

Revisiting the Expressiveness of Metric Temporal Logic

Mohammed Aristide Foughali 

Université Paris Cité, CNRS, IRIF, F-75013, Paris, France

Abstract

The expressiveness of Metric Temporal Logic (MTL) has been extensively studied throughout the last two decades. In particular, it has been shown that the *interval-based* semantics of MTL is strictly more expressive than the *pointwise* one. These results may suggest that enabling the evaluation of formulae at arbitrary time points *instead of* positions of timed events increases the expressive power of MTL. In this paper, we formally argue otherwise. We demonstrate that under standard models of finite or non-Zeno infinite (action-based) timed executions, the interval-based and the pointwise semantics are incomparable. We then propose a new *mixed* semantics that embeds both the pointwise and the interval-based ones.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Timed logics, Metric temporal logic

Digital Object Identifier 10.4230/OASICS.TIME.2026.1

Related Version *Full Version*: <https://arxiv.org/abs/2603.15379> [17]

Funding This work received partial support from the ANR project ERVoRS (ANR-25-CE48-6220).

1 Introduction

Metric Temporal Logic (MTL) [22] is a popular *timed* extension of the Linear-time Temporal Logic LTL [28]. The importance of MTL motivated a large body of research investigating its expressive power [14, 9, 10, 20, 27, 19]. In this paper, we focus on the expressiveness of the *pointwise* semantics of MTL as compared to its *interval-based* one. In order to present the existing results on such comparison, we first (informally) introduce the model of *timed words*.

A *timed event* is a couple containing a letter, representing an *action*, and a real number corresponding to its *timestamp*, i.e., the global time at which it occurred. A timed word is a finite or infinite sequence of timed events. Following the least constrained model of timed words [2, 25, 26], which we refer to hereafter as the *general model*, a (finite or infinite) timed word satisfies time monotonicity, i.e., timestamps are non-decreasing, and an infinite timed word obeys time divergence (also called non-Zenoness), that is global time does not converge towards some bounded value. For instance, over the alphabet $\{a, b, c\}$, the sequence $(a, 0)(b, 1)(c, 1)(a, 5.3)$ is a finite timed word whereas $(a, 0)(b, 1)(c, 0.8)$ is not (time monotonicity is violated). Similarly, the infinite timed word $(a, \frac{1}{2})(b, \frac{3}{4})(b, \frac{7}{8})(c, \frac{15}{16}) \dots$, i.e., where the difference between two consecutive timestamps is halved at every next action, is excluded (global time converges towards 1).

Under the pointwise semantics, formulae are evaluated at the *positions* of timed events within the timed word, whereas the interval-based semantics is given over continuous intervals within a *timed signal* [27]. To enable the comparison between both semantics, D’Souza & Prabhakar [14] interpret the interval-based semantics over timed words (instead of timed signals), allowing the evaluation of formulae at arbitrary time points (including those not corresponding to a timestamp). They demonstrate accordingly that the interval-based semantics



© Mohammed Aristide Foughali;

licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Temporal Representation and Reasoning (TIME 2026).

Editors: AndreA Orlandini and Sophie Pinchinat; Article No. 1; pp. 1:1–1:18

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is strictly more expressive than the pointwise one over finite timed words, using a constrained model where time monotonicity is strict (e.g., the timed word $(a, 0)(b, 1)(c, 1)(a, 5.3)$ used as an example above is excluded by design). We refer to this existing result as **ER**:

ER: The interval-based semantics is strictly more expressive than the pointwise one over strictly monotone finite timed words.

However, we argue that while strict monotonicity of time is reasonable in many cases, it is not justified in general. Indeed, simultaneous interleaving in timed systems, theorised since [13, 3], is of sheer importance when formally reasoning about concurrent behaviour, e.g., in multiprocessor real-time scheduling (see for instance [16, 11, 18]), and distributed applications [24, 15, 23]. There is therefore a need to complete the picture of **ER** and, in case of incomparability of the interval-based and the pointwise semantics, unify them in order to benefit from their respective advantages.

In a remotely related work, Bouyer et al. proved in two seminal papers [9, 10] that TPTL [4] is strictly more expressive than MTL. These two publications [9, 10] are of particular interest to us for the following two reasons:

- They use a clever encoding of timed words into *timed state sequences* to reason over the interval-based semantics. This encoding will inspire our incomparability proofs (contributions **C1**, **C2** and **C3** below).
- While considering the general model of infinite timed words [9, Sect. 2][10, Sect. 2], they make the following side claim [10, Section 1][9, Sect. 1]:

“As side results, we get that MTL is strictly more expressive under the interval-based semantics than under the pointwise one”

We formally show that this claim does not hold (Theorem 3.27).

Contributions. In this paper, we (i) answer the comparability question in the negative under the general model of timed words and (ii) devise a new mixed semantics that remedies such incomparability. More precisely, we provide the following contributions:

- **C1** The interval-based and the pointwise semantics are incomparable under the general model of finite timed words,
- **C2** The interval-based and the pointwise semantics are incomparable under the general model of infinite timed words,
- **C3** The incomparability holds even under reasonable restrictions such as *non-stuttering*,
- **C4** A mixed semantics embedding both the interval-based and the pointwise ones is proposed.

C1, **C2** and **C3** complete the picture of **ER**, showing that it does not hold under the more general (non-strict) monotonicity, regardless of whether timed words are finite or infinite. In addition, the mixed semantics (**C4**) conjugates the expressive powers of the interval-based and the pointwise ones, while embedding both semantics through automatic compilers.

Outline. Section 2 is dedicated to preliminaries, whereas the core of our contributions is detailed in Section 3 (contributions **C1**, **C2** and **C3**) and Section 4 (contribution **C4**). Section 5 wraps up with concluding remarks and future work directions.

2 Preliminaries

2.1 Binary relations and functions

Let X and Y be two non-empty sets.

► **Definition 2.1** (Binary relation). A binary relation $\mathcal{R}$ over X and Y is a subset of the cartesian product $X \times Y$. The inverse relation $\mathcal{R}^{-1}$ of $\mathcal{R}$ is defined as $\mathcal{R}^{-1} = \{(y, x) \in Y \times X \mid (x, y) \in \mathcal{R}\}$.

For any $x \in X$, we let $\mathcal{R}[x] = \{y \in Y \mid (x, y) \in \mathcal{R}\}$ denote the set of all images of x by $\mathcal{R}$. $\mathcal{R}$ is said left total iff every element of X has at least an image by $\mathcal{R}$, i.e., $\forall x \in X : \mathcal{R}[x] \neq \emptyset$, right total iff $\mathcal{R}^{-1}$ is left total, and total if it is both right and left total.

► **Definition 2.2** (Function). A binary relation $\mathcal{R}$ over X and Y is called a function from X to Y iff every element in X has exactly one image in Y , i.e., $\forall x \in X : |\mathcal{R}[x]| = 1$. We use the standard notation for functions, i.e., $\mathcal{R} : X \rightarrow Y$ for “ $\mathcal{R}$ a function from X to Y ” and $\mathcal{R}(x) = y$ (instead of $\mathcal{R}[x] = \{y\}$) for “ y is the image of x by $\mathcal{R}$ ”.

► **Definition 2.3** (Injective, surjective and bijective functions). A function $\mathcal{R}$ is said:

- injective iff all images are distinct: $\forall y \in Y : |\mathcal{R}^{-1}[y]| \leq 1$,
- surjective iff it is right total ($\mathcal{R}^{-1}$ is therefore a total relation),
- bijective iff it is both injective and surjective ($\mathcal{R}^{-1}$ is therefore a bijective function).

2.2 Timed words and timed state sequences

Let s be a sequence over a non-empty set X . We let sequences start at position 0, and $|s|$ denote the length of s defined in the usual way, with $|s| = \omega$ if s is infinite.

2.2.1 Timed words

The definitions provided in this subsection follow the general model of timed words, they can be seen as a generalisation of those given in [9] to both finite and infinite timed words.

► **Definition 2.4** (Time sequence). A time sequence τ is a non-empty sequence over $\mathbb{R}_{\geq 0}$ satisfying the following conditions:

- Monotonicity: $\forall i \in \mathbb{N}, 0 < i < |\tau| : \tau_{i-1} \leq \tau_i$,
- Time divergence: $|\tau| = \omega \Rightarrow (\forall t \in \mathbb{R}_{\geq 0} \exists i \in \mathbb{N} : \tau_i > t)$.

► **Remark 2.5.** In Definition 2.4, time divergence is only required for infinite time sequences. This is because every finite time sequence satisfying monotonicity is a prefix of a time-divergent infinite time sequence [5].

Let Σ be a non-empty, finite set of letters, Σ^+ and Σ^ω denote, respectively, the set of non-empty finite and infinite words over Σ , and $\Sigma^\infty = \Sigma^+ \cup \Sigma^\omega$.

► **Definition 2.6** (Timed word). A timed word is a pair $\rho = (\sigma, \tau)$ with $\sigma \in \Sigma^\infty$ and τ a time sequence such that $|\sigma| = |\tau|$. The length of ρ is $|\rho| = |\sigma|$, and each (σ_i, τ_i) is called a timed event. For some $i \in \mathbb{N}, i < |\rho|$, τ_i is called the timestamp of σ_i .

Accordingly, if $\rho = (\sigma, \tau)$ is finite, say of length $n \in \mathbb{N}_{>0}$ (resp. infinite), $\sigma = \sigma_0 \sigma_1 \dots \sigma_{n-1}$ is a finite word in Σ^+ and $\tau = \tau_0 \tau_1 \dots \tau_{n-1}$ is a finite time sequence (resp. $\sigma = \sigma_0 \sigma_1 \dots$ is an infinite word in Σ^ω and $\tau = \tau_0 \tau_1 \dots$ is an infinite time sequence). The duration of ρ , denoted $\mu(\rho)$, equals the timestamp of the last letter if ρ is finite, i.e., $\mu(\rho) = \tau_{|\rho|-1}$. Otherwise, $\mu(\rho) = \infty$ thanks to time divergence (Definition 2.4). We sometimes use the alternative notation of sequences of timed events (e.g., $(\sigma_0, \tau_0)(\sigma_1, \tau_1) \dots (\sigma_{|\rho|-1}, \tau_{|\rho|-1})$ instead of (σ, τ) for some finite timed word). We denote by ${}^m\rho$, $m < |\rho|$, the prefix of $\rho = (\sigma, \tau)$ up to and including position m , i.e., ${}^m\rho = (\sigma_0, \tau_0) \dots (\sigma_m, \tau_m)$.

