $\chi \in ac(\phi)$ such that $v \models_{\mathbb{R}} \chi$, for every $v \in R$. By definition, symbolic valuation sv_1 contains all comparisons between clocks (e.g., $x < y$), and between clocks and constants (e.g., $x < 1$) that are satisfied by v_1, and similarly for sv_2. By definition of $ac(\phi)$, χ contains, in addition to comparisons between clocks and between clocks and constants, diagonal constraints of the kind $x + c \sim y$ and similar ones. Since $v_1 \models_{\mathbb{R}} \chi$ and $v_2 \models_{\mathbb{R}} \chi$, both valuations satisfy the same constrains on $V \cup const(\phi)$, that is, $C_{v_1} = C_{v_2}$, hence $sv_1 = sv_2$.

Property 3 easily descends from properties 1 and 2. □

Lemma 4 *Let ϕ be a CLTLoc formula. Then, there is (π, σ) such that $(\pi, \sigma) \models \phi$ if, and only if, $\mathcal{A}_\phi^{\mathcal{R}}$ has an accepting run.*

Proof To prove the lemma, we first show that if $(\pi, \sigma) \models \phi$ then there is an accepting run of automaton $\mathcal{A}_\phi^{\mathcal{R}}$. From property 1 of Lemma 3, for every position i there is one, and only one, region $R_i \in \mathcal{R}_\phi$ such that $\sigma, i \models_{\mathbb{R}} R_i$. Also, from property 2 of the same lemma, we have $\sigma, i \models_{\mathbb{R}} sv_i$ and $\sigma, i \models_{\mathbb{R}} sv_i \cup R_i$. Therefore, model (π, σ) induces a sequence $\nu^{\mathcal{R}} = (sv_0 \cup R_0)(sv_1 \cup R_1)\dots$. By Lemma 1, sequence $sv_0 sv_1 \dots$ is recognized by $\mathcal{A}_\phi$ on which $\mathcal{A}_\phi^{\mathcal{R}}$ is based, so there is a run $(\Gamma_0, sv_0)(\Gamma_1, sv_1), \dots$ of $\mathcal{A}_\phi$ that is accepting. We need to show that for each R_i, R_{i+1} in $\nu^{\mathcal{R}}$ it holds that $(R_i, R_{i+1}) \in \delta$. In fact, for two adjacent positions i, $i+1$ there exists $\delta_i > 0$ such that either $x(i+1) = x(i) + \delta_i$ or $x(i+1) = 0$. Therefore, R_{i+1} is a time successor of R_i and the sequence $R_0 R_1 \dots$ is a sequence of successive regions recognized by the automaton $\mathcal{A}_{\mathcal{R}_\phi}$. Finally, by construction, run $(\Gamma_0, sv_0, R_0)(\Gamma_1, sv_1, R_1)\dots$ is recognized by $\mathcal{A}_\phi^{\mathcal{R}}$.

We now show that if automaton $\mathcal{A}_\phi^{\mathcal{R}}$ has an accepting run ρ then ϕ is satisfiable. By Lemma 1, the sequence $\nu = sv_0 sv_1 \dots$ of symbolic valuations occurring in ρ is a symbolic model of ϕ ($\nu \overset{sym}{\models} \phi$) when this is interpreted as a pure CLTL$((\mathbb{R}, <, =))$ formula. Then, to be able to conclude that there is (π, σ) such that $(\pi, \sigma) \models \phi$ we need to show that there exists an arithmetical model for sequence $(sv_0 \cup R_0)(sv_1 \cup R_1), \dots$ induced by ρ. The sequence of regions $R_0 R_1 R_2 \dots$ is such that, for each $i \geqslant 0$, $(R_i, R_{i+1}) \in \delta$. In addition, $sv_0 \cup R_0$ is satisfiable, so there is a valuation v_0 such that $v_0 \models_{\mathbb{R}} sv_0 \cup R_0$. For each $x \in V$ we define $\sigma(0, x) = v_0(x)$. Since $(R_0, R_1) \in \delta$, there exist δ_0 and a valuation v_1 such that $v_1 \models_{\mathbb{R}} R_1$ and, for all $x \in V$, either $v_1(x) = 0$, or $v_1(x) = v_0(x) + \delta_0$. We define, for each $x \in V$, $\sigma(1, x) = v_1(x)$. The process can be iterated ad infinitum, thus building a clock assignment σ that satisfies sequence $R_0 R_1, \dots$.

By Lemma 3, property 3, since by hypothesis $sv_i \cup R_i$ is satisfiable, it holds that $\sigma, i \models_{\mathbb{R}} sv_i$ and also that $\sigma, i \models_{\mathbb{R}} sv_i \cup R_i$; hence we obtain the desired result. □

As mentioned in [3], an infinite sequence of time-successive regions is not guaranteed to capture the notion of time progression – for example, if from position i on we loop ad infinitum in a region such that $1 < x < 2$, time never advances more than 1 time unit from the timestamp of i; hence, time progress is not guaranteed by the construction of $\mathcal{A}_{\phi}^{\mathcal{R}}$. However, this requirement is easily achieved by the CLTLoc formula $\mathbf{GF}\,(x = 0) \vee \mathbf{FG}\,(x > c(x))$ (where $c(x)$ is the biggest constant clock x is compared to), for all clocks $x \in V$. The same condition is considered in [3] to guarantee time progression for timed automata.

Finally, the main result of this section is a direct consequence of Lemma 4.

Theorem 1 *Satisfiability of CLTLoc is PSPACE-complete.*

Proof Given a CLTLoc formula ϕ, by Lemma 4, we can build a generalized Büchi automaton $\mathcal{A}_{\phi}^{\mathcal{R}}$, recognizing symbolic models of ϕ and which has an accepting run if, and only if, ϕ is satisfiable. Therefore, satisfiability of ϕ is decidable, since it is reduced to checking emptiness of $\mathcal{A}_{\phi}^{\mathcal{R}}$.
Satisfiability is PSPACE-hard, as every LTL formula (whose satisfiability problem is PSPACE-complete) is also a CLTLoc formula. PSPACE-membership of CLTLoc can be proved by applying arguments similar to those used in [3] to show that the transition relation of the automaton to be checked for emptiness is computable in PSPACE. Consider a CLTLoc formula ϕ. Let $|\phi|$ be the number of subformulae of ϕ, let N be the number of clock variables in ϕ, and let K be the biggest constant against which the clock variables of ϕ are compared. Since the number of clock regions is $O(N! \cdot K^N)$ [3], the number of states of $\mathcal{A}_{\phi}^{\mathcal{R}}$ is $O(2^{|\phi|} \cdot N! \cdot K^N)$. However, to check the language of $\mathcal{A}_{\phi}^{\mathcal{R}}$ for emptiness, we do not need to build the whole state space, but we can work on-the-fly by considering only a constant number of vertices at a time. Since the space needed to store a vertex of $\mathcal{A}_{\phi}^{\mathcal{R}}$, when using a binary encoding for K, is polynomial in $|\phi| \log(K)$, the algorithm for checking the emptiness of $\mathcal{A}_{\phi}^{\mathcal{R}}$ is in PSPACE. $\square$

4 An SMT-based procedure for solving CLTLoc satisfiability.

In this section, we outline a decision procedure for the satisfiability problem of CLTLoc, by means of an SMT-based technique instead of automata. The technique relies on encoding CLTLoc formulae into formulae of a decidable fragment of first-order logic, which can then be solved by off-the-shelf SMT solvers.

This approach is along the lines of previous works [9] and [8], where a complete procedure, called k-bounded satisfiability, was used to solve CLTL satisfiability by means of a polynomial reduction (in k and the size of the formula) to an SMT problem. This reduction has been implemented in the ae^{2}zot plugin of the Zot tool [2].

To deal with variables that behave like clocks, the method is here extended.

Preliminaries: k-bounded satisfiability for CLTLoc; the logic QF-EUF $\cup$ LRA

Given a CLTLoc formula ϕ on a set of clocks V and an integer $k > 0$, we say that ϕ is *k-bounded satisfiable* if there exists l, $0 \leqslant l \leqslant k$, such that:

1. there exists an ultimately periodic sequence of symbolic valuations of the form: $\nu = sv_0, \dots, sv_{l-1}(sv_l \dots sv_k)^\omega$ which is a symbolic model of ϕ, i.e., $\nu, 0 \overset{sym}{\models} \phi$;
2. there exists an ultimately periodic sequence of clock regions of the form: $R_0 \dots R_{l-1}(R_l \dots R_k)^\omega$ such that for all $0 \leqslant i \leqslant k-1$, R_{i+1} is a time-successor of R_i;
3. there exist $k+1$ valuations $v_0, v_1, \dots v_k$ such that $v_i \models_{\mathbb{R}} sv_i$ and $v_i \in R_i$ for all $0 \leqslant i \leqslant k$, i.e., they satisfy $sv_0, \dots, sv_k$ and they correspond to the sequence of regions $R_0 \dots R_k$;
4. for all $x, y \in V$, for all $0 \leqslant i < k$, if both $v_{i+1}(x) > 0, v_{i+1}(y) > 0$ then $v_{i+1}(x) - v_i(x) = v_{i+1}(y) - v_i(y)$, i.e., all clocks in V, when not reset, progress of the same amount from one position to the next.

The idea is that k-bounded satisfiability considers a finite sequence $sv_0, \dots, sv_k$ of symbolic valuations, which is representative of an ultimately periodic symbolic model for ϕ and which admits a so-called k-bounded arithmetical model $v_0, \dots, v_k$, which is a prefix of an infinite arithmetical model. Moreover, the sequence of traversed time-successive regions corresponding to the k-bounded arithmetical model is also ultimately periodic.

A peculiarity of the SMT-based approach when applied to CLTL and to CLTLoc is that, if the set $SV(\phi)$ of symbolic valuations partitions the space D^λ (with D the domain of the variables in V and λ the size of a symbolic valuation) then a sequence of valuations uniquely induces a sequence of symbolic valuations. This is the case for CLTLoc (and for many interesting constraint systems for CLTL). By solving the k-bounded satisfiability problem for a formula ϕ, we obtain a finite prefix σ_k of some infinite model σ that satisfies the formula. Prefix σ_k is a sequence of valuations, complying with the constraints in the formula, that induces an ultimately periodic symbolic model for ϕ. Hence, unlike automata-based techniques, our approach does not require the explicit construction of the set $SV(\phi)$ because symbolic valuations of the model are deduced from σ_k.

We can avoid building the set of clock regions induced by CLTLoc formulae. In fact, the bounded model σ_k can be constrained to go through time successive regions just by enforcing that clocks behave correctly, i.e., if clocks verify Condition (4) of k-bounded satisfiability, then Condition (2) is also verified for the corresponding regions.

The results of Section 3 are crucial to see that k-bounded satisfiability is indeed bounded, correct and complete. In fact, by Lemma 4, satisfiability checking of ϕ is equivalent to emptiness checking of a (generalized) Büchi automaton $A_\phi^{\mathcal{R}}$. Hence, the language accepted by $A_\phi^{\mathcal{R}}$ is ω-regular, therefore it is nonempty if, and only if, $A_\phi^{\mathcal{R}}$ accepts an ultimately periodic word. A run ρ of $A_\phi^{\mathcal{R}}$ accepting an ultimately periodic word has the form:

$$(\Gamma_0, sv_0, R_0) \dots (\Gamma_{l-1}, sv_{l-1}, R_{l-1})((\Gamma_l, sv_l, R_l) \dots (\Gamma_h, sv_h, R_h))^\omega$$

for some $h > 0$ and some $l \leqslant h$. Run ρ corresponds to a model of the formula ϕ. Therefore, a procedure for solving k-bounded satisfiability may just examine a finite number of bounded models of length k, and it is guaranteed to find a solution, if it exists, whenever $k \geqslant h$. Moreover, since $A_\phi^{\mathcal{R}}$ is finite state, there exists a well-defined bound h_{max} on the value of h, determined by the number of states of $A_\phi^{\mathcal{R}}$. Therefore, the procedure is also complete, since it can be started with a smaller bound k that can then be increased until $k = h_{max}$.

In [8] we show how to solve k-bounded satisfiability, without time progress and regions, for the case of CLTL over a class of arithmetical constraints that include the family of clock constraints of CLTLoc. The k-bounded satisfiability problem is solved through a reduction to the satisfiability problem of the logic composed of the union of the Quantifier-Free Equality Logic with Uninterpreted Functions (QF-EUF for short) and of the Quantifier-Free Linear Real Arithmetic (QF-LRA for short).

QF-EUF is the quantifier-free fragment of first-order logic with built-in equality. Equality is interpreted as identity, while the logic admits also function symbols which are not interpreted. Its syntax is defined by using Boolean operators $\neg, \wedge, \vee$, the equality binary relation $=$, any number of variables and a set of function symbols of various arities. An example is the formula $(x = y) \wedge \neg(y = z) \wedge f(x, z) = f(z, y)$. QF-LRA has Boolean operators, variables interpreted over real numbers, integer constants and the predicate symbols $+, -, <, =$. An example is the formula $(x + y > 0) \wedge \neg(y - 1 < z + 5)$. The syntax of the union QF-EUF $\cup$ LRA of these two logics is defined as the Boolean combination of formulae in QF-EUF and of formulae in QF-LRA. While satisfiability for the combination of theories QF-EUF $\cup$ LRA is, in the general case, NP-complete, the satisfiability for conjunctions of literals is polynomial, as proved in [31] and [26], respectively. The combination of the two theories is decidable [29,31] and its decision procedure is implemented by many SMT-solvers.

Reducing k-bounded satisfiability of CLTLoc to satisfiability of QF-EUF $\cup$ LRA

We now sketch the reduction of k-bounded satisfiability of CLTLoc to QF-EUF $\cup$ LRA. Let ϕ be a CLTLoc formula on the set V of clocks. The goal of the reduction is to define an encoding of ϕ into a QF-EUF $\cup$ LRA formula. The encoding explicitly considers positions $0, \dots, l \dots, k, k+1$, enforcing periodicity by imposing that regions, symbolic valuations and the truh values of all subformulae of ϕ at position l are the same of those at position $k + 1$.

To encode every clock $x \in V$ we introduce an *arithmetic formula function* $\boldsymbol{x} : \mathbb{N} \to \mathbb{R}$; e.g., if x is a clock in ϕ then $\boldsymbol{x}$ is the function associated with it. The intended meaning is that $\boldsymbol{x}(i)$ is the value $v_i(x)$ of x at position i of the k-bounded arithmetical model. To enforce that $\boldsymbol{x}$ behaves like clock x, first we constrain all clocks to be nonnegative at position 0. Then, time progress of the clocks is represented by a function $\boldsymbol{\delta} : \mathbb{N} \to \mathbb{R}$ that forces, from position i to position $i + 1$, every clock in ϕ either to progress of $\boldsymbol{\delta}(i)$ or to be reset at position $i + 1$. Strict monotonicity of time is enforced by making $\boldsymbol{\delta}$ be strictly positive.

We indicate with $Adv(V)$ the following QF-EUF formula:

$$\bigwedge_{x \in V} \boldsymbol{x}(0) \geqslant 0 \wedge \bigwedge_{i=0}^{k} \left(\boldsymbol{\delta}(i) > 0 \quad \wedge \bigwedge_{x \in V} (\boldsymbol{x}(i+1) = \boldsymbol{x}(i) + \boldsymbol{\delta}(i) \vee \boldsymbol{x}(i+1) = 0) \right).$$

Let $\theta \in ac(\phi) \cup S$, where set $ac(\phi)$ was defined in Section 3 and S is the set of constraints occurring in the symbolic valuations in $SV(\phi)$. Let $\boldsymbol{\theta}(i)$ denote the formula θ where all clocks are replaced by their associated function at position i; e.g., if θ is $x \sim y$ then $\boldsymbol{\theta}(i)$ is $\boldsymbol{x}(i) \sim \boldsymbol{y}(i)$.

As in [8], we introduce a variable $\boldsymbol{loop}$ representing position $l \in \mathbb{N}$ in the definition of k-bounded satisfiability.

