

Generalizing Non-punctuality for Timed Temporal Logic with Freeze Quantifiers

Krishna, Shankara Narayanan; Madnani, Khushraj; Mazo, Manuel; Pandya, Paritosh K.

DOI

[10.1007/978-3-030-90870-6_10](https://doi.org/10.1007/978-3-030-90870-6_10)

Publication date

2021

Document Version

Final published version

Published in

Formal Methods

Citation (APA)

Krishna, S. N., Madnani, K., Mazo, M., & Pandya, P. K. (2021). Generalizing Non-punctuality for Timed Temporal Logic with Freeze Quantifiers. In M. Huisman, C. Păsăreanu, & N. Zhan (Eds.), *Formal Methods : Proceedings of the 24th International Symposium, FM 2021* (pp. 182-199). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 13047 LNCS). Springer. https://doi.org/10.1007/978-3-030-90870-6_10

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.



Generalizing Non-punctuality for Timed Temporal Logic with Freeze Quantifiers

Shankara Narayanan Krishna¹, Khushraj Madnani^{2(✉)}, Manuel Mazo Jr.²,
and Paritosh K. Pandya¹

¹ IIT Bombay, Mumbai, India

{krishnas,pandya58}@cse.iitb.ac.in

² Delft University of Technology, Delft, The Netherlands

{k.n.madnani-1,m.mazo}@tudelft.nl

Abstract. Metric Temporal Logic (MTL) and Timed Propositional Temporal Logic (TPTL) are prominent real-time extensions of Linear Temporal Logic (LTL). In general, the satisfiability checking problem for these extensions is undecidable when both the future U and the past S modalities are used. In a classical result, the satisfiability checking for MITL[U,S], a non-punctual fragment of MTL[U,S], is shown to be decidable with EXPSPACE complete complexity. Given that this notion of non-punctuality does not recover decidability in the case of TPTL[U,S], we propose a generalization of non-punctuality called *non-adjacency* for TPTL[U,S], and focus on its 1-variable fragment, 1-TPTL[U,S]. While non-adjacent 1-TPTL[U,S] appears to be a very small fragment, it is strictly more expressive than MITL. As our main result, we show that the satisfiability checking problem for non-adjacent 1-TPTL[U,S] is decidable with EXPSPACE complete complexity.

1 Introduction

Metric Temporal Logic (MTL) and Timed Propositional Temporal Logic (TPTL) are natural extensions of Linear Temporal Logic (LTL) for specifying real-time properties [3]. MTL extends the U and S modalities of LTL by associating a timing interval with these. $aU_I b$ describes behaviours modeled as timed words consisting of a sequence of a 's followed by a b which occurs at a time within (relative) interval I . On the other hand, TPTL uses freeze quantification to store the current time stamp. A Freeze quantifier with clock variable x has the form $x.\varphi$. When it is evaluated at a point i on a timed word, the time stamp τ_i at i is frozen in x , and the formula φ is evaluated using this value for x . Variable x is used in φ in a constraint of the form $T - x \in I$; this constraint, when evaluated at a point j , checks if $\tau_j - \tau_i \in I$, where τ_j is the time stamp at point j .

This work is partially supported by the European Research Council through the SENTIENT project (ERC-2017-STG #755953).

© Springer Nature Switzerland AG 2021

M. Huisman et al. (Eds.): FM 2021, LNCS 13047, pp. 182–199, 2021.

https://doi.org/10.1007/978-3-030-90870-6_10

For example, the formula $Fx.(a \wedge F(b \wedge T - x \in [1, 2] \wedge F(c \wedge T - x \in [1, 2])))^1$ asserts that there is a point in future where a holds and in its future within interval $[1, 2]$, b and c occur, and the former occurs before the latter. This property is not expressible in $MTL[U, S]$ [4, 18]. Moreover, every property in $MTL[U, S]$ can be expressed in $1\text{-}TPTL[U, S]$. Thus, $1\text{-}TPTL[U, S]$ is strictly more expressive than $MTL[U, S]$. Unfortunately, both the logics have an undecidable satisfiability problem, making automated analysis for these logics theoretically impossible.

Exploring natural decidable variants of these logics has been an active area of research since the advent of these logics [2, 8–10, 21–23]. One line of work restricted itself to the future only fragments $MTL[U]$ and $1\text{-}TPTL[U]$ which have both been shown to have decidable satisfiability over finite timed words, under a pointwise interpretation [7, 17]. The complexity however is non-primitive recursive. Reducing the complexity to elementary has been challenging. One of the most celebrated of such logics is the Metric Interval Temporal Logic ($MITL[U, S]$) [1], a subclass of $MTL[U, S]$ where the timing intervals are restricted to be non-punctual (i.e. intervals of the form $\langle x, y \rangle$ where $x < y$). The satisfiability checking for $MITL$ formulae is decidable with $EXPSPACE$ complete complexity [1]. While non-punctuality helps to recover the decidability of $MTL[U, S]$, it does not help $TPTL[U, S]$. The freeze quantifiers of $TPTL$ enables us to trivially express punctual timing constraints using only the non-punctual intervals: for instance the $1\text{-}TPTL$ formula $x.(aU(a \wedge T - x \in [1, \infty) \wedge T - x \in [0, 1]))$ uses only non-punctual intervals but captures the MTL formula $aU_{[1,1]}b$. Thus, a more refined notion of non-punctuality is needed to recover the decidability of $1\text{-}TPTL[U, S]$.

Contributions. With the above observations, to obtain a decidable class of $1\text{-}TPTL[U, S]$ akin to $MITL[U, S]$, we revisit the notion of non-punctuality as it stands currently. As our first contribution, we propose *non-adjacency*, a refined version of non-punctuality. Two intervals, I_1 and I_2 are non-adjacent if the supremum of I_1 is not equal to the infimum of I_2 . Non-adjacent $1\text{-}TPTL[U, S]$ is the subclass of $1\text{-}TPTL[U, S]$ where, every interval used in clock constraints within the same freeze quantifier is non-adjacent to itself and to every other timing interval that appears within the same scope. (Wlog, we consider formulae in negation normal form only.) The non-adjacency restriction disallows punctual timing intervals: every punctual timing interval is adjacent to itself. It can be shown (Theorem 2) that non-adjacent $1\text{-}TPTL[U, S]$, while seemingly very restrictive, is strictly more expressive than $MITL$ and it can also express the counting and the Pnueli modalities [9]. Thus, the logic is of considerable interest in practical real-time specification. See the full version for an example.

Our second contribution is to give a decision procedure for the satisfiability checking of non-adjacent $1\text{-}TPTL[U, S]$. We do this in two steps. 1) We introduce a logic $PnEMTL$ which combines and generalises the automata modalities of [11, 22, 23] and the Pnueli modalities of [9, 10, 21], and has not been studied before to the best of our knowledge. We show that a formula of non-adjacent $1\text{-}TPTL[U, S]$ can be reduced to an equivalent formula of non-adjacent $PnEMTL$ (Theorem 5).

¹ Here T is a special symbol denoting the timestamp of the present point and x is the clock that was frozen when x . was asserted.

2) We prove that the satisfiability of non-adjacent PnEMTL is decidable with EXPSPACE complete complexity (Theorem 6). For brevity, some of the proof details are omitted here and can be found in the full version [13].

Related Work and Discussion. Much of the related work has already been discussed. MITL with counting and Pnueli modalities has been shown to have EXPSPACE-complete satisfiability [20, 21]. Here, we tackle even more expressive logics: namely non-adjacent 1-TPTL[U, S] and non-adjacent PnEMTL. We show that EXPSPACE-completeness of satisfiability checking is retained in spite of the additional expressive power. These decidability results are proved by equisatisfiable reductions to logic EMITL_{0,∞} of Ho [11]. As argued by Ho, it is quite practicable to extend the existing model checking tools like UPPAAL to logic EMITL_{0,∞} and hence to our logics too.

Addition of regular expression based modalities to untimed logics like LTL has been found to be quite useful for practical specification; even the IEEE standard temporal logic PSL has this feature. With a similar motivation, there has been considerable recent work on adding regular expression/automata based modalities to MTL and MITL. Raskin as well as Wilke added automata modalities to MITL as well as an Event-Clock logic *ECL* [22, 23] and showed the decidability of satisfaction. The current authors showed that MTL[U, S_{NP}] (where U can use punctual intervals but S is restricted to non-punctual intervals), when extended with counting as well as regular expression modalities preserves decidability of satisfaction [12, 14–16]. Recently, Ferrère showed the EXPSPACE decidability of MIDL which is LTL[U] extended with a fragment of timed regular expression modality [5]. Moreover, Ho has investigated a PSPACE-complete fragment EMITL_{0,∞} [11]. Our non-adjacent PnEMTL is a novel extension of MITL with modalities which combine the features of EMITL [11, 22, 23] and the Pnueli modalities [9, 10, 21].

2 Preliminaries

Let Σ be a finite set of propositions, and let $\Gamma = 2^\Sigma \setminus \emptyset$. A word over Σ is a finite sequence $\sigma = \sigma_1\sigma_2 \dots \sigma_n$, where $\sigma_i \in \Gamma$. A timed word ρ over Σ is a finite sequence of pairs $(\sigma, \tau) \in \Sigma \times \mathbb{R}_{\geq 0}$; $\rho = (\sigma_1, \tau_1) \dots (\sigma_n, \tau_n) \in (\Sigma \times \mathbb{R}_{\geq 0})^*$ where $\tau_1 = 0$ and $\tau_i \leq \tau_j$ for all $1 \leq i \leq j \leq n$. The τ_i are called time stamps. For a timed or untimed word ρ , let $\text{dom}(\rho) = \{i \mid 1 \leq i \leq |\rho|\}$ where $|\rho|$ denotes the number of (event, timestamp) pairs composing the word ρ , and $\sigma[i]$ denotes the symbol at position $i \in \text{dom}(\rho)$. The set of timed words over Σ is denoted $T\Sigma^*$. Given a (timed) word ρ and $i \in \text{dom}(\rho)$, a pointed (timed) word is the pair ρ, i . Let $\mathcal{I}_{\text{int}}$ ($\mathcal{I}_{\text{nat}}$) be the set of open, half-open or closed time intervals, such that the end points of these intervals are in $\mathbb{Z} \cup \{-\infty, \infty\}$ ($\mathbb{N} \cup \{0, \infty\}$, respectively). We assume familiarity with LTL.