► **Remark 2.7.** A timed word $\rho = (\sigma, \tau)$ inherits the properties of the time sequence τ , i.e., time monotonicity and, if ρ is infinite, time divergence (Definition 2.4).

► **Example 2.8.** Let $\Sigma = \{a, b, c\}$. The sequence $\rho_1 = (a, 1)(b, 1.3)(c, 3.5)(b, 3.5)$ is a timed word of length 4 and duration 3.5. The sequence $\rho_2 = (a, 1)(b, 1.3)(c, 1.2)(b, 3.5)$ is not a timed word (violation of time monotonicity). The sequence $\rho_3 = (\sigma_i, \tau_i)$, $i \in \mathbb{N}$ defined next is an infinite timed word: $\sigma_{2i} = a$, $\sigma_{2i+1} = b$; $\tau_0 = 0$ and for all i in $\mathbb{N}_{>0}$: $\tau_i = \tau_{i-1} + 1$ if $i \bmod 3 = 0$ and $\tau_i = \tau_{i-1}$ otherwise. Its prefix ${}^6\rho_3$ equals $(a, 0)(b, 0)(a, 0)(b, 1)(a, 1)(b, 1)(a, 2)$.

We now define some subclasses of timed words. A strongly stutter-free (resp. strictly monotone) timed word is exempt of the behaviour where two occurrences of the same letter (resp. of any two letters) happen at different positions with the same timestamp.

► **Definition 2.9** (Stutter freeness and strict monotonicity). *A timed word $\rho = (\sigma, \tau)$ is said:*

1. stutter free *iff* $\forall i \neq j \in \mathbb{N}, i, j < |\rho| : \tau_i = \tau_j \Rightarrow \sigma_i \neq \sigma_j$.
2. strictly monotone *iff* τ is strictly monotone: $\forall i \in \mathbb{N}, 0 < i < |\tau| : \tau_{i-1} < \tau_i$.

► **Remark 2.10.** While strict monotonicity implies stutter freeness, none of them is related to time-divergence. For instance, the infinite timed word ρ_3 of Example 2.8 is time divergent but not stutter free (and therefore not strictly monotone).

► **Example 2.11.** $(a, 0)(c, 3)$ is strictly monotone (and therefore stutter free), $(a, 0)(b, 1)(a, 1)(c, 3)$ is stutter free, and $(a, 0)(b, 1)(a, 1)(b, 1)(c, 3)$ is not stutter free.

2.2.2 Timed state sequences

The definitions given in this subsection generalise the model of infinite timed state sequences given in [9, 10] to both the finite and infinite cases.

A *non-empty interval* I over $\mathbb{R}_{\geq 0}$ has the form $[l, u]$, $(l, u]$, $[l, u)$ or (l, u) with its left bound $I^l = l \in \mathbb{R}_{\geq 0}$, right bound $I^u = u \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, $I^l \leq I^u$ if I is right- and left-closed (i.e., has the form $[l, u]$) and $I^l < I^u$ otherwise, and ∞ excluded; let $\mathcal{I}$ be the set of all such intervals. An element of $\mathcal{I}$ denotes a non-empty convex subset of $\mathbb{R}_{\geq 0}$ defined uniquely in the usual way. Two intervals $I_1, I_2 \in \mathcal{I}$ are *adjacent* iff $I_1 \cap I_2 = \emptyset$, $I_1 \cup I_2 \in \mathcal{I}$ and $x < y$ for every $x \in I_1$ and $y \in I_2$. An interval I is *punctual* iff it denotes a singleton set, i.e., it has the form $[l, l]$.

► **Definition 2.12** (Interval sequence). *An interval sequence ι is a non-empty sequence over $\mathcal{I}$ satisfying:*

- *Adjacency:* $\forall i \in \mathbb{N}, 0 < i < |\iota| : \iota_{i-1}$ and ι_i are adjacent,
- *Progress:* $|\iota| = \omega \Rightarrow \forall t \in \mathbb{R}_{\geq 0} \exists i \in \mathbb{N} : t \in \iota_i$.

► **Definition 2.13** (Timed state sequence). *Let AP be a finite set of atomic propositions. A timed state sequence over AP is a pair $\kappa = (\eta, \iota)$ where η is a sequence over $\mathcal{P}(AP)$ (the powerset of AP), ι an interval sequence and $|\eta| = |\iota|$.*

Similarly to timed words, given a timed state sequence $\kappa = (\eta, \iota)$, we (i) sometimes use the alternative sequence notation (η_i, ι_i) , $i \in \mathbb{N}$, $i < |\kappa|$ instead of (η, ι) and (ii) define $\mu(\kappa)$, the duration of κ as $\iota_{|\kappa|-1}^u$ if κ is finite (requiring, without loss of generality, $\iota_{|\kappa|-1}$ to be right closed), and ∞ otherwise (thanks to the progress condition, Definition 2.12). Moreover, given t in $\mathbb{R}_{\geq 0}$ (satisfying $t \leq \mu(\kappa)$ if κ is finite), we denote by $\kappa(t)$ the set η_i where $t \in \iota_i$ (such ι_i is uniquely defined thanks to the adjacency property).

► **Example 2.14.** Let $AP = \{p, q\}$. The sequence $(\{p\}, [1, 2])(\{p, q\}, [2, 3])(\{p\}, [3, 3.4])$ is not a timed state sequence (violation of interval adjacency). The sequence $\kappa = (\{p\}, [1, 2])(\{p, q\}, [2, 3])(\{q\}, [3, 3]), (\{p\}, (3, 3.4])$ is a timed state sequence with length 4 and duration 3.4, and we have $\kappa(1.2) = \kappa(3.1) = \{p\}$, $\kappa(2.2) = \{p, q\}$ and $\kappa(3) = \{q\}$.

2.3 Timed words vs. timed state sequences

A timed word is a natural representation of an *action-based* execution of a timed system. For example, $(a, 2)(b, 2)(c, 4.1)$ corresponds to an execution where the system waited for two time units, performed action a then immediately b , then waited for 2.1 time units and finally performed c . A timed state sequence, on the other hand, is suitable for signals: e.g., $(\{p\}, [0, 2])(\{q, r\}, [2, 3])$ represents a signal where p held for two time units (excluding 2), then q and r held for one time unit. While there is a two-way translation between the untimed versions of these models modulo bisimulation [12], this is not the case in the timed setting, see e.g., the discussion on signals vs. timed words in [7] (this heterogeneity led to the proposition of signal-event frameworks as in [8]). We adopt an action-based view for the simple reason that it is the only way to enable any comparison involving the pointwise semantics. We still need timed state sequences as their use to encode timed words [9, 10] inspire our incomparability proofs.

2.4 Metric Temporal Logic (MTL)

Let $\mathcal{I}_q$ be the set of all non-empty intervals over $\mathbb{R}_{\geq 0}$ with rational endpoints, i.e., defined similarly to $\mathcal{I}$ (Section 2.2.2) but with $l \in \mathbb{Q}_{\geq 0}$, $u \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$. In order to define clean syntax and semantics of MTL under both the pointwise and the interval-based semantics, we equate hereafter the sets of atomic propositions AP and letters (actions) Σ and refer to both as Σ . The syntax of MTL is given below with $a \in \Sigma$ and $I \in \mathcal{I}_q$.

$$\phi, \psi ::= a \mid \neg\phi \mid \phi \wedge \psi \mid \phi \mathcal{U}_I \psi$$

where $\mathcal{U}_I$ denotes the *timed Until* operator (note that we use its standard strict version [9, 27, 25]). The syntactic sugar is defined as usual: $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$, $\top \equiv a \vee \neg a$, $\bigcirc_I \phi \equiv (\neg \bigvee_{a \in \Sigma} a) \mathcal{U}_I \phi$ (*next* ϕ), $\Diamond_I \phi \equiv \top \mathcal{U}_I \phi$ (*eventually* ϕ) and $\Box_I \phi \equiv \neg \Diamond_I \neg\phi$ (*globally* ϕ). We will use often the handy formula $\varphi_\Sigma = \bigvee_{a \in \Sigma} a$ (utilised to define the next operator above) hereafter. We omit the subscript I when $I = [0, \infty)$.

Let $\rho = (\sigma, \tau)$ be a timed word and $\kappa = (\eta, \iota)$ a timed state sequence. Following [9, 14], we force τ_0 to equal 0 and ι_0 to be left closed at 0 without loss of generality. We let P and K be, respectively, the set of all such timed words and timed state sequences. We give below the pointwise semantics (over elements of P) and the two versions of the interval-based semantics (one over P as in [14], and one over K following [9]).

