Monitoring Real-Time Systems under Parametric Delay*

Martin Fränzle[1 [0000−0002−9138−8340]], Thomas M. Grosen[2 ✉ [0009−0008−7719−6482]], Kim G. Larsen[2 [0000−0002−5953−3384]], and Martin Zimmermann[2 [0000−0002−8038−2453]]

[1] Carl von Ossietzky Universität, Oldenburg, Germany martin.fraenzle@uol.de
[2] Aalborg University, Aalborg, Denmark {tmgr,kgl,mzi}@cs.aau.dk

Abstract. Timed Büchi automata provide a very expressive formalism for expressing requirements of real-time systems. Online monitoring of embedded real-time systems can then be achieved by symbolic execution of such automata on the trace observed from the system. This direct construction however only is faithful if observation of the trace is immediate in the sense that the monitor can assign exact time stamps to the actions it observes, which is rarely true in practice due to the substantial and fluctuating parametric delays introduced by the circuitry connecting the observed system to its monitoring device. We present a purely zone-based online monitoring algorithm, which handles such parametric delays exactly without recurrence to costly verification procedures for parametric timed automata. We have implemented our monitoring algorithm on top of the real-time model checking tool UPPAAL, and report on encouraging initial results.

Keywords: Monitoring · Timing uncertainty · Timed Büchi Automata.

1 Introduction

Online monitoring is an important tool to ensure functional correctness of safety-critical systems. It analyses the execution traces observed from the system during its runtime by determining in real-time whether the observed traces satisfy the system's specification. Continuous online monitoring consequently is concerned with unbounded time horizons, unlike offline monitoring where a fixed finite trace is analysed after the execution has terminated. Hence, specifications for online monitoring are typically defined over infinite traces, with the most significant approach being temporal logics. As specifications often include real-time requirements, e.g., "every request is answered within 10 milliseconds (ms)", we focus here on metric-time temporal logics over timed words. More precisely, we consider

* M. Fränzle has been funded by by the State of Lower Saxony, Zukunftslabor Mobilität, and by Deutsche Forschungsgemeinschaft, grants FR 2715/5-1 and FR 2715/6-1. T.M. Grosen and K.G. Larsen have been funded by the VILLUM Investigator grant S4OS, and together with M. Zimmermann they have been supported by DIREC - Digital Research Centre Denmark.

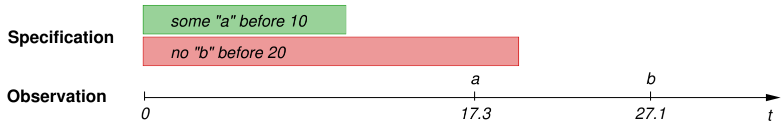

Fig. 1: Monitoring under observation delay: at time $t = 27.1$ we can conclusively decide that the MITL property $F_{[0,10]}a \wedge G_{[0,20]}\neg b$ is violated irrespective of the latency of the observation channel, provided the jitter is less than 0.2.

Metric Interval Temporal Logic (MITL) [2], which offers a good balance between expressiveness and algorithmic properties. For example, the request-response specification above is expressed by the MITL formula $G_{\geq 0}(\texttt{req} \rightarrow F_{\leq 10}\texttt{resp})$.

While the specifications classify infinite traces, the traces observed online and to be checked against the specification remain finite. Nevertheless, one can still return verdicts [7]: for example, *every* infinite extension of a finite trace with some request that is not answered within 10 ms violates the request-response specification above. Hence, violation of the specification is already witnessed by such a finite trace. Dually, consider the specification "system calibration is completed within 500 ms", expressed by the formula $F_{\leq 500}\texttt{cc}$ with the proposition $\texttt{cc}$ representing the completion of calibration. Every infinite extension of a finite trace on which the calibration is completed within 400 ms satisfies the specification. Hence, satisfaction of the specification is already witnessed by such a finite trace. However, there are also traces and specifications for which no verdict can currently be drawn, like in the situation where no calibration has been observed yet at current time of 350 ms. As usual, we capture these three situations with the three verdicts $\top$ (satisfaction for every extension), $\perp$ (violation for every extension), and **?** (inconclusive).

