Expressive Equivalence Between Decidable Freeze and Metric Timed Temporal Logics.

Hsi-Ming Ho
University of Sussex, UK

Shankara Narayanan Krishna
Indian Institute of Technology Bombay, Mumbai, India

Khushraj Madnani
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany
Indian Institute of Technology Guwahati, India

Rupak Majumdar
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Paritosh Pandya
Indian Institute of Technology Bombay, Mumbai, India

Abstract

We demonstrate a surprising and first-of-its-kind expressive equivalence between decidable metric and freeze logics over timed words in pointwise semantics. Our main result states that Metric Interval Temporal Logic with future, past and Pnueli modalities, MITPPL, and full unilateral timed propositional temporal logic with both future and past temporal modalities, UPTL, have identical expressiveness. One of the highlights of this paper, which allows for this equivalence, is to prove that UPTL formulas admit monadic decomposition. Our result also implies that several decidable logics for real-time specifications, such as one-variable UPTL, unilateral MITPPL, and Q2MLO, are all expressively equivalent, and the reductions between them are effective. Hence, our result unifies the fragmented expressiveness boundary of timed temporal logics. As corollaries, we resolve the open question of the decidability for full UPTL, and the variable or clock hierarchy problem for the future fragment of UPTL.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases Timed Propositional Temporal Logic, Expressiveness, Metric Interval Temporal Logic, Satisfiability, Decidability

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2025.24

Funding This research was funded in part by the Deutsche Forschungsgemeinschaft (DFG) project 389792660 TRR 248-CPEC.

1 Introduction

Timed logics provide means to declaratively specify properties of real-time systems. They extend discrete time temporal logics such as linear temporal logic (LTL) by associating real-time intervals with temporal modalities, or with freeze quantifiers. An important research goal in the algorithmic verification of real-time systems is to balance the desire for more expressive logics with their decidability and tractability. Timed logics have two main semantics : *pointwise* (or event-based), where it is represented as a sequence of (events, timestamp) pairs called a timed word, and *continuous*, where it is represented as a signal (as a function from non-negative real numbers to a set of events).

There has been a long quest for highly expressive, yet decidable real-time temporal logics [1, 32, 9, 11, 26, 21, 24, 15, 8, 22, 23]. It is well known that timed propositional temporal logic TPTL, which extends LTL with "freeze" quantifiers to record the time elapsed [3], and

36th International Conference on Concurrency Theory (CONCUR 2025).
Editors: Patricia Bouyer and Jaco van de Pol; Article No. 24; pp. 24:1–24:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

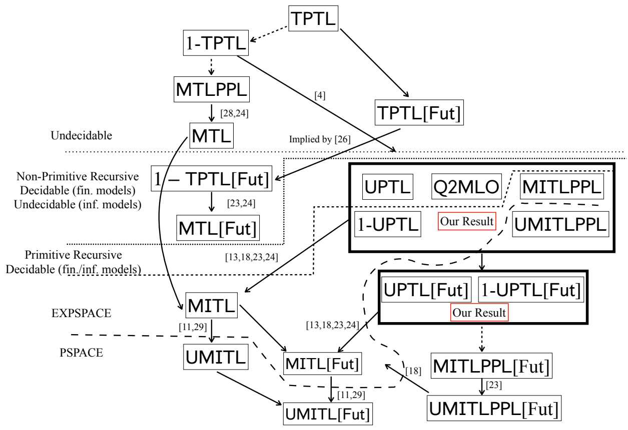

Figure 1 The expressiveness comparison (over timed words) of various timed logics studied in the literature. An arrow from L_1 to L_2 means that L_1 is more expressive than L_2. A dashed arrow between L_1 and L_2 means that strict containment wrt. expressiveness is still open. A solid arrow implies strict containment. Logics within the same box are equi-expressive. Our main contribution is to establish the equivalences shown in this figure.

metric temporal logic MTL, which extends LTL with temporal operators indexed by real-time intervals [20], are expressive but undecidable [4, 28] in both pointwise and continuous semantics. Moreover, TPTL (respectively, TPTL[Fut] the future only fragment of TPTL) is strictly more expressive than MTL (and MTL[Fut], respectively) in both pointwise and continuous semantics [5, 29]. This has led to a search for logics that restrict the expressiveness of TPTL and MTL in various ways to recover the decidability.

The continuous semantics naturally extends the first order theory of untimed words, and techniques from the untimed setting can be generalized, albeit with non-trivial extensions, such as adapting the separation lemma for expressive completeness [19]. In the pointwise semantics, these expressive completeness (and other equivalences) break down, leading to fragmented sublogics with varying expressiveness. This makes adapting traditional techniques difficult, and expressive equivalences in pointwise semantics are rare. Even though the continuous semantics is theoretically more well-behaved [12], the pointwise semantics is more amenable to practical implementations for verification applications [7, 10]. Currently, the expressiveness landscape for decidable real-time temporal logics – especially under the pointwise semantics – is quite fragmented: there are a number of different proposals that maintain decidability, but many questions remain unanswered about their relative expressive power. This paper attempts to fill in this gap.

Figure 1 shows a reasonably complete view of the current decidability/expressiveness landscape in the pointwise semantics[1]. The arrows $\mathsf{L} \to \mathsf{L}'$ in the figure indicate that logic L is strictly more expressive than logic L'. At the top of the figure sits the undecidable logics TPTL and MTL. Towards the bottom sits MITL, with an EXPSPACE-complete satisfiability problem [1, 2]. More recently, a number of other decidable logics have been proposed and studied independently. These logics either extend MITL with counting operators (Pnueli modalities) MITPPL [14, 13], or restrict TPTL to *unilateral* constraints of the form $(T{-}x){\le}u$ or $(T{-}x){\ge}l$ (UPTL) [24] or define syntactic fragments of a monadic first-order logic with metric and order restricting the number of variables (Q2MLO)[11, 9, 30, 32].

[1] In this paper, we implicitly assume logics have both future and past modalities. When we restrict a logic L to only future modalities, we explicitly denote it as L[Fut]. Our results span finite and infinite words.

Our Contributions.

1. **Equivalence between Metric and Freeze timed logics**. We show a surprising expressive equivalence between the metric logic MITPPL and the freeze logic UPTL. To the best of our knowledge, this is the first expressive equivalence result for a metric temporal logic and a freeze logic which works in both pointwise and continuous semantics. The key technical ingredient that enables us to prove our main result is that the formulas of UPTL (and UPTL[Fut]) admit *monadic decomposition.* That is, every UPTL (or UPTL[Fut]) formula with multiple free variables can be reduced to a Boolean combination of formulas containing only one free variable preserving equivalence. The only other such known equivalences are in continuous semantics between TPTL[$\mathbb{Q}$] and MTL[$\mathbb{Q}$] [19], and TPTL and MTL with counting [18]. Together with known equivalences from the literature [17], this implies that the logics MITPPL, UPTL, and Q2MLO are all expressively equivalent (see the box in the middle of Figure 1).
2. **Potential Applications**. The reduction of freeze logics to metric logics has practical implications for algorithmic verification. A key feature of MITL (and MITPPL) is that the satisfaction of any subformula is "local", i.e. can be defined with respect to a single point – the point of assertion. This property simplifies compositional verification for decidable metric logics and has enabled tools like MightyL [6] for MITL[Fut] and the more recent MightyPPL [27] for MITPPL. Since our reduction translates freeze logics into metric logics, it facilitates the use of these tools for verifying UPTL properties, broadening their applicability in practice.
3. **Refining the Expressiveness Boundary of Freeze Timed Logics**. We address open problems from [24] regarding the decidability of the full logic UPTL (with future and past modalities). In particular, we show that (1) the satisfiability and model-checking problems for UPTL are decidable, and (2) UPTL has the same expressive power as its one-variable fragment, indicating that increasing the number of variables in UPTL does not enhance expressiveness.

2 Preliminaries

Let $\mathbb{R}_{\geq 0}$, $\mathbb{N}$, $\mathbb{N}_{>0}$, $\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$, and $\mathbb{Z}$ respectively represent the set of non-negative reals, naturals (including 0), naturals (excluding 0), rationals, non-negative rationals, and integers. Let $\langle$ denote left open "(" or left closed "[", and $\rangle$ denote right open ")" or right closed "]", and let $\mathbb{I}$ denote the set of all intervals $\langle l, u\rangle$ for $l \leq u$, $l \in \mathbb{N}$, $u \in \mathbb{N} \cup \{\infty\}$. Let true and false be denoted by $\top$ and $\bot$, respectively. Given any sequence $\eth = a_1, a_2, \ldots,$, $|\eth|$ denote length of the sequence.

Let AP be a finite set of propositional variables, and let $\Sigma_{\mathsf{AP}} = 2^{\mathsf{AP}}$ be the finite set containing all subsets of AP. An infinite timed word (or just timed words) ρ over Σ_{AP} is an infinite sequence of pairs of letters and non-decreasing time stamps of the form $\rho = (\sigma_1, \tau_1)(\sigma_2, \tau_2)\ldots$ where for all $i > 0$, $\sigma_i \in \Sigma_{\mathsf{AP}}, \tau_i \in \mathbb{R}_{\geq 0}$, and $0 \leq \tau_i \leq \tau_{i+1}$. A finite timed word is a finite sequence of pairs of letters and non-decreasing time stamps.

For $i > 0$, we define $\rho[i] = \sigma_i$ for the element of Σ_{AP} at the i^{th} position of ρ. The set of all timed words over Σ is denoted $T\Sigma^\omega$. A timed word is called *Zeno* if the sequence $(\tau_i)_{i\geq 0}$ converges, and non-Zeno otherwise. We restrict ourselves to non-Zeno infinite timed words (henceforth called timed words), which is the usual convention, as Zeno words allow infinite actions within a finite duration, which does not model a natural behaviour.

2.1 Metric Temporal Logic with Pnueli Modalities, MTLPPL

Formulas of MTLPPL are defined over a set of propositional variables AP as follows:
$\varphi := a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\, \mathbf{U}_I\, \varphi \mid \varphi\, \mathbf{S}_I\, \psi \mid \mathbf{Pn}_I(\varphi_1, \ldots, \varphi_k) \mid \overleftarrow{\mathbf{Pn}}_I(\varphi_1, \ldots, \varphi_k)$
where $a \in \mathsf{AP}$ and I is an interval in $\mathbb{I}$. Given a timed word $\rho = (\sigma_1, \tau_1)(\sigma_2, \tau_2)\ldots$ over Σ_{AP}, a position $i \in \mathbb{N}_{>0}$, and any $a \in \mathsf{AP}$, we define the *pointwise-semantics* of MTLPPL inductively. We define $\rho, i \models a$ iff $a \in \sigma_i$, and conjunction, disjunction, and negation in the usual way. For temporal modalities, we define:

- $\rho, i \models \varphi_1\, \mathbf{U}_I\, \varphi_2$ iff $\exists j > i$ s.t. $\tau_j - \tau_i \in I$, $\rho, j \models \varphi_2$, and $\forall i < k < j, \rho, k \models \varphi_1$;
- $\rho, i \models \varphi_1\, \mathbf{S}_I\, \varphi_2$ iff $\exists j < i$ s.t. $\tau_i - \tau_j \in I$, $\rho, j \models \varphi_2$, and $\forall j < k < i, \rho, k \models \varphi_1$;
- $\rho, j \models \mathbf{Pn}_I(\varphi_1, \ldots, \varphi_k)$ iff $\exists i_k > i_{k-1} > \ldots > i_1 > j$ s.t. $\forall 1 \le n \le k, \tau_{i_n} - \tau_j \in I$, and $\rho, i_n \models \varphi_n$;
- $\rho, j \models \overleftarrow{\mathbf{Pn}}_I(\varphi_1, \ldots, \varphi_k)$ iff $\exists i_k < i_{k-1} < \ldots < i_1 < j$ s.t. $\forall 1 \le n \le k, \tau_j - \tau_{i_n} \in I$, and $\rho, i_n \models \varphi_n$.

We define the language of φ as $[\![\varphi]\!] = \{(\rho, i) | \rho, i \models \varphi\}$. Notice that we use strict semantics for both $\mathbf{U}$ and $\mathbf{S}$, as it can easily express "next" and "previous" modalities, unlike the non-strict variants. Derived operators $\Diamond_I$ (eventually), $\Box_I$ (globally), $\overleftarrow{\Diamond}_I$ (past), $\overleftarrow{\Box}_I$ (globally in the past), O (next) and $\overleftarrow{\mathsf{O}}$ (previous) are written in terms of $\mathbf{U}$ and $\mathbf{S}$ as follows. $\Diamond_I \varphi = \top\, \mathbf{U}_I\, \varphi$, $\Box_I \varphi = \neg \Diamond_I \neg\varphi$, $\overleftarrow{\Diamond}_I \varphi = \top\, \mathbf{S}_I\, \varphi$, $\overleftarrow{\Box}_I \varphi = \neg \overleftarrow{\Diamond}_I \neg\varphi$, $\mathsf{O}\varphi = \bot\, \mathbf{U}_{[0,\infty)}\, \varphi$ and $\overleftarrow{\mathsf{O}}\, \varphi = \bot\, \mathbf{S}_{[0,\infty)}\, \varphi$. For the sake of succinctness, we omit the subscript when the intervals are $[0, \infty)$. Hence, $\varphi\, \mathbf{U}_{[0,\infty)}\, \psi$ and $\varphi\, \mathbf{S}_{[0,\infty)}\, \psi$ are written as $\varphi\, \mathbf{U}\, \psi$ and $\varphi\, \mathbf{S}\, \psi$, respectively. We define some important subclasses of MTLPPL. The subclass *Metric Interval Temporal Logic with Pnueli modalities*, written MITPPL, consists of all MTLPPL formulas where the intervals I are non-singular (i.e., of the form $\langle l, u\rangle$, where $l < u$). Metric Temporal Logic (MTL) and Metric Interval Temporal Logic (MITL) are fragments of MTLPPL and MITPPL, respectively, without Pnueli modalities. Unilateral fragments of MITL and MITPPL (denoted by UMITL and UMITPPL, respectively) are subclasses where the intervals are either of the form $\langle 0, u\rangle$ or $\langle l, \infty)$.