2.4.1 Pointwise semantics

Formulae are evaluated over a timed word $\rho = (\sigma, \tau) \in P$ and a position $i \in \mathbb{N}$ with $i < |\rho|$. To avoid unnecessary cluttering of notation, $j < |\rho|$ is assumed in the semantics of $\phi \mathcal{U}_I \psi$.

$$\begin{array}{ll} \rho, i \models_{pw} a & \text{iff } \sigma_i = a \\ \rho, i \models_{pw} \neg\phi & \text{iff } \rho, i \not\models_{pw} \phi \\ \rho, i \models_{pw} \phi \wedge \psi & \text{iff } \rho, i \models_{pw} \phi \text{ and } \rho, i \models_{pw} \psi \\ \rho, i \models_{pw} \phi \mathcal{U}_I \psi & \text{iff } \exists j > i : \tau_j - \tau_i \in I \text{ and } \rho, j \models_{pw} \psi \text{ and } \forall i < k < j : \rho, k \not\models_{pw} \phi \end{array}$$

2.4.2 Interval-based semantics

Over timed words. Formulae are evaluated over a timed word $\rho = (\sigma, \tau) \in P$ and a timepoint $t \in \mathbb{R}_{\geq 0}$ (with $t \leq \mu(\rho)$ if ρ is finite). To avoid heavy notation, $|\rho| \neq \omega \Rightarrow t' \leq \mu(\rho)$ is assumed in the semantics of $\phi \mathcal{U}_I \psi$.

$$\begin{aligned}
\rho, t \models_{itw} a & \quad \text{iff} \quad \exists i < |\rho| : \sigma_i = a \text{ and } \tau_i = t \\
\rho, t \models_{itw} \neg\phi & \quad \text{iff} \quad \rho, t \not\models_{itw} \phi \\
\rho, t \models_{itw} \phi \wedge \psi & \quad \text{iff} \quad \rho, t \models_{itw} \phi \text{ and } \rho, t \models_{itw} \psi \\
\rho, t \models_{itw} \phi \mathcal{U}_I \psi & \quad \text{iff} \quad \exists t' > t : t' - t \in I \text{ and } \rho, t' \models_{itw} \psi \text{ and } \forall t < t'' < t' : \rho, t'' \models_{itw} \phi
\end{aligned}$$

Over timed state sequences. Formulae are interpreted over a timed state sequence $\kappa = (\eta, \iota) \in \mathbf{K}$ and a timepoint $t \in \mathbb{R}_{\geq 0}$ (with $t \leq \mu(\kappa)$ if κ is finite). We give the definition for $\phi = a$ only, the remaining cases can be obtained from $\models_{itw}$ by replacing ρ with κ .

$$\kappa, t \models_{its} a \quad \text{iff} \quad a \in \kappa(t)$$

We say that ρ satisfies ϕ , written $\rho \models_{sem} \phi$ iff $\rho, 0 \models_{sem} \phi$ with $sem \in \{pw, itw\}$ and similarly $\kappa \models_{its} \phi$ iff $\kappa, 0 \models_{its} \phi$. Let $sem \in \{pw, itw, its\}$. We write $\llbracket \phi \rrbracket_{sem}^f$ (resp. $\llbracket \phi \rrbracket_{sem}^{inf}$) to denote the set of all finite (resp. infinite) sequences s such that $s \models_{sem} \phi$, and let $\llbracket \phi \rrbracket_{sem} = \llbracket \phi \rrbracket_{sem}^f \cup \llbracket \phi \rrbracket_{sem}^{inf}$. For instance, $\llbracket \phi \rrbracket_{pw} = \{\rho \in \mathbf{P} \mid \rho \models_{pw} \phi\}$. When $sem \in \{pw, itw\}$, each of $\llbracket \phi \rrbracket_{sem}^f$, $\llbracket \phi \rrbracket_{sem}^{inf}$ and $\llbracket \phi \rrbracket_{sem}$ defines a *timed language*.

► **Example 2.15.** Let $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$, $\zeta_1 = \Diamond(b \wedge \bigcirc_{[0,0]} a)$ and $\zeta_2 = \Diamond_{(0,1)} \Diamond_{[0,3.5]} c$. For the formula ζ_1 , that reads “eventually b will occur followed immediately by an a ”, we have $\rho_1 \models_{pw} \zeta_1$ but $\rho_1 \not\models_{itw} \zeta_1$ (the closest formula to ζ_1 under $\models_{itw}$ is $\Diamond(a \wedge b)$ where the order is lost, see below). On the other hand, for ζ_2 that reads “from some time within $(0, 1)$, c will occur between 0 and 3.5 time units” we have $\rho_1 \models_{itw} \zeta_2$ but $\rho_1 \not\models_{pw} \zeta_2$.

The above example is an excellent motivator for the remainder of this paper: its counterintuitive outcome (as ρ_1 “should” satisfy both formulae) owes to the fact that the pointwise semantics can reason only at positions of timed events, whereas the interval-based one loses the notion of order over simultaneous events as we will see throughout Section 3.

3 The Pointwise and Interval-Based Semantics are Incomparable

3.1 Main observation

Over finite words. To explain the intuition behind the incomparability of both semantics, an excellent starting point is the existing result **ER** (Section 1), since D’Souza and Prabhakar [14] use a clean version of pointwise and interval-based semantics both interpreted over finite timed words, i.e., the satisfaction relations $\models_{pw}$ and $\models_{itw}$ (Section 2.4) where $|\rho| < \omega$. To arrive at **ER**, the authors use the formula ϕ_{dp} defined below¹ over $\Sigma = \{a, b\}$ and prove formally that there exists no formula ψ such that $\llbracket \psi \rrbracket_{pw}^f = \llbracket \phi_{dp} \rrbracket_{itw}^f$.

$$\phi_{dp} = \neg \Diamond (a \wedge (\neg \varphi_{\Sigma} \mathcal{U} a) \wedge (\neg a \mathcal{U} (\neg a \wedge \Diamond_{[1,1]} (a \vee b))))$$

The language $\llbracket \phi_{dp} \rrbracket_{itw}^f$ contains all finite timed words such that for every two consecutive occurrences of the action a at times t and t' , there is no action that occurs within the interval $[t + 1, t' + 1]$. This is an important expressiveness result, yet we argue that half of the picture is missing: the authors require timed words to be strictly monotone [14][Sect. 1], which impairs the expressive power of the pointwise semantics. To explain this, let us fall back on the general model of finite timed words, i.e., consider the set $\{\rho \in \mathbf{P} \mid |\rho| < \omega\}$. Consider now

¹ We slightly rewrote this formula in terms of the standard (strict) Until used in this paper.

the formula $\zeta_1 = \Diamond(b \wedge \bigcirc_{[0,0]} a)$ of Example 2.15. In the pointwise semantics, this formula captures exactly its informal description (Example 2.15), i.e., $\llbracket \zeta_1 \rrbracket_{pw}^f$ defines the language containing all finite timed words such that there exists a position $i > 0$ with $\tau_i = \tau_{i+1}$, $\sigma_i = b$ and $\sigma_{i+1} = a$. Under $\models_{itw}$, however, the language $\llbracket \zeta_1 \rrbracket_{itw}^f$ is empty, and the closest language to $\llbracket \zeta_1 \rrbracket_{pw}^f$ is $\llbracket \varphi \rrbracket_{itw}^f$ with $\varphi = \Diamond(a \wedge b)$. Yet, $\llbracket \varphi \rrbracket_{itw}^f$ is more permissive than $\llbracket \zeta_1 \rrbracket_{pw}^f$. Let us show this through an example.

► **Example 3.1.** Consider $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$ from Example 2.15 and $\rho_2 = (a, 0)(a, 1)(b, 1)(c, 3.3)$. Then we have $\rho_1 \in \llbracket \zeta_1 \rrbracket_{pw}^f$ and $\rho_2 \notin \llbracket \zeta_1 \rrbracket_{pw}^f$, but both ρ_1 and ρ_2 belong to the language $\llbracket \varphi \rrbracket_{itw}^f$.

That is, ζ_1 can distinguish between ρ_1 and ρ_2 under the pointwise semantics, whereas φ cannot under the interval-based one. As we will demonstrate later, there is in fact no MTL formula that distinguishes between ρ_1 and ρ_2 under the interval-based semantics.

Over infinite timed words. The same observation above continues to hold for infinite timed words as illustrated in the following example.

► **Example 3.2.** Consider the two infinite timed words $\rho = (\sigma, \tau)$ and $\rho' = (\sigma', \tau')$ s.t. ${}^3\rho = \rho_1$, ${}^3\rho' = \rho_2$ and $\forall i > 3 : \sigma'_i = \sigma_i = c \wedge \tau'_i = \tau_i = \tau_{i-1} + 1$, with ρ_1 and ρ_2 as defined in Example 3.1. That is, ρ and ρ' have respectively ρ_1 and ρ_2 as a prefix and agree otherwise on an infinite suffix containing only c actions. The above observation carries on to ρ and ρ' as we have $\rho \in \llbracket \zeta_1 \rrbracket_{pw}^{inf}$, $\rho' \notin \llbracket \zeta_1 \rrbracket_{pw}^{inf}$ but no formula can distinguish between ρ and ρ' under $\models_{itw}$ (this will be proven formally later on).

We observe the same weakness of the interval-based semantics over timed state sequences, i.e., when considering the relation $\models_{its}$ instead of $\models_{itw}$. To explain this, we rely on the clever remark in Bouyer et al. [9][Sect. 2], [10][Example 1]:

“an [infinite] timed word can be seen as a special case of timed state sequences, e.g., $\rho = (a, 0)(a, 1.1)(b, 2) \dots$ corresponds to the timed state sequence $\kappa = (\{a\}, [0, 0])(\emptyset, (0, 1.1))(\{a\}, [1.1, 1.1])(\emptyset, (1.1, 2))(\{b\}, [2, 2]) \dots$ ”

Which clearly holds under strict monotonicity. However, under the general model, the clearly different ρ and ρ' of Example 3.2 will correspond to the same timed state sequence starting with $(\{a\}, [0, 0])(\emptyset, (0, 1))(\{a, b\}, [1, 1])(\emptyset, (1, 3.3))(\{c\}, [3.3, 3.3])$. That is, under non-strict monotonicity of time, the encoding from infinite timed words to infinite timed state sequences, suggested in Bouyer et al.’s remark above, cannot correspond to an injective function, and therefore timed words become no longer a special case of timed state sequences. If one tries to have two different images for ρ and ρ' , e.g., by enforcing an order for the counterpart of ρ , they would obtain the $\hat{\kappa} = (\{a\}, [0, 0])(\emptyset, (0, 1))(\{b\}, [1, 1])(\{a\}, [1, 1]) \dots$, where interval adjacency is violated and $\hat{\kappa}(1)$ is no longer defined.