We indicate with $Per(ac(\phi) \cup S)$ the following QF-EUF formula:

$$\bigwedge_{\theta \in ac(\phi) \cup S} \boldsymbol{\theta}(k+1) \Leftrightarrow \boldsymbol{\theta}(\boldsymbol{loop})$$

that constrains both regions and symbolic valuations at position $\boldsymbol{loop}$ to repeat at position $k+1$.

Let $|\phi|_k$ be the bounded representation of ϕ described in [8]. Formula $|\phi|_k$ considers the Boolean and temporal operators in ϕ and encodes them using standard bounded model checking techniques [16] for encoding LTL into Boolean logic, adapted for the logic QF-EUF $\cup$ LRA. For instance, if ϕ has a subformula of the form $\psi_1 \wedge \psi_2$ then $|\phi|_k$ includes a formula constraining the predicate $\boldsymbol{\psi_1 \wedge \psi_2}$:

$$\bigwedge_{i=0}^{k} \left(\boldsymbol{\psi_1 \wedge \psi_2}\ (i) \Leftrightarrow \boldsymbol{\psi_1}(i) \wedge \boldsymbol{\psi_2}(i)\right).$$

Similarly, if ϕ has a subformula of the form $\psi_1 \mathbf{U} \psi_2$ then $|\phi|_k$ includes the following formula, constraining the predicate $\boldsymbol{\psi_1 \mathbf{U} \psi_2}$:

$$\bigwedge_{i=0}^{k} \left(\boldsymbol{\psi_1 \mathbf{U} \psi_2}\ (i) \Leftrightarrow (\boldsymbol{\psi_2}(i) \vee (\boldsymbol{\psi_1}(i) \wedge \boldsymbol{\psi_1 \mathbf{U} \psi_2}(i+1)))\right)$$

obtained by the fixed point definition of $\psi_1 \mathbf{U} \psi_2$, i.e., $\psi_2 \vee (\psi_1 \wedge \mathbf{X}(\psi_1 \mathbf{U} \psi_2))$, over all positions between 0 and k; in this case, $|\phi_k|$ must also include a formula imposing the eventuality of Until, i.e., that ψ_2 is satisfied at a position i', $\boldsymbol{loop} \leqslant i' \leqslant k$, if ψ_1 holds at k.

Formula $|\phi|_k$ also enforces that position $\boldsymbol{loop}$ is valid (i.e., $1 \leqslant \boldsymbol{loop} \leqslant k$), and it imposes the periodicity of positions $\boldsymbol{loop}$ and $k+1$ (i.e., every subformula ϕ' of ϕ must have the same truth value at positions $k+1$ and l):

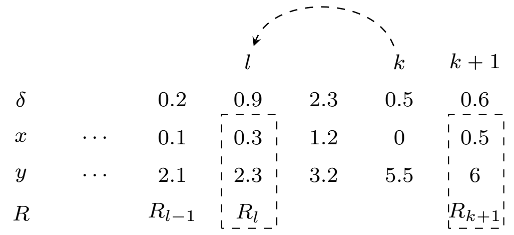
$$\bigwedge_{\phi' \text{ in } \phi} \Big(\boldsymbol{\phi'}(\boldsymbol{loop}) \Leftrightarrow \boldsymbol{\phi'}(k+1)\Big).$$

Finally, we may check the satisfiability of CLTLoc formula ϕ by feeding the SMT solver with the formula:

$$|\phi|_k \wedge Per(ac(\phi) \cup S) \wedge Adv(V). \tag{1}$$

The procedure that checks the satisfiability of a given CLTLoc formula first defines the translation $|\phi|_k$ and then builds the whole Formula (1). Satisfiability must be verified by an SMT-solver implementing a procedure for QF-EUF$\cup$LRA. The outcome of the solver is either "sat", if Formula (1) is satisfiable, or "unsat". Since the translation preserves equisatisfiability, in the former case the original CLTLoc formula ϕ is satisfiable and the model for Formula (1), provided by the solver, can be used to build the model of the original CLTLoc formula; in the latter case, formula ϕ is also unsatisfiable.

While PSPACE-complete, in practice k-bounded satisfiability can be solved efficiently, at least when the value of k is small: checking k-bounded satisfiability is then equivalent to solve a few SMT problems in NP. Obviously, the upper bound for k is in general exponential in the size of the formula.



			l		k	$k+1$
δ		0.2	0.9	2.3	0.5	0.6
x	$\cdots$	0.1	0.3	1.2	0	0.5
y	$\cdots$	2.1	2.3	3.2	5.5	6
R		R_{l-1}	R_l			R_{k+1}

Fig. 1 A (portion of a) bounded model satisfying infinitely often sub formula $(x < y)\mathbf{U}(x = 0)$. Dashed rectangles represents clock regions and the dashed arrow determines the periodic part of the word. Position $k + 1$ is not part of the model of the formula but it is required to build correctly the periodicity of the model.

An example of translation

Figure 1 shows a portion of a model satisfying the CLTLoc formula $(x < y)\mathbf{U}(x = 0)$ infinitely often. Only the periodic part is depicted, since the prefix is straightforward. To make the example non-trivial let us consider $c(x) = 1$ and $c(y) = 2$. Set $ac(\phi)$ of clock constraints induced by the formula (see Section 3) is $\{x = 0, 0 < x, x < 1, x = 1, 1 < x, y = 0, 0 < y, y < 1, y = 1, 1 < y, y < 2, y = 2, 2 < y, x < y, x = y, y < x, x + 1 < y, x + 1 = y, y < x + 1\}$. Set $SV(\phi)$ is not entirely defined here, for the sake of space, but the set of constraints defining the symbolic valuations in $SV(\phi)$ is $S = \{x = 0, 0 < x, x < 1, x = 1, 1 < x, y = 0, 0 < y, y < 1, y = 1, 1 < y, y < 2, y = 2, 2 < y, x < y, x = y, y < x\}$.

Formula $Adv(V)$ is straightforward. We show the definition of $Per(ac(\phi) \cup S)$, for all $\theta \in ac(\phi)$, which enforces that regions R_l and R_{k+1} (dashed rectangles) are equal. The similar formulae for $\theta \in S$, to enforce also the periodicity of symbolic valuations, are omitted.

$$\bigwedge \left(\begin{array}{c} \boldsymbol{x}(\boldsymbol{loop}) = 0 \Leftrightarrow \boldsymbol{x}(k+1) = 0 \\ \boldsymbol{x}(\boldsymbol{loop}) < 1 \Leftrightarrow \boldsymbol{x}(k+1) < 1 \\ \dots \\ \boldsymbol{x}(\boldsymbol{loop}) < \boldsymbol{y}(\boldsymbol{loop}) \Leftrightarrow \boldsymbol{x}(k+1) < \boldsymbol{y}(k+1) \\ \dots \\ \boldsymbol{x}(\boldsymbol{loop}) + 1 < \boldsymbol{y}(\boldsymbol{loop}) \Leftrightarrow \boldsymbol{x}(k+1) + 1 < \boldsymbol{y}(k+1) \\ \dots \end{array} \right)$$

In Figure 1, $R_l = \{x(l) < y(l), 0 < x(l) < 1, 1 < y(l)\}$ and $R_{k+1} = \{x(k+1) < y(k+1), 0 < x(k+1) < 1, 1 < y(k+1)\}$.

To deal with the Until operator in the example, $|\phi|_k$ will include the formula:

$$\bigwedge_{i=0}^{k} \left(\boldsymbol{\phi}(i) \Leftrightarrow (\boldsymbol{x}(i) = 0) \vee (\boldsymbol{x}(i) < \boldsymbol{y}(i) \wedge \boldsymbol{\phi}(i+1))\right).$$

The eventuality of Until, i.e., a positive occurrence of subformula $(x = 0)$ in the loop, is guaranteed by the formula, using another variable i_ϕ:

$$\boldsymbol{\phi}(k) \Rightarrow \boldsymbol{loop} \leqslant i_\phi \leqslant k \wedge (\boldsymbol{x}(i_\phi) = 0)$$

that imposes subformula $(x = 0)$ to be satisfied at position i_ϕ, between l and k, when $\boldsymbol{\phi}(k)$ holds.

In Figure 1, the value of clock x at position k, $x_k = 0$, satisfies the eventuality for ϕ and for all the positions from l to k it holds that $x < y$. Hence, ϕ is satisfied infinitely often in the loop.

5 MITL, MTL$_{(0,\infty)}$, QTL, and pMITL

Let $\mathcal{I}$ be the set of intervals (i.e., convex sets over $\mathbb{R}$) of the form $\langle a, b\rangle$ or of the form $\langle a, +\infty)$, where $0 \leqslant a < b$ are integer constants, $\langle$ is either (or [, and $\rangle$ is) or]. Given a finite alphabet AP of atomic propositions, the syntax of (well-formed) formulae of MITL is defined as:

$$\phi := p \mid \phi \wedge \phi \mid \neg\phi \mid \phi \mathbf{U}_I \phi$$

where $p \in AP$ and $I \in \mathcal{I}$. We often write, as customary, $\mathbf{U}$ instead of $\mathbf{U}_{(0,+\infty)}$.

Boolean operators $\vee, \top, \bot, \Rightarrow$ and the *globally* $\mathbf{G}_I$ and *eventually* $\mathbf{F}_I$ operators can be defined by the usual abbreviations, e.g. $\mathbf{F}_I\phi = \top\mathbf{U}_I\phi$ and $\mathbf{G}_I\phi = \neg\mathbf{F}_I(\neg\phi)$.

A *signal* is a function $M : \mathbb{R}_+ \to \wp(AP)$ which is assumed to be finitely variable, i.e., such that in every finite interval there is a finite number of changes in the value of the atomic propositions in AP.

The continuous semantics of MITL is defined in Table 2, for every $t \in \mathbb{R}_+$ and for every signal M. Notice that in the definition of the semantics an interval I is interpreted as the corresponding set of real numbers.

$$\begin{aligned}
M, t \models p &\Leftrightarrow p \in M(t) \qquad p \in AP\\
M, t \models \neg\phi &\Leftrightarrow M, t \not\models \phi\\
M, t \models \phi \wedge \psi &\Leftrightarrow M, t \models \phi \text{ and } M, t \models \psi\\
M, t \models \phi\mathbf{U}_I\psi &\Leftrightarrow \exists t' > t\ t' - t \in I, M, t' \models \psi \text{ and } \forall\ t < t'' < t'\ M, t'' \models \phi
\end{aligned}$$

Table 2 Continuous semantics of MITL.

An MITL formula ϕ is *satisfiable in the continuous semantics* if there exists a signal M such that $M, 0 \models \phi$. In this case, M is called a *continuous model* of ϕ.

The pointwise semantics is defined by introducing a relation $\models$, defined in Table 3 for every timed ω-word (π, τ) and for every position $i \in \mathbb{N}$. An MITL formula ϕ is *satisfiable in the pointwise semantics* if there exists a timed ω-word (π, τ) such that $(\pi, \tau), 0 \models \phi$ – also written as $(\pi, \tau) \models \phi$. In this case, (π, τ) is called a *pointwise model* of ϕ.

A useful operator on timed words is "next" $\mathbf{X}_I$, with the intuitive meaning that $\mathbf{X}_I\phi$ holds at position i if ϕ is true at position $i + 1$, and the difference of timestamps $\tau(i+1) - \tau(i)$ is in I. Since we adopted the strict version of $\mathbf{U}_I$, $\mathbf{X}_I$ can be defined as $\mathbf{X}_I\phi = \bot\mathbf{U}_I\phi$. It is also possible to define MITL with the non-strict version of $\mathbf{U}_I$, but in this case it is necessary to introduce also the (non-metric) next operator $\mathbf{X}$ as primitive.

$$
\begin{aligned}
(\pi,\tau),i \models p &\Leftrightarrow p \in \pi(i) \text{ for } p \in AP\\
(\pi,\tau),i \models \neg\phi &\Leftrightarrow (\pi,\tau),i \not\models \phi\\
(\pi,\tau),i \models \phi \wedge \psi &\Leftrightarrow (\pi,\tau),i \models \phi \text{ and } (\pi,\tau),i \models \psi\\
(\pi,\tau),i \models \phi\mathbf{U}_I\psi &\Leftrightarrow \exists j > i : \tau(j) - \tau(i) \in I, (\pi,\tau),j \models \psi \text{ and } \forall\, i < k < j \;\; (\pi,\tau),k \models \phi
\end{aligned}
$$

Table 3 Pointwise semantics of MITL.

It is sometimes useful to extend MITL with past-time operators, and in particular with the "since" temporal operator $\phi\mathbf{S}_I\psi$, whose semantics is the dual of the one of $\mathbf{U}_I$ ($\mathbf{P}_I$ is the past-time dual of $\mathbf{F}_I$, so $\mathbf{P}_I\phi = \top\mathbf{S}_I\phi$). MITL with past-time operators (MITL+Past) is strictly more expressive than MITL [17] over both the continuous and pointwise semantics. Nevertheless, our encoding for the continuous semantics, presented in Sect. 7, can also deal with past-time operators, so the examples of Sect. 8 will also include them.

In general, MITL formulae may give different results when interpreted over the pointwise semantics and over the continuous semantics. An exhaustive discussion of these two cases can be found in [20].

MTL$_{(0,\infty)}$ and QTL. A syntactic restriction of MITL, called MTL$_{(0,\infty)}$, is one in which in intervals $I = \langle a,b\rangle$ either $a = 0$ or $b = \infty$. The logic QTL is MTL$_{(0,\infty)}$ in which intervals are only of the form $(0,1)$. Despite their apparent simplicity, MTL$_{(0,\infty)}$ and QTL have the same expressive power of MITL [25].

Projection-closed MITL. Finally, we define an extension of MITL, here called *projection-closed MITL*, pMITL for short [23]. This logic, defined on timed words, is obtained by adding a set pAP of $n \geqslant 0$ propositional variables $q_1,\ldots,q_n$, which can be existentially quantified. The logic is called "projection-closed", since the actual extension to MITL is its capacity of adding new propositional variables, which can then be eliminated ("projected" away) by an external existential quantification, hence without extending the alphabet. This allows the definition of timed ω-languages that are not counter-free [23] (e.g., "the number of occurrences of event a is even"), which cannot be defined in MITL.

The syntax of pMITL is defined by the clause: $\exists q_1 \ldots q_n\phi$, where $pAP = \{q_1,\ldots,q_n\}$ for some $n \geqslant 0$ and ϕ is a MITL formula on the alphabet $AP \cup pAP$. To follow our definitions of pointwise semantics, the semantics of pMITL may, e.g., be defined by the semantic clause:

$$
\begin{aligned}
(\pi,\tau),0 \models \exists q_1 \ldots q_n\phi \Leftrightarrow \text{ there exists } \pi' :\mathbb{N} \to \wp(AP \cup pAP) \;\mid\; &(\pi',\tau),0 \models \phi \text{ and}\\
&\forall j \in \mathbb{N}\; \pi'(j) - pAP = \pi(j)
\end{aligned}
$$

The meaning is that $(\pi,\tau),0 \models \exists q_1 \ldots q_n\phi$ if there exists a mapping $\pi' : \mathbb{N} \to \wp(AP \cup pAP)$ such that $(\pi',\tau),0 \models \phi$ and, at every position j, $\pi'(j)$ may differ from $\pi(j)$ only in the presence of a subset of $q_1,\ldots,q_n$. We remark that, since the existential quantification cannot be nested within temporal operators, the quantified formula is only evaluated in the origin.

Notice that the alphabet of pMITL is only apparently extended with the set pAP: every proposition in pAP must be existentially quantified, hence the actual alphabet is still set AP. This is reflected in the semantics, where in a timed word (π, τ) the mapping π still considers alphabet AP, i.e., $\pi : \mathbb{N} \to \wp(AP)$. This point is important when comparing the expressiveness of pMITL with CLTLoc.

Complexity It is well-known that satisfiability of MITL and pMITL is EXPSPACE-complete. This is consistent with PSPACE-completeness of CLTLoc, since our translation from pMITL to CLTLoc of Section 6 is indeed exponential.

6 Comparing CLTLoc and pMITL on the pointwise semantics

In this section we show that, over timed words, CLTLoc and pMITL have the same expressive power. To this end, we devise two new semantics-preserving transformations, from pMITL formulae to CLTLoc ones and vice-versa.