Example. For the UMITPPL formula $\varphi = \mathbf{Pn}_{(0,2)}(a, b, c)$, $\rho, 1 \models \varphi$ for $\rho = (\{a\}, 0)(\{a\}, .5)$ $(\emptyset, .9)$ $(\{b, c\}, 1.1)(\{a, b, c\}, 1.8)\ldots$, since $\tau_5 - \tau_1 = 1.8, \tau_4 - \tau_1 = 1.1, \tau_2 - \tau_1 = 0.5$ all lie in $(0, 2)$, and $\rho, 2 \models a, \rho, 4 \models b$ and $\rho, 5 \models c$. However, $\rho', 1 \not\models \varphi$ for $\rho' = (\{a\}, 0)(\{c\}, .1)$ $(\{b, c\}, 1.1)(\{a, b\}, 1.9)(\emptyset, 2)\ldots$.

2.2 Timed Propositional Temporal Logic, TPTL

TPTL [4] is an extension of linear temporal logic (LTL) with real-valued variables and *freeze quantifiers* which allow subformulas to freeze the timestamps of the present point to a particular variable. We define the TPTL formulas in negation normal form, i.e. where the negation operator is only applied to atomic formulas. It is well known [24] that any formula in TPTL can be converted to an equivalent formula in negation normal form (NNF), and as such, we introduce TPTL in NNF, where all negations appear only at the level of propositional variables. Formulas in TPTL (in negation normal form) over a finite set X of freeze variables and AP is defined as[2]:
$\varphi := a \mid \neg a \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid x.\varphi \mid (T - x) \sim c \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{S} \psi \mid \Box\varphi \mid \overleftarrow{\Box}\varphi$, $a \in \mathsf{AP}$, $x \in X$, $c \in \mathbb{Z}$.

Given a timed word $\rho = (\sigma_1, \tau_1)(\sigma_2, \tau_2)\ldots$ over Σ_{AP}, a position $i \in \mathbb{N}_{>0}$, a valuation $\nu : X \to \mathbb{R}_{\ge 0}$ of the variables, and any $a \in \mathsf{AP}$, we define the *pointwise-semantics* of TPTL inductively. Propositions $a \in \mathsf{AP}$, conjunctions, disjunctions, $\mathbf{U}$, $\mathbf{S}$, $\Box$, and $\overleftarrow{\Box}$ formulas are evaluated as in LTL. In addition,

---

[2] In literature, the negation normal form usually admits a "Release" modality (and its past counterpart), but they can be expressed using "$\mathbf{U}$" and "$\Box$" (respectively, $\mathbf{S}$ and $\overleftarrow{\Box}$) without negations. Moreover, we use strict versions of $\mathbf{U}$ and $\mathbf{S}$ modalities.

- $\rho, i, \nu \models x.\varphi$ iff $\rho, i, \nu' \models \varphi$, where $\nu' = \nu[x \leftarrow \tau_i]$ is a valuation s.t. $\nu'(y) = \nu(y)$ for $y \neq x$, and $\nu'(x) = \tau_i$. Further, $x.\varphi$ is called a *freeze quantifier*. The timestamp τ_i of ρ is "frozen" in $x \in X$ before evaluating φ.
- $\rho, i, \nu \models (T - x) \sim c$ iff $\tau_i - \nu(x) \sim c$. Intuitively, at the ith position of ρ, the time elapsed since the last time x was frozen (i.e. $\tau_i - \nu(x)$) is compared with $c \in \mathbb{Z}$. If the last time x was frozen happens to be at position $j \leq i$, then $\nu(x) = \tau_j$ and $\tau_i - \tau_j$ is the time elapse since freezing x.

We denote by k-TPTL the class of TPTL formulas with k freeze variables; so TPTL=$\bigcup_{k\geq 0}$ k-TPTL. The derived modalities next (O), eventually ($\Diamond$), sometime ago ($\overleftarrow{\Diamond}$), and previous ($\overleftarrow{\mathsf{O}}$) are defined as usual.

Open and Closed TPTL Formulas. A constraint $(T - x) \sim c$ appearing in a TPTL formula φ is called *closed* if it lies within the scope of a freeze variable $x.$, and *open* otherwise. A freeze variable x in a TPTL formula φ is called a *free variable* if any occurrence of x in φ appears in an open constraint. We write $\varphi(x_1, \ldots, x_k)$ to denote that the free variables in φ are $x_1, \ldots, x_k$. When the free variables in a formula are clear from the context, we just write φ for $\varphi(x_1, \ldots, x_k)$. A TPTL formula is *closed* iff it does not contain any free variable; an *open* formula is one which is not closed.

▶ **Example.** *The formula* $\psi{=}x.((a{\wedge}(T - z){\leq}2)\ \mathbf{U}\ [b{\wedge}y.(\Diamond\, a{\wedge}(T{-}y){<}1 \wedge (T{-}x){<}2)])$ *is open.* $\varphi_{cnt}{=}x.(\Diamond(a{\wedge}\Diamond(a{\wedge}(T - x){<}1)))$ *is closed.*

The satisfaction of a closed TPTL formula is independent of any valuation ν of its variables since each constraint $(T - x) \sim c$ lies in the scope of a freeze quantifier $x.$, the value of x only depends on the time elapsed since it was frozen. Hence, we define the language of a TPTL formula φ only if it is closed. We define the language of a closed formula φ as $[\![\varphi]\!] = \{(\rho, i) | \rho, i \models \varphi\}$. The formula φ_{cnt} above encodes the property that there are at least two a in the next time interval of unit length; it is known [11] that φ_{cnt} does not have an equivalent MTL formula. Then $(\rho, 1){\in}[\![\varphi_{cnt}]\!]$ where $\rho{=}(\{a\}, 0.2)(\{a, b\}, 0.45)(\emptyset, .75)(\{b\}, .8)(\{c\}, .95)\ldots$.

Polarity of a TPTL formula. Let $(T - x) \sim c$ be an open constraint in formula φ. The *polarity* of $(T - x) \sim c$ denoted $\mathsf{Pol}((T - x) \sim c)$ is defined to be the singleton set $\{\leq\}$ if $\sim \in \{<, \leq\}$, and $\{\geq\}$ if $\sim \in \{>, \geq\}$. The polarity of a TPTL formula φ denoted $\mathsf{Pol}(\varphi)$ is defined when all open constraints in φ have the *same* polarity. $\mathsf{Pol}(\varphi)$ is the polarity of the open constraints in φ. We define the polarity of a closed TPTL formula as the set $\{\leq, \geq\}$ since it has no open constraints.

Unilateral Timed Propositional Temporal Logic, UPTL. A TPTL formula φ is UPTL iff $\mathsf{Pol}(\psi)$ is defined for all subformulas ψ of φ. Let k-UPTL denote UPTL formulas using k freeze variables, and UPTL=$\bigcup_{k\geq 0}$ k-UPTL.

▶ **Example.** *Consider* $\varphi = x.[\Diamond(a \wedge (T - x) > 2 \wedge (T - y) > 3)]$. $\mathsf{Pol}((T - y) > 3) = \{\geq\} = \mathsf{Pol}(\varphi)$*. Likewise,* $\mathsf{Pol}(\kappa) = \{\leq\}$ *for* $\kappa = x.(\Diamond(a \wedge (T - x) > 2 \wedge (T - y) < 3))$*. However,* $\mathsf{Pol}(\psi)$ *is undefined for the subformula* $\psi = \Diamond(a \wedge (T - x) > 2 \wedge (T - y) < 3)$ *of* κ*, since* $\mathsf{Pol}((T - x) > 2) = \{\geq\}$ *while* $\mathsf{Pol}((T - y) < 3) = \{\leq\}$.

Future Fragments of Logics. For any logic L defined so far, L[Fut] denotes its future fragment, one which does not involve any past (or past Pnueli) modalities. For instance, UPTL[Fut] is the fragment of UPTL with only future modalities [24].

▶ Remark 2.1. Notice that MTLPPL (and UMITPPL) properties can be translated to 1-TPTL (1-UPTL) preserving the languages by applying the following reduction inductively. $\mathbf{Pn}_I(a_1,\dots,a_k) \equiv x.(\Diamond(a_1 \wedge (T{-}x){\in}I \wedge \Diamond(a_2 \wedge (T-x){\in}I \wedge \Diamond(\dots\Diamond(a_k \wedge (T-x){\in}I)\dots)$. Moreover, $a_1 \mathbf{U}_I\, a_2 \equiv x.(a_1 \mathbf{U}\, (a_2 \wedge (T{-}x){\in}I))$. Similar reductions apply for $\mathbf{S}$ and $\overleftarrow{\mathbf{Pn}}$.

2.3 First Order Logic of Metric and Order, FOMLO

Formulas of FOMLO are defined over a signature containing *monadic* predicates $a(x)$ where $a \in \mathsf{AP}$, *order* predicates $x \sim y$ for $\sim \in \{<,>,=,\leq,\geq\}$ and a set of *metric* predicates $\Delta = \{(x \ominus y) \sim c \mid c \in \mathbb{Z}, \sim \in \{\leq,\geq,<,>,=\}\}$, where x, y are first-order variables. Syntax of FOMLO formula $\varphi(x_1,\dots,x_k)$ is defined as:
$\varphi(x_1,\dots,x_k) :: a(x_i) \mid x_i \sim x_j \mid (x_i{\ominus}x_j){\sim}c \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathcal{Q}x.\varphi(x,x_1,\dots,x_k)$,
where $\mathcal{Q} \in \{\exists,\forall\}$ and is interpreted over timed words ρ along with valuation
$\nu : \{x_1,\dots,x_k\} \rightarrow \mathbb{N}_{>0}$. $\rho,\nu \models (x{\ominus}y){\sim}c$ for $\nu(x) = j, \nu(y) = i$ iff $\tau_j - \tau_i \sim c$, where τ_j and τ_i are timestamps of points i, j in ρ, respectively. Hence, the variables range over positions of timed words. The semantics of the remaining operators are identical to that of first-order logic with $<$ predicate (FO[<]) over words(see [31][11] for more details). For a FOMLO formula $\varphi(x)$ with one free variable, the language $[\![\varphi(x)]\!]$ is defined as $\{(\rho,i) \mid \rho,\nu \models \varphi(x)$ with $\nu(x) = i\}$.

Q2MLO [11] is a syntactic fragment of FOMLO where, in the grammar of FOMLO, metric predicates $(x \ominus y) \sim c$ are replaced by the timed monadic formula $\psi(x)$ whose syntax is $\psi(x) := \exists y.[(y{\ominus}x < u) \wedge (y{\ominus}x > l) \wedge \varphi(x,y)]$ where $l, u \in \mathbb{Z}$, $l \neq u$, and $\varphi(x,y)$ is a well-formed Q2MLO formula. Q2MLO can be seen as an extension of FO[<] with timed monadic formulas.

Example. The formula $\exists y.\exists z.[x{>}y{>}z \wedge a(y) \wedge b(z) \wedge (z \ominus x) > 15 \wedge (z \ominus y) < 30]$
is in FOMLO but not in Q2MLO, as there is a subformula containing time constraints involving two free variables.

2.4 Expressive Equivalence

Let ϕ and ψ be in TPTL. Then ϕ is said to be equivalent to ψ iff for any timed word ρ, any point i of ρ, and any valuation ν over free variables of ϕ and ψ, $\rho,i,\nu \models \phi$ iff $\rho,i,\nu \models \psi$. The equivalence between two Q2MLO formulas is defined in the classical sense. We say that two formulas φ and ψ are *language equivalent* iff $[\![\varphi]\!] = [\![\psi]\!]$. Logic L_1 is at least as expressive as L_2 (denoted by $\mathsf{L}_2 \preceq \mathsf{L}_1$) iff for any formula φ in L_1, there exists a language equivalent ψ in L_2. L_1 and L_2 are said to be equally expressive (denoted by $\mathsf{L}_1 \equiv \mathsf{L}_2$) iff $\mathsf{L}_1 \preceq \mathsf{L}_2$ and $\mathsf{L}_2 \preceq \mathsf{L}_1$. Finally, L_1 is said to be strictly more expressive than L_2 (denoted by $\mathsf{L}_2 \prec \mathsf{L}_1$) iff $\mathsf{L}_2 \preceq \mathsf{L}_1$, and L_2 and L_1 are not equally expressive.

Extension to rational constants. For any logic L, let $\mathsf{L}[\mathbb{Q}]$ denote the extension of L where the constants used in the timing constraints are from the set $\mathbb{Q}_{\geq 0}$. For example, $a\, \mathbf{U}_{[0,\frac{1}{2}]}\, b$ is not in MITL but is in MITL[$\mathbb{Q}$].

3 Main Results

In this section, we discuss our main result (Theorem 3.1) connecting UPTL and MITPPL. A "guided tour" to navigate supporting results for each theorem is given in Figure 2.

▶ **Theorem 3.1.**
(1) UPTL $\equiv$ 1-UPTL $\equiv$ MITPPL $\equiv$ UMITPPL $\equiv$ Q2MLO.
(2) MITL[$\mathbb{Q}$]$\equiv$UPTL[$\mathbb{Q}$]$\equiv$1-UPTL[$\mathbb{Q}$]$\equiv$MITPPL[$\mathbb{Q}$]$\equiv$UMITPPL[$\mathbb{Q}$]$\equiv$Q2MLO[$\mathbb{Q}$].

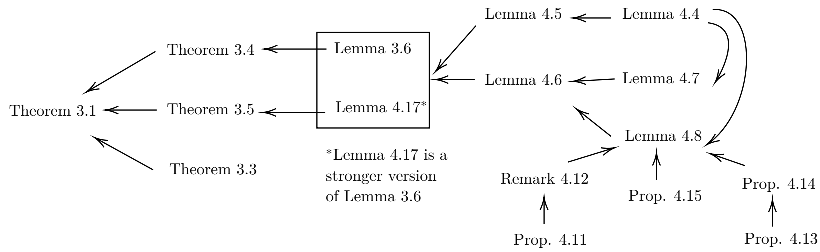

Figure 2 Dependencies between theorems and lemmas. $A \leftarrow B$ means proof of A uses B.