3.2 Approach

3.2.1 Equating $\models_{its}$ and $\models_{itw}$

In order to prove the incomparability between the pointwise and interval-based semantics, we first need to deal with the heterogeneity between timed words and timed state sequences. In other words, there is no way to compare $\models_{pw}$ and $\models_{its}$ if we allow arbitrary timed state sequences, due to the notable difference in nature with timed words, as explained in Section 2.3. For example, the timed state sequence $(\{a, b\}, [0, 1])(\{c\}, [1, 1])$ does not make much sense in the context of action-based timed executions: actions a and b , atomic by

definition, cannot remain true continuously over the dense interval $[0, 1)$. We need therefore to isolate the largest subset of timed state sequences that faithfully represents action-based timed executions modulo the interval-based semantics.

We show that such subset, which we call K_a , can be obtained using natural properties of action-based timed executions. In mathematical terms, K_a is the largest set of images of elements of P by a total relation $\mathcal{F}$ guaranteeing that $\models_{its}$ and $\models_{itw}$ coincide.

Action-based timed state sequences. To construct K_a , we use the fact that actions occur only at punctual intervals. Then, since the first action occurs at 0 and the execution, if finite, terminates with an action, the first interval is $[0, 0]$ and the last interval, if exists, is punctual. Finally, between every two punctual intervals, we have a non-punctual one with no action associated. This gives us an alternation of punctual intervals with sets of actions at even positions and non-punctual ones with empty sets at odd positions.

► **Definition 3.3** (Action-based timed state sequences). *The set of action-based timed state sequences K_a is defined as follows:*

$$\begin{aligned} K_a = \{ \kappa = (\eta, \iota) \in K \mid & \iota_0 = [0, 0] \text{ and} \\ & |\kappa| \in \mathbb{N} \Rightarrow \iota_{|\kappa|-1} \text{ is punctual and} \\ & \forall i \in \mathbb{N}, i < |\kappa|, i \bmod 2 = 0 : \eta_i \neq \emptyset \text{ and } \iota_i \text{ is punctual and} \\ & \forall i \in \mathbb{N}, i < |\kappa|, i \bmod 2 = 1 : \eta_i = \emptyset \text{ and } \iota_i \text{ is not punctual} \} \end{aligned}$$

► **Example 3.4.** $(\{a\}, [0, 0])(\emptyset, (0, 1))(\{a, b\}, [1, 1])$ is an element of K_a , corresponding to an action-based execution where a occurred at time 0, and a and b at time 1. In contrast, $(\{a\}, [0, 0])(\{a, b\}, (0, 1))(\emptyset, [1, 1])$ and $(\emptyset, [0, 0])(\{b\}, (0, 1))(\{a, b\}, [1, 1])$ do not represent any action-based execution and are naturally not elements of K_a .

► **Lemma 3.5.** *The length of any finite element in K_a is an odd number.*

Relating P and K_a . In order to define a relation between P and K_a , and to avoid heavy notations, we use an intermediary representation that allows to group events happening simultaneously while preserving their order. We refer to such representation, which uses the notions of *flat time sequences* (Definition 3.6) and *maximal flat subsequences* (Definition 3.7), as *compact timed words* (Definition 3.8).

► **Definition 3.6** (Flat time sequence). *A finite time sequence τ is said flat iff $\forall i \in \mathbb{N}, 0 < i < |\tau| : \tau_{i-1} = \tau_i$.*

► **Definition 3.7** (Maximal flat subsequence). *Let τ be a time sequence and $i, j \in \mathbb{N}$ such that $i \leq j < |\tau|$. The finite sequence $\tau' = \tau_i \tau_{i+1} \dots \tau_j$ is called a maximal flat subsequence (MFS) of τ iff τ' is a flat sequence and:*

- $i > 0 \Rightarrow \tau_i > \tau_{i-1}$ and
- $j + 1 < |\tau| \Rightarrow \tau_j < \tau_{j+1}$.

Note that when $i = j$ the MFS is *trivial* (contains only one element). It is easy to see that every time sequence τ is a sequence of MFSs. The number of MFSs is finite if τ is finite and infinite otherwise. If τ is infinite, the finiteness of each MFS is guaranteed by time divergence. Examples will follow.

Let $s\tau$ be the sequence over $\mathbb{N}$ defined as follows:

- $s\tau_0 = j$ s.t. $\tau_0 \dots \tau_j$ is an MFS of τ ,
- $s\tau_{i+1} = k$ s.t. $\tau_{s\tau_i+1} \dots \tau_k$ is an MFS of τ .

That is, $s\tau_i$ corresponds to the index of τ delimiting its i^{th} MFS, and $|s\tau|$ corresponds to the number of MFSs in τ . Now we are ready to define the compact version of an arbitrary timed word. In the following, let $\tilde{P}$ denote the set of all pairs $(\tilde{\sigma}, \tilde{\tau})$ where $\tilde{\sigma}$ is a (finite or infinite) sequence of non-empty finite sequences over Σ , $\tilde{\tau}$ a *strictly monotone* time sequence starting at 0 and $|\tilde{\sigma}| = |\tilde{\tau}|$.

► **Definition 3.8** (Compact timed word). *We define the function $\mathcal{C} : P \rightarrow \tilde{P}$ as follows. $\mathcal{C}(\rho = (\sigma, \tau)) = (\tilde{\rho} = (\tilde{\sigma}, \tilde{\tau}))$ iff:*

1. $|\tilde{\rho}| = |s\tau|$,
2. $\forall 0 \leq i < |\tilde{\rho}| : \tilde{\tau}_i = \tau_{s\tau_i}$,
3. $\tilde{\sigma}_0 = \sigma_0 \dots \sigma_{s\tau_0}$ and $\forall 0 < i < |\tilde{\rho}| : \tilde{\sigma}_i = \sigma_{s\tau_{i-1}+1} \dots \sigma_{s\tau_i}$.

That is, $\mathcal{C}$ relates every $\rho = (\sigma, \tau) \in P$ to its compact version, i.e., where every $\tau_i \dots \tau_j$ MFS of τ is reduced into a single element associated with the sequence of actions $\sigma_i \dots \sigma_j$.

► **Example 3.9.** Consider $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$ and $\rho_2 = (a, 0)(a, 1)(b, 1)(c, 3.3)$ from Example 3.1. Then we have $\tilde{\rho}_1 = ((a), 0)((b, a), 1)((c), 3.3)$ and $\tilde{\rho}_2 = ((a), 0)((a, b), 1)((c), 3.3)$.

► **Lemma 3.10.** $\mathcal{C}$ is bijective.

Relating $\tilde{P}$ and K_a . We now move to define a relation between compact timed words and action-based timed state sequences, where we use the function $setof()$ that transforms a sequence over Σ to a set preserving its elements.

► **Definition 3.11** (Relation between $\tilde{P}$ and K_a). *Let $\mathcal{R} : \tilde{P} \rightarrow K_a$ be the function defined as follows. $\mathcal{R}(\tilde{\rho} = (\tilde{\sigma}, \tilde{\tau})) = (\kappa = (\eta, \iota))$ iff:*

1. $|\kappa| = |\tilde{\rho}| = \omega$ if $\tilde{\rho}$ is infinite, $|\kappa| = 2|\tilde{\rho}| - 1$ otherwise and
2. $\forall i \in \{1 \dots |\tilde{\rho}| - 1\} : \eta_{2i} = setof(\tilde{\sigma}_i), \iota_{2i} = [\tilde{\tau}_i, \tilde{\tau}_i], \eta_{2i-1} = \emptyset$ and $\iota_{2i-1} = (\tilde{\tau}_{i-1}, \tilde{\tau}_i)$ and
3. $\eta_0 = setof(\tilde{\sigma}_0), \iota_0 = [\tilde{\tau}_0, \tilde{\tau}_0]$.

► **Example 3.12.** Consider $\tilde{\rho}_1 = ((a), 0)((b, a), 1)((c), 3.3)$ and $\tilde{\rho}_2 = ((a), 0)((a, b), 1)((c), 3.3)$ from Example 3.9. We have $\mathcal{R}(\tilde{\rho}_1) = \mathcal{R}(\tilde{\rho}_2) = (\{a\}, [0, 0])(\emptyset, (0, 1))(\{a, b\}, [1, 1])(\emptyset, (1, 3.3))(\{c\}, [3.3, 3.3])$.

In other words, the “holes” between the timestamps in $\tilde{\rho}$ are “filled” by the timed state sequence $\mathcal{R}(\tilde{\rho})$ (i.e., thanks to the non-punctual intervals between the punctual ones).

Notice how $\mathcal{R}$ is *not* injective due to the first component of a timed state sequence being a sequence of sets. We will show that this captures exactly the weakness of the relations $\models_{its}$ and $\models_{itw}$ alike, hinted at in Section 3.1.

► **Lemma 3.13.** $\mathcal{R}$ is surjective.

► **Lemma 3.14.** $\mathcal{R}$ is not injective.

Proof. The proof is straightforward through the counterexample given in Example 3.12. ◀

In the following, we let $\mathcal{F} = \mathcal{R} \circ \mathcal{C}$.

► **Lemma 3.15.** $\mathcal{F}$ is total.

Proof. $\mathcal{F}$ is a surjective function since (i) $\mathcal{C}$ is a function and (ii) $\mathcal{R}$ is a surjective function (Lemma 3.13). Therefore $\mathcal{F}$ is right and left total and consequently total. ◀

$\models_{itw}$ and $\models_{its}$ are equivalent. We are now ready to equate $\models_{its}$ and $\models_{itw}$. The following proposition is central as it shows that $\models_{itw}$ and $\models_{its}$ enjoy exactly the same expressive power over *all* timed words and *all* action-based timed state sequences. Note that we do not need to prove the equivalence for the version with every $\kappa \in K_a$ and elements of $\mathcal{F}^{-1}[\kappa]$ since $\mathcal{F}$ is total (Lemma 3.15).

► **Proposition 3.16.** *For every $\phi \in MTL$, $\rho \in P$ and $t \in \mathbb{R}_{\geq 0}$ ($t \leq \mu(\rho)$ if $|\rho| \in \mathbb{N}$):*

$$\rho, t \models_{itw} \phi \Leftrightarrow \mathcal{F}(\rho), t \models_{its} \phi$$

The main takeaway is summed by the following theorem, a direct consequence of Proposition 3.16 and Lemma 3.15, stating that $\mathcal{F}$ preserves the full expressiveness of $\models_{its}$ and $\models_{itw}$ over all timed words and all action-based timed state sequences.