Metric Temporal Logic (MTL). MTL is a real-time extension of LTL where the modalities (U and S) are guarded with intervals. Formulae of MTL are built from Σ using Boolean connectives and time constrained versions U_I and S_I of

the standard $\mathbf{U}, \mathbf{S}$ modalities, where $I \in \mathcal{I}_{\text{nat}}$. Intervals of the form $[x, x]$ are called punctual; a non-punctual interval is one which is not punctual. Formulae in MTL are defined as follows. $\varphi ::= a \mid \top \mid \varphi \wedge \varphi \mid \neg \varphi \mid \varphi \mathbf{U}_I \varphi \mid \varphi \mathbf{S}_I \varphi$, where $a \in \Sigma$ and $I \in \mathcal{I}_{\text{nat}}$. For a timed word $\rho = (\sigma_1, \tau_1)(\sigma_2, \tau_2) \dots (\sigma_n, \tau_n) \in T\Sigma^*$, a position $i \in \text{dom}(\rho)$, an MTL formula φ , the satisfaction of φ at a position i of ρ , denoted $\rho, i \models \varphi$, is defined below. We discuss the time constrained modalities.

- $\rho, i \models \varphi_1 \mathbf{U}_I \varphi_2$ iff $\exists j > i, \rho, j \models \varphi_2, \tau_j - \tau_i \in I$, and $\rho, k \models \varphi_1 \forall i < k < j$,
- $\rho, i \models \varphi_1 \mathbf{S}_I \varphi_2$ iff $\exists j < i, \rho, j \models \varphi_2, \tau_j - \tau_i \in I$, and $\rho, k \models \varphi_1 \forall j < k < i$.

The language of an MTL formula φ is defined as $L(\varphi) = \{\rho \mid \rho, 1 \models \varphi\}$. Using the above, we obtain some derived formulae: the *constrained eventual* operator $\mathbf{F}_I \varphi \equiv \text{true} \mathbf{U}_I \varphi$ and its dual is $\mathcal{G}_I \varphi \equiv \neg \mathbf{F}_I \neg \varphi$. Similarly $\mathcal{H}_I \varphi \equiv \text{true} \mathbf{S}_I \varphi$. The *next* operator is defined as $\oplus_I \varphi \equiv \perp \mathbf{U}_I \varphi$. The non-strict versions of $\mathbf{F}, \mathcal{G}$ are respectively denoted $\mathbf{F}^w$ and $\mathcal{G}^w$ and include the present point. Symmetric non-strict versions for past operators are also allowed. The subclass of MTL obtained by restricting the intervals I in the until and since modalities to **non-punctual intervals** is denoted MITL. We say that a formula φ is satisfiable iff $L(\varphi) \neq \emptyset$.

Theorem 1. *Satisfiability checking for MTL[$\mathbf{U}, \mathbf{S}$] is undecidable [2]. Satisfiability Checking for MITL is EXPSPACE-complete [1].*

Time Propositional Temporal Logic (TPTL). The logic TPTL also extends LTL using freeze quantifiers. Like MTL, TPTL is also evaluated on timed words. Formulae of TPTL are built from Σ using Boolean connectives, modalities $\mathbf{U}$ and $\mathbf{S}$ of LTL. In addition, TPTL uses a finite set of real valued clock variables $X = \{x_1, \dots, x_n\}$. Let $\nu : X \rightarrow \mathbb{R}_{\geq 0}$ represent a valuation assigning a non-negative real value to each clock variable. The formulae of TPTL are defined as follows. Without loss of generality we work with TPTL in the negation normal form. $\varphi ::= a \mid \neg a \mid \top \mid \perp \mid x.\varphi \mid T-x \in I \mid x-T \in I \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{S} \varphi \mid \mathcal{G} \varphi \mid \mathcal{H} \varphi$, where $x \in X, a \in \Sigma, I \in \mathcal{I}_{\text{int}}$. Here T denotes the time stamp of the point where the formula is being evaluated. $x.\varphi$ is the freeze quantification construct which remembers the time stamp of the current point in variable x and evaluates φ .

For a timed word $\rho = (\sigma_1, \tau_1) \dots (\sigma_n, \tau_n)$, $i \in \text{dom}(\rho)$ and a TPTL formula φ , we define the satisfiability relation, $\rho, i, \nu \models \varphi$ with valuation ν of all the clock variables. We omit the semantics of Boolean, $\mathbf{U}$ and $\mathbf{S}$ operators as they are similar to those of LTL.

- $\rho, i, \nu \models a$ iff $a \in \sigma_i$, and $\rho, i, \nu \models x.\varphi$ iff $\rho, i, \nu[x \leftarrow \tau_i] \models \varphi$
- $\rho, i, \nu \models T-x \in I$ iff $\tau_i - \nu(x) \in I$, and $\rho, i, \nu \models x-T \in I$ iff $\nu(x) - \tau_i \in I$
- $\rho, i, \nu \models \mathcal{G} \varphi$ iff $\forall j > i, \rho, j, \nu \models \varphi$, and
- $\rho, i, \nu \models \mathcal{H} \varphi$ iff $\forall j < i, \rho, j, \nu \models \varphi$

Let $\bar{0} = (0, 0, \dots, 0)$ represent the initial valuation of all clock variables. For a timed word ρ and $i \in \text{dom}(\rho)$, we say that ρ, i satisfies φ denoted $\rho, i \models \varphi$ iff $\rho, i, \bar{0} \models \varphi$. The language of φ , $L(\varphi) = \{\rho \mid \rho, 1 \models \varphi\}$. The Pointed Language of φ is defined as $L_{pt}(\varphi) = \{\rho, i \mid \rho, i \models \varphi\}$. Subclass of TPTL that uses

only 1 clock variable (i.e. $|X| = 1$) is known as 1-TPTL. The satisfiability checking for 1-TPTL[U, S] is undecidable, which is implied by Theorem 1 and the fact that 1-TPTL[U, S] trivially generalizes MTL[U, S]. As an example, the formula $\varphi = x.(aU(bU(c \wedge T - x \in [1, 2])))$ is satisfied by the timed word $\rho = (a, 0)(a, 0.2)(b, 1.1)(b, 1.9)(c, 1.91)(c, 2.1)$ since $\rho, 1 \models \varphi$. The word $\rho' = (a, 0)(a, 0.3)(b, 1.4)(c, 2.1)(c, 2.5)$ does not satisfy φ . However, $\rho', 2 \models \varphi$: if we start from the second position of ρ' , we assign $\nu(x) = 0.3$, and when we reach the position 4 of ρ' with $\tau_4 = 2.1$ we obtain $T - x = 2.1 - 0.3 \in [1, 2]$.

3 Introducing Non-adjacent 1-TPTL and PnEMTL

In this section, we define non-adjacent 1-TPTL and PnEMTL logics. Let x denote the unique freeze variable we use in 1-TPTL.

Non-Adjacent 1-TPTL is defined as a subclass of 1-TPTL where adjacent intervals within the scope of any freeze quantifier is disallowed. Two intervals $I_1, I_2 \in \mathcal{I}_{\text{int}}$ are non-adjacent iff $\sup(I_1) = \inf(I_2) \Rightarrow \sup(I_1) = 0$. A set $\mathcal{I}_{na}$ of intervals is non-adjacent iff any two intervals in $\mathcal{I}_{na}$ are non-adjacent. It does not contain punctual intervals other than $[0, 0]$ as every punctual interval is adjacent to itself. For example, the set $\{[1, 2], (2, 3], [5, 6]\}$ is not a non-adjacent set, while $\{[0, 0], [0, 1], (3, 4], [5, 6]\}$ is. Let $\mathcal{I}_{na}$ denote a set of non-adjacent intervals with end points in $\mathbb{Z} \cup \{-\infty, \infty\}$. See full version for an example specification using this logic. The freeze depth of a TPTL formula φ , $\text{fd}(\varphi)$ is defined inductively. For a ptopositional formula prop , $\text{fd}(\text{prop}) = 0$. Also, $\text{fd}(x.\varphi) = \text{fd}(\varphi) + 1$, and $\text{fd}(\varphi_1 U \varphi_2) = \text{fd}(\varphi_1 S \varphi_2) = \text{fd}(\varphi_1 \wedge \varphi_2) = \text{fd}(\varphi_1 \vee \varphi_2) = \text{Max}(\text{fd}(\varphi_1), \text{fd}(\varphi_2))$, $\text{fd}(\mathcal{G}(\varphi)) = \text{fd}(\mathcal{H}(\varphi)) = \text{fd}(\varphi)$.

Theorem 2. *Non-Adjacent 1-TPTL[U, S] is more expressive than MITL[U, S]. It can also express the Counting and the Pnueli modalities of [9, 10].*

The straightforward translation of MITL into TPTL in fact gives rise to non-adjacent 1-TPTL formula. Let $\hat{I}$ abbreviate $T - x \in I$. E.g. MITL formula $aU_{[2,3]}(bU_{[3,4]}c)$ translates to $x.(aU([2, 3] \wedge x.(bU([3, 4] \wedge c)))$. It has been previously shown that $F[x.(a \wedge F(b \wedge \widehat{(1, 2)} \wedge F(c \wedge \widehat{(1, 2)})))]$, which is in fact a formula of non-adjacent 1-TPTL, is inexpressible in MTL[U, S] [18]. The Pnueli modality $\text{Pn}_I(\phi_1, \dots, \phi_k)$ expresses that there exist positions $i_1 \leq \dots \leq i_k$ within (relative) interval I where each i_j satisfies ϕ_j . This is equivalent to the non-adjacent 1-TPTL formula $x.(F(\hat{I} \wedge \phi_1 \wedge F(\hat{I} \wedge \phi_2 \wedge F(\dots))))$. Similarly the (simpler) counting modality can also be expressed.

Pnueli EMTL: There have been several attempts to extend logic MTL[U] with regular expression/automaton modalities [5, 11, 14, 23]. We use a generalization of these existing modalities to give the logic PnEMTL. For any finite automaton A , let $L(A)$ denote the language of A .