▶ Remark 3.2. Since MITPPL has no freeze variables, we compare it with closed UPTL formulas, similar to how Kamp's theorem compares LTL with FO[<] having one free variable. Our equivalence between 1-UPTL and UPTL also holds for open formulas. A reduction from open 1-UPTL formulas to an extension of Q2MLO with two free variables also holds, but it would lengthen the proof unnecessarily.

3.1 Overview of the proof of Theorem 3.1

Theorem 3.1 is a direct consequence of Theorem 3.3, Theorem 3.4 and Theorem 3.5 below. Theorem 3.3 below is a result of [16, 17].

▶ **Theorem 3.3** ([16, 17]).
(1) UMITPPL $\equiv$ MITPPL $\equiv$ Q2MLO.
(2) MITL[$\mathbb{Q}$] $\equiv$ UMITPPL[$\mathbb{Q}$] $\equiv$ MITPPL[$\mathbb{Q}$] $\equiv$ Q2MLO[$\mathbb{Q}$].

Thanks to the above theorem, we just need to show that UPTL $\equiv$ 1-UPTL $\equiv$ Q2MLO, and UPTL[$\mathbb{Q}$] $\equiv$ 1-UPTL[$\mathbb{Q}$] $\equiv$ Q2MLO[$\mathbb{Q}$]. The former is implied by two of our main results, namely Theorems 3.4 and 3.5. The latter is implied by the fact that the proofs of Theorems 3.4 and 3.5 do not assume the constants to be integers.

▶ **Theorem 3.4.** UPTL $\equiv$ 1-UPTL.

▶ **Theorem 3.5.** 1-UPTL $\equiv$ Q2MLO.

Theorems 3.4 and 3.5 rely on the following main technical lemma, Lemma 3.6.

▶ **Lemma 3.6** (Monadic Decomposition of UPTL). *There is an effective translation from any $\varphi(x_1, \ldots, x_k) \in$ UPTL to an equivalent UPTL formula ψ, where ψ is a Boolean combination of UPTL formulas with at most one free variable.*

We first discuss the proof of Theorem 3.4 assuming Lemma 3.6. The proof of Lemma 3.6 and Theorem 3.5 are deferred to sections 4, 5, respectively.

3.2 Proof of Theorem 3.4

1-UPTL $\preceq$ UPTL follows trivially from the fact that the former is a syntactic fragment of the latter. Hence, it suffices to show that UPTL $\preceq$ 1-UPTL. Let φ be any closed UPTL formula using variables $x_1, \ldots, x_k$. We construct a closed 1-UPTL formula ψ which uses only a single clock x using the following steps.

1. **Start with the innermost freeze quantifier over a variable,** x_i. Consider a subformula of the form $x_i.\varphi'(x_1, \ldots, x_k)$, such that φ' does not contain any subformula of the form $x_j.\varphi''$, where $x_i, x_j \in \{x_1, \ldots, x_k\}$.

2. **Monadically decompose the formula** φ'. By Lemma 3.6, we can construct an equivalent formula ψ' which is a Boolean combination of formulas of the form $\psi_j^m(x_j)$.
3. **Distribute the freeze quantifier** x_i. Freeze quantifiers distribute over $\wedge$ and $\vee$. Hence, $x_i.\varphi'(x_1, \ldots, x_k)$ can be equivalently written as the Boolean combination of formulas of the form $x_i.\psi_j^m(x_j)$. Moreover, $x_i.\psi_j^m(x_j)$ is equivalent to (i) $\psi_j^m(x_j)$ when $i \neq j$, and (ii) to $x.\psi_j^m(x)$ when $i = j$. Hence, we either remove the freeze quantifier $(x_i.)$ in case $i \neq j$, or rename the variable x_i to x. This step strictly reduces the number of freeze quantifiers over some variable $x_i \in \{x_1, \ldots, x_k\}$.
4. Repeat steps 1, 2, and 3 until we remove all the freeze quantifiers over variables $\{x_1, \ldots, x_k\}$. Hence, after performing the above steps at most n times, where n is the number of freeze quantifiers in φ, we get an equivalent formula ψ containing only one variable x. ◀

4 Proof of Lemma 3.6: Monadic Decomposition of UPTL

The main ingredient to achieve monadic decomposition is to come up with a novel syntactic subclass of UPTL called *Conjunctive Timed Clauses* (CoT, defined later) which has the following properties:

P1 CoT formulas are "easily" monadically decomposable;
P2 Any $\varphi \in$ UPTL can be expressed as a Boolean combination of CoT formulas.
Notice that Lemma 3.6 follows trivially if we can prove **P1** and **P2**.

▶ **Definition 4.1.** *(Conjunctive Timed Clauses (CoT)) A UPTL formula is a* conjunctive timed clause *(CoT) if it is generated by the following grammar:*

$$\psi ::= \phi \mid (T - x) \sim c \mid \psi \wedge \psi \mid \phi\, \mathbf{U}\, \psi \mid \phi\, \mathbf{S}\, \psi$$

where ϕ is any closed UPTL formula, $x \in X$ is a freeze variable, and $c \in \mathbb{Z}$.

Note that all open time constraints in CoT formulas appear on the right side of **U** and **S** modalities and/or conjunctions. Intuitively, we make sure that all the open time constraints are existentially quantified, and do not appear within the scope of a disjunction. Further, $\Diamond(\psi)$, $\overleftarrow{\Diamond}(\psi)$, $\mathsf{O}(\psi)$, and $\overleftarrow{\mathsf{O}}(\psi)$ are valid production rules as they are restricted forms of **U** and **S**. Moreover, $\Box, \vee$ formulas appears in a CoT only in ϕ, and hence all instances of $\Box, \vee$ in a CoT are within a closed subformula.

Example. $x.\Box(\Diamond((T - x) < 2))$ is a well-formed CoT formula but $\Box(\Diamond((T - x) < 2))$ is not. $\phi = \mathsf{O}((T - x) < 1 \wedge (T - y) < 3) \wedge [\{\Box(z.\Diamond(b \vee (T - z) < 3))\}\, \mathbf{U}\, \{(T - x) < 1\}]$ is in CoT. However, $\psi = (b \wedge (T - y) < 1)\, \mathbf{U}\, [x.(\Diamond(a \wedge (T - x) < 2))]$ is not in CoT; even though $x.(\Diamond(a \wedge (T - x) < 2))$ is in CoT, $(b \wedge (T - y) < 1)$ is not closed.

Untiming Property. To show that CoT satisfies properties **P1** and **P2**, we rely on a semantic property of CoT: they can be "untimed". Intuitively, formulas in CoT of polarity $\{\leq\}$ (resp. $\{\geq\}$) enjoy a certain form of "downward closure" (resp. "upward closure") w.r.t. their satisfaction by removing the open constraints appearing in them. This is described by the *Untiming Lemma* (Lemma 4.4), the name suggestive of the fact that they can be untimed by removing the open constraints under certain conditions. Given a TPTL formula ψ, let $\mathsf{Untime}(\psi)$ denote the set of all formulas obtained by substituting $\top$ for some/all/none of the open time constraints occurring in ψ.

Example. For $\psi = (T - x) < u_1 \wedge \Diamond((T - y) < u_2) \wedge x.\Diamond((T - x) < u_3)$, $\mathsf{Untime}(\psi) = \{\psi,\ \Diamond((T - y) < u_2) \wedge x.\Diamond((T - x) < u_3),\ (T - x) < u_1 \wedge x.\Diamond((T - x) < u_3),\ x.\Diamond((T - x) < u_3)\}$.

▶ **Definition 4.2** (The Untiming Property). *Let* $\bowtie \in \{\leq, \geq\}$. *A* TPTL *formula* $\psi(x_1, \ldots, x_k)$ *satisfies the* $\bowtie$ untiming property *iff for any* $\psi' \in \mathsf{Untime}(\psi)$, *a timed word* ρ, *a position* $j \in \mathbb{N}_{>0}$, *and a valuation* $\nu : \{x_1, \ldots, x_k\} \mapsto \mathbb{R}_{\geq 0}$, *we have* $\rho, j, \nu \models \psi$ *implies for all* $i \in \mathbb{N}_{>0}$ *with* $i \bowtie j$ *we have* $\rho, i, \nu \models \psi'$ *iff* $\rho, i, \nu \models \psi$.

▶ Remark 4.3. A formula of the form CoT $\vee$ CoT may not satisfy the untiming property. Consider $\psi_{ex} = a \vee (b \wedge ((T - x) \leq 2))$, $\mathsf{Pol}(\psi_{ex}) = \{\leq\}$, and $a \vee b$ is in $\mathsf{Untime}(\psi_{ex})$. Let ρ be any timed word, and j be a point in ρ where a holds. Then, $\rho, j, \nu \models \psi_{ex}$ for any clock valuation ν. However, for any point $i < j$ and any valuation ν over x, $\rho, i, \nu \models a \vee b$ iff $\rho, i, \nu \models \psi_{ex}$ need not be true; for instance, consider $b \wedge \neg a$ is true at i, $\tau_i = 50$ and $\nu(x) = 1$.

▶ **Lemma 4.4** (The Untiming Lemma). *Let* $\bowtie \in \{\leq, \geq\}$. *Any* CoT *formula* ψ *of polarity* $\bowtie$ *satisfies the* $\bowtie$ *untiming property.*

Proof. Let ψ be a formula in CoT. We prove for the case $\leq \in \mathsf{Pol}(\psi)$; a symmetric argument follows when $\geq \in \mathsf{Pol}(\psi)$. Fix a timed word $\rho = (a_1, \tau_1)(a_2, \tau_2)\ldots$, a point j of ρ and a valuation ν over the variables in ψ. Let $\psi' \in \mathsf{Untime}(\psi)$. Assuming that (A) $\rho, j, \nu \models \psi$, we now show that (B) for any point $i \leq j$, $\rho, i, \nu \models \psi$ iff $\rho, i, \nu \models \psi'$. We apply induction on the structure of the formula ψ. From now, we assume that (A) holds. Notice that (B-1) implies (B-2) holds trivially. This is because all the negations in the formula ψ apply only to propositional variables. Hence, all the timing constraints appear positively within ψ. Thus, replacing some/all timing constraints by true should make satisfaction easier. Hence, it suffices to show that for any point $i \leq j$, (B-2) implies (B-1).
Base Case. Let ψ be either a closed formula or an atomic clock constraint of the form $(T - x) \sim u$ where $\sim \in \{<, \leq\}$. If ψ is a closed formula then $\mathsf{Untime}(\psi) = \{\psi\}$. Hence, the lemma holds trivially. If $\psi = (T - x) \sim u$, then $\mathsf{Untime}(\psi) = \{\psi, \top\}$. For any $\psi' \in \mathsf{Untime}(\psi)$, (B-1) implies (B-2) is a tautology as discussed above. To show that for all $i \leq j$ (B-2) implies (B-1), we need to show that $\rho, i, \nu \models \top$ implies $\rho, i, \nu \models (T - x) \sim u$ for all $i \leq j$. Note that for any $i \leq j$, $\tau_i \leq \tau_j$. As (A) holds, we know $\tau_j - \nu(x) \sim u$; hence for any $i \leq j$, $\tau_i - \nu(x) \leq \tau_j - \nu(x) \sim u$, giving $\rho, i, \nu \models (T - x) \sim u$.
Induction Hypothesis. Assume the lemma holds for all strict subformulas of ψ.
Induction. We now show that given (A) and induction hypothesis for any $i \leq j$ (B-2) implies (B-1) given the induction hypothesis and (A).

There are three possible cases (1) $\psi = \psi_1 \wedge \psi_2$, (2) $\psi = \psi_1 \mathbf{U} \psi_2$, and (3) $\psi = \psi_1 \mathbf{S} \psi_2$. If ψ is an open formula, then it cannot be of the form $\psi_1 \vee \psi_2$ or $\Box(\psi_1)$ by definition of CoT.

We only discuss case (2), with case (3) being symmetric. Details of all cases are in Appendix A.

Case (2). $\psi = \psi_1 \mathbf{U} \psi_2$ (*Argument Sketch*). As ψ is in CoT, ψ_1 is a closed formula. Hence, $\mathsf{Untime}(\psi_1) = \{\psi_1\}$, and ψ' is of the form $\psi_1 \mathbf{U} \psi_2'$ where $\psi_2' \in \mathsf{Untime}(\psi_2)$. For any $\psi_2' \in \mathsf{Untime}(\psi_2)$, we now show that for any $i \leq j$, (B-2) $\rho, i, \nu \models \psi_1 \mathbf{U} \psi_2'$ implies (B-1)$\rho, i, \nu \models \psi_1 \mathbf{U} \psi_2$.

Intuitively, (A) implies that there is a point $j_{min} > j$ such that $\rho, j_{min}, \nu \models \psi_2$, and all the points between j and j_{min} satisfy ψ_1. Notice, by induction hypothesis, $\rho, j_{min}, \nu \models \psi_2'$. Moreover,(B-2) implies that there exists a point $i_m > i$ such that $\rho, i_m, \nu \models \psi_2'$ and all the points between i and i_m satisfy ψ_1. Either $i_m < j_{min}$, in which case, by induction hypothesis $\rho, i_m, \nu \models \psi_2'$ iff $\rho, i_m, \nu \models \psi_2$, and hence (B-1) holds. Or, $i_m > j_{min} > i$ such that $\rho, j_{min}, \nu \models \psi_2'$ (recall by induction hypothesis), and all the points between i and i_m (thus between i and j_{min}) satisfy ψ_1. Hence, $\rho, i, \nu \models \psi_1 \mathbf{U} \psi_2$ which is equivalent to (B-1). ◀

The following sections 4.1 and 4.2 show that the subclass CoT of UPTL satisfies the properties **P1** and **P2**, respectively.