► **Theorem 3.17.** *For every $\phi \in MTL$:*

- $\llbracket \phi \rrbracket_{its}^{fi} \cap K_a = \{\mathcal{F}(\rho) \mid \rho \in \llbracket \phi \rrbracket_{itw}^{fi}\}$ and $\llbracket \phi \rrbracket_{itw}^{fi} = \bigcup \{\mathcal{F}^{-1}[\kappa] \mid \kappa \in \llbracket \phi \rrbracket_{its}^{fi} \cap K_a\}$ with $fi \in \{f, inf\}$,
- $\llbracket \phi \rrbracket_{its} \cap K_a = \{\mathcal{F}(\rho) \mid \rho \in \llbracket \phi \rrbracket_{itw}\}$ and $\llbracket \phi \rrbracket_{itw} = \bigcup \{\mathcal{F}^{-1}[\kappa] \mid \kappa \in \llbracket \phi \rrbracket_{its} \cap K_a\}$.

3.2.2 Proving Incomparability

Since $\models_{its}$ and $\models_{itw}$ are proven equivalent, we can devise our incomparability proofs for $\models_{pw}$ vs. $\models_{itw}$ only, while exploiting the equalities of Theorem 3.17.

We first formalise the notion of incomparability. Let $\llbracket MTL \rrbracket_{sem}^f$ denote the set of all languages over finite timed words (i.e., a set of sets) defined by MTL formulae under the semantics corresponding to $sem \in \{pw, itw\}$: $\llbracket MTL \rrbracket_{sem}^f = \{\llbracket \phi \rrbracket_{sem}^f \mid \phi \in MTL\}$. Similarly, $\llbracket MTL \rrbracket_{sem}^{inf} = \{\llbracket \phi \rrbracket_{sem}^{inf} \mid \phi \in MTL\}$ and $\llbracket MTL \rrbracket_{sem} = \llbracket MTL \rrbracket_{sem}^f \cup \llbracket MTL \rrbracket_{sem}^{inf}$. Let $\#$ be the binary operator defined over sets as: $S_1 \# S_2 \equiv S_1 \not\subseteq S_2 \wedge S_2 \not\subseteq S_1$, i.e., none of its arguments is included in the other. We can now formally define incomparability of $\models_{pw}$ and $\models_{itw}$ over MTL.

► **Definition 3.18** (Incomparability). *The MTL semantics defined by $\models_{pw}$ and $\models_{itw}$ are incomparable over:*

- Finite timed words iff $\llbracket MTL \rrbracket_{pw}^f \# \llbracket MTL \rrbracket_{itw}^f$,
- Infinite timed words iff $\llbracket MTL \rrbracket_{pw}^{inf} \# \llbracket MTL \rrbracket_{itw}^{inf}$.

The semantics defined by $\models_{pw}$ and $\models_{itw}$ are incomparable iff they are incomparable over finite and infinite timed words, i.e., $\llbracket MTL \rrbracket_{pw} \# \llbracket MTL \rrbracket_{itw}$.

In order to show that e.g., $\llbracket MTL \rrbracket_{pw}^f \# \llbracket MTL \rrbracket_{itw}^f$, it suffices to find two formulae $\phi_1, \phi_2 \in MTL$ such that for all $\psi \in MTL$, the language of finite timed words defined by ϕ_1 (resp. ϕ_2) under $\models_{pw}$ (resp. $\models_{itw}$) is different from the language defined by ψ under $\models_{itw}$ (resp. $\models_{pw}$) as generalised by the following proposition.

► **Proposition 3.19.** $\llbracket MTL \rrbracket_{pw}^f \# \llbracket MTL \rrbracket_{itw}^f$ iff $\exists \phi_1, \phi_2 \in MTL \forall \psi \in MTL$:

- $\llbracket \phi_1 \rrbracket_{pw}^f \neq \llbracket \psi \rrbracket_{itw}^f$ and
- $\llbracket \phi_2 \rrbracket_{itw}^f \neq \llbracket \psi \rrbracket_{pw}^f$.

(The equivalence with $\llbracket MTL \rrbracket_{pw}^{inf} \# \llbracket MTL \rrbracket_{itw}^{inf}$ is defined analogously).

Proving $\llbracket MTL \rrbracket_{pw}^f \# \llbracket MTL \rrbracket_{itw}^f$. We start with proving $\llbracket MTL \rrbracket_{pw}^f \not\subseteq \llbracket MTL \rrbracket_{itw}^f$. It suffices to find two distinct finite timed words that have the same image by $\mathcal{F}$ and a formula ϕ that distinguishes between them under $\models_{pw}$. By definition, no formula under $\models_{its}$ can distinguish between their images through $\mathcal{F}$ (since it is the same image) and we conclude that no formula can distinguish between ρ_1 and ρ_2 under $\models_{itw}$ since $\models_{its}$ and $\models_{itw}$ are equally expressive.

► **Lemma 3.20.** *Let $\zeta_1 = \Diamond(b \wedge \bigcirc_{[0,0]} a)$ from Example 2.15. The following statement holds:*

$$\forall \psi \in MTL : \llbracket \zeta_1 \rrbracket_{pw}^f \neq \llbracket \psi \rrbracket_{itw}^f$$

Proof. Let $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$ and $\rho_2 = (a, 0)(a, 1)(b, 1)(c, 3.3)$ from Example 3.1. We have $\rho_1 \in \llbracket \zeta_1 \rrbracket_{pw}^f$ and $\rho_2 \notin \llbracket \zeta_1 \rrbracket_{pw}^f$. Now, ρ_1 and ρ_2 have the same image by $\mathcal{F}$ that we will refer to as κ . By definition, for every $\psi \in MTL$, we have either $\kappa \in \llbracket \psi \rrbracket_{its}^f$, and by Theorem 3.17, $\rho_1 \in \llbracket \psi \rrbracket_{itw}^f$ and $\rho_2 \in \llbracket \psi \rrbracket_{itw}^f$, or $\kappa \notin \llbracket \psi \rrbracket_{its}^f$, and again by Theorem 3.17 $\rho_1 \notin \llbracket \psi \rrbracket_{itw}^f$ and $\rho_2 \notin \llbracket \psi \rrbracket_{itw}^f$. And therefore $\llbracket \zeta_1 \rrbracket_{pw}^f \neq \llbracket \psi \rrbracket_{itw}^f$. ◀

► **Remark 3.21.** Note that this weakness of the interval-based semantics has a direct consequence on the coherence between MTL and its untimed version, LTL. Consider e.g., ρ_1 and ρ_2 above. Their untimed counterparts are, respectively, the words *abac* and *aabc* which are temporally different (in terms of LTL). Yet, under $\models_{itw}$ or $\models_{its}$ alike, it is impossible to distinguish between them whenever the two letters in the middle occur simultaneously.

For the part $\llbracket MTL \rrbracket_{itw}^f \not\subseteq \llbracket MTL \rrbracket_{pw}^f$, we depart from

$$\phi_{dp} = \neg \Diamond(a \wedge (\neg(\varphi_\Sigma \mathcal{U} a) \wedge (\neg a \mathcal{U} (\neg a \wedge \Diamond_{[1,1]}(a \vee b))))$$

The formula proven to have no equivalent under $\models_{pw}$ over strictly monotone finite timed words [14] (Section 3.1). The idea here is rather simple: find a formula γ_1 that captures exactly the language of ϕ_{dp} under the general model of finite timed words then conclude.

► **Lemma 3.22.** *Let $\gamma_1 = \phi_{dp} \wedge \phi_s$ with² $\phi_s = (\neg \bigvee_{a \neq b \in \Sigma} (a \wedge b)) \wedge \Box (\neg \bigvee_{a \neq b \in \Sigma} (a \wedge b))$. The following statement holds: $\forall \psi \in MTL : \llbracket \gamma_1 \rrbracket_{itw}^f \neq \llbracket \psi \rrbracket_{pw}^f$*

Proof. The language $\llbracket \gamma_1 \rrbracket_{itw}^f$ contains the set of strictly monotone (satisfying ϕ_s) finite timed words satisfying ϕ_{dp} . The proof then follows directly from [14][Sect. 3]. ◀

► **Proposition 3.23.** *The pointwise and the interval based semantics are incomparable over finite timed words.*

Proof. From Lemma 3.20, Lemma 3.22 and Proposition 3.19 we get $\llbracket MTL \rrbracket_{pw}^f \# \llbracket MTL \rrbracket_{itw}^f$ and conclude by Definition 3.18. ◀

Proving $\llbracket MTL \rrbracket_{pw}^{inf} \# \llbracket MTL \rrbracket_{itw}^{inf}$. We first prove $\llbracket MTL \rrbracket_{pw}^f \not\subseteq \llbracket MTL \rrbracket_{itw}^f$, using the same formula from Lemma 3.20.

► **Lemma 3.24.** *Let $\zeta_1 = \Diamond(b \wedge \bigcirc_{[0,0]} a)$. The following statement holds:*

$$\forall \psi \in MTL : \llbracket \zeta_1 \rrbracket_{pw}^{inf} \neq \llbracket \psi \rrbracket_{itw}^{inf}$$

Proof. The proof is very similar to that of Lemma 3.20, with $\rho \in \llbracket \zeta_1 \rrbracket_{pw}^{inf}$ and $\rho' \notin \llbracket \zeta_1 \rrbracket_{pw}^{inf}$ (Example 3.2) having the same image by $\mathcal{F}$. ◀

² The first clause of ϕ_s is necessary due to the strict nature of the Until operator, from which $\Box$ is derived.