Given a finite alphabet Σ , formulae of PnEMTL have the following syntax:

$$\varphi ::= a \mid \varphi \wedge \varphi \mid \neg \varphi \mid \mathcal{F}_{I_1, \dots, I_k}^k(A_1, \dots, A_{k+1})(S) \mid \mathcal{P}_{I_1, \dots, I_k}^k(A_1, \dots, A_{k+1})(S)$$

where $a \in \Sigma$, $I_1, I_2, \dots, I_k \in \mathcal{I}_{\text{nat}}$ and $A_1, \dots, A_{k+1}$ are automata over 2^S where

S is a set of formulae from PnEMTL. $\mathcal{F}^k$ and $\mathcal{P}^k$ are the new modalities called future and past **Pnueli Automata** Modalities, respectively, where k is the arity of these modalities.

Let $\rho = (a_1, \tau_1), \dots, (a_n, \tau_n) \in T\Sigma^*$, $x, y \in \text{dom}(\rho)$, $x \leq y$ and $S = \{\varphi_1, \dots, \varphi_n\}$ be a given set of PnEMTL formulae. Let S_i be the exact subset of formulae from S evaluating to true at ρ, i , and let $\text{Seg}^+(\rho, x, y, S)$ and $\text{Seg}^-(\rho, y, x, S)$ be the untimed words $S_x S_{x+1} \dots S_y$ and $S_y S_{y-1} \dots S_x$ respectively. Then, the satisfaction relation for ρ, i_0 satisfying a PnEMTL formula φ is defined recursively as follows:

- $\rho, i_0 \models \mathcal{F}_{I_1, \dots, I_k}^k(A_1, \dots, A_{k+1})(S)$ iff $\exists i_0 \leq i_1 \leq i_2 \dots \leq i_k \leq n$ s.t.

$$\bigwedge_{w=1}^k [(\tau_{i_w} - \tau_{i_0} \in I_w) \wedge \text{Seg}^+(\rho, i_{w-1}, i_w, S) \in L(A_w)] \wedge \text{Seg}^+(\rho, i_k, n, S) \in L(A_{k+1})$$
- $\rho, i_0 \models \mathcal{P}_{I_1, I_2, \dots, I_k}^k(A_1, \dots, A_k, A_{k+1})(S)$ iff $\exists i_0 \geq i_1 \geq i_2 \dots \geq i_k \geq n$ s.t.

$$\bigwedge_{w=1}^k [(\tau_{i_0} - \tau_{i_w} \in I_w) \wedge \text{Seg}^-(\rho, i_{w-1}, i_w, S) \in L(A_w)] \wedge \text{Seg}^-(\rho, i_k, n, S) \in L(A_{k+1}).$$

Language of any PnEMTL formula φ , as $L(\varphi) = \{\rho | \rho, 1 \models \varphi\}$. The Pointed Language of φ is defined as $L_{pt}(\varphi) = \{\rho, i | \rho, i \models \varphi\}$. Given a PnEMTL formula φ , its arity is the maximum number of intervals appearing in any $\mathcal{F}, \mathcal{P}$ modality of φ . For example, the arity of $\varphi = \mathcal{F}_{I_1, I_2}^2(A_1, A_2, A_3)(S_1) \wedge \mathcal{P}_{I_1}^1(A_1, A_2)(S_2)$ for some sets of formulae S_1, S_2 is 2. For the sake of brevity, $\mathcal{F}_{I_1, \dots, I_k}^k(A_1, \dots, A_k)(S)$ denotes $\mathcal{F}_{I_1, \dots, I_k}^k(A_1, \dots, A_k, A_{k+1})(S)$ where automata A_{k+1} accepts all the strings over S . We define **non-adjacent** PnEMTL, as a subclass where every modality $\mathcal{F}_{I_1, \dots, I_k}^k$ and $\mathcal{P}_{I_1, \dots, I_k}^k$ is such that $\{I_1, \dots, I_k\}$ is a non-adjacent set of intervals.

EMITL of [23] (and variants of it studied in [5, 11, 14, 15]) are special cases of the non-adjacent PnEMTL modality where the arity is restricted to 1 and the second automata in the argument accepts all the strings. Hence, automaton modality of [23] is of the form $\mathcal{F}_1(A)(S)$. Let EMITL_{0,∞} denote the logic EMITL extended with $\mathcal{F}$ and $\mathcal{P}$ modality where the timing intervals are restricted to be of the form $\langle l, \infty \rangle$ or $\langle 0, u \rangle$.

We conclude this section defining size of a temporal logic formula.

Size of Formulae. Size of a formula φ denoted by $|\varphi|$ is a measure of how many bits are required to store it. The size of a TPTL formula is defined as the sum of the number of U, S and Boolean operators and freeze quantifiers in it. For PnEMTL formulae, $|\text{op}|$ is defined as the number of Boolean operators and variables used in it. $|\mathcal{F}_{I_1, \dots, I_k}^k(A_1, \dots, A_{k+1})(S)| = \sum_{\varphi \in S} (|\varphi|) + |A_1| + \dots + |A_{k+1}|$

where $|A|$ denotes the size of the automaton A given by sum of number of its states and transitions.

4 Anchored Interval Word Abstraction

All the logics considered here have the feature that a sub-formula asserts timing constraints on various positions relative to an anchor position; e.g. the position of freezing the clock in TPTL. Such constraints can be symbolically represented as an interval word with a unique anchor position and all other positions carry a set of time intervals constraining the time stamp of the position relative to the time stamp of the anchor. See interval word κ in Fig. 1. We now define these interval words formally. Let $I_\nu \subseteq \mathcal{I}_{\text{int}}$. An I_ν -interval word over Σ is a word κ of the form $a_1 a_2 \dots a_n \in (2^{\Sigma \cup \{\text{anch}\} \cup I_\nu})^*$. There is a unique $i \in \text{dom}(\kappa)$ called the *anchor* of κ and denoted by $\text{anch}(\kappa)$. At the anchor position i , $a_i \subseteq \Sigma \cup \{\text{anch}\}$, and $\text{anch} \in a_i$. Let J be any interval in I_ν . We say that a point $i \in \text{dom}(\kappa)$ is a J -time restricted point if and only if, $J \in a_i$. i is called time restricted point if and only if either i is J -time restricted for some interval J in I_ν or $\text{anch} \in a_i$.

From I_ν -interval word to Timed Words: Given a I_ν -interval word $\kappa = a_1 \dots a_n$ over Σ and a timed word $\rho = (b_1, \tau_1) \dots (b_m, \tau_m)$, the pointed timed word ρ, i is consistent with κ iff $\text{dom}(\rho) = \text{dom}(\kappa)$, $i = \text{anch}(\kappa)$, for all $j \in \text{dom}(\kappa)$, $b_j = a_j \cap \Sigma$ and for $j \neq i$, $I \in a_j \cap I_\nu$ implies $\tau_j - \tau_i \in I$. Thus, κ and ρ, i agree on propositions at all positions, and the time stamp of a non-anchor position j in ρ satisfies every interval constraint in a_j relative to τ_i , the time stamp of anchor position. $\text{Time}(\kappa)$ denotes the set of all the pointed timed words consistent with a given interval word κ , and $\text{Time}(\Omega) = \bigcup_{\kappa \in \Omega} (\text{Time}(\kappa))$ for a set of interval words Ω . Note that the “consistency relation” is a many-to-many relation.

Example. Let $\kappa = \{a, b, (-1, 0)\} \{b, (-1, 0)\} \{a, \text{anch}\} \{b, [2, 3]\}$ be an interval word over the set of intervals $\{(-1, 0), [2, 3]\}$. Consider timed words ρ and ρ' s.t. $\rho = (\{a, b\}, 0)(\{b\}, .5)(\{a\}, .95)(\{b\}, 3)$, $\rho' = (\{a, b\}, 0)(\{b\}, 0.8)(\{a\}, 0.9)(\{b\}, 2.9)$.

Then $\rho, 3$ as well as $\rho', 3$ are consistent with κ while $\rho, 2$ is not. Likewise, for the timed word $\rho'' = (\{a, b\}, 0)(\{b\}, 0.5)(\{a\}, 1.1)(\{b\}, 3)$, $\rho'', 3$ is not consistent with κ as $\tau_1 - \tau_3 \notin (-1, 0)$, as also $\tau_4 - \tau_3 \notin [2, 3]$.

Let $I_\nu, I'_\nu \subseteq \mathcal{I}_{\text{int}}$. Let $\kappa = a_1 \dots a_n$ and $\kappa' = b_1 \dots b_m$ be I_ν and I'_ν -interval words, respectively. κ is *similar* to κ' , denoted by $\kappa \sim \kappa'$ if and only if,

(i) $\text{dom}(\kappa) = \text{dom}(\kappa')$, (ii) for all $i \in \text{dom}(\kappa)$, $a_i \cap \Sigma = b_i \cap \Sigma$, and (iii) $\text{anch}(\kappa) = \text{anch}(\kappa')$. Additionally, κ is *congruent* to κ' , denoted by $\kappa \cong \kappa'$, iff $\text{Time}(\kappa) = \text{Time}(\kappa')$. I.e., κ and κ' abstract the same set of pointed timed words.

Collapsed Interval Words. The set of interval constraints at a position can be collapsed into a single interval by taking the intersection of all the intervals at that position giving a Collapsed Interval Word. Given an I_ν -interval word $\kappa = a_1 \dots a_n$, let $\mathcal{I}_j = a_j \cap I_\nu$. Let $\kappa' = \text{Col}(\kappa)$ be the word obtained by replacing $\mathcal{I}_j$ with $\bigcap_{I \in \mathcal{I}_j} I$ in a_j , for all $j \in \text{dom}(\kappa)$. Note that κ' is an interval word over $\text{CL}(I_\nu) = \{I | I = \bigcap I', I' \subseteq I_\nu\}$. Note that if for any j , the set $\mathcal{I}_j$ contains two disjoint intervals (like $[1, 2]$ and $[3, 4]$) then $\text{Col}(\kappa)$ is undefined. It is clear that $\text{Time}(\kappa) = \text{Time}(\kappa')$. An interval word κ is called *collapsed* iff $\kappa = \text{Col}(\kappa)$.