4.1 Monadic Decomposition of CoT formulas

In this section, we show that CoT admits monadic decomposition, i.e. satisfies property **P1**. Let ψ be any TPTL formula. Let $c_j = (T - x_{i_j}) \sim u_{i_j}$, $1 \leq j \leq k$ be the open constraints appearing in ψ. Let $\psi{\downarrow_{c_j}}$ be the formula obtained from $\psi(x_{i_1}, \ldots, x_{i_k})$ by substituting "true" for all open constraints except c_j with $\top$. Hence, $\psi{\downarrow_{c_j}}$ for any $1 \leq j \leq k$ can have at most one open constraint, and thus one free variable x_{i_j}.

▶ **Lemma 4.5.** *Any formula ψ in* CoT *is equivalent to $\psi{\downarrow_{c_1}}(x_{i_1}) \wedge \ldots \wedge \psi{\downarrow_{c_k}}(x_{i_k})$.*

Proof. Let $\psi(x_1, \ldots, x_k)$ be a UPTL formula in CoT. We prove for $\leq \in$ Pol(φ), and the argument is symmetric for $\geq \in$ Pol(φ). The proof follows by structural induction on ψ.

Base Case. If ψ is a closed formula or a conjunction of atomic time constraints, then $\psi{\downarrow_{c_1}}(x_{i_1}) \wedge \ldots \wedge \psi{\downarrow_{c_k}}(x_{i_k})$ is the same as ψ. Hence, the lemma trivially holds.

Induction Hypothesis. Let the lemma hold for all the strict subformulas of ψ.

Induction. There are three possible cases.

- $\psi = \alpha \wedge \beta$. Notice that $\alpha{\downarrow_{c_j}} \wedge \beta{\downarrow_{c_j}} = \psi{\downarrow_{c_j}}(x_{i_j})$ for all $1 \leq j \leq k$. By induction hypothesis, $\alpha \equiv \alpha{\downarrow_{c_1}} \wedge \alpha{\downarrow_{c_2}} \wedge \ldots \wedge \alpha{\downarrow_{c_k}}$, and $\beta \equiv \beta{\downarrow_{c_1}} \wedge \beta{\downarrow_{c_2}} \wedge \ldots \wedge \beta{\downarrow_{c_k}}$. Hence, $\psi = \psi{\downarrow_{c_1}}(x_{i_1}) \wedge \ldots \wedge \psi{\downarrow_{c_k}}(x_{i_k})$.
- $\psi = \Box(\alpha)$ or $\psi = \overleftarrow{\Box}(\alpha)$. Then ψ is closed as it is in CoT, and the lemma trivially holds (implied by the base case).
- $\psi = \alpha\,\mathbf{U}\,\beta$. A symmetric argument holds for $\alpha\,\mathbf{S}\,\beta$. For an overall idea, assume that ψ has only two open constraints c_1, c_2 (this argument easily generalizes for more open constraints, see Appendix B). By induction hypothesis, $\psi \equiv \psi'$, where $\psi' = \alpha\,\mathbf{U}\,(\beta{\downarrow_{c_1}} \wedge \beta{\downarrow_{c_2}})$. As α is a closed formula, for any $j \in \{1, 2\}$, $\psi{\downarrow_{c_j}} = \alpha\,\mathbf{U}\,\beta{\downarrow_{c_j}}$. We need to show that $\psi' \leftrightarrow \psi{\downarrow_{c_1}} \wedge \psi{\downarrow_{c_2}}$ is a tautology. $\psi' \rightarrow \psi{\downarrow_{c_1}} \wedge \psi{\downarrow_{c_2}}$ is trivially implied by the "until" semantics.

  Let ρ be any timed word, $i \in \mathbb{N}$ and ν be any valuation over free variables of ψ. $\rho, i, \nu \models \psi{\downarrow_{c_1}} \wedge \psi{\downarrow_{c_2}}$, implies that for $j \in \{1, 2\}$, there exists point $i_j > i$ such that $\rho, i_j, \nu \models \beta{\downarrow_{c_j}}$ and all the points between i and i_j satisfies α. It suffices to show that for some $j \in \{1, 2\}$, $\rho, i_j, \nu \models \beta{\downarrow_{c_1}} \wedge \beta{\downarrow_{c_2}}$. Without loss of generality, assume $i_1 \leq i_2$. Then, by Lemma 4.4 and $\rho, i_2, \nu \models \beta{\downarrow_{c_2}}$, we have $\rho, i_1, \nu \models \beta{\downarrow_{c_2}}$, thus $\rho, i_1, \nu \models \beta{\downarrow_{c_1}} \wedge \beta{\downarrow_{c_2}}$. ◀

4.2 Reducing UPTL to finite disjunctions of CoT

Let $\bigvee$ CoT denote the class of UPTL formulas which are finite disjunctions of CoT formulas. In this section, we show that for any UPTL formula φ we can construct an equivalent formula ψ in $\bigvee$ CoT. Hence, CoT satisfies the property **P2**. This is the most technically involved part of the paper; the rest of this section is devoted to the proof of Lemma 4.6.

▶ **Lemma 4.6.** *Any* UPTL *formula $\varphi(x_1, \ldots, x_k)$ such that $\sim \in$* Pol($\varphi(x_1, \ldots, x_k)$) *can be expressed as a finite disjunction of* CoT *formulas ψ such that $\sim \in$* Pol(ψ)*.*

Fix a UPTL formula φ. Let $\bigvee \mathsf{CoT}^{\varphi}$ be the class of UPTL formulas such that Pol(φ) = Pol(ψ) for all $\psi \in \bigvee$ CoT. By structural induction on φ, we construct an equivalent Θ in $\bigvee \mathsf{CoT}^{\varphi}$.

Base Case. Assume φ is a UPTL formula which is either a closed formula or an atomic time constraint of the form $(T - x) \sim u$. The lemma trivially holds in this case.

Induction Hypothesis. Assume that the result holds for all the strict subformulas of φ.

Induction. We show that φ can be reduced to an equivalent formula in $\bigvee \mathsf{CoT}^\varphi$. There are three possible cases. (Case 0) : $\varphi = \alpha \wedge \beta$ or $\varphi = \alpha \vee \beta$, (Case 1) : $\varphi = \alpha \,\mathbf{U}\, \beta$ or $\alpha \,\mathbf{S}\, \beta$, (Case 2) : $\varphi = \Box\alpha$ or $\overleftarrow{\Box}\alpha$. We discuss the more interesting cases 1 and 2. For a full formal proof, see Appendix C. We observe a simplification for cases 1, 2. By induction hypothesis, α and β can be reduced to $\alpha' = \bigvee_m \alpha_m$ and $\beta' = \bigvee_l \beta_l$ in $\bigvee \mathsf{CoT}^\varphi$, respectively. Thus, without loss of generality, we consider cases 1 and 2 to be of the forms $\varphi = \alpha' \,\mathbf{U}\, \beta'$ (or $\alpha' \,\mathbf{S}\, \beta'$) and $\Box\alpha'$ (or $\overleftarrow{\Box}\alpha'$) respectively.

For case 1, we argue about $\mathbf{U}$ formulas (case of $\mathbf{S}$ is symmetric). Notice that $\phi \mathbf{U} (\psi \vee \psi') \equiv (\phi \,\mathbf{U}\, \psi) \vee (\phi \,\mathbf{U}\, \psi')$. Hence, φ in case 1 can be further simplified by pulling the top-level disjunctions of β'. Hence, $\varphi \equiv \bigvee_l (\alpha' \,\mathbf{U}\, \beta_l)$, where each β_l is in CoT having the same polarity as φ. As $\bigvee \mathsf{CoT}^\varphi$ is closed under disjunctions, it suffices to show that each $(\alpha' \,\mathbf{U}\, \beta_l)$ is in CoT. This is shown by Lemma 4.7. Moreover φ in case 2 can be reduced to φ in case 1 by Lemma 4.8. The following lemma handles the case 1.

▶ **Lemma 4.7.** *Any* UPTL *formula φ of the form $(\bigvee_{g=1}^{n} \alpha_g) \,\mathbf{U}\, (\beta)$ or $(\bigvee_{g=1}^{n} \alpha_g) \,\mathbf{S}\, (\beta)$, where $\alpha_i, \beta \in \bigvee \mathsf{CoT}^\varphi$, can be reduced to an equivalent $\bigvee \mathsf{CoT}^\varphi$ formula.*

Proof. We argue for $\mathbf{U}$, and $\mathbf{S}$ is symmetric. We induct on the number of open α_gs in φ. If there are no open α_gs in φ, then φ is, by definition, in $\bigvee \mathsf{CoT}^\varphi$.

Induction Hypothesis. Suppose the lemma holds for the case where k of the α_gs are open. We show the lemma holds for formulas having $(k+1)$ open α_gs. Without loss of generality, assume α_1 is open. $\rho, i, \nu \models \varphi$ is equivalent to (A) $\exists j > i$ s.t. $\rho, j, \nu \models \beta$, and $\forall i < j' < j$, $\rho, j', \nu \models \bigvee_{g=1}^{n} \alpha_g$.

Case 1.1. $\leq \in \mathsf{Pol}(\varphi)$:

- Notice that (A) is equivalent to (B1) or (B2), where
  - (B1) $\rho, i, \nu \models (\bigvee_{g=2}^{n} \alpha_g) \,\mathbf{U}\, \beta$ (this is when $\rho, i, \nu \models \neg\alpha_1 \,\mathbf{U}\, \beta$) or
  - (B2) $\exists i' {>} i$ such that (B2,1)$\rho, i', \nu {\models} \alpha_1$, (B2,2)$\forall i {<} i'' {<} i' {:} \rho, i'', \nu {\models} \bigvee_{g=1}^{n} \alpha_g$, (B2,3) $\exists j > i'$ such that $\rho, j, \nu \models \beta$, (B2,4) $\forall i' {<} j' {<} j {:} \rho, j', \nu \models \bigvee_{g=2}^{n} \alpha_g$. Intuitively, after the point i', α_1 need not hold for $\bigvee_{g=1}^{n} \alpha_g$ to hold at point i for a given valuation ν. The last occurrence of α_1 is a valid choice for i'. See Figure 4 in Appendix C.1 for intuition.
- Lemma 4.4 and (B2,1) imply that for all points i'' strictly between i and i', α_1 can be replaced by the closed formula α'_1 where $\alpha'_1 \in \mathsf{Untime}(\alpha_1)$ is obtained by replacing all open constraints of α_1 with $\top$.
- Hence, the given formula φ is equivalent to $\varphi_1 \vee \varphi_2$ where:
  - $\varphi_1 = [(\bigvee_{g=2}^{n} \alpha_g) \,\mathbf{U}\, \beta]$, $\qquad \varphi_2 = [(\bigvee_{g=2}^{n} \alpha_g \vee \alpha'_1) \,\mathbf{U}\, (\alpha_1 \wedge \varphi_1)]$
- By induction hypothesis, φ_1 can be reduced to a formula of the form $\bigvee_{h=1}^{m} (\Theta_{1,h})$ where each $\Theta_{1,h}$ is in $\bigvee \mathsf{CoT}^\varphi$ having the same polarity as φ. Substituting this for φ_1 in φ_2 and pushing all the disjunctions at the top, we get $\Theta_2 = \bigvee_{h=1}^{m} [(\bigvee_{g=2}^{n} \alpha_g \vee \alpha'_1) \,\mathbf{U}\, (\alpha_1 \wedge \Theta_{1,h})]$
- It suffices to show that each $\Theta_{2,h} = [(\bigvee_{g=2}^{n} \alpha_g \vee \alpha'_1) \,\mathbf{U}\, (\alpha_1 \wedge \Theta_{1,h})]$ can be reduced to an equivalent $\bigvee \mathsf{CoT}^\varphi$ as formulas in $\bigvee \mathsf{CoT}^\varphi$ are closed under disjunctions. Moreover, $\Theta_{2,h}$

has k open α_i formulas as one of the open formula α_1 is replaced with a closed formula α_1'. Hence, by induction hypothesis, this could be reduced to an equivalent formula in $\bigvee \mathsf{CoT}^{\varphi}$, proving the result.

Case 1.2. $\geq \in \mathsf{Pol}(\varphi)$: We discuss this case in Appendix C.1. ◀

Case 2 is handled by the following Lemma 4.8.

▶ **Lemma 4.8.** *Let $\varphi \in \mathsf{UPTL}$ be of the form $\Box(\bigvee_{g=1}^{n} \alpha_g)$ or $\overleftarrow{\Box}(\bigvee_{g=1}^{n} \alpha_g)$, where each $\alpha_g \in \mathsf{CoT}$. Then, we can construct a $\bigvee \mathsf{CoT}^{\varphi}$ formula equivalent to φ.*

Notice that $\overleftarrow{\Box}(\bigvee_{g=1}^{n} \alpha_g) \equiv (\bigvee_{g=1}^{n} \alpha_g)\, \mathbf{S}\, (start)$, where $start = \overleftarrow{\Box}\bot$, is true only at the first point of the timed word. Hence, now applying Lemma 4.7, we get the required result.

Let us now consider $\varphi = \Box(\bigvee_{g=1}^{n} \alpha_g)$. There are two possible subcases:

- **Case 2.1** $\leq \in \mathsf{Pol}(\varphi)$, and **Case 2.2** $\geq \in \mathsf{Pol}(\varphi)$. The latter can be reduced to a formula in Case 1.2, and the proof appears in the Appendix D.
- **Case 2.1** $\leq \in \mathsf{Pol}(\varphi)$. This is the most interesting of all the cases. The idea is to reduce this to Case 1.1 of Lemma 4.7 wherever possible. Let $\alpha' = \alpha_1 \vee \ldots \vee \alpha_n$, and $\varphi = \Box(\alpha')$. If we restrict to finite timed words then this problem is trivially reduced to Case 1.1 of Lemma 4.7 as $\Box(\alpha') \equiv \alpha'\, \mathbf{U}\, Last$, where $Last = \Box(\bot)$ is the formula which evaluates to true only at the last point. The infinite word case needs more machinery.