Satisfiability of CLTLoc over timed words.

To compare CLTLoc with pMITL we introduce the satisfiability of CLTLoc formulae over *timed ω-words*. A timed ω-word over $\wp(AP)$ is a pair (π, τ) where $\pi : \mathbb{N} \to \wp(AP)$ and τ is a monotonic function $\tau : \mathbb{N} \to \mathbb{R}_+$ such that $\forall i\ \tau(i) < \tau(i{+}1)$ (strong monotonicity). The value $\tau(i)$ is called the *timestamp* at position i, $i \in \mathbb{N}$. Given a CLTLoc interpretation (π, σ), let τ be such that $\tau(i) = \sigma(i, \mathit{Now})$. Then, (π, τ) is called the timed ω-word associated with (π, σ) and it is denoted by $[(\pi, \sigma)]$.

A relation $\models$ can be defined for every timed ω-word (π, τ) as follows. Let ϕ be a CLTLoc formula and $x_1, \dots, x_n \in V$ be the clocks occurring in it. Define:

$$(\pi, \tau), 0 \models \exists x_1 \cdots x_n \phi \Leftrightarrow \text{ there exists } \sigma \mid (\pi, \sigma), 0 \models \phi \text{ and } (\pi, \tau) = [(\pi, \sigma)].$$

A CLTLoc formula ϕ is satisfiable *over timed ω-words* if $(\pi, \tau), 0 \models \exists x_1 \cdots x_n \phi$, for some (π, τ), equivalently written $(\pi, \tau) \models \phi$. The definition of satisfiability over timed ω-words requires that all clocks are existentially quantified and the (quantified) formula is in prenex normal form.

For example, let $\phi = \mathbf{G}(p \wedge x = 0)$, and let (π, τ) be the timed ω-word where, at every position, p occurs and time is increasing by one: $\pi(i) = \{p\}$, $\tau(i+1) = \tau(i) + 1$ for every $i \geqslant 0$. It is enough to let $\sigma(i, x) = 0$ for each position $i \geqslant 0$, with $[(\pi, \sigma)] = (\pi, \tau)$, to have $(\pi, \sigma) \models \phi$: hence, $(\pi, \tau) \models \phi$.

The definition of satisfiability over timed words allows for pathological cases when the truth of CLTLoc formulae depends only on the values of clocks. For instance, let $\psi = \mathbf{F}(x = 0)$, and let (π, τ) be a timed ω-word where at each position π is empty: $\pi(i) = \varnothing$, for every $i \geqslant 0$. It is enough to let $\sigma(0, x) = 0$ for a σ such that $[(\pi, \sigma)] = (\pi, \tau)$, to have $(\pi, \sigma) \models \psi$: hence, $(\pi, \tau) \models \psi$. Let now σ' be a clock assignment such that $\sigma'(i, x) > 0$ for every $i \geqslant 0$ and $[(\pi, \sigma')] = (\pi, \tau)$. Therefore, formula ψ is false on (π, σ'), i.e., $(\pi, \tau) \models \neg\psi$. Hence, ψ and $\neg\psi$ may be satisfied on the same timed word (π, τ), although on different models (π, σ) and (π, σ'), because both $(\pi, \tau), 0 \models \exists x(\psi)$ and $(\pi, \tau), 0 \models \exists x(\neg\psi)$ hold. This occurs because the definition of satisfiability over timed ω-words requires the (implicit) existential quantification of all clocks.

We now define the notion of equivalence between pMITL and CLTLoc formulae.

Definition 2 A pMITL formula $\exists q_1 \ldots q_n \phi$ and a CLTLoc formula ψ with set of clocks V are *equivalent* if, for all timed words (π, τ) on alphabet π, $(\pi, \tau), 0 \models \exists q_1 \ldots q_n \phi$ if, and only if, $(\pi, \tau), 0 \models \exists x_1 \cdots x_{|V|} \psi$.

6.1 From pMITL to CLTLoc

To transform pMITL formulae into CLTLoc ones, we first remark that the following standard equivalences hold for MITL (and pMITL) formulae, where $\mathbf{U}$ is an abbreviation for $\mathbf{U}_{(0,\infty)}$ – note that $\mathbf{U}_{(0,\infty)}\phi \equiv \mathbf{U}_{[0,\infty)}\phi$ because we adopted the strict version of the until operator.

Lemma 5 (From Sec. 9 of [21]) *Let (π, τ) be a timed word and $0 < a \leqslant b$. Then, for all $i \geqslant 0$,*

$$(1) \quad (\pi, \tau), i \models \phi \mathbf{U}_{[a,b\rangle}\psi \Leftrightarrow (\pi, \tau), i \models \phi \mathbf{U}\psi \wedge \mathbf{G}_{(0,a)}(\phi \wedge \phi \mathbf{U}\psi) \wedge \mathbf{F}_{[a,b\rangle}(\psi)$$

$$(2) \quad (\pi, \tau), i \models \phi \mathbf{U}_{(a,b\rangle}\psi \Leftrightarrow (\pi, \tau), i \models \phi \mathbf{U}\psi \wedge \mathbf{G}_{(0,a]}(\phi \wedge \phi \mathbf{U}\psi) \wedge \mathbf{F}_{(a,b\rangle}(\psi)$$

$$(3) \quad (\pi, \tau), i \models \phi \mathbf{U}_{(0,b\rangle}\psi \Leftrightarrow (\pi, \tau), i \models \phi \mathbf{U}\psi \wedge \mathbf{F}_{(0,b\rangle}(\psi)$$

When b is ∞, equivalences (1), (2) can be simplified, respectively, to $\phi \mathbf{U}_{[a,\infty)}\psi \equiv \phi \mathbf{U}\psi \wedge \mathbf{G}_{(0,a)}(\phi \wedge \phi \mathbf{U}\psi)$ and $\phi \mathbf{U}_{(a,\infty)}\psi \equiv \phi \mathbf{U}\psi \wedge \mathbf{G}_{(0,a]}(\phi \wedge \phi \mathbf{U}\psi)$.

Thanks to Lemma 5, we can focus only on temporal operators $\mathbf{U}$ and $\mathbf{F}_I$.

We also have the following result, which shows that a formula $\mathbf{F}_{\langle a,b\rangle}(\psi)$ must stay true for at least $b - a$ time units.

Lemma 6 *Consider formula $\mathbf{F}_{\langle a,b\rangle}(\psi)$. For any timed word (π, τ) there cannot be two positions $i < j$ such that $\tau(j) - \tau(i) < b - a$, $(\pi, \tau), i \not\models \mathbf{F}_{\langle a,b\rangle}(\psi)$, $(\pi, \tau), j \not\models \mathbf{F}_{\langle a,b\rangle}(\psi)$, and there is $i < k < j$ such that $(\pi, \tau), k \models \mathbf{F}_{\langle a,b\rangle}(\psi)$.*

Proof Assume that there are three positions $i < k < j$ in (π, τ) that violate the property. Then, there is position $k' > k$ such that $\tau(k') - \tau(k) \in \langle a, b\rangle$ and $(\pi, \tau), k' \models \psi$; in addition, for any position k'' such that $\tau(k'') \in \langle a + \tau(i), b + \tau(i)\rangle$ or $\tau(k'') \in \langle a + \tau(j), b + \tau(j)\rangle$ it is $(\pi, \tau), k'' \not\models \psi$. Since $\tau(j) - \tau(i) < b - a$, then $a + \tau(j) < b + \tau(i)$, hence for any position k'' such that $\tau(k'') \in \langle a + \tau(i), b + \tau(j)\rangle$ it is $(\pi, \tau), k'' \not\models \psi$. But $a + \tau(i) < \tau(k') < b + \tau(j)$, which leads to a contradiction. □

The following corollary descends from Lemma 6, and is exemplified in Figure 2.

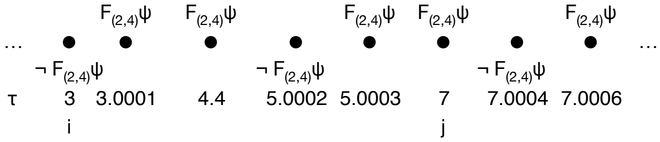


Fig. 2 Example of maximum number of changes in the value of formula $\mathbf{F}_{(2,4)}\psi$ in interval of length 4.

Corollary 1 *For any timed word (π, τ) and a pair of positions i, j such that $\tau(j) - \tau(i) \leqslant b$, between i (included) and j (excluded) there cannot be more than $2\left\lceil \frac{b}{b-a} \right\rceil$ positions k such that the value of $\mathbf{F}_{\langle a,b \rangle}(\psi)$ differs in k and $k+1$.*

Proof Suppose $\mathbf{F}_{\langle a,b \rangle}(\psi)$ is false at position i and true in $i+1$. Then, by Lemma 6, $\mathbf{F}_{\langle a,b \rangle}(\psi)$ cannot be false again until a position i' with timestamp $\tau(i') = \tau(i)+b-a$. Hence, in interval $[\tau(i), \tau(i)+b-a)$ formula $\mathbf{F}_{\langle a,b \rangle}(\psi)$ has at most 2 change points, from false to true at i, and then from true to false at $i'-1$. The proof is concluded by remarking that in an interval of length b there are at most $\left\lceil \frac{b}{b-a} \right\rceil$ intervals of length $b-a$. □

Let us consider a pMITL formula $\exists q_1 \dots q_n \phi$ such that AP is the set of its non-quantified propositional letters, and $pAP = \{q_1 \dots q_n\}$ is the set of quantified ones. The corresponding CLTLoc formula ϕ' is built upon the set AP, plus a set of constraints on freshly introduced clocks V_{pAP} that correspond to set pAP; we use clocks and not propositional letters for set pAP to mimic the semantics of the existential quantification, as shown in Theorem 2. Moreover, we introduce additional clocks V_{C_ϕ} which help capture the semantics of the subformulae of ϕ by measuring the passing of time.

The CLTLoc formula ϕ' corresponding to $\exists q_1 \dots q_n \phi$ is made of two parts: a formula m_ϕ which is a syntactic transformation of ϕ that is defined recursively (i.e., each subformula θ of ϕ has a corresponding m_θ), and a formula capturing the semantics of the clocks V_{C_ϕ}. We will inductively show that CLTLoc formula ϕ' is such that, for any subformula θ of ϕ, m_θ is true in a position of a timed word if, and only if, θ is also true there.

We first describe the new clocks in set V_{C_ϕ} and their semantics.

- For each subformula θ, V_{C_ϕ} contains two clocks, z^0_θ and z^1_θ, which are reset, in an alternate manner, at every position i in which θ holds and it does not hold in $i-1$ or in $i+1$. We use these clocks to measure the time distance between changes in the value of θ. This is needed, as shown below, when θ is an argument of a $\mathbf{F}_{\langle a,b \rangle}$ formula.
- V_{C_ϕ}, includes two clocks z^0_δ and z^1_δ, which measure the distance between two consecutive elements of the timed word; that is, at each position they are alternatively reset. In addition, V_{C_ϕ} contains clock *Now* described in Section 2, which is reset in the origin and never again, which measures absolute time.
- If θ is of the form $\mathbf{F}_{\langle a,b \rangle}\psi$, V_{C_ϕ} includes $2d$ auxiliary clocks, where $d = 2\lceil \frac{b}{b-a} \rceil + 1$, $x^0_\theta, \hat{x}^0_\theta, \dots x^{d-1}_\theta, \hat{x}^{d-1}_\theta$, whose role and behavior is described later on.

The following CLTLoc formulae capture the properties of clocks $\{z^n_\theta\}_{n \in \{0,1\}}$ and $\{z^n_\delta\}_{n \in \{0,1\}}$.

Formula (2) enforces that the occurrence of a change in the truth of subformula θ entails the reset of one of z^0_θ, z^1_θ, and that clock z^0_θ is reset in the origin.

$$z^0_\theta = 0 \wedge \mathbf{XG}\Big(m_\theta \wedge (\neg \mathbf{X} m_\theta \vee \neg \mathbf{Y} m_\theta) \quad \Leftrightarrow \quad z^0_\theta = 0 \vee z^1_\theta = 0\Big). \tag{2}$$

Let $a \in \mathbb{N}$ and value $\overline{a}_k$ be $(a \bmod k)$. The clocks associated with a subformula θ are alternatively reset. Hence, between any two resets of clock z^0_θ there must be a reset of clock z^1_θ, and vice-versa:

$$\mathbf{G}\left(\bigwedge_{i\in\{0,1\}}\left(z^i_\theta = 0 \Rightarrow z^{\overline{(i+1)}_2}_\theta > 0 \wedge \mathbf{X}\left((z^{\overline{(i+1)}_2}_\theta = 0)\mathbf{R}(z^i_\theta \neq 0)\right)\right)\right). \tag{3}$$

Formula 4 defines that clock $z^0_\delta = 0$ (resp. $z^1_\delta = 0$) is reset in all even (resp. odd) positions, and that clock *Now* is reset only in the origin.

$$z^0_\delta = 0 \wedge \mathbf{G}\left(\bigwedge_{i\in\{0,1\}}\left(z^i_\delta = 0 \Rightarrow z^{\overline{(i+1)}_2}_\delta > 0 \wedge \mathbf{X}\left(z^{\overline{(i+1)}_2}_\delta = 0\right)\right)\right) \wedge \mathit{Now} = 0 \wedge \mathbf{XG}(\mathit{Now} > 0). \tag{4}$$

We abbreviate by $\mathtt{ck}_\theta$ the conjunction of formulae (2)–(3), and we call `zddef` Formula (4).

We now define the syntactic transformation m_θ. The transformation is inductively defined, with the cases for propositional letters in $AP \cup pAP$ being the base ones.

- $\theta = p \in AP$: we simply have $m_p \triangleq p$.
- $\theta = q \in pAP$: we introduce clock $c_q \in V_{pAP}$ and define $m_q \triangleq (c_q = 0)$.
- $\theta = \neg\psi$: in this case it is $m_{\neg\psi} \triangleq \neg m_\psi$.
- $\theta = \gamma \wedge \psi$: we have: $m_{\gamma\wedge\psi} = m_\gamma \wedge m_\psi$.
- $\theta = \gamma\mathbf{U}_{(0,\infty)}\psi$: we need to take into account that the $\mathbf{U}$ operator in pMITL is strict, whereas it is not in CLTLoc, hence we have the following:

$$m_{\gamma\mathbf{U}_{(0,\infty)}\psi} = \mathbf{X}\big(m_\gamma \mathbf{U} m_\psi\big)\,.$$

In all cases above, we replace each occurrence of m_ψ and m_γ with the corresponding definition. For example, consider the pMITL formula $\phi = \exists q(p\mathbf{U}_{(0,\infty)}(\neg p \wedge q) \wedge q\mathbf{U}_{(0,\infty)}\neg r)$. Then, $m_\phi = \mathbf{X}(p\mathbf{U}(\neg p \wedge c_q = 0)) \wedge \mathbf{X}((c_q = 0)\mathbf{U}\neg r)$.

- $\theta = \mathbf{F}_{\langle a,b\rangle}\psi$: As mentioned above, to capture the semantics of the $\mathbf{F}_{\langle a,b\rangle}$ operator we need to introduce $2d$ associated auxiliary clocks $\{x^n_\theta, \hat{x}^n_\theta\}_{n\in[0,d-1]}$. Then, before defining the transformation $m_{\mathbf{F}_{\langle a,b\rangle}\psi}$, we present the formulae that formalize the behavior of these auxiliary clocks. For simplicity, we focus on the case in which $\langle a,b\rangle = (a,b)$, the other cases being similar.