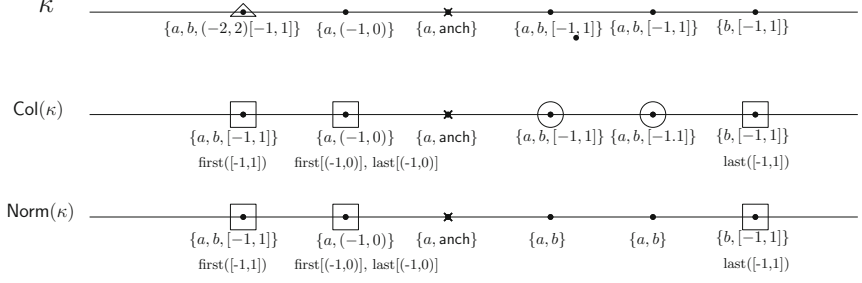


Fig. 1. Point within the triangle has more than one interval. The encircled points are intermediate points and carry redundant information. The required timing constraint is encoded by first and last time restricted points of all the intervals (within boxes).

Normalization of Interval Words. An interval I may repeat many times in a collapsed interval word κ . Some of these occurrences are redundant and we can only keep the first and last occurrence of the interval in the normalized form of κ . See Fig. 1. For a collapsed interval word κ and any $I \in I_\nu$, let $\text{first}(\kappa, I)$ and $\text{last}(\kappa, I)$ denote the positions of first and last occurrence of I in κ . If κ does not contain any occurrence of I , then both $\text{first}(\kappa, I) = \text{last}(\kappa, I) = \perp$. We define, $\text{Boundary}(\kappa) = \{i | i \in \text{dom}(\kappa) \wedge \exists I \in I_\nu \text{ s.t. } (i = \text{first}(\kappa, I) \vee i = \text{last}(\kappa, I) \vee i = \text{anch}(\kappa))\}$

The normalized interval word corresponding to κ , denoted $\text{Norm}(\kappa)$, is defined as $\kappa_{\text{nor}} = b_1 \dots b_m$, such that (i) $\kappa_{\text{nor}} \sim \text{Col}(\kappa)$, (ii) for all $I \in \text{CL}(I_\nu)$, $\text{first}(\kappa, I) = \text{first}(\kappa_{\text{nor}}, I)$, $\text{last}(\kappa, I) = \text{last}(\kappa_{\text{nor}}, I)$, and for all points $j \in \text{dom}(\kappa_{\text{nor}})$ with $\text{first}(\kappa, I) < j < \text{last}(\kappa, I)$, j is not a I -time constrained point. See Fig. 1. Hence, a normalized word is a collapsed word where for any $J \in \text{CL}(I_\nu)$ there are at most two J -time restricted points. This is the key property which will be used to reduce 1-TPTL to a finite length PnEMTL formulae.

Lemma 1. $\kappa \cong \text{Norm}(\kappa)$. Note, $\text{Norm}(\kappa)$ has at most $2 \times |I_\nu|^2 + 1$ restricted points.

The proof follows from the fact that $\kappa \cong \text{Col}(\kappa)$ and since $\text{Col}(\kappa) \sim \text{Norm}(\kappa)$, the set of timed words consistent with any of them will have identical untimed behaviour. For the timed part, the key observation is as follows. For some interval $I \in I_\nu$, let $i' = \text{first}(\kappa, I)$, $j' = \text{last}(\kappa, I)$. Then for any ρ, i in $\text{Time}(\kappa)$, points i' and j' are within the interval I from point i . Hence, any point $i' \leq i'' \leq j'$ is also within interval I from i . Thus, the interval I need not be explicitly mentioned at intermediate points. The full proof can be found in the full version.

5 1-TPTL to PnEMTL

In this section, we reduce a 1-TPTL formula into an equisatisfiable PnEMTL formula. First, we consider 1-TPTL formula with a single outermost freeze quantifier (call these simple TPTL formulae) and give the reduction. More complex

formulae can be handled by applying the same reduction recursively as shown in the first step. For any set of formulae S , let $\bigvee S$ denote $\bigvee_{s \in S} s$. A TPTL formula is said to be *simple* if it is of the form $x.\varphi$ where, φ is a 1-TPTL formula with no freeze quantifiers. Let $\mathcal{I}_\nu \subseteq \mathcal{I}_{\text{int}}$. Let $\psi = x.\varphi$ be a simple $\mathcal{I}_\nu$ -TPTL formula and let $\text{CL}(\mathcal{I}_\nu) = I_\nu$. We construct a PnEMTL formula ϕ , such that $\rho, i \models \psi \iff \rho, i \models \phi$. We break this construction into the following steps:

- 1) We construct an LTL formula α s.t. $L(\alpha)$ contains only I_ν -interval words and $\rho, i \models \psi$ iff $\rho, i \in \text{Time}(L(\alpha))$. Let A be the NFA s.t. $L(A) = L(\alpha)$. Let W be the set of all I_ν -interval words.
- 2) We partition W into finitely many types, each *type*, capturing a certain relative ordering between first and last occurrences of intervals from I_ν as well as *anch*. Let $\mathcal{T}(I_\nu)$ be the finite set of all types.
- 3) For each type $\text{seq} \in \mathcal{T}$, we construct an NFA A_{seq} such that $L(A_{\text{seq}}) = \text{Norm}(L(A) \cap W_{\text{seq}})$, where W_{seq} is the set of all the I_ν -interval words of type seq .
- 4) For every type seq , using the A_{seq} above, we construct a PnEMTL formula ϕ_{seq} such that, $\rho, i \models \phi_{\text{seq}}$ if and only if $\rho, i \in \text{Time}(L(A_{\text{seq}}))$. The desired $\phi = \bigvee_{\text{seq} \in \mathcal{T}(I_\nu)} \phi_{\text{seq}}$. Hence, $L_{\text{pt}}(\phi) = \bigcup_{\text{seq} \in \mathcal{T}} \text{Time}(L(A_{\text{seq}})) = \text{Time}(L(A)) = L_{\text{pt}}(\psi)$.

1a) Simple TPTL to LTL over Interval Words: As above, $\psi = x.\varphi$. Consider an LTL formula $\alpha = \text{F}[\text{LTL}(\varphi) \wedge \text{anch} \wedge \neg(\text{F}(\text{anch}) \vee \text{P}(\text{anch}))] \wedge \mathcal{G}(\bigvee \Sigma)$ over $\Sigma' = \Sigma \cup \mathcal{I}_\nu \cup \{\text{anch}\}$, where $\text{LTL}(\varphi)$ is the LTL formula obtained from φ by replacing clock constraints $T - x \in I$ with I and $x - T \in I$ with $-I$. Then all words in $L(\alpha)$ are $\mathcal{I}_\nu$ -interval words.

Theorem 3. *For any timed word $\rho, i \in \text{dom}(\rho)$, and any clock valuation v , $\rho, i, v \models \psi \iff \rho, i \in \text{Time}(L(\alpha))$.*

Proof Sketch. Note that for any timed word ρ and $i \in \text{dom}(\rho)$, $\rho, i, [x \leftarrow \tau_i] \models \varphi$ is equivalent to $\rho, i \models \psi$ since $\psi = x.\varphi$. Let κ be any $\mathcal{I}_\nu$ -interval word over Σ with $\text{anch}(\kappa) = i$. It can be seen that if $\kappa, i \models \text{LTL}(\varphi)$ then for all $\rho, i \in \text{Time}(\kappa)$ we have $\rho, i \models \psi$. Likewise, if $\rho, i \models \psi$ for a timed word ρ , then there exists some $\mathcal{I}_\nu$ -interval word over Σ such that $\rho, i \in \text{Time}(\kappa)$ and $\kappa, i \models \text{LTL}(\varphi)$.

Illustrated on an example, if $\psi = x.\varphi$ and $\varphi = \text{F}(x \in I \wedge a)$. Then $\rho, i \models \varphi$ iff there exists a point j within an interval I from i , where a holds. Now consider $\alpha = \text{F}^w[(I \wedge a) \wedge \text{anch} \wedge \neg(\text{F}(\text{anch}) \vee \text{P}(\text{anch}))] \wedge \mathcal{G}^w(\bigvee \Sigma)$ whose language consists of interval words κ such that there is a point ahead of the anchor point i where both a and I holds. Clearly, words in $\text{Time}(\kappa)$ are such that they contain a point $j > i$ within an interval I from point i where a holds. Hence, $\rho, i \models \psi$ if and only if $\rho, i \in \text{Time}(\{\kappa \mid \kappa, i \models \text{LTL}(\varphi)\})$. Moreover, $\kappa \in L(\alpha)$ if and only if $\kappa, i \models \text{LTL}(\varphi)$ and $\text{anch}(\kappa) = i$. The full proof is in the full version.

1b) LTL to NFA over Collapsed Interval Words. It is known that for any $\text{LTL}[\text{U}, \text{S}]$ formula, one can construct an equivalent NFA with at most exponential number of states [6]. We reduce the LTL formula α to an equivalent NFA $A_\alpha = (Q, \text{init}, 2^{\Sigma'}, \delta', F)$ over I_ν -interval words, where $\Sigma' = 2^{\Sigma \cup I_\nu \cup \{\text{anch}\}}$. From

A_α , we construct an automaton $A = (Q, \text{init}, 2^{\Sigma'}, \delta, F)$ s.t. $L(A) = \text{Col}(L(A_\alpha))$. Automaton A is obtained from A_α by replacing the set of intervals $\mathcal{I}$ on the transitions by the single interval $\bigcap \mathcal{I}$. In case $\exists I_1, I_2 \in \mathcal{I}$ s.t. $I_1 \cap I_2 = \emptyset$ (i.e. with contradictory interval constraints), the transition is omitted in A . This gives $L(A) = \text{Col}(L(A_\alpha))$.