4.3 Machinery for Case 2.1, Lemma 4.8 for Infinite Words

We define some terminology which would be useful in the rest of the proof.

▶ **Definition 4.9** (Finitely Occurring Formulas). *Given $\psi(x_1, \ldots, x_k) \in \mathsf{TPTL}$, a timed word ρ, and a valuation for free variables of ψ, we say that ψ is finitely occurring in ρ, ν iff there are only finitely many points i of ρ such that $\rho, i, \nu \models \psi$. That is, there are only finitely many points in ρ where ψ is satisfied for a given valuation.*

▶ **Definition 4.10** (Maximal Finitely Occurring Open Subformula (MFOS)). *Given any TPTL formula $\varphi(x_1, \ldots, x_k)$, a timed word ρ, and a valuation ν for free variables of φ, we say that a subformula ψ of φ is a "finitely occurring open subformula" (FOS) of φ in ρ, ν iff*

1. *ψ is not a subformula of any closed subformula ψ' of φ. This implies ψ is open.*
2. *ψ is finitely occurring in ρ, ν.*

ψ is said to be a "maximal finitely occurring open subformula" (MFOS) of φ in ρ, ν iff ψ is a maximal formula that satisfies the above two conditions: that is, ψ is not a strict subformula of an open, finitely occurring formula in ρ, ν, and ψ is not a subformula of a closed formula.

▶ **Proposition 4.11.** *For any timed word ρ and a valuation ν, formulas of the form $(T - x) \leq u$ (or $(T - x) < u$) are finitely occurring formulas in ρ, ν.*

Proof. This is trivial, as otherwise this would imply that there are infinitely many points within at most u distance apart, implying the Zeno condition. ◀

▶ Remark 4.12. Any formula of φ that has a FOS also has a MFOS. Hence, by Proposition 4.11, any TPTL formula φ that contains an open constraint of the form $(T - x) \leq u$ or $(T - x) < u$ has an MFOS.

▶ **Proposition 4.13.** *Given* TPTL *formulas* φ, ψ, *a timed word* ρ *and a valuation* ν *over the free variables of* φ *and* ψ, *if* ψ *is finitely occurring in* ρ, ν, *then both* $\varphi \,\mathbf{U}\, \psi$, *and* $\varphi \wedge \psi$ *are finitely occurring.*

Proof. If ψ holds at finitely many points of ρ for the given valuation, then there exists a point j in ρ after which ψ ceases to hold for the given valuation ν. By semantics of $\mathbf{U}$ and conjunction, any point after j can not satisfy $\varphi \,\mathbf{U}\, \psi$ and $\varphi \wedge \psi$ for the given valuation ν. Hence, these are also finitely occurring in ρ, ν. ◀

▶ **Proposition 4.14.** *Given any* CoT *formulas* φ, ψ, *a timed word* ρ, *and a valuation* ν *over the free variables of* φ *and* ψ, *if* ψ *is MFOS of* φ *in* ρ, ν, *then either* $\varphi = \psi$ *or* ψ *appears in a subformula* $\beta \,\mathbf{S}\, \psi$ *of* φ. *In other words, either* ψ *is the same as* φ *or* ψ *appears as the right argument of a* $\mathbf{S}$ *modality. Moreover, this* $\beta \,\mathbf{S}\, \psi$ *is not finitely occurring.*

Proof. Suppose this is not the case. If ψ is in the scope of $\vee$, $\Box$, or $\overleftarrow{\Box}$ then φ is not CoT as ψ is an open subformula. Then ψ is in the scope of $\mathbf{U}$, $\mathbf{S}$ or $\wedge$: φ has a subformula of the form $\beta \,\mathbf{U}\, \psi$, $\beta \wedge \psi$ or $\beta \,\mathbf{S}\, \psi$. For the first and second cases, as ψ is finitely occurring in ρ, ν, by Proposition 4.13, both $\beta \,\mathbf{U}\, \psi$ and $\beta \wedge \psi$ are finitely occurring in ρ, ν. This implies ψ is not a MFOS of φ in ρ, ν, a contradiction. Thus, the only option is $\beta \,\mathbf{S}\, \psi$. If $\beta \,\mathbf{S}\, \psi$ is finitely occurring in ρ, ν, then ψ is not MFOS of φ in ρ, ν. Hence, $\beta \,\mathbf{S}\, \psi$ is not finitely occurring. ◀

▶ **Proposition 4.15.** *Given* TPTL *formulas* φ, ψ, *a timed word* ρ, *and a valuation* ν *over the free variables of* φ *and* ψ, *if* ψ *is finitely occurring for* ρ, ν *then for any point* i *of* ρ,

I. $\rho, i, \nu \models \Diamond\Box(\varphi \,\mathbf{S}\, \psi) \iff \rho, i, \nu \models \Box\Diamond(\varphi \,\mathbf{S}\, \psi)$, *and*

II. $\rho, i, \nu \models \Box(\varphi \,\mathbf{S}\, \psi) \iff \rho, i, \nu \models \chi$ *where* $\chi = \chi^+ \vee \chi^- \vee \chi^0$ *and*
$\chi^+ = (\varphi \,\mathbf{S}\, \psi) \,\mathbf{U}\, (\psi \wedge \Box(\varphi))$, $\quad \chi^- = (\varphi \,\mathbf{S}\, \psi) \,\mathbf{S}\, (\psi \wedge \Box(\varphi))$, $\quad \chi^0 = \psi \wedge \Box(\varphi)$.

Proof. Let i be any point of ρ. As ψ is finitely occurring, there exists a point j such that (a) $\rho, j, \nu \models \psi$, and $\rho, j', \nu \not\models \psi$ for any $j' > j$.

Proving I. The forward direction is trivially a tautology. For the reverse direction, suppose j satisfies $\rho, j, \nu \models \Box\Diamond(\varphi \,\mathbf{S}\, \psi)$. That is, there are infinitely many points after j satisfying $\varphi \mathbf{S} \psi$. Hence, for any point $i > j$, we can find a point $j'' > i > j$ such that (b) $\rho, j'', \nu \models \varphi \mathbf{S} \psi$. (see Figure 3 along with the proof for intuition).

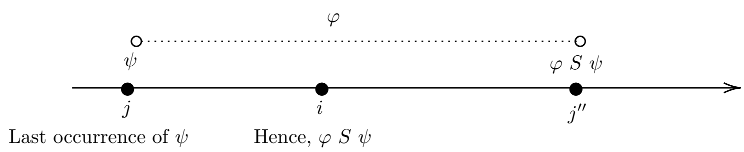

Figure 3 Figure depicting intuition behind Proposition 4.15 (I).

But, by (a), the closest point where ψ holds before j'' is j. Hence, (a) and (b) imply $\rho, i, \nu \models \varphi$. Hence, every point after i necessarily satisfies φ and hence $\varphi \,\mathbf{S}\, \psi$. Thus, every point after j also satisfies $\varphi \,\mathbf{S}\, \psi$. Hence, (c) : for any point i in ρ, there is a point k in the future of i, such that all the points $k' \geq k$ satisfy $\varphi \,\mathbf{S}\, \psi$ for the valuation ν. Notice that $k = i$ in case $i > j$ and $k = j + 1$ in case $i \leq j$. (c) shows the reverse direction of I.

Proving II. In case of II, the reverse direction is a tautology (by semantics). For the forward direction: let (Assumption 1) $\rho, i, \nu \models \Box(\varphi \,\mathbf{S}\, \psi)$ and (Assumption 2) ψ be finitely occurring in ρ, ν. By I, (Assumption 1) and (Assumption 2), we know that there exists a point i' such that $\rho, i', \nu \models \chi^0$, where $\chi^0 = \psi \wedge \Box(\varphi)$. One such point is j, the last occurrence of ψ (see (c) in the proof of I). If $i < i'$, then $\chi^+ = (\varphi \,\mathbf{S}\, \psi) \,\mathbf{U}\, (\psi \wedge \Box(\varphi))$ holds at i. For $i > i'$, we encode the fact that i comes after i', by asserting $\chi^- = \overleftarrow{\Diamond}((\psi \wedge \Box(\varphi))$ at i. For $i = j$, we encode that j is same as i : $\chi^0 = \psi \wedge \Box(\varphi)$. ◀

4.3.1 Case 2.1: Handling Infinite Words

Recall that $\leq \in \mathsf{Pol}(\varphi)$ and $\varphi = \Box\alpha'$ with $\alpha' = \alpha_1 \vee \cdots \vee \alpha_n$. We apply induction on the formula size of α'. If the size of α' is 1, the construction is trivial. We either get a closed formula of the form $\Box((T-x) \leq c)$ or $\Box((T-x) < c)$. Both these evaluate to $\bot$ (which is a closed formula) as they imply zenoness (see Proposition 4.11), hence getting a formula in CoT.

- If all α_i are closed then φ is closed, and hence trivially in CoT. Hence, we consider the case where at least one of the α_i is open. Wlg, assume α_1 is open. We now show that we can construct an equivalent formula for $\varphi = \Box(\alpha')$ in $\bigvee \mathsf{CoT}^{\varphi}$, assuming such a construction exists for all formulas smaller than φ.

  Let ρ be any timed word, i any point in ρ, and ν any valuation over free variables of φ. When we mention about the satisfaction of a TPTL formula at any point in ρ, we assume it is for the given valuation ν. There are three possible cases. We only discuss the case (called Case C below) when α_1 is not finitely occurring in ρ, ν, which mandates the use of MFOS. Appendix D.1 has the other two (called cases A and B) and are similar to the finite word case.
- Case C : α_1 *is not finitely occurring in* ρ, ν. Let i be any point of ρ. Let δ be any subformula of α_1 such that **[I]** $\rho, i, \nu \models \Diamond\Box(\delta)$. That is, $\exists j > i$ such that $\rho, j, \nu \models \Box(\delta)$. Hence, after point j we can replace the check of α_1 with α_{1_δ} where α_{1_δ} is obtained from α_1 by replacing δ to $\top$. Thus, $\rho, i, \nu \models \varphi$ iff $\rho, i, \nu \models \varphi_{3,\delta}$ where
  $\varphi_{3,\delta} = \alpha' \, \mathbf{U} \, [\alpha' \wedge \Box(\delta) \wedge \varphi_\delta]$, and $\varphi_\delta = \Box(\alpha_{1_\delta} \vee \alpha_2 \vee \ldots \vee \alpha_n)$.
  Moreover, by TPTL/LTL semantics, we get **[II]** if $\rho, i, \nu \not\models \Diamond\Box(\delta)$, then $\varphi_{3,\delta}$ is equivalent to false. That is, $\rho, i, \nu \not\models \varphi_{3,\delta}$. This is because $\varphi_{3,\delta}$ implies $\rho, i, \nu \models \Diamond\Box(\delta)$. Let $SFS(\alpha_1)$ be all the open subformulas of α_1 with $\mathbf{S}$ being the top level modality, i.e. formulas of the form $\delta = \beta \, \mathbf{S} \, \gamma$. We now show that : $\varphi \equiv \bigvee_{\delta \in SFS(\alpha_1)} \varphi_{3,\delta}$.
  For that, it suffices to show **[III]** Case C implies there is at least one $\delta \in SFS(\alpha_1)$ such that $\rho, i, \nu \models \Diamond(\Box(\delta))$ for any $i \in dom(\rho)$. **[II]** and **[III]** imply for any $i \in dom(\rho)$ there is at least one disjunct equivalent to φ, and every disjunct is either equivalent to $\bot$ or φ. We now prove **[III]**. As α_1 is not a closed formula, it has at least one MFOS with respect to ρ, ν (Remark 4.12). By Proposition 4.14, any MFOS of α_1 with respect to ρ, ν should occur within the scope of temporal operator $\mathbf{S}$; hence, α_1 has a subformula of the form $\delta = \beta \, \mathbf{S} \, \gamma$ where γ is an MFOS with respect to ρ, ν. In other words, there is at least one formula $\beta \, \mathbf{S} \, \gamma \in SFS(\alpha_1)$. Moreover, as γ is "maximal" finitely occurring subformulas, $\delta = \beta \mathbf{S} \gamma$ is not finitely occurring. This along with Proposition 4.15 imply $\rho, i, \nu \models \Diamond\Box(\delta)$, hence proving **[III]**. Thus $\varphi \equiv \bigvee_{\delta \in SFS(\alpha_1)} \varphi_{3,\delta}$. As $\bigvee \mathsf{CoT}^{\varphi}$is closed under disjunctions among the same polarity formulas, we just need to show that $\varphi_{3,\delta}$ for any δ is in $\bigvee \mathsf{CoT}^{\varphi}$, which is what we show now. Since φ' is strictly smaller than φ, by induction hypothesis, there is a formula ψ'_δ in $\bigvee \mathsf{CoT}^{\varphi}$equivalent to φ'. Moreover, by Proposition 4.15(II), there is $\chi = \chi^+ \vee \chi^- \vee \chi^0$ equivalent to $\Box(\beta \, \mathbf{S} \, \gamma)$. χ^0 is already in CoT with the same polarity as φ. χ^+ and χ^- are $\mathbf{U}$ and $\mathbf{S}$ formulas not necessarily in $\bigvee \mathsf{CoT}^{\varphi}$. But, it reduces to the **Case 1.1**. Thus, using the construction in **Case 1.1**, we can construct an equivalent formula χ' in $\bigvee \mathsf{CoT}^{\varphi}$ which is equivalent to χ. Hence, we get the following formula equivalent to $\varphi_{3,\delta}$, namely, $\varphi'_{3,\delta} = \alpha' \, \mathbf{U} \, [\alpha' \wedge \chi' \wedge \psi'_\delta]$. This again falls under **Case 1.1** of Lemma 4.7. Hence, we can construct a ψ_δ, for a given δ which is in $\bigvee \mathsf{CoT}^{\varphi}$.
- The formula we are looking for is obtained by disjuncting the formulas constructed in all three cases A, B, C (of which we have Case C here, and Cases A,B in Appendix D.1).