The $2d$ clocks $\{x^n_\theta, \hat{x}^n_\theta\}_{n\in[0,d-1]}$ associated with a subformula $\mathbf{F}_{(a,b)}\psi$ are reset in pairs, i.e., a reset of x^j_θ is immediately followed by a reset of $\hat{x}^j_\theta$, as defined by Formula 5. In addition, clock $\hat{x}^{d-1}_\theta$ is reset in the origin (where no other clock $\hat{x}^j_\theta$ is reset).

$$\mathbf{G}\left(\bigwedge_{j=0}^{d-1}\left(x^j_\theta = 0 \Leftrightarrow \mathbf{X}\left(\hat{x}^j_\theta = 0\right)\right)\right) \wedge \hat{x}^{d-1}_\theta = 0 \wedge \bigwedge_{j=0}^{d-2}\hat{x}^j_\theta > 0. \tag{5}$$

The order in which auxiliary clocks are reset is defined by Formulae (6) and (7). More precisely, Formula (6) states that no two auxiliary clocks are reset at the same time. In addition, clock x^0_θ is reset in the origin.

$$x^0_\theta = 0 \wedge \mathbf{XG}\left(\bigwedge_{i=0}^{d-1}\bigwedge_{j=0,i\neq j}^{d-1}\neg(x^i_\theta = 0 \wedge x^j_\theta = 0)\right). \tag{6}$$

Formula (7), instead, states that the resets of clocks x_θ^i are circularly ordered. That is, if $x_\theta^i = 0$, then, from the next position, all clocks are strictly greater than 0 until $x_\theta^{\overline{i+1}_d} = 0$ occurs.

$$\mathbf{G}\left(\bigwedge_{i=0}^{d-1}\left(x_\theta^i = 0 \Rightarrow \mathbf{X}\left((x_\theta^{\overline{i+1}_d} = 0)\mathbf{R}\bigwedge_{j\in[0,d-1],\, j\neq\overline{i+1}_d}(x_\theta^j > 0)\right)\right)\right). \tag{7}$$

Let us now explain when clocks $\{x_\theta^n\}_n$ (hence also clocks $\{\hat{x}_\theta^n\}_n$) are reset.

The idea is that a clock x_θ^j is reset at a position k when in interval $[\tau(k), \tau(k+1)]$ there is a timestamp t where the conditions for $\mathbf{F}_{(a,b)}\psi$ to become true or false would be met. More precisely, this corresponds to the following two – possibly overlapping – cases for resetting x_θ^j in k, which are also exemplified in Figure 3 (in the conditions m_ψ appears instead of ψ, because m_ψ is the CLTLoc formula that holds wherever pMITL formula ψ does).

(a) There is a timestamp t such that: (i) $\tau(k) \leqslant t < \tau(k+1)$; (ii) there is a position k' such that $(\pi, \tau), k' \models m_\psi$ and $\tau(k') - t = b$; and (iii) there is no position $k'' < k'$ such that $(\pi, \tau), k'' \models m_\psi$ and $\tau(k'') - t \in (a, b)$ (i.e., $t + a < \tau(k'') < t + b$ or, equivalently, $\tau(k') - \tau(k'') < b - a$).

(b) There is a timestamp t such that: (i) $\tau(k) < t \leqslant \tau(k+1)$; (ii) there is a position k' such that $(\pi, \tau), k' \models m_\psi$ and $\tau(k') - t = a$; and (iii) there is no position $k'' > k'$ such that $(\pi, \tau), k'' \models m_\psi$ and $\tau(k'') - t \in (a, b)$ (i.e., $\tau(k'') - \tau(k') < b - a$).

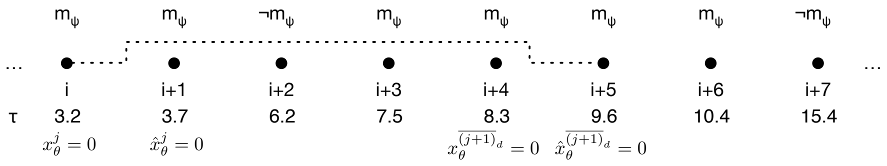


Fig. 3 Example of reset of clocks x_θ^j and $\hat{x}_\theta^j$ for $\theta = \mathbf{F}_{(2,4)}\psi$.

Let us consider Figure 3. Position i is such that condition (a) holds. In fact, for $t = 3.5$ and $k' = i + 3$ we have that m_ψ holds in k', $\tau(k') - t = 7.5 - 3.5 = 4$, and there is no position $k'' < k'$ where m_ψ holds such that $\tau(k'') - t \in (2, 4)$. In other words, there is a timestamp in $[\tau(i), \tau(i + 1))$ in which $\mathbf{F}_{(2,4)}\psi$ would become true, as informally depicted by the dotted line in Figure 3. Hence, there is a clock x_θ^j that is reset at position i, and the associated $\hat{x}_\theta^j$ is reset at position $i + 1$. Since all timestamps t' in $(3.5, 8.3)$ are such that there is a position i' such that $\tau(i') - t' \in (2, 4)$, clock $\hat{x}_\theta^{\overline{(j+1)_d}}$ is not reset again before position $i + 4$ (recall that clocks are reset circularly). Position $i + 4$, however, satisfies condition (b) above. In fact, if one considers $t = 8.4$ and $k' = i + 6$, then m_ψ holds in k', $\tau(k') - t = 10.4 - 8.4 = 2$, and there is no $k'' > k'$ where m_ψ holds such that $\tau(k'') - t \in (2, 4)$. In other words, there is a timestamp in $(\tau(i + 4), \tau(i + 5)]$ in which $\mathbf{F}_{(2,4)}\psi$ would become false, as represented by the dotted line in Figure 3.

Clocks x^j_θ and $\hat{x}^j_\theta$ are used to establish when $\mathbf{F}_{(a,b)}\psi$ is true or false. We now formalize conditions (a) and (b) by means of CLTLoc formulae.

Formula (8) defines the necessary condition for a clock x^j_θ to be reset at a position k.

$$\mathbf{G}\left(\bigwedge_{j=0}^{d-1}\left(x^j_\theta = 0 \Rightarrow \mathtt{up@b}^j_\theta \vee \mathtt{down@a}^j_\theta\right)\right). \tag{8}$$

This condition rests on two possibilities, which are captured by abbreviations $\mathtt{up@b}^j_\theta$ and $\mathtt{down@a}^j_\theta$ defined below, and which correspond, respectively, to conditions (a) and (b).[1]

More precisely, $\mathtt{up@b}^j_\theta$ is the following:

$$\mathtt{up@b}^j_\theta = \mathbf{X}\left(x^j_\theta > 0\,\mathbf{U}\left(m_\psi \wedge \begin{pmatrix} z_\delta \geqslant b-a & \vee \\ \neg\mathbf{Y}m_\psi \wedge z_\psi \geqslant b-a \end{pmatrix} \wedge \begin{pmatrix} x^j_\theta \geqslant b \ \wedge \\ \hat{x}^j_\theta < b \end{pmatrix}\right)\right).$$

It states that there is a position $k' > k$ where the CLTLoc formula m_ψ holds, and there is no position $k'' < k'$ where m_ψ holds and such that $\tau(k') - \tau(k'') < b - a$ or, equivalently, such that $\tau(k'')$ belongs to interval $(\tau(k') - (b-a), \tau(k'))$. This, in turn, can occur in two cases: either

(i) there is no k'' such that $\tau(k'') \in (\tau(k')-(b-a), \tau(k'))$, i.e. $\tau(k')-\tau(k'-1) \geqslant b-a$, which in turn corresponds to one of z^0_δ, z^1_δ being $\geqslant b - a$;
or
(ii) for all k'' such that $\tau(k'') \in (\tau(k')-(b-a), \tau(k'))$, m_ψ does not hold there, which corresponds to saying that m_ψ does not hold in $k' - 1$ (i.e., $\neg\mathbf{Y}m_\psi$ holds in k', so one of the clocks z^0_ψ, z^1_ψ is reset there), and the last time when m_ψ switched from true to false was at least $b - a$ time units before $\tau(k')$ – that is, the clock between z^0_ψ and z^1_ψ that is not reset in k' is $\geqslant b - a$.

Because of Corollary 1, between $\tau(k)$ and $\tau(k) + b$ $\mathbf{F}_{(a,b)}\psi$ cannot change value more than $d - 1$ times, and we have d clocks $x^0_\theta, \ldots x^{d-1}_\theta$, so x^j_θ is not reset again before k'. Hence, the condition that there is a timestamp $t \in [\tau(k), \tau(k + 1))$ such that $\tau(k') - t = b$ corresponds to saying that in k' clock x^j_θ is $\geqslant b$ and $\hat{x}^j_\theta$ is $< b$.

Let us consider Figure 3 again. Clock x^j_θ is reset in position i, and position $i + 3$ is such that the previous one, $i + 2$, is closer than $b - a = 2$ time units, but m_ψ does not hold there (i.e., $\neg\mathbf{Y}m_\psi$ holds in $i + 3$). In addition, clocks z^0_ψ and z^1_ψ, by Formula (2), are reset at positions $i + 1$ and $i + 3$, hence the clock that is not reset in $i + 3$ has value $3.8 > 2 = b - a$, so $z_\psi \geqslant b - a$ holds in $i + 3$.

Abbreviation $\mathtt{down@a}^j_\theta$ is the following:

$$\mathtt{down@a}^j_\theta = \mathbf{X}\left(x^j_\theta > 0\,\mathbf{U}\left(m_\psi \wedge \left(\begin{matrix} \mathbf{X}(z_\delta \geqslant b-a) \qquad \vee \\ \left(\begin{matrix} \neg\mathbf{X}m_\psi \wedge \\ \neg\mathbf{X}\left(\neg m_\psi\mathbf{U}\begin{pmatrix} m_\psi \wedge \\ \bigvee_{i=0}^{1} 0 < z^i_\psi < b-a \end{pmatrix}\right)\end{matrix}\right) \wedge \begin{pmatrix} x^j_\theta > a \ \wedge \\ \hat{x}^j_\theta \leqslant a \end{pmatrix}\end{matrix}\right)\right)\right)$$

[1] Note that, in the formulae of this section, we write $z_\delta \sim c$ as an abbreviation for $\bigvee_{i\in\{0,1\}} z^i_\delta \sim c$ (similarly for $z_\psi \sim c$).

The formula defines that there is a position $k' > k$ where the CLTLoc formula m_ψ holds, and there is no position $k'' > k'$ where m_ψ holds and such that $\tau(k'') - \tau(k') < b-a$ (i.e, such that $\tau(k'')$ belongs to interval $(\tau(k'), \tau(k') + (b-a))$. Similarly to the case of $\mathtt{up@b}^j_\theta$ this occurs in two cases: either

(i) there is no k'' such that $\tau(k'') \in (\tau(k'), \tau(k') + (b-a))$, i.e. $\tau(k'+1) - \tau(k') \geqslant b-a$, which in turn corresponds to one of z^0_δ, z^1_δ being $\geqslant b-a$ in $k'+1$ (i.e., $\mathbf{X}(z_\delta \geqslant b-a)$ holds in k');
or
(ii) for all k'' such that $\tau(k'') \in (\tau(k'), \tau(k') + (b-a))$, m_ψ does not hold there, which corresponds to saying that m_ψ does not hold in $k'+1$ (i.e., $\neg\mathbf{X}m_\psi$ holds in k'), and it is not true that the next time (if any) that m_ψ switches from false to true (when one of the clocks z^0_ψ, z^1_ψ is reset, by Formula (2)) is less than $b-a$ time units after $\tau(k')$ – that is, the clock between z^0_ψ and z^1_ψ that is not reset when m_ψ becomes true is $< b-a$.

As before, because of Corollary 1, x^j_θ is not reset again before k', hence the condition that there is a timestamp $t \in (\tau(k), \tau(k+1)]$ such that $\tau(k') - t = a$ corresponds to saying that in k' clock x^j_θ is $> a$ and $\hat{x}^j_\theta$ is $\leqslant a$.

It is easy to see that, in Figure 3, $\mathtt{down@a}^{\overline{(j+1)}_d}_\theta$ holds in $i+4$, when one considers $k' = i+6$, where both m_ψ and $\mathbf{X}(z_\delta \geqslant b-a)$ hold.

Formulae (9) and (10) define the sufficient conditions for one of the d auxiliary clocks x^j_θ, with $j \in [0, d-1]$ to be reset. They are, in a sense, the dual of abbreviations $\mathtt{up@b}^j_\theta$ and $\mathtt{down@a}^j_\theta$. More precisely, consider Formula (9); if m_ψ holds at a position k and there is no position $k' < k$ such that $\tau(k) - \tau(k') < b-a$ and m_ψ holds in k', then θ would have become true at time instant $\tau(k) - b$, so — unless $\tau(k) < b$ — there is a clock x^j_θ that was reset sometime in the past (recall that operator $\mathbf{P}$ is the past-time dual of $\mathbf{F}$) and that in k has value $\geqslant b$, with its associated clock $\hat{x}^j_\theta$ that is $< b$.

$$\mathbf{G}\left(m_\psi \wedge \big(z_\delta \geqslant b-a \quad \vee \quad \neg\mathbf{Y}(\psi) \wedge z_\psi \geqslant b-a\big) \Rightarrow \begin{matrix} Now < b \qquad \vee \\ \displaystyle\bigvee_{j=0}^{d-1} \left(\mathbf{P}\left(x^j_\theta = 0\right) \wedge x^j_\theta \geqslant b \wedge \hat{x}^j_\theta < b\right) \end{matrix} \right). \tag{9}$$

The condition captured by Formula (10) is similar to the one of abbreviation $\mathtt{down@a}^j_\theta$.

$$\mathbf{G}\left(m_\psi \wedge \left(\begin{matrix} \mathbf{X}(z_\delta \geqslant (b-a)) \qquad \vee \\ \neg\mathbf{X}m_\psi \wedge \neg\mathbf{X}\left(\neg m_\psi \mathbf{U} \left(\begin{matrix} m_\psi \wedge \\ \displaystyle\bigvee_{i=0}^{1} 0 < z^i_\psi < b-a \end{matrix} \right)\right) \end{matrix} \right) \Rightarrow \begin{matrix} Now \leqslant a \qquad \vee \\ \displaystyle\bigvee_{j=0}^{d-1} \left(\begin{matrix} \mathbf{P}\left(x^j_\theta = 0\right) \wedge \\ x^j_\theta > a \wedge \\ \hat{x}^j_\theta \leqslant a \end{matrix} \right) \end{matrix} \right). \tag{10}$$

For example, consider Figure 3. The antecedent of Formula (9) holds at position $i+3$. Hence, there must be a clock x^j_θ such that in $i+3$ both $x^j_\theta \geqslant b$ and $\hat{x}^j_\theta < b$

hold, which entails that x_θ^j is reset at i. Similarly, the antecedent of Formula (10) holds at position $i+6$, so there must be a clock $x_\theta^{j'}$ such that in $i+6$ both $x_\theta^{j'} > a$ and $\hat{x}_\theta^{j'} \leqslant a$ are true, so $x_\theta^{j'}$ is reset in $i+4$.

We abbreviate by $\mathtt{auxck}_\theta$ the conjunction of formulae (5)–(10).

We now introduce the CLTLoc formula $m_{\mathbf{F}_{(a,b)}}\psi$ which captures the semantics of the $\mathbf{F}_{(a,b)}$ operator.

$$m_{\mathbf{F}_{(a,b)}\psi} \triangleq \left(\neg \bigvee_{i=0}^{d-1} \hat{x}_\theta^i = 0\right) \mathbf{S} \bigvee_{j=0}^{d-1} \left(\hat{x}_\theta^j = 0 \wedge \mathbf{X}\left(\hat{x}_\theta^j > 0 \,\mathbf{U}\left(m_\psi \wedge a < \hat{x}_\theta^j < b\right)\right)\right)$$

Formula $m_{\mathbf{F}_{(a,b)}}\psi$ is true at position k if, and only if, in the greatest position, say k', in which one of the $\{\hat{x}_\theta^n\}_n$ clocks was reset (with possibly $k' = k$) the conditions for which $\mathbf{F}_{(a,b)}\psi$ is true held. These conditions are captured by the second argument of the $\mathbf{S}$ subformula, which state that, if $\hat{x}_\theta^j$ is the clock that is reset in k', then there is a position $k'' \geqslant k'$ in which m_ψ holds and such that $\hat{x}_\theta^j$ is in (a,b) there (i.e., $\tau(k') + a < \tau(k'') < \tau(k') + b$). Notice that, since by Formula (6) clock $\hat{x}_\theta^{d-1}$ is reset in the origin, for any position $k \in \mathbb{N}$ there is a $k' \leqslant k$ in which one of the $\{\hat{x}_\theta^n\}_n$ clocks is reset.