2) Partitioning Interval Words. We discuss here how to partition W , the set of all collapsed I_ν -interval words, into finitely many classes. Each class is characterized by its **type** given as a finite sequences **seq** over $I_\nu \cup \{\text{anch}\}$. For any collapsed $w \in W$, its type **seq** gives an ordering between $\text{anch}(w)$, $\text{first}(w, I)$ and $\text{last}(w, I)$ for all $I \in I_\nu$, such that, any $I \in I_\nu$ appears at most twice and **anch** appears exactly once in **seq**. For instance, $\text{seq} = I_1 I_1 \text{anch} I_2 I_2$ is a sequence different from $\text{seq}' = I_1 I_2 \text{anch} I_2 I_1$ since the relative orderings between the first and last occurrences of I_1, I_2 and **anch** differ in both. Let the set of types $\mathcal{T}(I_\nu)$ be the set of all such sequences; by definition, $\mathcal{T}(I_\nu)$ is finite. Given $w \in W$, let $\text{Boundary}(w) = \{i_1, i_2, \dots, i_k\}$ be the positions of w which are either $\text{first}(w, I)$ or $\text{last}(w, I)$ for some $I \in I_\nu$ or is $\text{anch}(w)$. Let $w \downarrow_{\text{Boundary}(w)}$ be the subword of w obtained by projecting w to the positions in $\text{Boundary}(w)$, restricted to the sub alphabet $2^{I_\nu} \cup \{\text{anch}\}$. For example, $w = \{a, I_1\}\{b, I_1\}\{c, I_2\}\{\text{anch}, a\}\{b, I_1\}\{b, I_2\}\{c, I_2\}$ gives $w \downarrow_{\text{Boundary}(w)}$ as $I_1 I_2 \text{anch} I_1 I_2$. Then w is in the partition W_{seq} iff $w \downarrow_{\text{Boundary}(w)} = \text{seq}$. Clearly, $W = \bigcup_{\text{seq} \in \mathcal{T}(I_\nu)} W_{\text{seq}}$. Continuing with the example above, w is a collapsed $\{I_1, I_2\}$ -interval word over $\{a, b, c\}$, with $\text{Boundary}(w) = \{1, 3, 4, 5, 7\}$, and $w \in W_{\text{seq}}$ for $\text{seq} = I_1 I_2 \text{anch} I_1 I_2$, while $w \notin W_{\text{seq}'}$ for $\text{seq}' = I_1 I_1 \text{anch} I_2 I_2$.

3) Construction of NFA for each type: Let **seq** be any sequence in $\mathcal{T}(I_\nu)$. In this section, given $A = (Q, \text{init}, 2^{\Sigma'}, \delta, F)$ as constructed above, we construct an NFA $A_{\text{seq}} = (Q \times \{1, 2, \dots, |\text{seq}| + 1\} \cup \{\perp\}, (\text{init}, 1), 2^{\Sigma'}, \delta_{\text{seq}}, F \times \{|\text{seq}| + 1\})$ such that $L(A_{\text{seq}}) = \text{Norm}(L(A) \cap W_{\text{seq}})$. Thus, $\bigcup_{\text{seq} \in \mathcal{T}(I_\nu)} L(A_{\text{seq}}) = \text{Norm}(L(A))$. Thus, $\bigcup_{\text{seq} \in \mathcal{T}(I_\nu)} \text{Time}(L(A_{\text{seq}})) = \text{Time}(\text{Norm}(L(A))) = \text{Time}(L(A)) = L(\psi)$. Intuitively, the second element of the state makes sure that only normalized words of type **seq** are accepted. From (q, j) , A_{seq} is allowed to read a set $S \subseteq \Sigma$ (containing no time interval or **anch** and hence an unrestricted point) or it can read a set $S \cup \{I\}$ where $S \subseteq \Sigma$ and $J = \text{seq}[j]$ (containing time interval/anchor $\text{seq}[j]$). In case of latter, the A_{seq} ends up with a state of the form $(q', j + 1)$ if and only if there is a transition in A of the form $q \xrightarrow{S \cup J} q'$. In case of the former, it non-deterministically proceeds to a state (q', j) if and only if, in automaton A , there is a transition of the form $q \xrightarrow{S} q'$ or $q \xrightarrow{S \cup J} q'$ where $\text{first}(J, w)$ has already been read and $\text{last}(J, w)$ is yet to be read in the future (that is, $\exists j'' j', j' < j \leq j'' \wedge \text{seq}[j'] = \text{seq}[j''] = J$). The detailed construction as well as the proof for Lemma 2 can be found in the full version. Let W_{seq} denote set of I_ν -interval words of type **seq**.

Lemma 2. $L(A_{\text{seq}}) = \text{Norm}(L(A) \cap W_{\text{seq}})$. Hence, $\bigcup_{\text{seq} \in \mathcal{T}(I_\nu)} L(A_{\text{seq}}) = \text{Norm}(L(A))$.

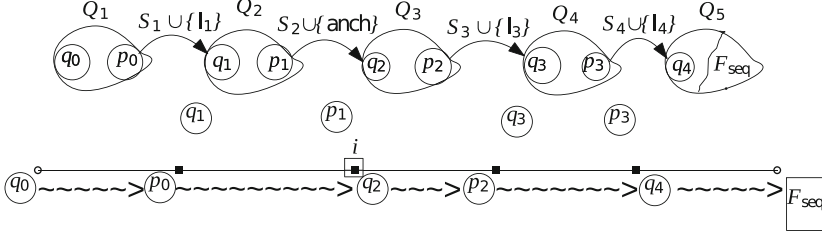


Fig. 2. Figure representing set of runs $A_{I_1 \text{ anch } I_3 I_4}$ of type $Qseq$ where each $S_i \subseteq \Sigma$ and each sub-automaton Q_i has only transitions without any intervals. Here $Qseq = T_1 T_2 T_3 T_4$, for $1 \leq i \leq 4$, $T_i = (p_{i-1} \xrightarrow{S_i \cup \{I_i\}} q_i)$, $I_2 = \{\text{anch}\}$.

Our next step is to reduce the NFAs A_{seq} corresponding to each type seq to PnEMTL. The words in $L(A_{seq})$ are all normalized, and have at most $2|I_\nu| + 1$ -time restricted points. Thanks to this, its corresponding timed language can be expressed using PnEMTL formulae with arity at most $2|I_\nu|$.

4) Reducing NFA of each type to PnEMTL: Next, for each A_{seq} we construct PnEMTL formula ϕ_{seq} such that, for a timed word ρ with $i \in \text{dom}(\rho)$, $\rho, i \models \phi_{seq}$ iff $\rho, i \in \text{Time}(L(A_{seq}))$. For any NFA $N = (St, \Sigma, i, Fin, \Delta)$, $q \in Q$, $F' \subseteq Q$, let $N[q, F'] = (St, \Sigma, q, F', \Delta)$. For brevity, we denote $N[q, \{q'\}]$ as $N[q, q']$. We denote by $\text{Rev}(N)$, the NFA N' that accepts the reverse of $L(N)$. The right/left concatenation of $a \in \Sigma$ with $L(N)$ is denoted $N \cdot a$ and $a \cdot N$ respectively.

Lemma 3. *We can construct a PnEMTL formula ϕ_{seq} with arity $\leq |I_\nu|^2$ and size $\mathcal{O}(|A_{seq}|^{|seq|})$ containing intervals from I_ν s.t. $\rho, i \models \phi_{seq}$ iff $\rho, i \in \text{Time}(L(A_{seq}))$.*

Proof. Let $seq = I_1 I_2 \dots I_n$, and $I_j = \text{anch}$ for some $1 \leq j \leq n$. Let $\Gamma = 2^\Sigma$ and $Qseq = T_1 T_2 \dots T_n$ be a sequence of transitions of A_{seq} where for any $1 \leq i \leq n$,

$T_i = p_{i-1} \xrightarrow{S'_i} q_i$, $S'_i = S_i \cup \{I_i\}$, $S_i \subseteq \Sigma$, $p_{i-1} \in Q \times \{i-1\}$, $q_i \in Q \times \{i\}$. Let $q_0 = (\text{init}, 1)$. We define R_{Qseq} as set of accepting runs containing transitions $T_1 T_2 \dots T_n$. Hence the runs in R_{Qseq} are of the following form:

$T_{0,1} T_{0,2} \dots T_{0,m_0} T_1 T_{1,1} \dots T_{1,m_1} T_2 \dots T_{n-1,1} T_{n-1,2} \dots T_n T_{n,1} \dots T_{n+1}$ where the source of the transition $T_{0,1}$ is q_0 and the target of the transition T_{n+1} is any accepting state of A_{seq} . Moreover, all the transitions $T_{i,j}$ for $0 \leq$

$i \leq n$, $1 \leq j \leq n_i$ are of the form $(p' \xrightarrow{S_{i,j}} q')$ where $S_{i,j} \subseteq \Sigma$ and $p', q' \in Q_{i+1}$. Hence, only $T_1, T_2, \dots, T_n$ are labelled by any interval from I_ν . Moreover, only on these transitions the counter (second element of the state) increments. Let $A_i = (Q_i, 2^\Sigma, q_{i-1}, \{p_{i-1}\}, \delta_{seq}) \equiv A_{seq}[q_{i-1}, p_{i-1}]$ for $1 \leq i \leq n$ and $A_{n+1} = (Q_{n+1}, 2^\Sigma, q_n, F_{seq}, \delta_{seq}) \equiv A_{seq}[q_n, F]$. Let $\mathcal{W}_{Qseq}$ be set of words associated with any run in R_{Qseq} . In other words, any word w in $\mathcal{W}_{Qseq}$ admits an accepting run on A which starts from q_0 reads letters without intervals (i.e., symbols of the form $S \subseteq \Sigma$) ends up at p_0 , reads S'_1 , ends up at q_1 reads letters without intervals, ends up and p_1 , reads S'_2 and so on. Refer Fig. 2 for illustration. Hence, $w \in \mathcal{W}_{Qseq}$ if and only if $w \in L(A_1) \cdot S'_1 \cdot L(A_2) \cdot S'_2 \cdot \dots \cdot L(A_n) \cdot S'_n \cdot L(A_{n+1})$.