Lemmas 4.5 and 4.6 imply Lemma 3.6. For any UPTL formula $\psi(x_1,\ldots,x_k)$, construct an equivalent $\psi'(x_1,\ldots,x_k) = \bigvee_{i=1}^{m} \psi_i(x_1,\ldots,x_k)$, where each ψ_i is in CoT with the same polarity as that of ψ. Then apply Lemma 4.5 and reduce each $\psi_i(x_1,\ldots,x_k)$ to $\bigwedge_{j=1}^{n} \psi_{i_j}(x_{i,j})$, where each $x_{i,j}$ is some variable in $\{x_1,\ldots,x_k\}$, and ψ_{i_j} contains at most one open time constraint. Then ψ is equivalent to $\varphi = \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} \psi_{i_j}(x_i)$, obtaining the result. ◀

▶ Remark 4.16. Each ψ_{i_j} is a CoT containing at most one open constraint. Thus, our construction gives a stronger result than Lemma 3.6, which is used in proving Theorem 3.5.

▶ **Lemma 4.17** (Monadic Decomposition of UPTL (Stronger Form)). *There is an effective translation from any* UPTL $\varphi(x_1,\ldots,x_k)$ *to an equivalent* UPTL $\psi = (\Psi_1) \vee (\Psi_2) \vee \ldots \vee (\Psi_n)$ *where each* Ψ_i *is a conjunction of* CoT *formulas containing at most one open time constraint.*

5 1-UPTL to Q2MLO: Proof of Theorem 3.5

By Theorem 3.3, Q2MLO is equivalent to UMITPPL. Moreover, by Remark 2.1, 1-UPTL can efficiently express formulas in UMITPPL. Hence, to prove Theorem 3.5, it suffices to show that we can convert a closed 1-UPTL formula φ to an equivalent Q2MLO formula $\psi(y)$. We give a proof sketch along with a running example.

1. Let $\varphi{=}x.\varphi'(x)$ be a closed 1-UPTL formula. Using Lemma 4.17 on φ', and distributing $(x.)$ over disjunctions, we get an equivalent 1-UPTL formula $\varphi_{DTF} = [(\varphi_1) \vee (\varphi_2) \vee \ldots \vee (\varphi_k)]$, where each φ_i has the form $x.\varphi_{1,i} \wedge \ldots \wedge x.\varphi_{k,i}$ and each $\varphi_{j,i}$ is a formula in CoT containing at most one open time constraint. As Q2MLO is closed under $\vee$s and $\wedge$s, it suffices to show that each $x.\varphi_{j,i}$ is reducible to a Q2MLO formula $\psi'(y)$.

   Let $\varphi'(x) = a\,\mathbf{U}\,[\{((T{-}x) < u_1) \wedge (a\,\mathbf{S}\,(b \wedge ((T{-}x) < u_2)))\} \vee \{(T{-}x){<}u_3\}]$. The above reduction on φ' gives $\varphi_{DTF} = [\varphi_{1,1} \wedge \varphi_{1,2}] \vee [\varphi_{2,1}]$ where, $\varphi_{2,1} = a\,\mathbf{U}\,\{(T{-}x){<}u_3\}$, $\varphi_{1,1} = a\,\mathbf{U}\,\{((T{-}x) < u_1) \wedge (a\,\mathbf{S}\,b))\}$, $\varphi_{1,2} = a\,\mathbf{U}\,\{(a\,\mathbf{S}\,(b \wedge ((T{-}x) < u_2)))\}$.
2. Encode the semantics of $\varphi_{j,i}$ in first order logic. Notice that the single open constraint in a CoT formula $\varphi_{j,i}$ never appears on a left subtree of any $\mathbf{U}$ or $\mathbf{S}$ modality in the parse tree of $\varphi_{j,i}$. Thanks to this syntactic restriction of CoT, the variable guarded by metric predicates always appears within the scope of $\exists^*$ in the corresponding first-order formula. After encoding the semantics in first-order logic and pulling up the "metric guarded quantifiers", i.e. the quantifier $\exists y$ quantifying a variable appearing in the open timing constraint $y \ominus x \sim c$, to the top gives us an equivalent Q2MLO formula.

   To illustrate, we show the reduction on $x.\varphi_{1,2}$, where the open time constraint appears at modal depth two. Similar reductions can be applied for $x.\varphi_{1,1}$ and $x.\varphi_{2,1}$. By TPTL semantics, $\rho, j \models x.\varphi_{1,2}$ iff $\rho, \nu \models \psi_{1,2}(x)$ where $\psi_{1,2}(x)$ is the required Q2MLO defined by $\nu(x) = j$ and
   $\psi_{1,2}(x){=}\exists z.[(z \ominus x < u_2) \wedge \varphi(x,z)]$ where $\varphi(x,z)$ is defined as follows.
   $\varphi(x,z) = \exists y.[\{z{<}y \wedge x < y \wedge b(z)\} \wedge \{(\forall y'.(x{<}y'{<}y{\to}a(y')))\wedge\{\forall z'(z{<}z'{<}y{\to}a(z'))\}]$.

6 Applications

Theoretical Applications. We first discuss the theoretical implications of our results. First, our results resolve the open questions of [24] : (1) Does the expressiveness of UPTL[Fut] increase with more variables? We resolve this question negatively. Notice that the variable reduction in the proof of Theorem 3.4 also works for the future fragment of UPTL. Thus,

▶ **Corollary 6.1.** UPTL[Fut] $\equiv$ 1-UPTL[Fut]

(2) Is full UPTL decidable? We resolve this question positively. Theorem 3.1 and the decidability of the satisfiability checking problem of Q2MLO [11] implies:

▶ **Corollary 6.2.** *Satisfiability checking for* UPTL *is decidable.*

Practical Applications: Compositional Verification of UPTL. A fundamental challenge in verifying freeze logics like UPTL is the lack of compositional techniques due to the dependency of subformulas on multiple free variable valuations. Our reduction addresses this by transforming UPTL formulas into equivalent MITPPL formulas, enabling compositional verification via model checkers such as MightyL [7], TChecker [10] (both accept MITL[Fut] formulas) and MightyPPL [27] (accepts MITPPL formulas). Our reduction allows us to leverage small, modular timed automata representations, significantly improving scalability. Consider a food delivery service specification as follows. The food delivery should be quick (within 30 mins since the order) and fresh (within 15 mins since food is cooked/pick-up). The UPTL formula $\psi = \Box(\text{order} \Rightarrow \phi)$ over propositions $AP = \{\text{order, pickup, deliver}\}$ where $\phi = x.\Diamond(\text{pickup} \wedge y.\Diamond(\text{deliver} \wedge x < 30 \wedge y < 15))$ captures this. Our technique translates ϕ to the equivalent MITPPL formula $\phi' = \mathbf{Pn}_{[0,30]}(\text{pickup,deliver}) \wedge \Diamond(\text{pickup} \wedge \Diamond_{[0,15]}(\text{deliver}))$. MightyPPL took 10ms to construct the timed automaton corresponding to ϕ' and 1ms to verify that it is satisfiable.

7 Conclusion and Discussion

One of our key contributions is a *first equivalence result* between a *freeze logic* and a *metric logic* under *pointwise semantics*. Our results extend naturally to the *continuous semantics*. This equivalence serves as an analogue to known results in continuous semantics, such as the equivalence between MTL *with counting and* TPTL [18], and between MTL[$\mathbb{Q}$] *and* TPTL[$\mathbb{Q}$] [19]. These prior equivalences break down in *pointwise semantics*. An interesting consequence of our results is that adding variables does not increase the expressiveness of UPTL, while introducing past operators does (See Appendix E).

(1) Precise Complexities, Translation Sizes. We conjecture that the satisfiability and model checking of UPTL to be in EXPSPACE, based on the observation that many subformulas in the reduced 1-UPTL formula are repeated. Hence, one could hope that the upper bound on the *formula DAG* of the equivalent 1-UPTL could be only *exponential* in the size of the input UPTL formula. Additionally, if the given UPTL is already in the $\bigvee$CoT form, then the UPTL to 1-UPTL reduction is worst case quadratic (Lemma 4.5). Translating 1-UPTL to UMITPPL can get non-elementary (as Q2MLO generalizes FO[<] over words, and has a non-elementary satisfiability). While our focus is expressive equivalence, for efficient satisfiability checking, 1-UPTL can be reduced to equisatisfiable UMITPPL with only exponential blowup using ideas from [21]. Translating UPTL to $\bigvee$ CoT causes a non-elementary blowup on formula trees, but using DAGs and a Tseitin-like encoding, we conjecture an exponential equisatisfiability reduction. Our translations, along with tools like MightyPPL [27] can verify UPTL specifications.

Succinctness of UPTL vs. 1-UPTL. While we establish that increasing the number of variables does not improve *expressiveness*, we believe it may lead to an exponential increase in *succinctness*. We leave a formal proof of UPTL (or UPTL[Fut]) being more succinct than 1-UPTL (resp. 1-UPTL[Fut]) as future work.

References

1 Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. In Luigi Logrippo, editor, *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, August 19-21, 1991*, pages 139–152. ACM, 1991. doi:10.1145/112600.112613.

2 Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *J.ACM*, 43(1):116–146, 1996. doi:10.1145/227595.227602.

3 Rajeev Alur and Thomas A. Henzinger. A really temporal logic. In *30th Annual Symposium on Foundations of Computer Science*, pages 164–169, 1989. doi:10.1109/SFCS.1989.63473.

4 Rajeev Alur and Thomas A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993. doi:10.1006/inco.1993.1025.

5 Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the expressiveness of TPTL and MTL. In Sundar Sarukkai and Sandeep Sen, editors, *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science*, pages 432–443, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11590156_35.

6 Thomas Brihaye, Morgane Estiévenart, and Gilles Geeraerts. On mitl and alternating timed automata over infinite words. In *Formal Modeling and Analysis of Timed Systems: 12th International Conference, FORMATS 2014, Florence, Italy, September 8-10, 2014. Proceedings 12*, pages 69–84. Springer, 2014. doi:10.1007/978-3-319-10512-3_6.

7 Thomas Brihaye, Gilles Geeraerts, Hsi-Ming Ho, and Benjamin Monmege. Mightyl: A compositional translation from MITL to timed automata. In Rupak Majumdar and Viktor Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*, pages 421–440. Springer, 2017. doi:10.1007/978-3-319-63387-9_21.

8 Thomas Ferrère. The compound interest in relaxing punctuality. In Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik P. de Vink, editors, *Formal Methods - 22nd International Symposium, FM 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 15-17, 2018, Proceedings*, volume 10951 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2018. doi:10.1007/978-3-319-95582-7_9.

9 Thomas A. Henzinger, Jean-François Raskin, and Pierre-Yves Schobbens. The regular real-time languages. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 580–591. Springer, 1998. doi:10.1007/BFb0055086.

10 Frédéric Herbreteau and Gérald Point. The tchecker tool and libraries. https://github.com/ticktac-project/tchecker.

11 Yoram Hirshfeld and Alexander Moshe Rabinovich. A framework for decidable metrical logics. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 422–432. Springer, 1999. doi:10.1007/3-540-48523-6_39.

12 Yoram Hirshfeld and Alexander Moshe Rabinovich. Logics for real time: Decidability and complexity. *Fundamenta Informaticae*, 62(1):1–28, 2004. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi62-1-02.

13 Yoram Hirshfeld and Alexander Moshe Rabinovich. An expressive temporal logic for real time. In *MFCS*, pages 492–504, 2006. doi:10.1007/11821069_43.

14 Yoram Hirshfeld and Alexander Moshe Rabinovich. Expressiveness of metric modalities for continuous time. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science – Theory and Applications*, pages 211–220, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11753728_23.

15 Hsi-Ming Ho. Revisiting timed logics with automata modalities. In Necmiye Ozay and Pavithra Prabhakar, editors, *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2019, Montreal, QC, Canada, April 16-18, 2019*, pages 67–76. ACM, 2019. doi:10.1145/3302504.3311818.

16 Hsi-Ming Ho and Khushraj Madnani. When do you start counting? revisiting counting and pnueli modalities in timed logics. In *DCM*. EPTCS, 2023.

17 Hsi-Ming Ho and Khushraj Madnani. Metric quantifiers and counting in timed logics and automata. *Information and Computation*, 303:105268, 2025. doi:10.1016/j.ic.2025.105268.

18 Paul Hunter. When is metric temporal logic expressively complete? In *22nd EACSL Conference on Computer Science Logic, CSL*, pages 380–394, 2013. doi:10.4230/LIPICS.CSL.2013.380.

19 Paul Hunter, Joël Ouaknine, and James Worrell. Expressive completeness for metric temporal logic. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 349–357. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.41.

20 Ron Koymans. Specifying real-time properties with metric temporal logic. *Real Time Syst.*, 2(4):255–299, 1990. doi:10.1007/BF01995674.

21 Shankara Narayanan Krishna, Khushraj Madnani, Manuel Mazo Jr., and Paritosh K. Pandya. Generalizing non-punctuality for timed temporal logic with freeze quantifiers. In Marieke Huisman, Corina S. Pasareanu, and Naijun Zhan, editors, *Formal Methods - 24th International Symposium, FM 2021, Virtual Event, November 20-26, 2021, Proceedings*, volume 13047 of *Lecture Notes in Computer Science*, pages 182–199. Springer, 2021. doi:10.1007/978-3-030-90870-6_10.

22 Shankara Narayanan Krishna, Khushraj Madnani, and Paritosh K. Pandya. Metric temporal logic with counting. In *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, pages 335–352, 2016. doi:10.1007/978-3-662-49630-5_20.

23 Shankara Narayanan Krishna, Khushraj Madnani, and Paritosh K. Pandya. Making metric temporal logic rational. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, volume 83 of *LIPIcs*, pages 77:1–77:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.MFCS.2017.77.

24 Shankara Narayanan Krishna, Khushraj Nanik Madnani, Rupak Majumdar, and Paritosh K. Pandya. Satisfiability checking of multi-variable TPTL with unilateral intervals is pspace-complete. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory, CONCUR 2023, September 18-23, 2023, Antwerp, Belgium*, volume 279 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CONCUR.2023.23.