For example, let us consider Figure 3. In $i+1$ clock $\hat{x}_\theta^j$ is reset, there is a later position ($i+3$ in this case) where m_ψ holds, and the value of $\hat{x}_\theta^j$ there (3.8) is in (a,b); then, $m_{\mathbf{F}_{(a,b)}\psi}$ is true in $i+1$. In $i+2$, $i+3$ and $i+4$, instead, no clock $\{\hat{x}_\theta^n\}_n$ is reset, so the value of $m_{\mathbf{F}_{(a,b)}\psi}$ depends on the conditions at the greatest position before them in which one of those clocks was reset; this position happens to be $i+1$, so $m_{\mathbf{F}_{(a,b)}\psi}$ holds in all three positions.

Finally, given a pMITL formula $\exists q_1 \dots q_n \phi$, its equivalent CLTLoc formula $\mathtt{trans}_\phi$ is the following (where $sub(\phi)$ is the set of subformulae of ϕ which, for simplicity, we assume to include in operators $\mathbf{F}_{\langle a,b \rangle}$ only intervals of the form (a,b)):

$$\mathtt{trans}_\phi = m_\phi \wedge \mathtt{zddef} \wedge \bigwedge_{\theta \in sub(\phi)} \mathtt{ck}_\theta \wedge \bigwedge_{\theta \in sub(\phi), \theta = \mathbf{F}_{(a,b)}(\psi)} \mathtt{auxck}_\theta.$$

For example, consider pMITL formula $\phi = \exists q(\neg p \mathbf{U}_{(0,\infty)} q \wedge \mathbf{F}_{(2,4)} q)$. Formula m_ϕ is $\mathbf{X}(\neg p \mathbf{U}(c_q = 0)) \wedge m_{\mathbf{F}_{(2,4)}q}$, where $m_{\mathbf{F}_{(2,4)}q}$ must further be expanded according to the definition above — where m_ψ is replaced by $c_q = 0$ —, which we avoid for the sake of brevity. Similar expansions must be carried out for the subformulae $\mathtt{ck}_\theta$ and $\mathtt{auxck}_\theta$.

We have the following result.

Theorem 2 *Let $\exists q_1 \dots q_n \phi$ be a pMITL formula. CLTLoc formula $\mathtt{trans}_\phi$ is equivalent to $\exists q_1 \dots q_n \phi$.*

To prove Theorem 2 we rely on the following lemma, which shows equivalence between a non-quantified pMITL formula ϕ and the CLTLoc translation of ϕ where the atoms in pAP are not transformed into clock relations.

Lemma 7 *Consider a transformation m'_ϕ, which is the same as m_ϕ except that $m'_q \triangleq q$ even when $q \in pAP$, and define the following CLTLoc formula:*

$$\mathtt{trans}'_\phi = m'_\phi \wedge \mathtt{zddef} \wedge \bigwedge_{\theta \in sub(\phi)} \mathtt{ck}_\theta \wedge \bigwedge_{\theta \in sub(\phi), \theta = \mathbf{F}_{(a,b)}(\psi)} \mathtt{auxck}_\theta.$$

Let (π', τ) be a timed word such that $\pi' : \mathbb{N} \to \wp(AP \cup pAP)$ and $\exists q_1 \dots q_n \phi$ be a pMITL formula. Then $(\pi', \tau) \models \phi$ if, and only if, $(\pi', \tau) \models \mathtt{trans}'_\phi$. In addition, there is exactly one CLTLoc interpretation (π', σ) such that $\sigma : \mathbb{N} \times V_{C_\phi} \to \mathbb{R}_+$, $[(\pi', \sigma)] = (\pi', \tau)$ and $(\pi', \sigma) \models \mathtt{trans}'_\phi$.

Proof First of all, let be ξ be a CLTLoc formula. In the following we indicate by σ_ξ a clock valuation that only includes the clocks appearing in ξ; that is, such that $\sigma_\xi : \mathbb{N} \times V_\xi \to \mathbb{R}_+$, where V_ξ is the set of clocks referenced in ξ.

The proof is by induction on the structure of formula ϕ. For each step, we show that, for every subformula θ of ϕ:

P1. $(\pi', \tau) \models \mathtt{ck}_\theta$ and there is exactly one CLTLoc interpretation $(\pi', \sigma_{\mathtt{ck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{ck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{ck}_\theta}) \models \mathtt{ck}_\phi$;
P2. $(\pi', \tau) \models \mathtt{auxck}_\theta$ when θ is $\mathbf{F}_{(a,b)}\psi$; in addition, there is exactly one CLTLoc interpretation $(\pi', \sigma_{\mathtt{auxck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{auxck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{auxck}_\theta}) \models \mathtt{auxck}_\phi$;
P3. for every $i \in \mathbb{N}$, $(\pi', \tau), i \models \theta$ if, and only if, $(\pi', \tau), i \models m'_\theta$.

First, however, we need to show that every timed word (π', τ) is a model for $\mathtt{zddef}$. This is straightforward, as $(\pi', \tau) \models \mathtt{zddef}$ if, and only if, there is a CLTLoc interpretation $(\pi', \sigma_{\mathtt{zddef}})$ such that $[(\pi', \sigma_{\mathtt{zddef}})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{zddef}}) \models \mathtt{zddef}$; this is achieved simply by having, for all $k \in \mathbb{N}$, $\sigma_{\mathtt{zddef}}(2k, z^0_\delta) = 0$, $\sigma_{\mathtt{zddef}}(2k + 1, z^1_\delta) = 0$ and $\sigma_{\mathtt{zddef}}(k, \mathit{Now}) = 0$ if, and only if, $k = 0$. In addition, the CLTLoc interpretation $(\pi', \sigma_{\mathtt{zddef}})$ such that $[(\pi', \sigma_{\mathtt{zddef}})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{zddef}}) \models \mathtt{zddef}$ is unique, because the sequence of clock resets defined by $\mathtt{zddef}$ is uniquely determined (first z^0_δ and *Now*, then z^1_δ, then z^0_δ, and so on).

Let us tackle the base cases where $\theta = p \in AP$ and $\theta = q \in pAP$, which are in fact treated in the same manner, as $m'_p \triangleq p$ and $m'_q \triangleq q$. Concerning P1, we have that $(\pi', \tau) \models \mathtt{ck}_p$ if, and only if, there is a CLTLoc interpretation such that $[(\pi', \sigma_{\mathtt{ck}_p})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{ck}_p}) \models \mathtt{ck}_p$. Building the required $(\pi', \sigma_{\mathtt{ck}_p})$ from (π', τ) is straightforward. In fact, to satisfy Formulae (2) and (3), no matter when p holds in (π', τ), it is enough to define $\sigma_{\mathtt{ck}_p}$ so that: (i) $\sigma_{\mathtt{ck}_p}(0, z^0_\theta) = 0$; (ii) either $\sigma_{\mathtt{ck}_p}(i, z^0_\theta) = 0$ or $\sigma_{\mathtt{ck}_p}(i, z^1_\theta) = 0$ if p holds in i, but it does not $i - 1$ or in $i + 1$; and (iii) z^0_θ and z^1_θ are reset in an alternate manner. As before, interpretation $(\pi', \sigma_{\mathtt{ck}_p})$ such that $[(\pi', \sigma_{\mathtt{ck}_p})] = (\pi', \tau)$ is unique, since the sequence of clock resets is uniquely determined from π' (z^0_θ is reset in 0, then z^1_θ is reset the next time p is true, but it is false the position before or the position after it, and so on). P2 does not apply to the base case, whereas P3 is trivial.

Let us now consider the inductive case. Hence, let θ be a subformula of ϕ; we assume that properties P1, P2 and P3 hold for all subformulae of θ.

If $\theta = \neg\psi$ or $\theta = \gamma \wedge \psi$, P1 holds for the same reasons as the base case: no matter when m'_ψ and m'_γ hold, it is trivial to build a CLTLoc interpretation $(\pi', \sigma_{\mathtt{ck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{ck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{ck}_\theta}) \models \mathtt{ck}_\theta$. Showing that P3 holds is also straightforward.

If $\theta = \gamma \mathbf{U}_{(0,\infty)} \psi$, P1 holds for the same arguments as above. As far as $P3$ is concerned, by the semantics of Table 3, θ holds in i if, and only if, there is $i' > i$ such that $(\pi', \tau), i' \models \psi$, and for all $i < i'' < i'$ we have $(\pi', \tau), i'' \models \gamma$, which corresponds to the semantics of CLTLoc formula $\mathbf{X}\left(m'_\gamma \mathbf{U} m'_\psi\right)$ by inductive hypothesis.

Let us now tackle the case $\theta = \mathbf{F}_{(a,b)} \psi$.

P1 is similar to the other cases. By similar arguments as those used for showing that P1 holds one can prove P2. In fact, P2 only asserts that, given a timed word (π', τ), one can build a CLTLoc interpretation $(\pi', \sigma_{\mathtt{auxck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{auxck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{auxck}_\theta}) \models \mathtt{auxck}_\theta$. This is again straightforward, as Formulae (5), (6), and (7) are satisfied simply by having $\sigma_{\mathtt{auxck}_\theta}(0, x^0_\theta) = 0$, $\sigma_{\mathtt{auxck}_\theta}(0, \hat{x}^{d-1}_\theta) = 0$ (and the other $\hat{x}^j_\theta$ clocks greater than 0 in the origin), $\sigma_{\mathtt{auxck}_\theta}(i+1, \hat{x}^j_\theta) = 0$ whenever $\sigma_{\mathtt{auxck}_\theta}(i, x^j_\theta) = 0$, and the $\{x^n_\theta\}_n$ clocks are reset in a circular manner. In addition, it is always possible to satisfy Formulae (8), (9) and (10) given the truth value of m'_ψ. In fact, consider Formula (9). If its antecedent (which only depends on the subformulae of θ) is false no constraints are imposed on the $\{x^n_\theta, \hat{x}^j_\theta\}_n$ clocks. If the antecedent is true at position i, then either $\tau(i) < b$ (i.e., *Now* $< b$), or there is timestamp $\tau(i) - b \geqslant 0$. Necessarily there is a unique $i' < i$ such that $\tau(i') \leqslant \tau(i) - b < \tau(i'+1)$ (i.e., $\tau(i) - \tau(i') \geqslant b$ and $\tau(i) - \tau(i'+1) < b$), so it is enough to define that $\sigma_{\mathtt{auxck}_\theta}(i', x^j_\theta) = 0$ for the clock x^j_θ that respects the ordering defined by Formula (7). In addition, if the antecedent of Formula (9) is true in i, both the antecedent and the consequence of Formula (8) hold, since $\mathtt{up@b}^j_\theta$ holds at i'. Similarly for Formula (10). Notice that, as for formula $\mathtt{ck}_\theta$, given (π', τ) the interpretation $(\pi', \sigma_{\mathtt{auxck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{auxck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{auxck}_\theta}) \models \mathtt{auxck}_\theta$ is unique, because the sequence of resets of clocks $\{x^n_\theta\}_n$ is uniquely determined from π' and τ.

The key to prove P3 is to show that Formulae (8)–(10) define that, unless $i = 0$, no clock of set $\{\hat{x}^n_\theta\}_n$ is reset in i if in interval $[\tau(i-1), \tau(i))$ there is no timestamp t in which the conditions are met for θ to become true in t, and in $(\tau(i-1), \tau(i)]$ there is no timestamp t' in which the conditions are met for θ to become false at t'. More precisely, these conditions, which are exemplified by the rising and falling edges of the dotted line in Figure 3 are the following:

C1. For θ to become true at timestamp t there must be a position $i' > i$ such that ψ holds in i', $\tau(i') = t + b$, and there is no i'' such that $\tau(i'') \in (\tau(i') - (b-a), \tau(i'))$ and ψ holds in i''.

C2. For θ to become false at t' there must be a position $i' > i$ such that ψ holds in i', $\tau(i') = t + a$, and there is no i'' such that $\tau(i'') \in (\tau(i'), \tau(i') + (b-a))$ and ψ holds in i''.

As mentioned above, interpretation $(\pi', \sigma_{\mathtt{auxck}_\theta})$ such that $[(\pi', \sigma_{\mathtt{auxck}_\theta})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{auxck}_\theta}) \models \mathtt{auxck}_\theta$ is unique; then, with a slight abuse, we do not specify each time whether we are considering a position in (π', τ) or in $(\pi', \sigma_{\mathtt{auxck}_\theta})$.

Now, suppose $i > 0$ and no clock of set $\{\hat{x}^n_\theta\}_n$ is reset in i. Then, no clock of set $\{x^n_\theta\}_n$ is reset in $i-1$. Hence, there cannot be $i' > i-1$ such that (i) $\tau(i') - \tau(i-1) \geqslant b$, $\tau(i') - \tau(i) < b$ and the antecedent of Formula (9) holds, or (ii) $\tau(i') - \tau(i-1) > a$, $\tau(i') - \tau(i) \leqslant a$ and the antecedent of Formula (10) holds, otherwise one of the $\{x^n_\theta\}_n$ clocks should be reset in $i-1$, thus leading to a contradiction. It is easy to see that case (i) entails that there is no timestamp $t \in [\tau(i-1), \tau(i))$ where θ

can become true according to condition C1 above, and case (ii) implies that there is no $t' \in (\tau(i-1), \tau(i)]$ where θ can become false according to C2. Conversely, if a clock $\hat{x}^j_\theta$ is reset in $i > 0$, x^j_θ is reset in $i-1$, so the antecedent of Formula (8) holds in $i-1$; then, there is $i' > i-1$ where the second argument of the $\mathbf{U}$ operator in Formula $\mathtt{up@b}_\theta$ holds in i', or the second argument of $\mathbf{U}$ in Formula $\mathtt{down@a}_\theta$ does. In the first case i' is such that $\tau(i') - \tau(i-1) \geqslant b$, $\tau(i') - \tau(i) < b$ and the antecedent of Formula (9) holds, so again there is $t \in [\tau(i-1), \tau(i))$ where θ can become true according to condition C1 above; in the second case $\tau(i') - \tau(i-1) > a$, $\tau(i') - \tau(i) \leqslant a$ and the antecedent of Formula (10) holds, hence there is $t' \in (\tau(i-1), \tau(i)]$ where θ can become false according to C2.

Since we have established that no clock of set $\{\hat{x}^n_\theta\}_n$ is reset in $i > 0$ if the value of θ cannot change between i and $i-1$, it is easy to see that, by inductive hypothesis, m'_θ in i holds if, and only if, θ holds there. In fact, the value of m'_θ is that of the greatest $i' \leqslant i$ where one of the clocks of set $\{x^n_\theta\}_n$ is reset (i' is well-defined, because $\hat{x}^{d-1}_\theta$ is reset in the origin). Formula m'_θ captures i' through the $\mathbf{S}$ operator; more precisely, i' must be such that the second argument of $\mathbf{S}$ in m'_θ holds. Then, we need to show that, assuming $\hat{x}^j_\theta$ is reset in i', $\mathbf{X}\left(\hat{x}^j_\theta > 0 \,\mathbf{U}\left(m'_\psi \wedge a < \hat{x}^j_\theta < b\right)\right)$ correctly defines the value of θ there. This descends from the fact that, by Corollary 1, there cannot be more than $d-1$ resets of clocks of set $\{\hat{x}^n_\theta\}_n$ in an interval of length b, hence $\hat{x}^j_\theta$ is not reset again after i' before hitting value b. Since $\mathbf{F}_{(a,b)}\psi$ holds in i' if, and only if, there is i'' such that $\tau(i') + a < \tau(i'') < \tau(i') + b$ and ψ holds in i'', by inductive hypothesis this is equivalent to formula $\mathbf{X}\left(\hat{x}^j_\theta > 0 \,\mathbf{U}\left(m'_\psi \wedge a < \hat{x}^j_\theta < b\right)\right)$. Finally, since for $\theta = \phi$ we have that $V_{\mathtt{trans}'_\phi} = V_{\mathtt{zddef}} \cup V_{\mathtt{ck}_\phi} \cup V_{\mathtt{auxck}_\phi} = V_{C_\phi}$, the CLTLoc interpretation (π', σ) such that $\sigma : \mathbb{N} \times V_{C_\phi} \to \mathbb{R}_+$, $[(\pi', \sigma)] = (\pi', \tau)$ and $(\pi', \sigma) \models \mathtt{trans}'_\phi$ is unique. □

We can now prove Theorem 2 through a simple application of Lemma 7, by substituting atoms in pAP with clock relations.