Let $A'_k = S_{k-1} \cdot A_k \cdot S_k$ for $1 \leq k \leq n+1$, with $S_0 = S_{n+1} = \epsilon^2$. Let $\rho = (b_1, \tau_1) \dots (b_m, \tau_m)$ be a timed word over Γ . Then $\rho, i_j \in \text{Time}(W_{Qseq})$ iff $\exists \ 0 \leq i_1 \leq i_2 \leq \dots \leq i_{j-1} \leq i_j \leq i_{j+1} \leq \dots \leq i_n \leq m$ s.t. $\bigwedge_{k=1}^{j-1} [(\tau_{i_k} - \tau_{i_j} \in I_k) \wedge \text{Seg}^-(\rho, i_{k+1}, i_k, \Gamma) \in L(\text{Rev}(A'_k))] \wedge \bigwedge_{k=j}^n [(\tau_{i_k} - \tau_{i_j} \in I_k) \wedge \text{Seg}^+(\rho, i_k, i_{k+1}, \Gamma) \in L(A'_k)]$, where $i_0 = 0$ and $i_{n+1} = m$. Hence, by semantics of $\mathcal{F}^k$ and $\mathcal{P}^k$ modalities, $\rho, i \in \text{Time}(W_{Qseq})$ if and only if $\rho, i \models \phi_{qseq}$ where $\phi_{qseq} = \mathcal{P}_{I_{j-1}, \dots, I_1}^j (\text{Rev}(A'_1), \dots, \text{Rev}(A'_j))(\Gamma) \wedge \mathcal{F}_{I_{j+1}, \dots, I_n}^{n-j} (A'_{j+1}, \dots, A'_{n+1})(\Gamma)$. Let **State-seq** be set of all possible sequences of the form **Qseq**. As A_{seq} accepts only words which has exactly n time restricted points, the number of possible sequences of the form **Qseq** is bounded by $|Q|^n$. Hence any word $\rho, i \in \text{Time}(L(A_{seq}))$ iff $\rho, i \models \phi_{seq}$ where $\phi_{seq} = \bigvee_{qseq \in \text{State-seq}} \phi_{qseq}$. Disjuncting over all possible sequences $seq \in \mathcal{T}(I_\nu)$ we get formula ϕ and the following lemma.

Lemma 4. *Let $L(A)$ be the language of I_ν -interval words definable by a NFA A . We can construct a PnEMTL formula ϕ s.t. $\rho, i \models \phi$ iff $\rho, i \in \text{Time}(L(A))$.*

Note that, if ψ is a simple 1-TPTL formula with intervals in $\mathcal{I}_\nu$, then the equivalent PnEMTL formula ϕ constructed above contains only interval in $\text{CL}(\mathcal{I}_\nu)$. Hence, we have the following theorem.

Theorem 4. *For a simple non-adjacent 1-TPTL formula ψ containing intervals from $\mathcal{I}_\nu$, we can construct a non-adjacent PnEMTL formula ϕ , s.t. for any valuation v , $\rho, i, v \models \psi$ iff $\rho, i \models \phi$ where, $|\phi| = O(2^{\text{Poly}(|\psi|)})$ and arity of ϕ is $O(|\mathcal{I}_\nu|^2)$.*

This is a consequence of Theorem 3, Lemma 2 and Lemma 4. A formal proof appears in the full version. For the complexity: The size LTL formula α constructed from ψ (in 1a)) is linear in ψ . The translation from LTL formula α to NFA A has a complexity $\mathcal{O}(2^{|\alpha|}) = \mathcal{O}(2^{|\psi|})$. Let $I_\mu = \text{CL}(\mathcal{I}_\nu)$. Hence, $|I_\mu| = \mathcal{O}(|\mathcal{I}_\nu|^2)$. The size of A_{seq} is $\mathcal{O}(|seq| \times 2^{(|\psi|)}) = \mathcal{O}(2^{\text{Poly}(|\psi|)})$ as $|seq| \leq 2 \times |I_\mu| = \mathcal{O}(|\mathcal{I}_\nu|^2) = \mathcal{O}(|\psi|^2)$. Next, $|\phi_{seq}| = \mathcal{O}(|A_{seq}|^{|seq|}) = \mathcal{O}(2^{\text{Poly}(|\psi|)})$. $|T(\mathcal{I}_\nu)| = \mathcal{O}(2^{\text{Poly}(n)})$. Hence, $|\phi| = \mathcal{O}(2^{\text{Poly}(n, |\psi|)}) = \mathcal{O}(2^{\text{Poly}(|A|)})$. Moreover, the arity of ϕ is also bounded by $2 \times |\text{CL}(\mathcal{I}_\nu)|$. Note that, $|\text{CL}(\mathcal{I}_\nu)| \leq |\mathcal{I}_\nu|^2$. Moreover, $\text{CL}(\mathcal{I}_\nu)$ is non-adjacent iff I_ν is. This result is lifted to a (non-simple) 1-TPTL formula ψ as follows: for each occurrence of a subformula $x.\varphi_i$ in ψ , introduce a new propositional variable a_i and replace $x.\varphi_i$ with a_i . After replacing all such, we are left with the outermost freeze quantifier. Conjoin $\bigwedge_{i=1}^m \mathcal{G}^w(a_i \leftrightarrow x.\varphi_i)$ to the replaced formula obtaining a simple 1-TPTL formula ψ' , equisatisfiable to ψ . Apply the procedure above to each of the $m+1$ conjuncts of ψ' resulting in $m+1$ equivalent non-adjacent PnEMTL formulae φ'_i . The conjunction of φ'_i is the non-adjacent PnEMTL formula equisatisfiable with ψ , giving Theorem 5.

² We A'_k instead of A_k in the formulae below due to the strict inequalities in the semantics of PnEMTL modalities.

Theorem 5. *Any non-adjacent 1-TPTL formula ψ with intervals in $\mathcal{I}_\nu$, can be reduced to a non-adjacent PnEMTL, ϕ , with $|\phi| = 2^{\text{Poly}(|\psi|)}$ and arity of $\phi = O(|\mathcal{I}_\nu|^2)$ such that ψ is satisfiable if and only if ϕ is.*

6 Satisfiability Checking for Non-adjacent PnEMTL

Theorem 6. *Satisfiability Checking for non-adjacent PnEMTL and non-adjacent 1-TPTL are decidable with EXPSPACE complete complexity.*

The proof is via a satisfiability preserving reduction to logic $\text{EMITL}_{(0,\infty)}$ resulting in a formula whose size is at most exponential in the size of the input non-adjacent PnEMTL formula. Satisfiability checking for $\text{EMITL}_{0,\infty}$ is PSPACE complete [11]. This along with our construction implies an EXPSPACE decision procedure for satisfiability checking of non-adjacent PnEMTL. The EXPSPACE lower bound follows from the EXPSPACE hardness of sublogic MITL. The same complexity also applies to non-adjacent 1-TPTL, using the reduction in the previous section. We now describe the technicalities associated with our reduction. We use the technique of equisatisfiability modulo oversampling [12, 16]. Let Σ and OVS be disjoint set of propositions. Given any timed word ρ over Σ , we say that a word ρ' over $\Sigma \cup \text{OVS}$ is an oversampling of ρ if $|\rho| \leq |\rho'|$ and when we delete the symbols in OVS from ρ' we get back ρ . Intuitively, OVS are set of propositions which are used to label oversampling points only. Informally, a formulae α is equisatisfiable modulo oversampling to formulae β if and only if for every timed word ρ excepted by β there exists an oversampling of ρ accepted by α and, for every timed word ρ' accepted by α its projection is accepted by β . Note that when $|\rho'| > |\rho|$, ρ' will have some time points where no proposition from Σ is true. These new points are called oversampling points. Moreover, we say that any point $i' \in \text{dom}(\rho')$ is an old point of ρ' corresponding to i iff i' is the i^{th} point of ρ' when we remove all the oversampling points. For the rest of this section, let ϕ be a non-adjacent PnEMTL formula over Σ . We break down the construction of an $\text{EMITL}_{0,\infty}$ formula ψ as follows.

- 1) Add oversampling points at every integer timestamp using φ_{ovs} below,
- 2) Flatten the PnEMTL modalities to get rid of nested automata modalities, obtaining an equisatisfiable formula ϕ_{flat} ,
- 3) With the help of oversampling points, assert the properties expressed by PnEMTL subformulae ϕ_i of ϕ_{flat} using only EMITL formulae,
- 4) Get rid of bounded intervals with non-zero lower bound, getting the required $\text{EMITL}_{0,\infty}$ formula ψ_i . Replace ϕ_i with ψ_i in ϕ_{flat} getting ψ .

Let $\text{Last} = \mathcal{G}\perp$ and $\text{LastTS} = \mathcal{G}\perp \vee (\perp \text{U}_{(0,\infty)} \top)$. Last is true only at the last point of any timed word. Similarly, LastTS , is true at a point i if there is no next point $i + 1$ with the same timestamp τ_i . Let cmax be the maximum constant used in the intervals appearing in ϕ .

1) Behaviour of Oversampling Points. We oversample timed words over Σ by adding new points where only propositions from Int holds, where $\text{Int} \cap$

$\Sigma = \emptyset$. Given a timed word ρ over Σ , consider an extension of ρ called ρ' , by extending the alphabet Σ of ρ to $\Sigma' = \Sigma \cup \text{Int}$. Compared to ρ , ρ' has extra points called *oversampling* points, where $\neg \bigvee \Sigma$ (and $\bigvee \text{Int}$) hold. These extra points are added at all integer timestamps, in such a way that if ρ already has points with integer time stamps, then the oversampled point with the same time stamp appears last among all points with the same time stamp in ρ' . We will make use of these oversampling points to reduce the PnEMTL modalities into EMITL_{0,∞}. These oversampling points are labelled with a modulo counter $\text{Int} = \{\text{int}_0, \text{int}_1, \dots, \text{int}_{\text{cmax}-1}\}$. The counter is initialized to be 0 at the first oversampled point with timestamp 0 and is incremented, modulo cmax , after exactly one time unit till the last point of ρ . Let $i \oplus j = (i + j) \% \text{cmax}$. The oversampled behaviours are expressed using the formula φ_{ovs} : $\{\neg F_{(0,1)} \bigvee \text{Int} \wedge F_{[0,1)} \text{int}_0\} \wedge \{ \bigwedge_{i=0}^{\text{cmax}-1} \mathcal{G}^w \{ (\text{int}_i \wedge F(\bigvee \Sigma)) \rightarrow (\neg F_{(0,1)}(\bigvee \text{Int}) \wedge F_{(0,1)}(\text{int}_{i \oplus 1} \wedge (\neg \bigvee \Sigma) \wedge \text{LastTS})) \}$. to an extension ρ' given by $\text{ext}(\rho) = \rho'$ iff (i) ρ can be obtained from ρ' by deleting oversampling points and (ii) $\rho' \models \varphi_{\text{ovs}}$. Map ext is well defined as for any ρ , $\rho' = \text{ext}(\rho)$ if and only if ρ' can be constructed from ρ by appending oversampling points at integer timestamps and labelling k^{th} such oversampling point (appearing at time $k-1$) with $\text{int}_{k \% \text{cmax}}$.