25 Slawomir Lasota and Igor Walukiewicz. Alternating timed automata. *ACM Trans. Comput. Log.*, 9(2):10:1–10:27, 2008. doi:10.1145/1342991.1342994.

26 Nicolas Markey. Temporal logic with past is exponentially more succinct. *Bulletin-European Association for Theoretical Computer Science*, 79:122–128, 2003.

27 MightyPPL. https://github.com/hsimho/MightyPPL.

28 Joël Ouaknine and James Worrell. On the decidability of metric temporal logic. In *LICS*, pages 188–197, 2005. doi:10.1109/LICS.2005.33.

29 Paritosh K. Pandya and Simoni S. Shah. On expressive powers of timed logics: Comparing boundedness, non-punctuality, and deterministic freezing. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 60–75. Springer, 2011. doi:10.1007/978-3-642-23217-6_5.

30 Jean Francois Raskin. *Logics, Automata and Classical Theories for Deciding Real Time*. PhD thesis, Universite de Namur, 1999.
31 Howard Straubing. *Finite Automata, Formal Logic and Circuit Complexity*. Birkhauser, 1994.
32 Thomas Wilke. Specifying timed state sequences in powerful decidable logics and timed automata. In *Formal Techniques in Real-Time and Fault-Tolerant Systems, Third International Symposium Organized Jointly with the Working Group Provably Correct Systems - ProCoS, Lübeck, Germany, September 19-23, Proceedings*, pages 694–715, 1994. doi:10.1007/3-540-58468-4_191.

A Proof of Lemma 4.4

Proof. Let ψ be a formula in CoT. We prove for the case $\leq \in \mathsf{Pol}(\psi)$; a symmetric argument follows when $\geq \in \mathsf{Pol}(\psi)$. Fix a timed word $\rho = (a_1, \tau_1)(a_2, \tau_2)\ldots$, a point j of ρ and a valuation ν over the variables appearing in ψ. Let $\psi' \in \mathsf{Untime}(\psi)$. Assuming that (A) $\rho, j, \nu \models \psi$, we now show that (B) for any point $i \leq j$, $\rho, i, \nu \models \psi$ iff $\rho, i, \nu \models \psi'$. We apply induction on the structure of the formula ψ. From now, we assume that (A) holds.

Base Case. Let ψ be either a closed formula or an atomic clock constraint of the form $T - x \sim u$ where $\sim \in \{<, \leq\}$. If ψ is a closed formula then $\mathsf{Untime}(\psi) = \{\psi\}$. Hence, the lemma holds trivially. If $\psi = (T - x) \sim u$, then $\mathsf{Untime}(\psi) = \{\psi, \top\}$. It suffices to show that (I) $\rho, i, \nu \models \top$ iff (II) $\rho, i, \nu \models (T - x) \sim u$ for all $i \leq j$. (II) implies (I) is a tautology. Conversely, for any $i \leq j$, $\tau_i \leq \tau_j$. As (A) holds, we know $\tau_j - \nu(x) \leq u$; hence for any $i \leq j$, $\tau_i - \nu(x) \leq u$, giving $\rho, i, \nu \models (T - x) \leq u$.

Induction Hypothesis. Assume the lemma holds for all subformulas of ψ.

Induction. We now show that (B) holds given the induction hypothesis and (A). There are three possible cases.

1. $\psi = \psi_1 \wedge \psi_2$. By definition of Untime, $\psi' = \psi_1' \wedge \psi_2'$ such that $\psi_1' \in \mathsf{Untime}(\psi_1)$ and $\psi_2' \in \mathsf{Untime}(\psi_2)$. Now, (A) iff (A1.1) $\rho, j, \nu \models \psi_1$ and (A1.2) $\rho, j, \nu \models \psi_2$. As ψ is a formula in CoT, both ψ_1 and ψ_2 are in CoT. Hence, by induction hypothesis, (A1.1) and (A1.2) implies that for all $i \leq j$, (B1.1) $\rho, i, \nu \models \psi_1$ iff $\rho, i, \nu \models \psi_1'$ and (B1.2) $\rho, i, \nu \models \psi_2$ iff $\rho, i, \nu \models \psi_2'$, respectively. Thus, (A) implies (B). Notice that this argument may not hold when $\psi = \psi_1 \vee \psi_2$. But in this case, ψ is not a CoT formula as disjunction in CoT are allowed only amongst closed formulas.
2. $\psi = \psi_1 \mathbf{U} \psi_2$. As ψ is in CoT, ψ_1 is a closed formula. Hence, $\mathsf{Untime}(\psi_1) = \{\psi_1\}$, and $\psi' = \psi_1 \mathbf{U} \psi_2'$ where $\psi_2' \in \mathsf{Untime}(\psi_2)$. We show that assuming (A), for any $i \leq j$, (B2.1) $\rho, i, \nu \models \psi$ iff (B2.2) $\rho, i, \nu \models \psi'$. By semantics of $\mathbf{U}$,
   A holds iff there exists a point j' such that (I) $j' > j$, $\rho, j', \nu \models \psi_2$, and for any point j'' between j and j', $\rho, j'', \nu \models \psi_1$.
   B2.1 holds iff there exists a point i' such that (II) $i' > i$, $\rho, i', \nu \models \psi_2$, and for any point i'' between i and i', $\rho, i'', \nu \models \psi_1$.
   B2.2 holds iff there exists a point i' such that (III) $i' > i$, $\rho, i', \nu \models \psi_2'$ and for any point i'' between i and i', $\rho, i'', \nu \models \psi_1$.
   Let $i \leq j$ be any arbitrary point. Let $j_{min} > j$ be the point closest to j satisfying (I). Similarly, let i_m be any point satisfying (II). Either $i_m \leq j_{min}$ or $i_m > j_{min}$. In the former case, $\rho, i', \nu \models \psi_2$ iff $\rho, i', \nu \models \psi_2'$, by induction hypothesis. Hence, in the former case, (B2.1) iff (B2.2). Now, suppose $i_m > j_{min}$. Then notice that (as $i \leq j$, and j_{min} satisfies II), $j_{min} > i$, (IV-a) $\rho, j_{min}, \nu \models \psi_2$. Moreover, as i_m satisfies (II), all the points between i and i_m of ρ satisfies ψ_1 for the given valuation ν. As $j_{min} < i_m$, all the points

between i and j_{min} also lies between i and i_m. Hence, (V) for all points j'' between i and j_{min} $\rho, j'', \nu \models \psi_1$. By induction hypothesis, (IV-a) iff (IV-b), where (IV(b)) states $\rho, j_{min}, \nu \models \psi_2'$. Hence, (IV-a) and (V) iff (IV-b) and (V). Notice that (IV-b) and (V) is equivalent to j_{min} satisfying (III). Moreover, (IV-a) and (V) is equivalent to j_{min} satisfying (II). Hence, if $j_{min} < i_m$, then (B2.1) iff B2.2). Hence, (A) implies (B2.1) iff (B2.2), in both cases. Thus, (A) implies that for all $i \leq j$ (B) holds.

3. $\psi = \psi_1 \, \mathbf{S} \, \psi_2$. This case is symmetrical to the previous case (i.e., $\psi = \psi_1 \, \mathbf{U} \, \psi_2$). ◀

B Proof of Lemma 4.5

Proof. Let $\psi(x_1, \ldots, x_k)$ be a UPTL formula in CoT. We prove the lemma for $\leq \in \mathsf{Pol}(\varphi)$, and the argument is symmetric for $\geq \in \mathsf{Pol}(\varphi)$. The proof follows by structural induction on ψ.

Base Case. If ψ is a closed formula or a conjunction of atomic time constraints, then $\psi_1(x_{i_1}) \wedge \ldots \wedge \psi_k(x_{i_k})$ is the same as ψ. Hence, the lemma trivially holds.

Induction Hypothesis. Let the lemma hold for all the subformulas of ψ.

Induction. There are three possible cases.

- $\psi = \alpha \wedge \beta$- Notice that $\alpha{\downarrow_{c_j}} \wedge \beta{\downarrow_{c_j}} = \psi{\downarrow_{c_j}}(x_{i_j})$ for all $1 \leq j \leq k$. By induction hypothesis, $\alpha \equiv \alpha_1 \wedge \alpha_2 \ldots \alpha_k$, and $\beta \equiv \beta_1 \wedge \beta_2 \wedge \ldots \wedge \beta_k$. Hence, $\psi = \psi{\downarrow_{c_1}}(x_{i_1}) \wedge \ldots \wedge \psi{\downarrow_{c_k}}(x_{i_k})$.
- $\psi = \Box(\alpha)$ or $\psi = \overleftarrow{\Box}(\alpha)$. In this case, ψ is a closed formula as it is in CoT. Hence, the lemma trivially holds.
- $\psi = \alpha \, \mathbf{U} \, \beta$. Symmetric argument holds for $\psi = \alpha \, \mathbf{S} \, \beta$. Notice that as ψ is in CoT, α is a closed formula and β is in CoT. Hence, For any $1 \leq j \leq k$, $\psi{\downarrow_{c_j}} = \alpha \, \mathbf{U} \, \beta{\downarrow_{c_j}}$. By induction hypothesis $\beta = \beta{\downarrow_{c_1}} \wedge \ldots \wedge \beta{\downarrow_{c_k}}$. Hence, it suffices to show that for any timed word ρ, point g of ρ, and any valuation ν over variables of ψ, (A) $\rho, g, \nu \models \alpha \, \mathbf{U} \, (\beta{\downarrow_{c_1}} \wedge \ldots \wedge \beta{\downarrow_{c_k}})$ iff (B-1) $\wedge$ (B-2) $\wedge \ldots \wedge$ (B-k), where for all $1 \leq j \leq k$, (B-j) $\rho, g, \nu \models \alpha \, \mathbf{U} \, \beta{\downarrow_{c_j}}$. Notice that by semantics of $\mathbf{U}$, $\phi_1 \, \mathbf{U} \, (\phi_2 \wedge \phi_3)$ implies $\phi_1 \, \mathbf{U} \, (\phi_3)$ and $\phi_1 \, \mathbf{U} \, (\phi_2)$. Hence, (A) implies (B-1) $\wedge$ (B-2) $\wedge \ldots \wedge$ (B-k) holds trivially. The other direction is more interesting and relies on the untiming property of CoT. Hence, assume that (B-j) holds for all $1 \leq j \leq k$ (Assumption B). Thus for all $1 \leq j \leq k$, there exists a point g' such that (I-j) $g' > g$, $\rho, g', \nu \models \beta{\downarrow_{c_j}}$, and for all g'' between g and g', $\rho, g'', \nu \models \alpha$. For all $1 \leq j \leq k$, let g_j be any point in ρ satisfying (I-j). Let g_{min} be the point minimum among $g_1, \ldots, g_k$. Let $\beta_u = \mathsf{Untime}(\beta)$. Hence, β_u is a closed formula. Notice that for any $1 \leq j \leq k$, $\beta{\downarrow_{c_j}} \in \mathsf{Untime}(\beta)$ and $\beta_u \in \mathsf{Untime}(\beta{\downarrow_{c_j}})$. By Lemma 4.4 and Assumption B, $\rho, g_{min}, \nu \models \beta_u$. Again by By Lemma 4.4, Assumption B and $\rho, g_{min}, \nu \models \beta_u$, $\rho, g_{min}, \nu \models \beta{\downarrow_{c_j}}$ for all $1 \leq j \leq k$. Hence, there exists a point $g_{min} > g$ such that $\rho, g_{min}, \nu \models \beta{\downarrow_{c_1}} \wedge \beta{\downarrow_{c_2}} \ldots \wedge \beta{\downarrow_{c_k}}$, and for all points g'' between g and g_{min}, $\rho, g'', \nu \models \alpha$. Hence, (A) holds. ◀

C Proof of Lemma 4.6

This section is dedicated to the proof of the above lemma 4.6. We fix a UPTL formula φ and let $\bigvee \mathsf{CoT}^{\varphi}$denote the set of UPTL formulas such that $\mathsf{Pol}(\varphi) = \mathsf{Pol}(\psi)$ for all $\psi \in \bigvee \mathsf{CoT}^{\varphi}$. We now show how to construct a formula Θ in $\bigvee \mathsf{CoT}^{\varphi}$. Let ϕ be any UPTL formula. We apply induction on the structure of ϕ.

For the base case, assume ϕ is a UPTL formula which is either a closed formula or an atomic timing constraint of the form $x \sim u$. Then $\phi \in \mathcal{B}(F)$ where F is a finite set consisting of UPTL formulas whose subformulas are (i) either closed, or (ii) are open constraints of the form $x_i \sim u_i$ such that all of them have the same polarity. Hence, the lemma trivially holds.

Induction Hypothesis. Assume that the result holds for all subformulas of φ.

- **(Case 0).** $\varphi = \alpha \wedge \beta$ or $\varphi = \alpha \vee \beta$. Notice that α and β has the same polarity as φ. Hence, by induction hypothesis, α and β can be reduced to an equivalent $\bigvee_i \alpha_i$ and $\bigvee_j \beta_j$, respectively, where each α_i and β_j are in $\bigvee \mathsf{CoT}^\varphi$. Hence, if $\varphi = \alpha \vee \beta$, then the required formula in $\bigvee \mathsf{CoT}^\varphi$ is $\bigvee_i \alpha_i \vee \bigvee_j \beta_j$. Similarly, if $\varphi = \alpha \wedge \beta$, then $\varphi = \bigvee_i \bigvee_j (\alpha_i \wedge \beta_j)$ is the required formula in CoT, as each α_i and β_j are in CoT having the same polarity as φ.
- **(Case 1).** $\varphi = \alpha \,\mathbf{U}\, \beta$ or $\alpha \,\mathbf{S}\, \beta$.
- **(Case 2).** $\varphi = \Box\alpha$ or $\overleftarrow{\Box}\alpha$.