Proof (of Theorem 2) Let (π, τ), with $\pi : \mathbb{N} \to \wp(AP)$, be a timed word. By definition of equivalence between pMITL and CLTLoc formulae we need to show that (π, τ) is a model for $\exists q_1 \dots q_n \phi$ if, and only if, it is also a model for $\mathtt{trans}_\phi$.

First, note that $(\pi, \tau) \models \exists q_1 \dots q_n \phi$ if, and only if, there is $\pi' : \mathbb{N} \to \wp(AP \cup pAP)$ such that $(\pi', \tau) \models \phi$ and $\pi'(i) - pAP = \pi(i)$ for all $i \in \mathbb{N}$. By Lemma 7, $(\pi', \tau) \models \phi$ if, and only if, $(\pi', \tau) \models \mathtt{trans}'_\phi$. In addition, $(\pi', \sigma_{\mathtt{trans}'_\phi})$ such that $[(\pi', \sigma_{\mathtt{trans}'_\phi})] = (\pi', \tau)$ and $(\pi', \sigma_{\mathtt{trans}'_\phi}) \models \mathtt{trans}'_\phi$ is unique, so $(\pi', \tau) \models \mathtt{trans}'_\phi$ if, and only if, $(\pi', \sigma_{\mathtt{trans}'_\phi}) \models \mathtt{trans}'_\phi$. If in m'_ϕ we substitute each occurrence of $q \in pAP$ with constraint $c_q = 0$, where $c_q \in V_{pAP}$ is a fresh clock, we obtain transformation m_ϕ. Consider interpretation $(\pi, \sigma_{\mathtt{trans}})$, where $\sigma_{\mathtt{trans}} : \mathbb{N} \times V_{C_\phi} \cup V_{pAP} \to \mathbb{R}_+$, such that for all $q \in pAP$ $\sigma_{\mathtt{trans}}(i, c_q) = 0$ if, and only if, $q \in \pi'(i)$ and for all $c \in V_{C_\phi}$ we have $\sigma_{\mathtt{trans}}(i, c) = \sigma_{\mathtt{trans}'}(i, c)$. We have that $(\pi, \sigma_{\mathtt{trans}}) \models \mathtt{trans}_\phi$ if, and only if, $(\pi', \sigma_{\mathtt{trans}'_\phi}) \models \mathtt{trans}'_\phi$ and $[(\pi, \sigma_{\mathtt{trans}})] = (\pi, \tau)$, which concludes the proof. □

Before concluding this section we remark that the encoding presented above can also be used to realize a decision procedure for the satisfiability of pMITL formulae. For this, it is enough, given a pMITL formula $\exists q_1 \dots q_n \phi$, to build the corresponding CLTLoc formula $\mathtt{trans}_\phi$, then solve it using the tool presented in Section 8. However, formula $\mathtt{trans}_\phi$ has been devised to show the equivalence of pMITL and CLTLoc formulae, and is not optimized for size. In fact, let us compute the size of $\mathtt{trans}_\phi$ with respect to the number of subformulae of ϕ, which we indicate by $|\phi|$, and to the value K of the biggest constant appearing in it. The size of formula $m_\theta \wedge \mathtt{ck}_\theta \wedge \mathtt{auxck}_\theta$ is biggest in the case $\theta = \mathbf{F}_{(a,b)}\psi$. In this case, the size of m_θ is d times the size of m_ψ; since in the worst case d is $2K$ (if (a,b) is $(K-1,K)$) and the number $|\psi|$ of subformulae in ψ is $|\theta|-1$, the size of m_θ is $O(K^{|\theta|})$. The biggest factor in the size of $\mathtt{auxck}_\theta$ is Formula (8), whose size is proportional to the size of m_θ times K. Hence, the size of $\mathtt{auxck}_\theta$ is $O(K^{|\theta|+1})$, which is also the size of $m_\theta \wedge \mathtt{ck}_\theta \wedge \mathtt{auxck}_\theta$. Finally, the size of formula $\mathtt{trans}_\phi$ is $O(K^{|\phi|} + K^{|\phi|+1}|\phi|)$ which is $O(K^{|\phi|+1}|\phi|)$. If, as customary, we consider a binary encoding for the constants appearing in ϕ, we indicate by s the number of bits for representing K, and we have $n = |\phi|$, the size of $\mathtt{trans}_\phi$ is $O(2^{s(n+1)}n)$.

However, if the goal is deciding the satisfiability of pMITL, although the exponential factor in s cannot be eliminated (recall that the satisfiability problem for MITL, which is a proper subset of pMITL, is EXPSPACE-complete, whereas it is PSPACE-complete for CLTLoc), the size of formula $\mathtt{trans}_\phi$ can be optimized. Without delving in many details if, for each subformula θ of ϕ, we introduce a fresh propositional letter $\boldsymbol{\theta}$, define m''_θ as m_θ where each occurrence of m_γ and m_ψ is replaced by propositions $\boldsymbol{\gamma}$ and $\boldsymbol{\psi}$, and introduce a constraint $\boldsymbol{\theta} \Leftrightarrow m''_\theta$, we obtain a satisfiability-preserving translation whose size is $O(K|\phi|)$, that is, $O(2^s n)$.

6.2 From CLTLoc to pMITL

To show the equivalence between pMITL and CLTLoc over timed words, in this section we build a model-preserving transformation from CLTLoc to pMITL formulae. We consider the future-only fragment of CLTLoc since, as shown in [14], over timed words it has the same expressiveness of the full language including past operators.

Let ϕ be a future-only CLTLoc formula, over a set V_ϕ of clocks. Without loss of generality, we assume that all clocks of V_ϕ are reset in the origin. In fact, suppose we need to build a formula $\bar{\phi}$ which states that, in the origin (i.e., the first meaningful position in the interpretation), the clocks of $V_{\bar{\phi}}$ have a certain relationship among them captured by formula ξ. For example, we might have $\xi = x_1 > x_2 > x_3$. We can build a CLTLoc formula ϕ whose models are isomorphic to those of $\bar{\phi}$, and in which all clocks are reset in the origin. The idea is to use the first $|V_{\bar{\phi}}|$ positions of the model of ϕ to create the required ordering among clocks, and then the rest of the positions to capture the semantics of $\bar{\phi}$. More precisely, we introduce a fresh clock x_{act} which is reset in each position $0 \leqslant i < |V_{\bar{\phi}}|$, and then never again (we do not show the corresponding CLTLoc formula, which is straightforward). Then, the first position $|V_{\bar{\phi}}|$ in the model of ϕ in which $x_{act} > 0$ corresponds to the origin of the model of $\bar{\phi}$. ϕ is a transformation of $\bar{\phi}$ obtained in the following way: $\phi = \mathbf{X}^{|V_{\bar{\phi}}|}\bar{\phi}'$ (where $\mathbf{X}^n$ stands for "$\mathbf{X}$ nested n times"), and $\bar{\phi}'$ is $\bar{\phi}$ where each formula of the form $\mathbf{Y}\psi$ is replaced by $\mathbf{Y}(\psi \wedge x_{act} > 0)$ and each formula of the form $\gamma\mathbf{S}\psi$ is replaced by $\gamma\mathbf{S}(\psi \wedge x_{act} > 0)$.

When all clocks are reset in the origin, it is easy to eliminate from a CLTLoc formula ϕ constraints of the form $x_i < x_j$ so that only constraints $x_h \sim c$, with c a constant, appear in ϕ. In fact, $x_i < x_j$ is equivalent to $(x_j > 0)\mathbf{S}(x_i = 0 \wedge x_j > 0)$ since for all $x_h \in V_\phi$ and $k \in \mathbb{N}$ there is $k' \leqslant k$ where $x_h = 0$.

Consider now a CLTLoc formula ϕ with clocks V_ϕ that are all reset in the origin. To capture its behavior in pMITL, we define the following quantified propositions in set pAP:

- For each $x_i \in V_\phi$ we introduce a propositional letter r_{x_i}, which holds when clock x_i is reset (i.e., if condition $x_i = 0$ holds).
- For each constraint $x_i \sim c$ (with $c > 0$), independent of the nature of relation $\sim$, we introduce two propositional letters $p_{x_i \leqslant c}$ and $p_{x_i < c}$, which capture, respectively, the conditions $x_i \leqslant c$ and $x_i < c$.

The behavior of the new propositional letters is defined by a formula $\mu_{x_i \sim c}$, which captures when $p_{x_i \leqslant c}$ and $p_{x_i < c}$ hold with respect to the truth of r_{x_i}. In the formula we introduce abbreviation $\mathbf{G}^i_I(\xi) = \xi \wedge \mathbf{G}_I(\xi)$ (resp. $\mathbf{F}^i_I(\xi) = \xi \vee \mathbf{F}_I(\xi)$), where i stands for "included", which requires (resp. allows) ξ to hold in the current position (hence $\mathbf{G}^i(\xi) = \mathbf{G}^i_{(0,\infty)}(\xi)$).

If $c > 0$ we have the following:

$$\mu_{x_i \sim c} := r_{x_i} \wedge$$
$$\mathbf{G}^i \left(r_{x_i} \Rightarrow \begin{pmatrix} \mathbf{G}^i_{(0,c]}(p_{x_i \leqslant c}) \wedge \mathbf{F}^i_{(0,c]} \left(\mathbf{G}_{(0,\infty)}(\neg p_{x_i \leqslant c}) \vee (\neg p_{x_i \leqslant c}) \mathbf{U}_{(0,\infty)} r_{x_i} \right) \wedge \\ \mathbf{G}^i_{(0,c)}(p_{x_i < c}) \wedge \mathbf{F}^i_{(0,c)} \left(\mathbf{G}_{(0,\infty)}(\neg p_{x_i < c}) \vee (\neg p_{x_i < c}) \mathbf{U}_{(0,\infty)} r_{x_i} \right) \end{pmatrix} \right).$$

For $c = 0$ we have simply $\mu_{x_i \sim 0} := r_{x_i}$ (i.e., we only require the clock to be reset in the origin), since $x_i \leqslant 0$ is true if, and only if x_i is reset (i.e., r_{x_i} holds) and $x_i < 0$ is trivially always false.

Theorem 3 *For any CLTLoc formula there is an equivalent pMITL formula.*

Proof For every CLTLoc formula ϕ we inductively define a transformed pMITL formula ϕ' as follows (recall that we can eliminate constraints of the form $x < y$ when x and y are clocks):

$$\begin{aligned} p &\longmapsto p \quad \text{ for } p \in AP \\ x = 0 &\longmapsto r_x \\ x < 0 &\longmapsto \bot \\ x = c &\longmapsto p_{x \leqslant c} \wedge \neg p_{x < c} \\ x < c &\longmapsto p_{x < c} \\ \neg\psi &\longmapsto \neg\psi' \\ \gamma \wedge \psi &\longmapsto \gamma' \wedge \psi' \\ \mathbf{X}\psi &\longmapsto \bot \mathbf{U}_{(0,\infty)} \psi' \\ \gamma \mathbf{U} \psi &\longmapsto \psi' \vee (\gamma' \wedge \gamma' \mathbf{U}_{(0,\infty)} \psi'). \end{aligned}$$

The final pMITL formula is

$$\exists r_{x_1}, p_{x_1 \leqslant c}, p_{x_1 < c}, \dots r_{x_n}, p_{x_n \leqslant c}, p_{x_n < c} (\phi' \wedge \bigwedge_{x_i \sim c \,\in\, sub(\phi)} \mu_{x_i \sim c}) \qquad (11)$$

where $n = |V_\phi|$.

To show that Formula (11) has the same models as ϕ we prove that for each timed word (π, τ) there is (π, σ_ϕ) such that $[(\pi, \sigma_\phi)] = (\pi, \tau)$ and $(\pi, \sigma_\phi) \models \phi$ if, and only if, there is (π', τ), with $\pi' : \mathbb{N} \to \wp(AP \cup pAP)$, such that for all $k \in \mathbb{N}$ $\pi'(k) - pAP = \pi(k)$ and $(\pi', \tau) \models \phi'$.

As usual, the proof is by induction on the structure of ϕ. The base case for $p \in AP$ is trivial, as it is simply $(\pi, \tau), k \models p$ if, and only if, $p \in \pi(k)$ in both CLTLoc and pMITL.

The inductive cases for $\neg\psi$, $\gamma \wedge \psi$, $\mathbf{X}\psi$ and $\gamma\mathbf{U}\psi$ are also straightforward, as they do not rely on quantified propositions except possibly those necessary for subformulae ψ and γ, which are covered by the inductive hypothesis.

The last case is the one of clock constraints $x_i = 0$ and $x_i \sim c$. First, consider the following relation $\mathcal{T}_\phi$ between CLTLoc interpretations of ϕ and pMITL interpretations of ϕ': $\mathcal{T}_\phi((\pi, \sigma_\phi), (\pi', \tau))$ if, and only if, $[(\pi, \sigma_\phi)] = (\pi, \tau)$, and for each $k \in \mathbb{N}$

- $\pi'(k) - pAP = \pi(k)$
- for each clock $x_i \in V_\phi$ we have $\sigma_\phi(k, x_i) = 0$ if, and only if, $r_{x_i} \in \pi'(k)$,
- for each constraint $x_i \sim c$ in ϕ we have $\sigma_\phi(k, x_i) \leqslant c$ if, and only if, $p_{x_i \leqslant c} \in \pi'(k)$ and $\sigma_\phi(k, x_i) < c$ if, and only if, $p_{x_i < c} \in \pi'(k)$.

Then, to prove the goal in this case we split it in two cases. We first show that, for each CLTLoc interpretation (π, σ_ϕ), if $\mathcal{T}_\phi((\pi, \sigma_\phi), (\pi', \tau))$, then $(\pi', \tau) \models \mu_{x_i \sim c}$. Second, we prove that, for each pMITL interpretation (π', τ), if $(\pi', \tau) \models \mu_{x_i \sim c}$, then there exists CLTLoc interpretation (π, σ_ϕ) such that $\mathcal{T}_\phi((\pi, \sigma_\phi), (\pi', \tau))$.

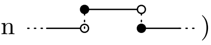
If (π, σ_ϕ) is a CLTLoc interpretation for ϕ, it is easy to see that there is a pMITL interpretation such that $\mathcal{T}_\phi((\pi, \sigma_\phi), (\pi', \tau))$. We show that $(\pi', \tau) \models \mu_{x_i \sim c}$. Since by hypothesis all clocks are reset in the origin, $r_{x_i} \in \pi'(0)$. If $\sigma_\phi(k, x_i) > 0$, then $r_{x_i} \notin \pi'(k)$, so the implication in $\mu_{x_i \sim c}$ holds. If $\sigma_\phi(k, x_i) = 0$, instead, then $r_{x_i} \in \pi'(k)$. For all k' such that $\tau(k') - \tau(k) \leqslant c$ we have that $x_i \leqslant c$ (even if x_i is reset again), hence $p_{x_i \leqslant c} \in \pi'(k)$, so $(\pi', \tau), k \models \mathbf{G}^i_{(0,c]}(p_{x_i \leqslant c})$. After k, either x_i is not reset anymore, or there is $k'' > k$ where it is reset, and it is not reset in between. In the first case after the last k' such that $\tau(k') - \tau(k) \leqslant c$ it always holds that $x_i > c$, i.e., $(\pi', \tau), k \models \mathbf{F}^i_{(0,c]}\mathbf{G}_{(0,\infty)}(\neg p_{x_i \leqslant c})$. In the second case, after the last k' such that $\tau(k') - \tau(k) \leqslant c$, until x_i is reset again it holds that $x_i > c$, i.e., $(\pi', \tau), k \models \mathbf{F}^i_{(0,c]}((\neg p_{x_i \leqslant c})\mathbf{U}_{(0,\infty)} r_{x_i})$. Similarly for the other conjunct in the consequence of the implication.