For $\llbracket MTL \rrbracket_{itw}^{inf} \not\subseteq \llbracket MTL \rrbracket_{pw}^{inf}$, we use a formula already proven in [9] [Prop. 1 and Sect. 3.2] to have a language over infinite words that is not expressible under $\models_{pw}$. This language is informally defined as “all infinite timed words containing an occurrence of b within the absolute interval $(0, 2)$ and an occurrence of c within the absolute interval $(1, 2]$, such that the latter happens *after* the former within the relative interval $(0, 1)$ ”.

► **Lemma 3.25.** *Let $\gamma_2 = \Diamond_{[0,1]} (\Diamond_{[0,1]} b \wedge \Diamond_{[1,1]} c)$. The following statement holds:*

$$\forall \psi \in MTL : \llbracket \gamma_2 \rrbracket_{itw}^{inf} \neq \llbracket \psi \rrbracket_{pw}^{inf}$$

► **Proposition 3.26.** *The pointwise and the interval-based semantics are incomparable over infinite timed words.*

Proof. From Lemma 3.24, Lemma 3.25 and Proposition 3.19 we get $\llbracket MTL \rrbracket_{pw}^{inf} \# \llbracket MTL \rrbracket_{itw}^{inf}$ and conclude by Definition 3.18. ◀

And finally we conclude with the following theorem, a direct consequence of Proposition 3.23 and Proposition 3.26.

► **Theorem 3.27.** *The pointwise and the interval-based semantics of MTL are incomparable.*

Incomparability under stutter freeness. Our incomparability results continue to hold under the reasonable restriction of stutter freeness (Definition 2.9, Item 1). For the finite case, the formula ζ_1 given in Lemma 3.20, as shown in the proof of the same lemma, distinguishes between two stutter-free finite timed words under $\models_{pw}$ whereas no formula under $\models_{itw}$ can do so (and the same goes for the infinite case, Lemma 3.24 and its proof). The only technical difference when disallowing stuttering is the fact that a finite timed state sequence, element of K_a , will then have an image of finite cardinality by the inverse relation $\mathcal{F}^{-1}$, which has no effect on the incomparability theorem.

4 Mixed Semantics

In this section, we present a new *mixed* semantics that embeds both the pointwise and the interval-based ones. We do so in three steps. First, we present an intuitive version of the mixed semantics over MTL that, leveraging the expressive power of the interval-based semantics, is strictly more expressive than the pointwise one (Section 4.1). Second, in Section 4.2, we conjecture that such version is incomparable with the interval-based one (due mainly to the fact that the latter is more suited for signals). Third, we show that extending the syntax of MTL with a single atomic formula unleashes the power of the mixed semantics leading to the embedding of both the pointwise and the interval-based semantics, conjugating therefore their full expressive powers (Section 4.3). The embedding is automated through syntax-driven compilers.

4.1 Dual reasoning on time and positions

In this subsection, we undertake the first step described above, i.e., we propose to increase the power of the pointwise semantics using the flexibility of the interval-based one. The reason we do it in this direction (rather than the other way around) is the fact that the interval-based semantics is more suitable for signals (Section 2.3) whereas we focus on action-based timed executions (another example on the counter-intuitiveness of the interval-based semantics over timed words will be given in Section 4.2). We still provide in Section 4.3 an extended version of the mixed semantics that embeds both the pointwise and the interval-based ones.

Given a compact timed word $\tilde{\rho} = (\tilde{\sigma}, \tilde{\tau})$, we define its duration in the obvious way, i.e., $\mu(\tilde{\rho}) = \mu(C^{-1}(\tilde{\rho}))$. For any $t \in \mathbb{R}_{\geq 0}$ (with $t \leq \mu(\tilde{\rho})$ if $\tilde{\rho}$ is finite), we let $\tilde{\rho}(t)$ equal $\tilde{\sigma}_i$ where $\tilde{\tau}_i = t$ (such $\tilde{\tau}_i$ is uniquely defined if it exists) and the singleton sequence $(\vdash)$ otherwise, where $\vdash$ is a “helper” symbol that represents “no action”, i.e., it will be semantically nothing else than equivalent to $\neg\varphi_\Sigma$.

The idea is to evaluate an MTL formula over a compact timed word $\tilde{\rho} = (\tilde{\sigma}, \tilde{\tau})$, a time point $t \in \mathbb{R}_{\geq 0}$ (with $t \leq \mu(\tilde{\rho})$ if $|\tilde{\rho}| < \omega$) and a position $0 \leq j < |\tilde{\rho}(t)|$. It is important to remark here that the condition $0 \leq j < |\tilde{\rho}(t)|$ is by no means impractical for verification purposes, as the satisfaction of some formula ϕ by some $\tilde{\rho}$ boils down to the satisfaction of ϕ by $\tilde{\rho}$, $t_0 = 0$ and $j_0 = 0$ (examples will follow).

Now we are ready to formalise the mixed semantics:

$$\begin{aligned} \tilde{\rho}, t, j \models_{mx} a & \quad \text{iff} \quad a = \tilde{\rho}(t)(j) \\ \tilde{\rho}, t, j \models_{mx} \neg\phi & \quad \text{iff} \quad \tilde{\rho}, t, j \not\models_{mx} \phi \\ \tilde{\rho}, t, j \models_{mx} \phi \wedge \psi & \quad \text{iff} \quad \tilde{\rho}, t, j \models_{mx} \phi \text{ and } \tilde{\rho}, t, j \models_{mx} \psi \\ \tilde{\rho}, t, j \models_{mx} \phi \mathcal{U}_I \psi & \quad \text{iff} \quad \exists (t', j') < (t', j') : t' - t \in I \text{ and } \tilde{\rho}, t', j' \models_{mx} \psi \text{ and} \\ & \quad \forall (t, j) < (t'', j'') < (t', j') : \tilde{\rho}, t'', j'' \models_{mx} \phi \end{aligned}$$

Where the relation $<$, used for the $\mathcal{U}_I$ operator, denotes a lexicographic order³ defined in the usual way. We say $\tilde{\rho}$ satisfies ϕ , written $\tilde{\rho} \models_{mx} \phi$, iff $\tilde{\rho}, 0, 0 \models_{mx} \phi$.

Let us explain, relying on the rules of the semantics above, how the relation $\models_{mx}$ increases the expressive power of $\models_{pw}$ (we will formally prove this soon). The real t enables the evaluation at any arbitrary time point, whereas the natural j allows to “access” the positions of ordered events occurring at t , if any. The semantics of $\mathcal{U}_I$ reflects this increased power of expressiveness: $\phi \mathcal{U}_I \psi$ is evaluated at any timepoint and any position of the sequence of events occurring at t . Such position is guaranteed to exist: in particular, if t belongs to a “hole” in $\tilde{\rho}$ (i.e., strictly comprised between two timestamps), then $\tilde{\rho}(t)(0)$ exists and is equal to the $\vdash$ symbol which plays otherwise a transparent role as, in such case, $\tilde{\rho}, t, 0 \models \neg\varphi_\Sigma$ and $\tilde{\rho}, t, 0 \models \top$ as expected. The following example illustrates the intuitive nature of the mixed semantics.

► **Example 4.1.** Consider the setting of Example 2.15 with $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$, $\zeta_1 = \Diamond(b \wedge \bigcirc_{[0,0]} a)$ and $\zeta_2 = \Diamond_{(0,1)} \Diamond_{[0,3.5]} c$. As explained within the same example, ρ_1 should, intuitively, satisfy both formulae, yet we have $\rho_1 \not\models_{itw} \zeta_1$ and $\rho_1 \not\models_{pw} \zeta_2$.

Under the mixed semantics, both formulae are satisfied thanks to the flexibility of $\models_{mx}$, taking both time points and positions into account. To illustrate this, let us first encode in the obvious way $\rho_1 = (a, 0)(b, 1)(a, 1)(c, 3.3)$ into its compact counterpart $\tilde{\rho}_1 = ((a), 0)((b, a), 1)((c), 3.3)$.

Then we have $\tilde{\rho}_1 \models_{mx} \zeta_1$ since $\tilde{\rho}_1, 0, 0 \models_{mx} \zeta_1$: we have $\tilde{\rho}_1, t', j'$ with $(t' = 1, j' = 0)$, greater than $(0, 0)$ in lexicographic order, satisfies b (since $\tilde{\rho}_1(1)(0) = b$) and in turn, $\tilde{\rho}_1, t'', j''$ with $(t'' = 1, j'' = 1)$, greater than $(1, 0)$ satisfies a within $[0, 0]$ and (i) for every $(0, 0) < (t_1, j_1) < (t', j')$, $\tilde{\rho}_1, t_1, j_1 \models_{mx} \top$ and (ii) for every $(t', j') < (t_2, j_2) < (t'', j'')$, $\tilde{\rho}_1, t_2, j_2 \models_{mx} \neg\varphi_\Sigma$ (vacuous truth).

Dually, $\tilde{\rho}_1, 0, 0 \models_{mx} \zeta_2$ because for any couple $(t', j' = 0)$ with $t' \in (0, 1)$, we have $\tilde{\rho}_1, t', j' \models_{mx} \Diamond_{[0,3.5]} c$ since at $(t'' = 3.3, j'' = 0)$ we have $\tilde{\rho}_1, t'', j'' \models_{mx} c$ (since $\tilde{\rho}_1(t'')(j'') = c$) and $t'' - t' \in [0, 3.5]$.

³ Compact timed words are similar to *weakly-monotonic observations* [3] over which the authors also mention lexicographic order. While we were unaware of these similarities when realising this work, all our contributions summarised in Section 1 and developed in Section 3 and Section 4 are novel.

4.1.1 $\models_{mx}$ is strictly more expressive than $\models_{pw}$

The mixed semantics is strictly more expressive than the pointwise one. To show this, we proceed as follows. First, we devise a compiler $(\cdot)_{pw}$ over MTL such that $(\phi)_{pw}$ under $\models_{mx}$ is equivalent to ϕ under $\models_{pw}$ (using the bijective function $\mathcal{C}$ to move between timed words and their compact version). Second, we give an example of a formula not expressible in the pointwise semantics (from Section 3.2) and show that it is expressible in the mixed semantics.