Online monitoring can be achieved by compiling the MITL specification into an equivalent timed Büchi automaton and then symbolically executing the automaton on the observed trace of the system [7,15]. However, this approach is correct only if the actions of the monitored system can be observed immediately by the monitor. In practice, there is usually a communication delay between the system and the monitor. This delay is induced by various types of circuitry at their interfaces, like technical sensors, conversion between analog and digital signals, and communication networks forwarding signals to the monitor. We follow the approach described in McGraw-Hill's Encyclopedia of Networking and Telecommunications [24] where a communication delay consists of a constant part (latency) and varying part (jitter).

Consequently, the monitored system and the symbolic execution are no longer synchronized but deviate by a delay, for which only bounds, yet not exact values tend to be known. But even then, one can still provide meaningful verdicts, see Fig. 1: The specification $F_{\leq 10}a \wedge G_{\leq 20}\neg b$ expresses that an a occurs within 10 ms and no b occurs within 20 ms. The observed trace shows the first a at 17.3 ms and the first b at 27.1 ms. This observation is only consistent with satisfaction of the constraint $F_{\leq 10}a$ if a's observation delay exceeds 7.3 ms, while satisfaction of $G_{\leq 20}\neg b$ requires a delay of at most 7.1 ms for b. Thus, if the jitter is strictly smaller than 0.2 ms, the specification is definitely violated. Note that the verdict "violated" is true independently of the actual value of the unknown, parametric communication latency.

On the other hand, if the parametric latency is known to be in the range $[4.5, 8]$ ms and the jitter is in $[0, 0.3]$ ms, then we cannot give a definitive verdict: The a may have occurred at 10 ms and then has been observed with 7 ms latency plus 0.3 ms jitter at $17.3 = 10 + 7 + 0.3$ ms, and the b may have occurred at 20.1 ms and then observed with the same latency (yet independent jitter) at $27.1 = 20.1 + 7 + 0$ ms. In this case, the property would be satisfied. But the a may also have occurred at 10.3 ms, violating the property, and still be observed with the same latency at $17.3 = 10.3 + 7 + 0$ ms. From the observations, we can nevertheless derive bounds on the parametric latency, as the property definitely is violated irrespective of the actual (unknown) value of the jitter whenever the actual latency is smaller than 7 ms or larger than 7.1 ms. It however cannot be guaranteed to be satisfied when the latency is in the range of $[7, 7.1]$ ms, as satisfaction then depends on the exact value of the jitter, which is not detectable. Thus, one can determine information beyond the verdicts $\top$, $\bot$, and **?** in terms of bounds on the delay that imply definitive verdicts.

Our Contribution. Based on previous work by Grosen et al. [15] on online monitoring of MITL specifications without delay via timed Büchi automata, we present a symbolic MITL monitoring algorithm that provides exact verdicts under unknown delay consisting of parametric (i.e. unknown within bounds) latency and jitter. While an unknown delay is a timing parameter, our construction avoids the semidecidability [3] of analysis for parameterized timed automata, and instead uses only classical clock zones [9].

In addition, our approach has the advantage that it is even more informative than typical monitoring algorithms, which only return a verdict in $\{\top, \bot, \textbf{?}\}$. Recall the example specification $F_{\leq 10}a \wedge G_{\leq 20}\neg b$ in the case where the jitter is constrained to [0,0.3] ms. As argued above, this specification can, given this bound on the jitter, only be satisfied if $7 \leq l \leq 7.1$, where l denotes the actual latency. Our algorithm, for which we also provide a prototype implementation and experimental evaluation, computes such parametric constraints on the set of potential latencies under which the specification can be satisfied as well as on the set of potential latencies under which the specification can be violated.

The implementation is built on top of the real-time model checking tool Uppaal [19] using the difference-bounded matrix (DBM) data structure allowing for representation of convex polytopes called zones. Most importantly, the DBM data structure can be used for efficient implementation of various geometrical operations over zones needed for the symbolic analysis of timed automata, such as testing for emptiness, inclusion, equality, and computing projection and intersection of zones [9]. Our experiments show encouraging initial results on an industrial gear controller model from [20].

All proofs omitted due to space restrictions can be found in the full version [14].