Let us now consider a pMITL interpretation for ϕ' such that $(\pi', \tau) \models \mu_{x_i \sim c}$. Then, $r_{x_i} \in \pi'(0)$, so x_i is reset in the origin, and the consequence of the implication appearing in $\mu_{x_i \sim c}$ also holds there. If r_{x_i} does not hold any more after 0, then it is enough, for all $k \geqslant 0$ to define $\sigma_\phi(k, x_i) = \tau(k) - \tau(0) = \tau(k)$; it is easy to see that if by $\mu_{x_i \sim c}$ it holds that $p_{x_i \leqslant c} \in \pi'(k)$, then $\sigma_\phi(k, x_i) \leqslant c$, and similarly for $p_{x_i < c}$. If, instead, $k > 0$ is the next position after 0 in which $r_{x_i} \in \pi'(k)$, then for all $0 \leqslant k' < k$ we define $\sigma_\phi(k', x_i) = \tau(k')$ and, as before, if by $\mu_{x_i \sim c}$ it holds that $p_{x_i \leqslant c} \in \pi'(k')$, then $\sigma_\phi(k', x_i) \leqslant c$ (similarly for $p_{x_i < c}$). Then, since $r_{x_i} \in \pi'(k)$, the antecedent of formula $\mu_{x_i \sim c}$ holds there, and the arguments above are applied again to build σ_ϕ. □

7 Encoding Metric Temporal Logics over the continuous semantics

We exploit the decision procedure for CLTLoc outlined in Sect. 4 to define mechanisms for deciding various metric temporal logics over continuous time. In [12], [13] and [11], we have defined several satisfiability-preserving reductions from metric temporal logics to CLTLoc; hence, satisfiability of formulae of these former logics can be determined by solving the corresponding problem for CLTLoc. In particular, MITL, $\mathrm{MITL}_{(0,\infty)}$, $\mathrm{MITL}_{(0,\infty)}$ with counting modalities [34], and their extensions with past operators are the logics we have targeted so far.

We now briefly show how to encode MITL and $\mathrm{MITL}_{(0,\infty)}$ (hence, QTL) formulae into CLTLoc ones, by providing some highlights of the reduction in a special case.

In general, in [4] it is shown that a signal can be seen as an infinite sequence of adjacent non-empty intervals starting from the origin. Each interval is a convex set of points over $\mathbb{R}$ that defines exactly the set of atomic propositions that are true in all the time instants in it. In our translation, we represent the truth of a MITL (or $\mathrm{MITL}_{(0,\infty)}$) formula ϕ, over the sequence of time intervals, by a CLTLoc formula that captures its semantics. We assume that signals are finitely variable. For these signals, time can be partitioned in a countable set of adjacent intervals such that the value of every subformula of ϕ is constant in each interval. In the case of MITL, we also restrict signals to intervals that are *left-closed and right-open* (l.c.r.o. for short), as in ). As in Section 5, let $\langle a, b\rangle$ be a bounded interval, where $0 \leqslant a < b$ are integer constants, $\langle$ is either (or [, and $\rangle$ is) or]. Given a signal $M : \mathbb{R}_+ \to \wp(AP)$ over a finite alphabet AP and a bounded interval I, denote with $p \in M_I$, the case where $p \in M(x)$ for all $x \in I$. A signal M is l.c.r.o if for all $p \in AP$, for all $a < b \in \mathbb{R}_+$ if $p \in M_{(a,b)}$ then $p \in M_{[a,b)}$ and for all $a \leqslant b \in \mathbb{R}_+$ if $p \in \; M_{[a,b]}$ then there exists $c > b$ such that $p \in M_{[a,c)}$. The l.c.r.o. assumption allowed us to devise a simpler translation, but it is not strictly necessary. For instance, under the l.c.r.o assumption, a formula cannot hold in isolated points, but if it holds at time instant t then it holds over a non-empty interval $[t, t+\varepsilon)$, for some $\varepsilon > 0$. Then, the case of isolated points can be disregarded for l.c.r.o signals.

Let ϕ be a MITL formula. For each subformula θ of ϕ, we introduce a CLTLoc predicate $\overleftarrow{\theta}$ that represents the value of θ in the intervals and the following abbreviations:

$$\underleftarrow{\xi} = \neg\, \overleftarrow{\xi} \qquad\qquad \lrcorner\!\ulcorner_{\xi} = \neg\mathbf{Y}(\overleftarrow{\xi}) \wedge \overleftarrow{\xi} \qquad\qquad \urcorner\!\llcorner_{\xi} = \neg\mathbf{Y}(\underleftarrow{\xi}) \wedge \underleftarrow{\xi}$$

where $\lrcorner\!\ulcorner_{\xi}$, for example, captures the situation in which ξ changes its value from false to true, with the formula being true in the current interval.

For simplicity, we focus our attention on temporal operators $\mathbf{F}_{(0,b]}\psi$ and $\mathbf{P}_{[0,b)}\psi$. We remark that it can be shown that, if ψ holds only in l.c.r.o. intervals, so do $\mathbf{F}_{(0,b]}\psi$ and $\mathbf{P}_{[0,b)}\psi$ (the same does not hold, for example, for $\mathbf{F}_{(0,b)}\psi$). For each subformula θ of ϕ, we introduce two clocks, z_θ^0 and z_θ^1, which measure the time from the last change point (either $\lrcorner\!\ulcorner_\theta$ or $\urcorner\!\llcorner_\theta$, so we have $\lrcorner\!\ulcorner_\theta \vee \urcorner\!\llcorner_\theta \Leftrightarrow z_\theta^0 = 0 \vee z_\theta^1 = 0$), and whose resets alternate. Our translation defines the (sufficient and necessary) conditions causing events $\lrcorner\!\ulcorner_\theta$ and $\urcorner\!\llcorner_\theta$ to occur, for all θ.

Formula (12), then, captures the condition in which formula $\theta = \mathbf{F}_{(0,b]}\psi$ becomes false: in this case, ψ must become false, and it cannot become true again for b instants (i.e., ψ cannot become true again before its associated clock that is reset when ψ becomes false hits b).

$$\urcorner\!\llcorner_\theta \Leftrightarrow \urcorner\!\llcorner_\psi \wedge \lrcorner\!\ulcorner_\psi \mathbf{R} \neg \left(\lrcorner\!\ulcorner_\psi \wedge \bigwedge_{i \in \{0,1\}} z_\psi^i \leqslant b \right). \tag{12}$$

The case for θ becoming true is not shown for brevity.

Consider the case $\theta = \mathbf{P}_{[0,b)}(\psi)$. Formula (13) captures the condition in which θ becomes true. This occurs when ψ becomes true and either the current instant is the origin (O is an abbreviation for $\neg\mathbf{Y}(\top)$), or ψ has never become true since the origin, or the last time ψ changed value (necessarily from false to true), this occurred more than b instants ago (i.e., the clock associated with ψ that is not reset now is $\geqslant b$).

$$\lrcorner\!\ulcorner_{\theta} \Leftrightarrow \lrcorner\!\ulcorner_{\psi} \wedge \left(O \vee \mathbf{Y}\left(\neg \lrcorner\!\ulcorner_{\psi} \mathbf{S}\ (O \wedge \underset{\bullet\!-}{\psi}) \right) \vee \bigvee_{i \in \{0,1\}} z_{\psi}^{i} \geqslant b \right) \tag{13}$$

To conclude this section, we provide an example of an MITL formula over two l.c.r.o. signals, whose model is intrinsically aperiodic in the values of the delays between change points. We say that a signal $M : \mathbb{R}_+ \to \wp(AP)$ is (ultimately) periodic (with respect to the time) if there exist two instants t_1 and t_2 such that $t_2 - t_1 = d > 0$ and $M(t + kd) = M(t)$ for all $t_1 \leqslant t \leqslant t_2$ and for all $k \in \mathbb{N}$. For example, let AP be set $\{a\}$. A signal M such that $M(t) = \{a\}$ for all $2k \leqslant t \leqslant 2k+1$ and $k \in \mathbb{N}$ and $M(t) = \varnothing$ elsewhere is periodic. A signal M' such that $M'(t) = \{a\}$ for all $2k \leqslant t \leqslant 2k + 1 + \frac{1}{k+1}$ and $k \in \mathbb{N}$ and $M'(t) = \varnothing$ elsewhere is not periodic. The existence of formulae admitting only aperiodic models shows that, in the decision procedure of Section 4, the periodicity must be enforced on the set of constraints defining regions, but not on the actual values of the clocks, nor on the time differences δ. Therefore, the encoding of a CLTLoc formula presented in Section 4 cannot include constraints of the form $\boldsymbol{\delta}(k+1) = \boldsymbol{\delta}(\boldsymbol{loop})$ and $\boldsymbol{x}(k+1) = \boldsymbol{x}(\boldsymbol{loop})$ for some $k \geqslant \boldsymbol{loop}$, where $\boldsymbol{loop}$ is the variable defining the position of the periodic part of the model. In other words, there are aperiodic models that do not admit a periodic sequence of time increments of the form $\delta_0\delta_1 \ldots (\delta_l \ldots \delta_k)^{\omega}$, even if the sequence of clock regions is periodic.

Example 2

Consider the behavior of two Boolean signals p and q in Figure 4. Signal p holds in $[2k, 2k + 1 + \varepsilon)$, for all $k \geqslant 0$, and it is false elsewhere, as formalized by the conjunction of the following MITL formulae (recall that $\mathbf{G}_I^i(\phi) = \phi \wedge \mathbf{G}_I(\phi)$):

$$\mathbf{G}_{\langle 0,1]} p \wedge \mathbf{G}^i(\mathbf{G}_{\langle 0,1]} p \Rightarrow \mathbf{G}_{\langle 2,3]} p) \tag{14}$$

$$\mathbf{G}_{\langle 0,1]} q \wedge \mathbf{G}^i(\mathbf{G}_{\langle 0,1]} q \Rightarrow \mathbf{G}_{\langle 2,3]} q) \tag{15}$$

$$\mathbf{G}^i(p \Rightarrow q). \tag{16}$$

Both signals p and q hold over intervals longer than one time unit, because of the l.c.r.o. assumption which says that if p (or q) holds at t then p holds in $[t, t + \varepsilon)$ for any $t \geqslant 0$ and any $\varepsilon > 0$. In fact, if formula $\mathbf{G}_{\langle 0,1]} p$ holds at t it imposes that p holds in $(t, t + 1]$. Assume that $\neg p$ holds in $(t + 1, t + 1 + \mu)$, with $\mu > 0$. Because of l.c.r.o. assumption, p must hold in $[t + 1, t + 1 + \varepsilon)$ which yields a contradiction. In addition, we require that q is at least as long as p by Formula (16). Formulae (14)-(15) above may, in general, admit periodic models with respect to the values; therefore, we have to add other formulae to rule out these models. This may be achieved by the following formulae (where $\gamma\mathbf{U}^i\phi = \gamma \wedge (\gamma\mathbf{U}\phi)$) enforcing that,

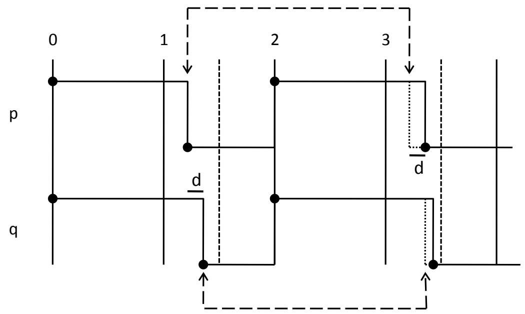


Fig. 4 Aperiodic model for the MITL formula of Example 2.

over interval $[2k, 2k+1]$, with $k \geqslant 0$, signal q is strictly longer than p, while over $[2k+1.5, 2k+2)$ both p and q are false and that each new occurrence of an interval where p holds, involving instant $2(k+1)$, is always longer than the previous one involving instant $2k$.[2]: Signals p and q become false before each time instant $2k+1.5$ because Formula (17) imposes that $\neg p \wedge q$ occurs until both p and q are false over an interval of length 0.5.

$$\mathbf{G}^i(\mathbf{G}_{\langle 0,1]}(p \wedge q) \Rightarrow (p \wedge q)\mathbf{U}((\neg p \wedge q)\mathbf{U}^i(\mathbf{G}_{\langle 0,0.5]}\neg q)) \tag{17}$$

$$\mathbf{G}^i(\neg p \wedge q \Rightarrow \mathbf{G}_{\langle 1,2]} p). \tag{18}$$

Now, we show that a signal that is model for the formula is not periodic. Let $t^p_k \in (2k+1, 2k+1.5)$ be the instant where p becomes false, and $d_k = t^p_k - 2k$ be the length of the k-th interval $[2k, t^p_k)$ in which p holds. Formula (18) lengthens of $\delta_k > 0$ time units the duration of p over the next interval starting at $2(k+1)$; that is, p holds in interval $[2(k+1), t^p_k + 2 + \delta_k)$, whose length d_{k+1} is $d_k + \delta_k$. Since δ_k is strictly positive, the sequence of values d_k is strictly monotonic increasing, i.e., $d_k < d_{k+1}$, for all $k \geqslant 0$. Therefore, the signal modeling the conjunction of MITL formulae (14)–(18) is not periodic with respect to the time, and the sequence of durations d_k is also not periodic in its CLTLoc counterpart. However, the sequence of clock regions induced by the clocks in the CLTLoc formula obtained through the translation shown above applied to the conjunction of MITL formulae (14)–(18) is periodic. In fact, it is easy to see that the length of intervals where p and q hold is always in $(1, 1.5)$. Then, the clocks measuring the time that elapses from the instant in which p or q start to hold are either reset to 0 – when the associated signal becomes true–, or first reach region $(1, 1.5)$ – when the signal changes truth value to false – and then region $[2, 2]$ when the signal becomes true again. The same arguments hold for clocks measuring the time that elapses since p or q become false, as the length of the intervals in which the propositions do not hold is always in $(0.5, 1)$.

[2] With slight abuse, we use rational bound 0.5; as customary, a formula with only integer bounds can be obtained by doubling all constants appearing in the formula.

8 Implementation and Experimental Results

The reductions from MITL to CLTLoc, outlined in Sect. 7, are implemented in the `qtlsolver` tool, available from [1]. The tool translates MITL and $\mathrm{MTL}_{(0,\infty)}$ into CLTLoc, which can be checked for satisfiability by `ae`^{2}`zot`.

The resulting toolkit has a 3-layered structure, where CLTLoc is the intermediate layer between SMT-solvers and various temporal formalisms that can be reduced to CLTLoc. This not only supports (bounded) satisfiability verification of different languages, but it also allows the expression of different degrees of abstraction. For instance, MITL abstracts away the notion of clocks, inherently encompassed within temporal modalities, which are instead explicit in CLTLoc (as witnessed by the example of the timed lamp in Sect. 2) and available to a user, e.g., to express or verify properties where clocks are very convenient. In fact, preliminary experimental results point out that the time required to solve CLTLoc may be significantly smaller than the one needed for more abstract classes of languages, such as MITL. This gap is caused by the "effort" required to capture the semantics of temporal modalities, which, on the other hand, allow for more concise and manageable high-level specifications. One can then take advantage of the layered structure, which allows the resolution of a formula to be compliant also with constraints imposed at lower layers, for instance by adding at the CLTLoc layer some extra formula limiting the set of valid models (e.g., by discarding certain edges of some events or by adding particular timing requirements). Also the third layer (the SMT solver) may be used to add further constraints, e.g., to force the occurrence of a proposition or of a certain clock value at a specific discrete position of the finite model.