We observe a simplification which can be applied to cases 1, 2 using the induction hypothesis. By induction hypothesis, α and β can be reduced to $\alpha' = \bigvee_m \alpha_m$ and $\beta' = \bigvee_l \beta_l$ in $\bigvee \mathsf{CoT}^\varphi$, respectively. Thus, without loss of generality, we consider cases 1 and 2 to be of the forms $\varphi = \alpha' \,\mathbf{U}\, \beta'$ $(\alpha' \,\mathbf{S}\, \beta')$ and $\Box\alpha'$ $(\overleftarrow{\Box}\alpha')$ respectively.

For case 1, we just argue about $\mathbf{U}$ formulas. The argument for $\mathbf{S}$ is symmetrical. Notice that $\phi \,\mathbf{U}\, (\psi \vee \psi') \equiv (\phi \,\mathbf{U}\, \psi) \vee (\phi \,\mathbf{U}\, \psi)'$. Hence, φ in case 1 can be further simplified by pulling the top-level disjunctions of β'. Hence, $\varphi \equiv \bigvee_l (\alpha' \,\mathbf{U}\, \beta_l)$, where each β_l is in CoT having the same polarity as φ. As $\bigvee \mathsf{CoT}^\varphi$ is closed under disjunctions, it suffices to show that each $(\alpha' \,\mathbf{U}\, \beta_l)$ is in CoT. This is shown by proof of Lemma 4.7. Moreover, φ in case 2 can be reduced to φ in case 1 by the construction in the proof of Lemma 4.8.

C.1 Proof of Lemma 4.7: Case 1.2

This section is dedicated to the proof of 4.7.

Proof. We will argue about $\mathbf{U}$. The case for $\mathbf{S}$ is symmetrical. We apply induction on the number of α_is in φ that are open.

Base case. If there are no α_is open in φ then φ is, by definition, in $\bigvee \mathsf{CoT}^\varphi$.

Induction hypothesis. Suppose the lemma holds for the case where k of the α_i's are open formulas.

Induction. We show the lemma holds for formulas having $k+1$ open α_i formulas. Without loss of generality assume that α_1 is an open formula. Then, $\rho, i, \nu \models \varphi$ is equivalent to

(A) $\exists j > i$ such that $\rho, j, \nu \models \beta$, and $\forall i < j' < j$, $\rho, j', \nu \models \bigvee_{g=1}^n \alpha_g$.

- **Case 1.1** This case has been discussed in the main paper (see Section 4.2). Please see the Figure 4 for intuition.
- **Case 1.2** $\geq \in \mathsf{Pol}(\varphi)$.
  - Without loss of generality, we assume the $\alpha_1, \ldots, \alpha_{k+1}$ are open formulas, and rest all the formulas are closed.
  - (A) is equivalent to (C0) $\rho, i, \nu \models (\bigvee_{g=k+2}^{n})\alpha_g \,\mathbf{U}\, \beta$,
    or for some $1 \leq h \leq k+1$, (C1) there exists a point i' such that $\rho, i', \nu \models \alpha_h$, and (C2)for all $i < i'' < i'$, $\rho, i'', \nu \models \bigvee_{l=k+2}^{n} \alpha_l$, and (C3) $\rho, j, \nu \models \beta$.
  - By Untiming Lemma 4.4, (C1) implies that for all $j' \geq i'$, α_h is equivalent to α'_h, where $\alpha'_h = \mathsf{Untime}(\alpha_h)$.
  - Hence, $\varphi \equiv \varphi_1 \vee \varphi_2$ where, $\varphi_1 = [(\bigvee_{g=k+2}^{n})\alpha_g) \,\mathbf{U}\, \beta]$, $\varphi_2 = \bigvee_{h=1}^{k+1} [(\bigvee_{g=k+2}^{n} \alpha_g) \,\mathbf{U}\, (\alpha_h \wedge \varphi'_h]$, $\varphi'_h = ((\bigvee_{1 \leq g \leq n \wedge g \neq h} \alpha_g) \vee \alpha'_h) \,\mathbf{U}\, \beta$, and $\alpha'_h = \mathsf{Untime}(\alpha_h)$. Hence, α'_h is a closed formula.

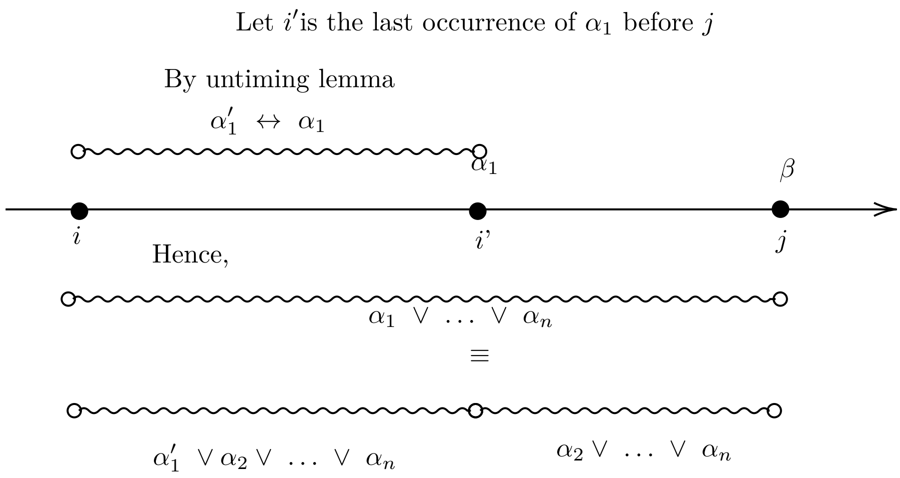

Figure 4 Figure Corresponding to the (B2) in Case 1.1 of Lemma 4.7.

- The left argument in $\mathbf{U}$ for φ_1 is closed (recall that $\alpha_1, \ldots, \alpha_{k+1}$ are the only open formulas) and the right argument (i.e. β) can be reduced to en equivalent formulas $\bigvee_g \beta_g$, where each β_g is in CoT. Hence, φ_1 is equivalent to Θ_1, where Θ_1 is $\Theta_1 = \bigvee_g [(\bigvee_{g=k+2}^{n})\alpha_g) \, \mathbf{U} \, \beta_g]$.
  Moreover φ'_h has one less open α_i subformula, as α'_h is closed. Hence, by induction hypothesis we can reduce φ'_h to an equivalent formula in $\bigvee \mathsf{CoT}^{\varphi}$, $\bigvee_l \psi_l$, where each ψ_l is in CoT and having the same polarity as φ. Pushing these disjunctions at the top, we get $\varphi_2 \equiv \Theta_2$ where $\Theta_2 = \bigvee_{h=1}^{k+1} \bigvee_l [(\bigvee_{g=k+2}^{n} \alpha_g) \, \mathbf{U} \, (\alpha'_h \land \psi_l]$.
  As $\bigvee \mathsf{CoT}^{\varphi}$ is closed under disjunction we just need to show that each $[(\bigvee_{g=k+2}^{n})\alpha_g) \, \mathbf{U} \, \beta_g]$ and $[(\bigvee_{g=k+2}^{n} \alpha_g) \, \mathbf{U} \, (\alpha'_h \land \psi_l]$ can be reduced to an equivalent formula in $\bigvee \mathsf{CoT}^{\varphi}$, which is implied by the induction hypothesis. ◀

D Proof of Lemma 4.8: Missing Cases

Notice that $\overleftarrow{\Box}(\bigvee_{g=1}^{n} \alpha_g) \equiv (\bigvee_{g=1}^{n} \alpha_g) \, \mathbf{S} \, (start)$, where $start = \overleftarrow{\Box} \bot$, is true only at the first point of the timed word. Hence, now applying Lemma 4.7, we get the required result.

Let us now consider $\varphi = \Box(\bigvee_{g=1}^{n} \alpha_i)$. We first discuss the Case 2.2, i.e. $\geq \in \mathsf{Pol}(\varphi)$, as it is not as involved as the Case 2.1 ($\leq \in \mathsf{Pol}(\varphi)$).

Case 2.2. Assume that $\geq \in \mathsf{Pol}(\varphi)$. We apply induction on the number of open α_is. If there are no open α_is we get a closed formula, and hence a $\bigvee \mathsf{CoT}^{\varphi}$ formula. Assume that for k open α_is, the lemma holds. We prove that the lemma holds for $k+1$ open α_is too.

- Without loss of generality we assume the $\alpha_1, \ldots, \alpha_{k+1}$ are open formulas, and the rest $\alpha_{k+2}, \ldots, \alpha_n$ are all closed.
- Then $\rho, i, \nu \models \varphi$ is equivalent to (C0) $\rho, i, \nu \models \Box[\bigvee_{g=k+2}^{n} \alpha_g]$, or,

- (C1) for some $1 \leq h \leq k+1$, there exists i' such that $\rho, i', \nu \models \alpha_h$, and (C2) for all $i < i'' < i'$, $\rho, i'', \nu \models \bigvee_{l=k+2}^{n} \alpha_l$, and (C3) $\rho, i', \nu \models \varphi = \Box(\bigvee_{g=1}^{n} \alpha_g)$.
- By untiming Lemma 4.4, (C1) implies that for all $j' \geq i'$, α_h is equivalent to α'_h, where $\alpha'_h = \mathsf{Untime}(\alpha_h)$.
- Hence, $\varphi \equiv \varphi_1 \vee \varphi_2$ where $\varphi_1 = [\Box(\bigvee_{g=k+2}^{n})\alpha_g)], \varphi_2 = \bigvee_{h=1}^{k+1} [(\bigvee_{g=k+2}^{n} \alpha_g)\, \mathbf{U}\, (\alpha_h \wedge \varphi'_h]$, $\varphi'_h = \Box(\bigvee_{1 \leq g \leq n \wedge g \neq h}(\alpha_g) \vee \alpha'_h)$, and $\alpha'_h = \mathsf{Untime}(\alpha_h)$. Notice that α'_h is a closed formula.
- Hence, φ'_h has at most k open α_is in its argument. Thus, by induction hypothesis, each φ'_h can be reduced into equivalent $\bigvee_l \psi_{h,l}$ where each $\psi_{h,l}$ is in CoT having same polarity as φ. Thus, φ_2 is equivalent to $\Theta_3 = \bigvee_{h=1}^{k+1} \bigvee_l [(\bigvee_{g=k+2}^{n} \alpha_g)\, \mathbf{U}\, (\alpha_h \wedge \psi_{h,l}]$. As $\bigvee \mathsf{CoT}^{\varphi}$ is closed under finite disjunctions, we just need to show that each $[(\bigvee_{g=k+2}^{n} \alpha_g)\, \mathbf{U}\, (\alpha_h \wedge \psi_{h,l}]$ can be reduced to an equivalent $\bigvee \mathsf{CoT}^{\varphi}$ formula. This is implied by Lemma 4.7.

Case 2.1. $\leq \in \mathsf{Pol}(\varphi)$. This case has already been discussed in the main paper. There are only couple of cases (Case A and Case B) left while arguing about the general case of infinite words in Section 4.3.1. We just discuss those cases here.

D.1 Handling the infinite word case: Missing Cases

- Case A - α_1 is not satisfied by any point $j > i$ of ρ for the given valuation ν. In this case, φ is equivalent to $\alpha_2 \vee \ldots \vee \alpha_n$ holds globally after i. Hence,
  - $\varphi_1 = \Box(\alpha_2 \vee \ldots \vee \alpha_n)$

  As φ_1 is smaller than φ, by induction hypothesis we can construct a formula $\delta = \bigvee_k \delta_k$, where each δ_k is in CoT.
- Case B - α_1 is finitely occurring in ρ, ν and is satisfied by some point $j > i$ for a given valuation ν. Then, φ implies α' holds until there exists a point in the future, where α_1 holds (possibly the last time) after which $\alpha_2 \vee \ldots \vee \alpha_n$ holds globally. Hence, we get $\varphi' = \alpha'\, \mathbf{U}\, \{\alpha_1 \wedge (\varphi_1)\}$. Substituting for φ_1 in φ' we get:
  - $\varphi_2 = \bigvee_k [\alpha'\, \mathbf{U}\, \{\alpha_1 \wedge \delta_k\}]$

  φ_2 falls in the category of case 1.1 of Lemma 4.7. Hence, we can construct an equivalent formula ζ in $\bigvee \mathsf{CoT}^{\varphi}$ equivalent to it.

E Past Modalities Add Expressiveness

▶ **Corollary E.1.** UMITPPL *is strictly more expressive than* UPTL[Fut].

Proof. By Lemma 3.4, UPTL is as expressive as 1-UPTL. It is known that 1-TPTL (and hence the subclass 1-UPTL) are less expressive than 1-clock Alternating Timed Automata (1-ATA) [19, 23, 24]. There exists a timed language L over a set of propositions $\{st, a, b\}$ that is not expressible in 1-ATA [25]. We write a UMITL formula expressing this language L, proving the result.

L consists of all timed words ρ for which (1) ρ begins with (st, τ_1), that is, proposition st is true at the first point of ρ with timestamp τ_1, (2) there are exactly two points in the next unit interval $(0, 1)$ from τ_1 with timestamps τ_2, τ_3 and a is true at both these points, and (3) there is a point in ρ with timestamp $\tau_j \in [\tau_2 + 1, \tau_3 + 1]$ where b holds. This is captured in MITL by the formula:

$$\varphi = \quad \neg \Diamond_{(0,1)}\{\overleftarrow{\Diamond}(\overleftarrow{\Diamond}(\overleftarrow{\Diamond}(\top)))\} \wedge \Diamond_{(0,1)}\{\overleftarrow{\Diamond}(\overleftarrow{\Diamond}(\top))\} \wedge \Diamond[b \wedge \overleftarrow{\Diamond}_{[0,1]}\{a \wedge (\overleftarrow{\mathsf{O}}\ (\overleftarrow{\mathsf{O}}\ (st)))\}] \\ \wedge \Diamond[b \wedge \overleftarrow{\Diamond}_{[1,2]}\{a \wedge (\overleftarrow{\mathsf{O}}\ (st))\}]$$

It can be seen that $[\![\varphi]\!] = L$. ◀