From $\models_{pw}$ to $\models_{mx}$. We reason following the main observation that, in $\models_{pw}$, formulae are evaluated only at timepoints where actions occur. Thus, $(\phi)_{pw}$ must satisfy φ_Σ , regardless of its argument. The case for $\mathcal{U}_I$ is the involved one. Given a formula $\phi \mathcal{U}_I \psi$ interpreted under $\models_{pw}$, to get its equivalent formula under $\models_{mx}$, we need also to ensure that (i) ψ holds at a timepoint corresponding to an action occurrence and (ii) ϕ does not need to hold at timepoints where no action occurs, as formalised by $(\phi \mathcal{U}_I \psi)_{pw}$ in the following definition.

► **Definition 4.2** ($\models_{pw}$ -preserving compilation). *The compiler $(\cdot)_{pw}$ is defined inductively:*

- $(a)_{pw} = a$,
- $(\neg\phi)_{pw} = \varphi_\Sigma \wedge \neg(\phi)_{pw}$,
- $(\phi \wedge \psi)_{pw} = \varphi_\Sigma \wedge (\phi)_{pw} \wedge (\psi)_{pw}$,
- $(\phi \mathcal{U}_I \psi)_{pw} = \varphi_\Sigma \wedge ((\phi)_{pw} \vee \neg\varphi_\Sigma) \mathcal{U}_I ((\psi)_{pw} \wedge \varphi_\Sigma)$.

► **Proposition 4.3.** *For every $\phi \in \text{MTL}$ and $\rho \in \mathbf{P} : \rho \models_{pw} \phi \Leftrightarrow \mathcal{C}(\rho) \models_{mx} (\phi)_{pw}$.*

A language definable under $\models_{mx}$ but not under $\models_{pw}$. Consider the formula $\gamma_2 = \Diamond_{[0,1]} (\Diamond_{[0,1]} b \wedge \Diamond_{[1,1]} c)$ from Lemma 3.25. As states the same lemma, γ_2 has no equivalent under the pointwise semantics over infinite timed words. The following lemma states that γ_2 does have an equivalent under $\models_{mx}$ over infinite timed words.

► **Lemma 4.4.** *Let $\gamma_2 = \Diamond_{[0,1]} (\Diamond_{[0,1]} b \wedge \Diamond_{[1,1]} c)$ and $\gamma_3 = \Diamond_{(0,1]} (\Diamond_{[0,1]} b \wedge \Diamond_{[1,1]} c)$. Then:*

$$\llbracket \gamma_2 \rrbracket_{itw}^{inf} = \{\mathcal{C}^{-1}(\tilde{\rho}) \mid \tilde{\rho} \in \llbracket \gamma_3 \rrbracket_{mx}^{inf}\}$$

► **Remark 4.5.** Note the subtlety in γ_3 where the interval of the outer eventually operator is left-open at 0. Under $\models_{itw}$, γ_2 and γ_3 are equivalent because the time point satisfying the formula must be greater than 0 by the definition of the (strict) Until operator from which $\Diamond_{[0,1]}$ is derived. Under $\models_{mx}$, on the other hand, we have $\llbracket \gamma_3 \rrbracket_{mx}^{inf} \subsetneq \llbracket \gamma_2 \rrbracket_{mx}^{inf}$ since e.g., $(t=0, j'=2)$ is greater than $(t=0, j=0)$ (but t is not greater than 0); we have therefore the compact version of any infinite timed word starting with $(a,0)(c,0)(b,0)(c,1)(c,3)$ in $\llbracket \gamma_2 \rrbracket_{mx}^{inf}$ but not in $\llbracket \gamma_3 \rrbracket_{mx}^{inf}$. Hence the explicit exclusion of 0 from the interval.

► **Theorem 4.6.** *The mixed semantics is more expressive than the pointwise one.*

Proof. Direct from Lemma 4.4 and Proposition 4.3. ◀

4.2 Mixed semantics vs. interval-based semantics

We now explain the fundamental difference between $\models_{itw}$ and our $\models_{mx}$ and why we argue that $\models_{mx}$ is more suitable for action-based executions. Let $\phi = (a \vee \neg\varphi_\Sigma) \mathcal{U}_{[1,2]} b$ and $\rho = (a,0)(a,0.5)(c,0.5)(c,1.5)(b,1.5)$. The (action-based) intuition would suggest that ρ should not satisfy ϕ , two occurrences of c at different dates take place (which correspond to neither an a nor “no action”) before b does, and indeed $\tilde{\rho}$ does not satisfy ϕ under the mixed

semantics. In contrast, we have $\rho \models_{itw} \phi$, since the interval-based semantics is more suitable for signals. If ρ was a signal, it would naturally make sense for it to satisfy ϕ since at every time point, either the proposition a held or no proposition did until b within $[1, 2]$.

Because of this “permissive” nature of the $\mathcal{U}_I$ operator at its first argument under $\models_{itw}$ (and its more “restrictive” nature at its second argument, see the second example below), we conjecture that the satisfaction relations $\models_{mx}$ and $\models_{itw}$ are incomparable over timed words. The proof of this conjecture is already halfway since we know that the $\models_{mx}$ is strictly more expressive than $\models_{pw}$ (Theorem 4.6), and given that $\models_{itw}$ and $\models_{pw}$ are incomparable (Theorem 3.27), $\models_{mx}$ is either more expressive than or incomparable with $\models_{itw}$. Actually, we can relatively easily construct an MTL formula defining a language under $\models_{mx}$ (after un-compacting with C^{-1}) that cannot be characterised by any MTL formula under $\models_{itw}$ or $\models_{pw}$ alike, as states the following lemma.

► **Lemma 4.7.** *Let $\gamma_3 = \Diamond_{(0,1]} (\Diamond_{[0,1]} b \wedge \Diamond_{[1,1]} c)$ from Lemma 4.4, $\zeta_1 = \Diamond (b \wedge \bigcirc_{[0,0]} a)$ from Lemma 3.20, $\gamma_4 = \Box_{(0,2]} \neg \varphi_\Sigma$ and $\gamma_5 = \gamma_3 \vee (\zeta_1 \wedge \gamma_4)$. Then for all $\phi \in MTL$:*

- $\llbracket \phi \rrbracket_{pw}^{inf} \neq \{C^{-1}(\tilde{\rho}) \mid \tilde{\rho} \in \llbracket \gamma_5 \rrbracket_{mx}^{inf}\}$
- $\llbracket \phi \rrbracket_{itw}^{inf} \neq \{C^{-1}(\rho) \mid \tilde{\rho} \in \llbracket \gamma_5 \rrbracket_{mx}^{inf}\}$

To have a less biased view, let us now look at a trickier example that rather pleads in favour of $\models_{itw}$. Let $\rho = (c, 0)(c, 0.5)(c, 1.5)(b, 1.5)$ and $\phi = (c \vee \neg \varphi_\Sigma) \mathcal{U}_{[1,2]} (b \wedge \neg c)$. Here, we have $\tilde{\rho} \models_{mx} \phi$ but $\rho \not\models_{itw} \phi$. This is because $b \wedge \neg c$ is trivially satisfied at $(t, j) = (1.5, 1)$ under $\models_{mx}$, whereas it is not at $t = 1.5$ under $\models_{itw}$. It is however legitimate for a user to mean by this formula “verify that only occurrences of c are seen until an occurrence of b takes place between times 1 and 2 and c does not occur simultaneously with such b ”. The mixed semantics fails to capture this, and at first glance the naive way to remedy this issue would be to use an immediate past operator that allows to look *behind* $(t, j) = (1.5, 1)$ while maintaining the same timepoint, i.e., at $(1.5, 0)$. In the next subsection, we show that this “immediate past” is an overkill as we can instead use a single atomic formula to extend the syntax of MTL in order to strictly embed the full expressive power of $\models_{itw}$ into $\models_{mx}$.

4.3 Extending mixed semantics

Following the examples given in Section 4.2, and in order to embed $\models_{itw}$, the satisfaction relation $\models_{mx}$ needs to be able to force the evaluation of a formula at index 0 within any $\tilde{\rho}(t)$, looking forward at the whole sequence returned by $\tilde{\rho}(t)$ (instead of complexifying the syntax of MTL, and $\models_{mx}$ thereof, with an immediate past operator). Let us consider the example $(a \vee \neg \varphi_\Sigma) \mathcal{U}_I b$ satisfied by some ρ, t under $\models_{itw}$, and let t correspond to no timestamp of ρ to simplify the explanation. To satisfy the same formula at $\tilde{\rho}, t, 0$ under $\models_{mx}$, intuitively we need to check for $(a \vee \Diamond_{[0,0]} a)$ only at position 0 for every $\tilde{\rho}(t') \neq (\vdash), t' > t$ until $(b \vee \Diamond_{[0,0]} b)$ becomes true at position 0 of some $\tilde{\rho}(t'')$ satisfying $t'' - t \in I$.

To do this in a simple and efficient way, the idea is to interpret the mixed semantics over $\Sigma \cup \{\beta\}$, where β is an atomic proposition not in Σ . The syntax of MTL is accordingly extended with the atomic formula β , we call such extended syntax MTL_β . The semantics of β is defined simply to hold only at the beginning of a sequence $\tilde{\rho}(t)$:

$$\tilde{\rho}, t, j \models_{mx} \beta \text{ iff } j = 0$$

This seemingly minuscule addition comes with a major expressiveness gain. The mixed semantics now embeds the interval-based one using the compiler $(\llbracket - \rrbracket)_{itw} : MTL \rightarrow MTL_\beta$ defined inductively as follows:

- $\langle a \rangle_{itw} = a \vee \Diamond_{[0,0]} a$,
- $\langle \neg \phi \rangle_{itw} = \neg \langle \phi \rangle_{itw}$,
- $\langle \phi \wedge \psi \rangle_{itw} = \langle \phi \rangle_{itw} \wedge \langle \psi \rangle_{itw}$,
- $\langle \phi \mathcal{U}_I \psi \rangle_{itw} = (\beta \Rightarrow \langle \phi \rangle_{itw}) \mathcal{U}_I (\beta \wedge \langle \psi \rangle_{itw})$.