The current implementation of `qtlsolver` supports various reductions. More precisely, it realizes the MITL-to-CLTLoc translation tailored to l.c.r.o. signals, as highlighted in Sect. 7. It also implements a translation from $\mathrm{MTL}_{(0,\infty)}$ to CLTLoc. This translation does not assume any special shape for signals, except that they be finitely variable; it natively supports operators $\mathbf{F}_{\langle 0,b\rangle}$ and $\mathbf{G}_{\langle 0,b\rangle}$ (and their past counterparts), where the bounds can be either included or excluded. These operators allow us to define concisely $\mathbf{F}_{\langle a,b\rangle}$ and $\mathbf{G}_{\langle a,b\rangle}$ as abbreviations. For instance, $\mathbf{G}_{(3,6)}(\phi)$ is equivalent to $\mathbf{G}_{(0,3)}\Big(\mathbf{F}_{(0,3)}\Big(\mathbf{G}_{(0,3)}(\phi)\Big)\Big)$; defining a similar equivalence using only the $\mathbf{F}_{(0,1)}$ and $\mathbf{G}_{(0,1)}$ modalities (see, e.g., [24]) involves the recursive expansions of each conjunct of $\mathbf{G}_{(3,4)}(\phi) \wedge \mathbf{G}_{[4,5)}(\phi) \wedge \mathbf{G}_{[5,6)}(\phi)$, where $\mathbf{G}_{(n,n+1)}(\phi)$ is equivalent to $\mathbf{G}_{(n-1,n)}\Big(\mathbf{F}_{(0,1)}\Big(\mathbf{G}_{(0,1)}(\phi)\Big)\Big)$.

The following two encodings are currently available (they both include past operators):

MITL: providing a direct definition of MITL operators, assuming l.c.r.o. intervals;
QTL: providing the definition of $\mathrm{MTL}_{(0,\infty)}$ operators with unrestricted signals (other than they be finitely variable), and MITL operators through abbreviations.

We used the above two encodings and the CLTLoc decision procedure to carry out some verification experiments on the example of the Timed Lamp described in Sect. 2. More precisely, we have built several descriptions of the behavior of the lamp: (i) the CLTLoc model presented in Sect. 2; (ii) an MITL specification assuming l.c.r.o. signals; (iii) an $\mathrm{MTL}_{(0,\infty)}$ specification in which predicates *on* and *off* are constrained to be true only in isolated instants. On each of these specifications we have carried out three experiments, assuming $\varDelta = 5$: a check of the satisfiability of the specification, to show that it is consistent (*sat*); the (dis)proof of property "the light never stays on for more than $\varDelta$ time units" (p_1); the proof of property "if at some point the light stays on for more than $\varDelta$ time units, then there is an instant when *on* is pressed, and then it is pressed again before $\varDelta$ time units" (p_2). Depending on the temporal logic and of the restrictions on the signals (l.c.r.o. or not) the formalization of the timed lamp and of the properties can change.

In the case of the CLTLoc specification of the timed lamp, in order to formalize properties p_1 and p_2 we introduce an auxiliary clock c_{aux}, which is reset every time the light is turned on, i.e., $c_{\text{aux}} \Leftrightarrow l \wedge \mathbf{Y}(\neg l)$. Then, in CLTLoc property p_1 is captured by formula $\mathbf{G}(\mathbf{Y}(l) \Rightarrow c_{\text{aux}} \leqslant \varDelta)$. In addition, property p_2 is formalized by the following formula:

$$\mathbf{F}(l \wedge c_{\text{aux}} \geqslant \varDelta) \Rightarrow \mathbf{F}(on \wedge \mathbf{X}(\neg rst\text{-}c\mathbf{U}(on \wedge test_{0<c\leqslant\varDelta}))) . \tag{19}$$

The behavior of the timed lamp can be captured by the following MITL formula over l.c.r.o. signals (we write $\gamma\mathbf{S}^i\psi$ for $\psi \vee (\gamma \wedge \gamma\mathbf{S}_{(0,\infty)}\psi)$ and $\mathbf{P}^i_I\psi$ for $\psi \vee \mathbf{P}_I\psi$):

$$\mathbf{G}^i\left(\left(l \Leftrightarrow (\ on\mathbf{S}^i\neg off\ ) \wedge \mathbf{P}^i_{(0,\varDelta)}(on)\right) \wedge (on \Rightarrow \neg off)\right) . \tag{20}$$

In MITL over l.c.r.o. signals, where predicates hold over non-null intervals, we limit the length of intervals in which *on* (and *off*) holds to be at most 1 by adding the following constraint:

$$\mathbf{G}^i\left(\neg\mathbf{G}_{(0,1]}(on) \wedge \neg\mathbf{G}_{(0,1]}(off)\right) . \tag{21}$$

Over unrestricted signals, instead, we force *on* to hold only in isolated instants by adding the following $\mathrm{MITL}_{(0,\infty)}$ constraint (and similarly for *off*)

$$\mathbf{G}^i\left(\neg(on\ \mathbf{U}_{(0,+\infty)}\top) \wedge \neg(on\ \mathbf{S}_{(0,\infty)}\top)\right) . \tag{22}$$

Properties p_1 and p_2 over unrestricted signals are captured by the following $\mathrm{MTL}_{(0,\infty)}$ formulae (where $\mathbf{F}^i$ stands for $\mathbf{F}^i_{(0,+\infty)}$):

$$\mathbf{G}^i\left(\mathbf{F}^i_{(0,\varDelta]}(\neg l)\right) \tag{23}$$

$$\mathbf{F}^i\left(\mathbf{G}^i_{(0,\varDelta]}(l)\right) \Rightarrow \mathbf{F}^i\left(on \wedge \mathbf{F}_{(0,\varDelta]}(on)\right) . \tag{24}$$

Over l.c.r.o. signals property p_1 is still captured by Formula (23); property p_2, instead, is more involved, and corresponds to the following formula:

$$\mathbf{F}^i\left(\mathbf{G}^i_{(0,\varDelta]}(l)\right) \Rightarrow \mathbf{F}^i\left((\neg on \wedge \mathbf{P}^i_{(0,\varDelta)}(on))\mathbf{U}^i on\right) . \tag{25}$$

Table 8 reports the time and space required for the checks outlined above (all tests have been done using the Common Lisp compiler SBCL 1.1.2 on a 2.13GHz Core2 Duo MacBook Air with MacOS X 10.7 and 4GB of RAM; the solver was z3 4.0). All bounded satisfiability checks have been performed using a bound $k = 20$. The first line of each row shows the total processing time (i.e., parsing and solving)

Table 4 Experimental results with the timed lamp, reporting Time (sec) and heap size (MB).

Problem	Satisfiable?	CLTL-o-c	MITL (l.c.r.o)	$\mathrm{MTL}_{(0,\infty)}$ (unrest.)
sat	Yes	0.48/0.33 5.63	15.5/13.84 66.45	4.24/3.04 27.12
$\mathbf{p_1}$	Yes	0.52/0.35 6.22	36.74/33.16 102.47	17.2/14.86 63.5
$\mathbf{p_2}$	No	0.67/0.49 6.55	6.61/5.09 110.27	257.1/240.88 58.66

and the time taken by the SMT-solver (both times in seconds). The second line reports the heap size (in Mbytes) required by z3. In every case the specification is satisfiable, property p_1 does not hold (the tool returns a counterexample), while property p_2 holds ("unsat" is returned). In addition to the results shown in the table, a variant of Formula (19) where $test_{0<c<\Delta}$ is used instead of $test_{0<c\leqslant\Delta}$ (i.e., $\leqslant$ is replaced by $<$) is shown to not hold, and a counterexample is obtained in less than 1 second.

Finally, we present an interesting behavior over unrestricted signals. The behavior is captured by the following formulae, which state that p and q only occur in isolated instants, with p occurring exactly every 80 time units, and q occurring within 80 time units in the past from each p (origin excluded).

$$\begin{gathered}\mathbf{G}^i\begin{pmatrix}\mathbf{G}_{(0,80)}(\neg p) \Rightarrow \mathbf{G}_{(80,160)}(\neg p) \quad \wedge \\ (p \Rightarrow \mathbf{F}_{(0,160)} p) \;\wedge\; (q \Rightarrow (\neg q)\,\mathbf{U}\,\top)\end{pmatrix} \wedge \\ p \;\wedge\; \mathbf{G}_{(0,80)}(\neg p) \;\wedge\; \mathbf{G}_{(0,\infty)}(p \Rightarrow \mathbf{P}_{(0,80)} q)\end{gathered} \tag{26}$$

In this case, the bound $k = 10$ is enough to prove that the formula is satisfiable: a model is produced in about 40 secs. In around the same time, the solver shows that property $\mathbf{G}^i(p \Rightarrow \mathbf{F}_{(0,80)}(q))$ holds for model (26) (up to the considered bound), whereas property $\mathbf{G}^i(q \Rightarrow \mathbf{F}_{(0,80)}(q))$ does not hold. It is worth noticing that, in Formula (26), the constants 80 and 160 occurring in the temporal modalities are significantly greater than the above bound $k = 10$, since in principle any value is possible for the clock increments between two consecutive positions. Therefore, the length of the intervals described by a CLTLoc model is independent of the bound k, as long as k is large enough to capture all change points that are necessary to build a periodic sequence of regions.

9 Conclusions

This paper investigates a bounded approach to satisfiability checking of an extension of CLTL where variables behave like clocks (CLTLoc). The decidability of the logic (by means of an automata-based technique) is shown first, followed by an encoding into a decidable SMT problem. This encoding, implemented in our $\mathtt{ae^2zot}$ tool, allows, both in principle and in practice, the use of SMT solvers to check the satisfiability of CLTLoc. We provide a short but non-trivial example of a CLTLoc specification describing a timed behavior over continuous time, which should demonstrate the effectiveness of this approach, as we are able to (dis)prove various properties of the specification. The paper also outlines continuous time, metric temporal logics, namely MITL and $\mathrm{MTL}_{(0,\infty)}$ (a generalization of QTL), showing that their extension pMITL, allowing existential propositional quantifiers, is as expressive as CLTLoc over the pointwise semantics. An encoding of MITL over the continuous semantics into CLTLoc is implemented in our `qtlsolver` tool. This shows that CLTLoc can be considered as a target language to reduce decision problems of various continuous-time formalisms, such as temporal logics, but in principle also Timed Automata or Timed Petri Nets.

To the best of our knowledge, our approach is the first allowing an effective implementation of a fully automated verification tool for continuous-time metric temporal logics such as MITL. The tool is still a non-optimized prototype, whose performance might also be substantially improved in future versions. Clearly, verification of formulae requiring many clocks may in general be infeasible, since satisfiability of MITL is EXPSPACE-complete (but we also support verification of an interesting, PSPACE-complete fragment of MITL). However, in practice a large number of clocks is not very frequent, and the examples of MITL formulae that we studied were verified in a fairly short time.

Acknowledgements We would like to thank the anonymous reviewers for their comments, which have greatly helped improve the presentation of the paper.

References

1. qtlsolver. available from `qtlsolver.googlecode.com`.
2. Zot: a bounded satisfiability checker. available from `zot.googlecode.com`.
3. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
4. R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.
5. G. Audemard, A. Cimatti, A. Kornilowicz, and R. Sebastiani. Bounded model checking for timed systems. In *Proc. of FORTE*, pages 243–259, 2002.
6. B. Badban and M. Lange. Exact incremental analysis of timed automata with an SMT-solver. In *FORMATS*, volume 6919 of *LNCS*, pages 177–192. 2011.
7. J. Bengtsson and W. Yi. Timed automata: Semantics, algorithms and tools. In *Lect. on Concurrency and Petri Nets*, volume 3098 of *LNCS*, pages 87–124. 2004.
8. M. M. Bersani, A. Frigeri, A. Morzenti, M. Pradella, M. Rossi, and P. S. Pietro. Constraint LTL satisfiability checking without automata. *Journal of Applied Logic*, 2014. In Press, available from `http://www.sciencedirect.com/science/article/pii/S1570868314000615`.
9. M. M. Bersani, A. Frigeri, A. Morzenti, M. Pradella, M. Rossi, and P. San Pietro. Bounded reachability for temporal logic over constraint systems. In *TIME 2010*, pages 43–50. IEEE Computer Society, 2010.
10. M. M. Bersani, A. Frigeri, M. Rossi, and P. San Pietro. Completeness of the bounded satisfiability problem for constraint LTL. In *Reachability Problems*, volume 6945 of *LNCS*, pages 58–71. 2011.
11. M. M. Bersani, M. Rossi, and P. San Pietro. Deciding continuous-time metric temporal logic with counting modalities. In *Reachability Problems*, volume 8169 of *LNCS*, pages 70–82. 2013.
12. M. M. Bersani, M. Rossi, and P. San Pietro. Deciding the satisfiability of MITL specifications. In *Proc. of the Int. Symp. on Games, Automata, Logics and Formal Verification (GandALF)*, pages 64–78, 2013.
13. M. M. Bersani, M. Rossi, and P. San Pietro. On the satisfiability of metric temporal logics over the reals. In *Proc. of the Int. Work. on Automated Verification of Critical Systems (AVOCS)*, pages 1–15, 2013.

14. M. M. Bersani, M. Rossi, and P. San Pietro. A logical characterization of timed (non-)regular languages. In *Mathematical Foundations of Computer Science*, volume 8634 of *Lecture Notes in Computer Science*, pages 75–86. 2014.
15. M. M. Bersani, M. Rossi, and P. San Pietro. An SMT-based approach to satisfiability checking of MITL. *Information and Computation*, 2015. To appear.
16. A. Biere, K. Heljanko, T. A. Junttila, T. Latvala, and V. Schuppan. Linear encodings of bounded LTL model checking. *Log. Met. in Comp. Sci.*, 2(5), 2006.
17. P. Bouyer, F. Chevalier, and N. Markey. On the expressiveness of TPTL and MTL. *Information and Computation*, 208(2):97–116, 2010.
18. E. M. Clarke, D. Kroening, J. Ouaknine, and O. Strichman. Completeness and complexity of bounded model checking. In *Verification Model Checking and Abstract Interpretation*, volume 2937 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2004.
19. S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. *Information and Computation*, 205(3):380–415, 2007.
20. D. D'Souza and P. Prabhakar. On the expressiveness of mtl in the pointwise and continuous semantics. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(1):1–4, 2007.
21. D. D'Souza and N. Tabareau. On timed automata with input-determined guards. In *FORMATS/FTRTFT '04*, volume 3253 of *Lecture Notes in Computer Science*, pages 68–83. 2004.
22. J. Ferrante and C. Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM J. Comput.*, 4(1):69–76, 1975.
23. T. A. Henzinger, J.-F. Raskin, and P.-Y. Schobbens. The regular real-time languages. In *In Proc. 25th Int. Coll. Automata, Languages, and Programming (ICALP'98*, pages 580–591. Springer-Verlag, 1998.
24. Y. Hirshfeld and A. Rabinovich. Timer formulas and decidable metric temporal logic. *Information and Computation*, 198(2):148 – 178, 2005.
25. Y. Hirshfeld and A. M. Rabinovich. Logics for real time: Decidability and complexity. *Fundamenta Informaticae*, 62(1):1–28, 2004.
26. L. Khachiyan. Polynomial algorithms in linear programming. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.
27. O. Maler, D. Nickovic, and A. Pnueli. From MITL to timed automata. In *Proc. of FORMATS*, volume 4202 of *LNCS*, pages 274–289. 2006.
28. Microsoft Research. Z3: An efficient SMT solver. http://research.microsoft.com/en-us/um/redmond/projects/z3/.
29. G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, Oct. 1979.
30. P. Niebert, M. Mahfoudh, E. Asarin, M. Bozga, O. Maler, and N. Jain. Verification of timed automata via satisfiability checking. In *FTRTFT*, volume 2469 of *LNCS*, pages 225–243. 2002.
31. D. C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12(3):291 – 302, 1980.
32. J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *LICS*, pages 188–197, 2005.
33. M. Pradella, A. Morzenti, and P. San Pietro. Bounded satisfiability checking of metric temporal logic specifications. *ACM Trans. on Softw. Eng.g and Meth. (TOSEM)*, 22(3):20, 2013.
34. A. Rabinovich. Complexity of metric temporal logics with counting and the Pnueli modalities. *Th. Comp. Sci.*, 411:2331–2342, 2010.
35. P.-Y. Schobbens, J.-F. Raskin, and T. A. Henzinger. Axioms for real-time logics. *Theor. Comput. Sci.*, 274(1-2):151–182, 2002.
36. A. Tarski. A decision method for elementary algebra and geometry. Univ. of California Press, second ed., Berkeley, 1951.