2) Flattening. Next, we flatten ϕ to eliminate the nested $\mathcal{F}_{l_1, \dots, l_k}^k$ and $\mathcal{P}_{l_1, \dots, l_k}^k$ modalities while preserving satisfiability. Flattening is well studied [11, 12, 16, 19]. The idea is to associate a fresh witness variable b_i to each subformula ϕ_i which needs to be flattened. This is achieved using the *temporal definition* $T_i = \mathcal{G}^w((\bigvee \Sigma \wedge \phi_i) \leftrightarrow b_i)$ and replacing ϕ_i with b_i in ϕ , $\phi'_i = \phi[b_i / \phi_i]$, where $\mathcal{G}^w$ is the weaker form of $\mathcal{G}$ asserting at the current point and strict future. Then, $\phi'_i = \phi'_i \wedge T_i \wedge \bigvee \Sigma$ is equisatisfiable to ϕ . Repeating this across all subformulae of ϕ , we obtain $\phi_{\text{flat}} = \phi_t \wedge T$ over the alphabet $\Sigma' = \Sigma \cup W$, where W is the set of the witness variables, $T = \bigwedge_i T_i$, ϕ_t is a propositional logic formula over W . Each T_i is of the form $\mathcal{G}^w(b_i \leftrightarrow (\phi_f \wedge \bigvee \Sigma))$ where $\phi_f = \mathcal{F}_{l_1, \dots, l_n}^n(A_1, \dots, A_{n+1})(S)$ (or uses $\mathcal{P}_{l_1, \dots, l_n}^n$) and $S \subseteq \Sigma'$. For example, consider the formula $\phi = \mathcal{F}_{(0,1)(2,3)}^2(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)(\{\phi_1, \phi_2\})$, where $\phi_1 = \mathcal{P}_{(0,2)(3,4)}^2(A_4, A_5, A_6)(\Sigma)$, $\phi_2 = \mathcal{P}_{(1,2)(4,5)}^2(A_7, A_8, A_9)(\Sigma)$. Replacing the ϕ_1, ϕ_2 modality with witness propositions b_1, b_2 , respectively, we get $\phi_t = \mathcal{F}_{(0,1)(2,3)}^2(A_1, A_2, A_3)(\{b_1, b_2\}) \wedge T$, where $T = \mathcal{G}^w(b_1 \leftrightarrow (\bigvee \Sigma \wedge \phi_1)) \wedge \mathcal{G}^w(b_2 \leftrightarrow (\bigvee \Sigma \wedge \phi_2))$, A_1, A_2, A_3 are automata constructed from $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, respectively, by replacing ϕ_1 by b_1 and ϕ_2 by b_2 in the labels of their transitions. Hence, $\phi_{\text{flat}} = \phi_t \wedge T$ is obtained by flattening the $\mathcal{F}_{l_1, \dots, l_k}^k, \mathcal{P}_{l_1, \dots, l_k}^k$ modalities.

3) Obtaining equisatisfiable EMITL formula ψ_f for the PnEMTL formula ϕ_f in each $T_i = \mathcal{G}^w(b_i \leftrightarrow (\phi_f \wedge \bigvee \Sigma))$. The next step is to replace all the PnEMTL formulae occurring in temporal definitions T_i . We use oversampling to construct the formula ψ_f : for a timed word ρ over Σ , $i \in \text{dom}(\rho)$, there is an extension $\rho' = \text{ext}(\rho)$ over an extended alphabet Σ' , and a point $i' \in \text{dom}(\rho')$ which is an old point corresponding to i such that $\rho', i' \models \psi_f$ iff $\rho, i \models \phi_f$. Consider $\phi_f = \mathcal{F}_{l_1, \dots, l_n}^n(A_1, \dots, A_{n+1})(S)$ where $S \subseteq \Sigma'$. Wlg, we assume:

- **[Assumption1]**: $\inf(l_1) \leq \inf(l_2) \leq \dots \leq \inf(l_n)$ and $\sup(l_1) \leq \dots \leq \sup(l_n)$. This is wlog, since the check for A_{j+1} cannot start before the check of A_j in case of $\mathcal{F}_{l_1, \dots, l_n}^n$ modality (and vice-versa for $\mathcal{P}_{l_1, \dots, l_n}^n$ modality) for any $1 \leq j \leq n$.
- **[Assumption 2]**: Intervals $l_1, \dots, l_{n-1}$ are bounded intervals. Interval l_n may or may not be bounded. This is also wlog³.

Let $\rho = (a_1, \tau_1) \dots (a_n, \tau_n) \in T\Sigma^*$, $i \in \text{dom}(\rho)$. Let $\rho' = \text{ext}(\rho)$ be defined by $(b_1, \tau'_1) \dots (b_m, \tau'_m)$ with $m \geq n$, and each τ'_i is either a new integer timestamp not among $\{\tau_1, \dots, \tau_n\}$ or is some τ_j . Let i' be an old point in ρ' corresponding to i . Let $i'_0 = i'$ and $i'_{n+1} = |\rho'|$. $\rho, i \models \phi_f$ iff $\text{cond} \equiv \exists i' \leq i'_1 \leq \dots \leq i'_{n+1} \bigwedge_{g=1}^n (\tau'_{i'_g} - \tau'_{i'_{g-1}} \in l_g \wedge \rho', i'_g \models \bigvee \Sigma \wedge \text{Seg}^+(\rho', i'_{g-1}, i'_g, S') \in L(A'_g)) \wedge$

$\text{Seg}^+(\rho', i'_n, i'_{n+1}, S') \in L(A'_{n+1})$ where for any $1 \leq j \leq n+1$, A'_j is the automata built from A_j by adding self loop on $\neg \bigvee \Sigma$ (oversampling points) and $S' = S \cup \{\neg \bigvee \Sigma\}$. This self loop makes sure that A'_j ignores(or skips) all the oversampling points while checking for A_j . Hence, A'_j allows arbitrary interleaving of oversampling points while checking for A_j . Hence, for any $g, h \in \text{dom}(\rho)$ with g', h' being old action points of ρ' corresponding to g, h , respectively, $\text{Seg}^s(\rho, g, h, S) \in L(A_i)$ iff $\text{Seg}^s(\rho', g', h', S \cup \{\neg \bigvee \Sigma\}) \in L(A'_i)$ for $s \in \{+, -\}$. Note that the question, “ $\rho, i \models \phi_f$?”, is now reduced to checking cond on ρ' .

Checking the conditions for $\rho, i \models \phi_f$. Let $I_g = \langle l_g, u_g \rangle$ for any $1 \leq g \leq n$ (Here, $\langle \rangle$ denotes half-open, closed, or open). We discuss only the case where $\{I_1, \dots, I_n\}$ are pairwise disjoint and $\inf(I_1) \neq 0$ in $\phi_f = \mathcal{F}_{l_1, \dots, l_n}^n(A_1, \dots, A_{n+1})(S)$. The case of overlapping intervals can be found in the full version. The disjoint interval assumption along with [Assumption 1] implies that for any $1 \leq g \leq n$, $u_{g-1} < l_g$. By construction of ρ' , between i'_{g-1} and i'_g , we have an oversampling point k_g . The point k_g is guaranteed to exist between i'_{g-1} and i'_g , since these two points lie within two distinct non-overlapping, non-adjacent intervals l_{g-1} and l_g from i' . Hence their timestamps have different integral parts, and there is always a uniquely labelled oversampling point k_g with timestamp $\lceil \tau'_{i'_{g-1}} \rceil$ between i'_{g-1} and i'_g for all $1 \leq g \leq n$. Let for all $1 \leq g \leq n+1$, $A'_g = (Q_g, 2^S, \text{init}_g, F_g, \delta'_g)$. Let the unique label for k_g be int_{j_g} . For any $1 \leq g \leq n$, we assert that the behaviour of propositions in S between points i'_{g-1} and i'_g (of ρ') should be accepted by A'_g . This is done by splitting the run at the oversampling point k_g (labelled as int_{j_g}) with timestamp $\tau'_{k_g} = \lceil \tau'_{i'_{g-1}} \rceil$, $i'_{g-1} < k_g < i'_g$.

- (1) Concretely, checking for cond , for each $1 \leq g \leq n$, we start at i'_{g-1} in ρ' , from the initial state init_g of A_g , and move to the state (say q_g) that is reached at the closest oversampling point k_g . Note that we use only A_g (we disallow the $\neg \bigvee \Sigma$ self loops) to move to the closest oversampling point.

³ Unbounded intervals can be eliminated using $\mathcal{F}_{l_1, l_2, \dots, l_{k-2}, [l_1, \infty)[l_2, \infty)}^k(A_1, \dots, A_{k+1}) \equiv \mathcal{F}_{l_1, l_2, \dots, l_{k-2}, [l_1, \text{cmax}][l_2, \infty)}^k(A_1, \dots, A_{k+1}) \vee \mathcal{F}_{l_1, l_2, \dots, l_{k-2}, [l_2, \infty)}^{k-1}(A_1, \dots, A_{k-1}, A_k \cdot A_{k+1})$.