The following proposition states that $\models_{mx}$ over MTL_β embeds $\models_{itw}$ over MTL.

► **Proposition 4.8.** *For all ϕ in MTL and $\rho \in P$: $\rho \models_{itw} \phi \Leftrightarrow C(\rho) \models_{mx} \langle \phi \rangle_{itw}$.*

Strict expressiveness superiority follows directly from Proposition 4.8 and Theorem 4.6.

► **Remark 4.9.** The embedding of $\models_{itw}$ into $\models_{mx}$ (Proposition 4.8) looks asymmetrical since we allow an extended syntax for $\models_{mx}$ but not for $\models_{itw}$. However, adding a straightforward interpretation of β under $\models_{itw}$, i.e., $\rho, t \models_{itw} \beta$ iff $t = 0$ is effectless: since we have $\rho \models_{itw} \phi$ iff $\rho, 0 \models_{itw} \phi$, we get $\rho \models_{itw} \beta$ iff $\rho, 0 \models_{itw} \beta$, and therefore $\llbracket \beta \rrbracket_{itw} = P = \llbracket \top \rrbracket_{itw}$. In other words, the expressive power of $\models_{itw}$ over MTL and MTL_β would be identical.

We conclude with showing that the set of languages characterised by $\models_{mx}$ over MTL_β is strictly larger than that characterised by the union of the two sets of languages characterised by $\models_{itw}$ and $\models_{pw}$ over MTL.

► **Theorem 4.10.**

$$\llbracket MTL \rrbracket_{pw} \cup \llbracket MTL \rrbracket_{itw} \subsetneq \{ \{ C^{-1}(\tilde{\rho}) \mid \tilde{\rho} \in \llbracket \phi \rrbracket_{mx} \} \mid \phi \in MTL_\beta \}$$

Proof. Direct from Proposition 4.3, Proposition 4.8 and Lemma 4.7. ◀

Note that the witness formula for the strict inclusion in Theorem 4.10 is γ_5 in Lemma 4.7, which does not use the atomic formula β . This is an interesting observation: $\models_{mx}$ over MTL already characterises at least one language that can be expressed neither in $\models_{itw}$ nor in $\models_{pw}$.

5 Conclusion

In this paper, we revisited an important aspect of the extensively studied expressiveness of MTL, namely that of its pointwise semantics as opposed to the interval-based one. We have demonstrated that the two semantics are incomparable over the general models of action-based timed executions, giving thereby a clear answer, in the negative, to the question on whether reasoning over arbitrary time points *instead of* ordered positions of timed events increases the expressive power of MTL. To get the better of both worlds over action-based executions, we proposed a new mixed semantics that embeds the interval-based and the pointwise ones, therefore conjugating their full expressive powers. For future work, the results presented within this paper call for further research on at least two fronts.

On the complexity front, an interesting future direction would be to investigate that of monitoring under the mixed semantics, already shown to be tractable for the pointwise semantics e.g., in [21]. In particular, offline monitoring is a promising venue to reason on executions of distributed systems where the order over simultaneous events is a key aspect [24, 15, 23]. On the expressiveness front, it would be interesting to consider extending our mixed semantics to reason on mixed signal-event executions while maintaining reasonable complexity. This would require to define a mixed structure, e.g., $((a, b), [0, 0])(\{p, q\}, (0, 1))(c, [1, 1])$ where a, b and c are in the actions set and p, q in the atomic propositions one, then further extend our mixed semantics to both sets. In the same vein, we aim to investigate relating the mixed semantics to the different semantics of more expressive logics such as TPTL [4]

and T^{lin} [6]. Finally, our incomparability results are likely to hold when restricting MTL to its decidable fragment MITL [1], where only non-punctual intervals are allowed in formulae. Surprisingly, the non-containments $\llbracket MITL \rrbracket_{pw}^f \not\subseteq \llbracket MITL \rrbracket_{itw}^f$ and $\llbracket MITL \rrbracket_{pw}^{inf} \not\subseteq \llbracket MITL \rrbracket_{itw}^{inf}$ are easy to prove: Lemma 3.20 and Lemma 3.24 still hold if we replace the interval $[0, 0]$ in ζ_1 by any other interval $I \in \mathcal{I}_q$ such that I is left-closed at 0. We leave it open whether $\llbracket MITL \rrbracket_{itw}^f \not\subseteq \llbracket MITL \rrbracket_{pw}^f$ and $\llbracket MITL \rrbracket_{itw}^{inf} \not\subseteq \llbracket MITL \rrbracket_{pw}^{inf}$.

References

- 1 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.
- 2 Rajeev Alur, Limor Fix, and Thomas A Henzinger. A determinizable class of timed automata. In *International conference on computer aided verification (CAV)*, pages 1–13. Springer, 1994. doi:10.1007/3-540-58179-0_39.
- 3 Rajeev Alur and Thomas A Henzinger. Logics and models of real time: A survey. In *Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems)*, pages 74–106. Springer, 1991. doi:10.1007/BFB0031988.
- 4 Rajeev Alur and Thomas A Henzinger. A really temporal logic. *Journal of the ACM (JACM)*, 41(1):181–203, 1994. doi:10.1145/174644.174651.
- 5 Rajeev Alur and Thomas A Henzinger. Modularity for timed and hybrid systems. In *International Conference on Concurrency Theory (CONCUR)*, pages 74–88. Springer, 1997. doi:10.1007/3-540-63141-0_6.
- 6 Mouloud Amara, Giovanni Bernardi, Mohammed Aristide Foughali, and Adrian Francalanza. A theory of (linear-time) timed monitors. In *European Conference on Object-Oriented Programming (ECOOP)*, volume 333, pages 1:1–1:30, 2025. doi:10.4230/LIPIcs.ECOOP.2025.1.
- 7 Eugene Asarin, Paul Caspi, and Oded Maler. Timed regular expressions. *Journal of the ACM*, 49(2):172–206, 2002. doi:10.1145/506147.506151.
- 8 Béatrice Bérard, Paul Gastin, and Antoine Petit. Refinements and abstractions of signal-event (timed) languages. In *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pages 67–81. Springer, 2006. doi:10.1007/11867340_6.
- 9 Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the expressiveness of TPTL and MTL. In *International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 432–443. Springer, 2005. doi:10.1007/11590156_35.
- 10 Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the expressiveness of TPTL and MTL. *Information and Computation*, 208(2):97–116, 2010. doi:10.1016/J.IC.2009.10.004.
- 11 Bjorn B Brandenburg. *Scheduling and locking in multiprocessor real-time operating systems*. PhD thesis, The University of North Carolina at Chapel Hill, 2011.
- 12 Rocco De Nicola and Frits Vaandrager. Action versus state based logics for transition systems. In *LITP Spring School on Theoretical Computer Science*, pages 407–419. Springer, 1990. doi:10.1007/3-540-53479-2_17.
- 13 David L Dill. Timing assumptions and verification of finite-state concurrent systems. In *International Conference on Computer Aided Verification (CAV)*, pages 197–212. Springer, 1989. doi:10.1007/3-540-52148-8_17.
- 14 Deepak D’Souza and Pavithra Prabhakar. On the expressiveness of MTL in the pointwise and continuous semantics. *International Journal on Software Tools for Technology Transfer*, 9(1):1–4, 2007. doi:10.1007/S10009-005-0214-9.
- 15 Colin Fidge. Logical time in distributed computing systems. *Computer*, 24(8):28–33, 2002.
- 16 Mohammed Foughali, Pierre-Emmanuel Hladik, and Alexander Zuepke. Compositional verification of embedded real-time systems. *J. Syst. Archit.*, 142:102928, 2023. doi:10.1016/J.SYSARC.2023.102928.

- 17 Mohammed Aristide Foughali. Revisiting the expressiveness of metric temporal logic: A tale of "je t'aime, moi non plus.". *arXiv preprint arXiv:2603.15379*, 2026. doi:10.48550/arXiv.2603.15379.
- 18 Mohammed Aristide Foughali, Marius Mikucionis, and Maryline Zhang. Scalable computation of inter-core bounds through exact abstractions. In *Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 61–70, 2024. doi:10.1109/COMPSAC61105.2024.00019.
- 19 Carlo Alberto Furia and Matteo Rossi. On the expressiveness of MTL variants over dense time. In *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pages 163–178. Springer, 2007.
- 20 Christoph Haase, Joël Ouaknine, and James Worrell. On process-algebraic extensions of metric temporal logic. *Reflections on the Work of CAR Hoare*, pages 283–300, 2010. doi:10.1007/978-1-84882-912-1_13.
- 21 Hsi-Ming Ho, Joël Ouaknine, and James Worrell. Online monitoring of metric temporal logic. In *International Conference on Runtime Verification (RV)*, pages 178–192. Springer, 2014. doi:10.1007/978-3-319-11164-3_15.
- 22 Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-time systems*, 2(4):255–299, 1990. doi:10.1007/BF01995674.
- 23 Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. In *Concurrency: the Works of Leslie Lamport*, pages 179–196. Association for Computing Machinery, 2019. doi:10.1145/3335772.3335934.
- 24 Christoph Liebig, Mariano Cilia, and Alejandro Buchmann. Event composition in time-dependent distributed systems. In *International Conference on Cooperative Information Systems (COOPIS)*, pages 70–78. IEEE, 1999. doi:10.1109/COOPIS.1999.792159.
- 25 Joël Ouaknine and James Worrell. On the decidability of metric temporal logic. In *Symposium on Logic in Computer Science (LICS)*, pages 188–197. IEEE, 2005. doi:10.1109/LICS.2005.33.
- 26 Joël Ouaknine and James Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3, 2007. doi:10.2168/LMCS-3(1:8)2007.
- 27 Joël Ouaknine and James Worrell. Some recent results in metric temporal logic. In *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pages 1–13. Springer, 2008. doi:10.1007/978-3-540-85778-5_1.
- 28 Amir Pnueli. The temporal logic of programs. In *Annual Symposium on Foundations of Computer Science (SFCS)*, pages 46–57. iee, 1977. doi:10.1109/SFCS.1977.32.