- (2) Reaching q_g from $init_g$ we have read a behaviour between i'_{g-1} and k_g ; this must to the full behaviour, and hence must also be accepted by A'_g (we use A'_g instead of A_g to ignore the oversampling points that could be encountered while checking the latter part). Towards this, we guess a point i'_g which is within interval I_g from i' , such that, the automaton A'_g starts from state q_g reading int_{k_g} and reaches a final state in F_g at point i'_g . Then indeed, the behaviour of propositions from S between i'_{g-1} and i'_g respect A'_g , and also $\tau'_{i'_g} - \tau'_{i'} \in I_g$.
- (1) amounts to $\text{Seg}^+(\rho', i'_{g-1}, k_g, S) \in L(A_g[init_g, q_g]) \cdot int_{j_g}$. This is defined by the formula $\psi_{g-1, int_{j_g}, q_g}^+$ which asserts $A_{g+1}[init_g, q_g] \cdot int_{j_g}$ from point i'_{g-1} to the next nearest oversampling point k_g where int_{j_g} holds.
- (2) amounts to checking from point i , within interval I_g in its future, the existence of a point i'_g such that $\text{Seg}^-(\rho', i'_g, k_g, S) \in L(\text{Rev}(int_{j_g} \cdot A'_g[q_g, F_g]))$. This is defined by the formula $\varphi_{g, int_{j_g}, q_g}^-$ which asserts $\text{Rev}(int_{j_g} \cdot A'_g[q_g, F_g])$, from point i'_g to an oversampling point k_g which is the earliest oversampling point s.t. $i'_{g-1} < k_g < i'_g$. For cond , we define the formula
- $$\psi = F_{[0,1)} int_{j_0} \wedge \bigvee_{g=1}^n [\psi_{g-1, int_{j_g}, q_g}^+ \wedge \psi_{g, int_{j_g}, q_g}^-] \wedge \psi_n^+.$$

- For $1 \leq g \leq n$, $\psi_{g-1, int_{j_g}, q_g}^+ = F_{I_g}(\bigvee \Sigma \wedge \mathcal{F}(A_g[init_g, q_g] \cdot \{int_{j_g}\})(S \cup \{int_{j_g}\}))$,
- $\psi_n^+ = F_{I_n}(\bigvee \Sigma \wedge \mathcal{F}(A_{n+1} \cdot \{\text{Last}\})(S \cup \{\text{Last}\}))$, and
- For $1 \leq g \leq n$, $\psi_{g, int_{j_g}, q_g}^- = F_{I_g}(\bigvee \Sigma \wedge \mathcal{P}(\text{Rev}(int_{j_g} \cdot A_g[q_g, F_g]))(S \cup \{int_{j_g}\}))$.

Note that there is a unique point between i'_{g-1} and i'_g labelled int_{j_g} . This is because, $\tau'_{i'_g} - \tau'_{i'_{g-1}} < \tau'_{i'_g} - \tau'_{i'} \leq \text{cmax}$. Hence, we can ensure that the meeting point for the check (1) and (2) is indeed characterized by a unique label. Note that there is exactly one point labeled int_y from any point within future cmax or past cmax time units (by φ_{ovs}). This is the reason we used the counter modulo cmax to label the oversampling points. We encourage the readers to see the Fig. 3. The full EMITL formula ψ_f , is obtained by disjuncting over all n length sequences of states reachable at oversampling points k_g between i'_{g-1} and i'_g , and all possible values of the unique label $int_{j_g} \in \text{Int}$ holding at point k_g .

4) Converting the EMITL to $\text{EMITL}_{0,\infty}$: We use the reduction from EMITL to equivalent $\text{EMITL}_{0,\infty}$ formula [16]. In ψ_f , only the F operators are timed with intervals of the form $\langle l, u \rangle$ where $l > 0$ and $u \neq \infty$, but the $\mathcal{F}_1$ and $\mathcal{P}_1$ modalities are untimed. We can reduce these time intervals into purely lower bound $(\langle l, \infty \rangle)$ or upper bound $(\langle 0, u \rangle)$ constraints using these oversampling points, preserving satisfiability, by reduction showed in [16] Chap. 5 lemma 5.5.2 Page 90–91.

The above 4 step construction shows that (i) the equisatisfiable $\text{EMITL}_{0,\infty}$ formula ψ is of the size $(\mathcal{O}(|\phi|^{Poly(n)}))$ where, n is the arity ϕ . (ii) For a non-adjacent 1-TPTL formula γ , applying the reduction in Sect. 5 yields ϕ of size $\mathcal{O}(2^{Poly(|\gamma|)})$ and, arity of $\phi = \mathcal{O}(|\gamma|^2)$. Also, after applying the reduction of Sect. 6 by plugging the value of $|\phi|$ from and its arity from (ii) in (i), we get the $\text{EMITL}_{0,\infty}$ formula ψ of size $\mathcal{O}(2^{Poly(|\gamma|) * Poly(n)}) = \mathcal{O}(2^{Poly(|\gamma|)})$.

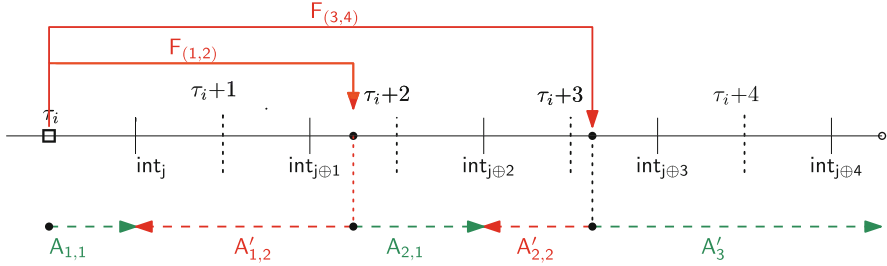


Fig. 3. Figure showing elimination of $\mathcal{F}^2$ modality from temporal definition of the form $\mathcal{G}^w(b \leftrightarrow \mathcal{F}_{(1,2),(3,4)}^2(A_1, A_2, A_3)(\Sigma'))$. This is done by (i) checking for the first part of A_1 , $A_{1,1}$, from present point to the next oversampling point at timestamp $\lceil \tau_i \rceil$, labelled, int_j , (ii) jumping to a non-deterministically chosen point within (1,2) and asserting the remaining part of A_1 skipping oversampling points, $A'_{1,2}$, in reverse till int_j , (iii) Following the steps similar to (i) and (ii) for checking A_2 but starting the check of first part of A_2 from the point chosen in (ii).

7 Conclusion

We generalized the notion of non-punctuality to non-adjacency in TPTL. We proved that satisfiability checking for non-adjacent 1-variable fragment of TPTL is EXPSpace Complete. This gives us a strictly more expressive logic than MITL while retaining its satisfaction complexity. An interesting open problem is to compare the expressive power of non-adjacent 1-TPTL with that of MITL with Pnueli modalities (and hence Q2MLO) of [10]. We also leave open the satisfiability checking problem for non-adjacent TPTL with multiple variables.

References

1. Alur, R., Feder, T., Henzinger, T.: The benefits of relaxing punctuality. J. ACM **43**(1), 116–146 (1996)
2. Alur, R., Henzinger, T.A.: Real-time logics: complexity and expressiveness. Inf. Comput. **104**(1), 35–77 (1993)
3. Alur, R., Henzinger, T.A.: A really temporal logic. J. ACM **41**(1), 181–203 (1994)
4. Bouyer, P., Chevalier, F., Markey, N.: On the expressiveness of TPTL and MTL. In: Sarukkai, S., Sen, S. (eds.) FSTTCS 2005. LNCS, vol. 3821, pp. 432–443. Springer, Heidelberg (2005). https://doi.org/10.1007/11590156_35
5. Ferrère, T.: The compound interest in relaxing punctuality. In: Havelund, K., Peleska, J., Roscoe, B., de Vink, E. (eds.) FM 2018. LNCS, vol. 10951, pp. 147–164. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95582-7_9
6. Gustin, P., Oddoux, D.: LTL with past and two-way very-weak alternating automata. In: Rován, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 439–448. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45138-9_38
7. Haase, C., Ouaknine, J., Worrell, J.: On process-algebraic extensions of metric temporal logic. In: Roscoe, A.W., Jones, C.B., Wood, K.R. (eds.) Reflections on the Work of C.A.R. Hoare, pp. 283–300. Springer, London (2010). https://doi.org/10.1007/978-1-84882-912-1_13

8. Henzinger, T.A., Raskin, J.-F., Schobbens, P.-Y.: The regular real-time languages. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 580–591. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0055086>
9. Hirshfeld, Y., Rabinovich, A.: An expressive temporal logic for real time. In: Kráľovič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 492–504. Springer, Heidelberg (2006). https://doi.org/10.1007/11821069_43
10. Hirshfeld, Y., Rabinovich, A.: Expressiveness of metric modalities for continuous time. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 211–220. Springer, Heidelberg (2006). https://doi.org/10.1007/11753728_23
11. Ho, H.-M.: Revisiting timed logics with automata modalities. In: Ozay, N., Prabhakar, P. (eds.) Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2019, Montreal, QC, Canada, 16–18 April 2019, pp. 67–76. ACM (2019)
12. Madnani, K., Krishna, S.N., Pandya, P.K.: Partially punctual metric temporal logic is decidable. In: TIME, pp. 174–183 (2014)
13. Krishna, S.N., Madnani, K., Mazo Jr., M., Pandya, P.K.: Generalizing non-punctuality for timed temporal logic with freeze quantifiers. CoRR, abs/2105.09534 (2021)
14. Krishna, S.N., Madnani, K., Pandya, P.K.: Making metric temporal logic rational. In: Larsen, K.G., Bodlaender, H.L., Raskin, J.-F. (eds.) 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, 21–25 August 2017 - Aalborg, Denmark. LIPIcs, vol. 83, pp. 77:1–77:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017)
15. Krishna, S.N., Madnani, K., Pandya, P.K.: Logics meet 1-clock alternating timed automata. In: Schewe, S., Zhang, L. (eds.) 29th International Conference on Concurrency Theory, CONCUR 2018, 4–7 September 2018, Beijing, China. LIPIcs, vol. 118, pp. 39:1–39:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)
16. Madnani, K.N.: On decidable extensions of metric temporal logic. Ph.D. thesis, Indian Institute of Technology Bombay, Mumbai, India (2019)
17. Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic. In: LICS, pp. 188–197 (2005)
18. Pandya, P.K., Shah, S.S.: On expressive powers of timed logics: comparing boundedness, non-punctuality, and deterministic freezing. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 60–75. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23217-6_5
19. Prabhakar, P., D’Souza, D.: On the expressiveness of MTL with past operators. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 322–336. Springer, Heidelberg (2006). https://doi.org/10.1007/11867340_23
20. Rabinovich, A.: Complexity of metric temporal logic with counting and pnueli modalities. In: FORMATS, pp. 93–108 (2008)
21. Rabinovich, A.: Complexity of metric temporal logics with counting and the pnueli modalities. Theor. Comput. Sci. **411**(22–24), 2331–2342 (2010)
22. Raskin, J.F.: Logics, automata and classical theories for deciding real time. Ph.D. thesis, Université de Namur (1999)
23. Wilke, T.: 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, 19–23 September, Proceedings, pp. 694–715 (1994)