Related Work. Our automata-based monitoring of finite traces against specifications over infinite words using the three verdicts $\{\top, \bot, ?\}$ follows the seminal work of Bauer et al. [7], who presented monitoring algorithms for LTL and timed LTL. Their algorithm for timed LTL is based on clock regions [1], while we follow the approach of Grosen et al. [15] and use clock zones [9], whose performance is an order of magnitude faster. Also, they translated timed LTL into event-clock automata, which are less expressive than the timed Büchi automata (TBA) used both by Grosen et al. [15] and here. More recently, the same approach has been used to monitor real-time properties under assumptions [11].

As our algorithms work with TBA, we also support MITL specifications, as these can be compiled into TBA. The monitoring problem for MITL (without delay) has been investigated before. Baldor et al. showed how to construct a monitor for dense-time MITL formulas by constructing a tree of timed transducers [5]. Ho et al. split unbounded and bounded parts of MITL formulas for monitoring, using traditional LTL monitoring for the unbounded parts and permitting a simpler construction for the (finite-word) bounded parts [16]. Bulychev et al. apply a technique of rewriting a given WMTL formula during monitoring as part of performing statistical model checking. None of the above works makes use of the efficient DBM datastructure or extends to the setting of TBA that provides the basis of our approach. Here we note, that as a specification formalism TBA exceeds the expressive power of MITL, which might be useful in certain applications (e.g. in the presence of counting properties).

There is also a large body of work on monitoring with finite-word semantics. Roşu et al. focussed on discrete-time finite-word MTL [25], while Basin et al. proposed algorithms for monitoring real-time finite-word properties [6] and compared the differences between different time models. André et al. consider monitoring finite logs of parameterized timed and hybrid systems [29]. Finally, Ulus et al. described monitoring timed regular expressions over finite words using unions of two-dimensional zones [26,27].

The problem of monitoring trace properties under uncertain observation has been addressed before [12,23,28,13,17], most notably based on Signal Temporal Logic (STL), exploiting STL's quantitative semantics [22] that characterizes robustness against variation in state variables. These approaches are mostly orthogonal to ours, as they tend to address uncertainty in the state observed at a time instant rather than uncertainty in the time stamps associated to state observations. It would consequently be interesting to combine the two approaches, thus permitting both state uncertainty due to inexact measurements and time uncertainty due to inexact clocks and fluctuating communication latencies. It should also be noted that robust STL monitoring comes in diverse variants representing different error models, starting from monitors that exploit the compositional real-valued robustness semantics [12,23]. This semantics however underapproximates the factual robustness of the verdict against state shifts in the observed trace such that monitoring algorithms based on this compositional semantics are sound and computationally efficient, yet incomplete. Due to the safe approximation, they may yield inconclusive verdicts in actually determined situations. Complete and thus optimally informed STL monitoring under uncertainty, which guarantees a verdict whenever the property is determined, has only recently been investigated. Visconti et al. in [28] developed sound and complete monitoring wrt. an interval model of state measurement error, where each single measurement features an independent displacement ranging over a bounded interval. Finkbeiner et al. in [13] address a refined model distinguishing between a constant, yet unknown up to bounds, offset and a time-varying, interval-bounded noise, as suggested by the pertinent ISO norm 5725 on measurement accuracy (there called "trueness" and "precision" of a measurement). We here adopt the latter, more refined model of measurement error and transfer it into the time domain, thus implementing sound and complete monitoring for the case when timestamps are affected by a parametric (unknown, yet constant) observation latency plus a fluctuating jitter that differs between observations. Closest to our approach is [18], which addresses a more confined model of observation delay comprising a fixed known (non-parametric) latency plus a varying jitter. It also covers clock drift, which is an additional source of (relative) jitter that we have excluded to simplify the exposition.

2 Preliminaries

The set of natural numbers (excluding zero) is $\mathbb{N}$, we define $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, the set of rational numbers is $\mathbb{Q}$, the set of non-negative rational numbers is $\mathbb{Q}_{\geq 0}$. the set of real numbers is $\mathbb{R}$, and the set of non-negative real numbers is $\mathbb{R}_{\geq 0}$. The powerset of a set S is denoted by 2^S.

Timed Words. A timed word over a finite alphabet Σ is a pair $\rho = (\sigma, \tau)$ where σ is a nonempty word over Σ and τ is a sequence of non-decreasing non-negative real numbers of the same length as σ. Timed words may be finite or infinite; in the latter case, we require $\limsup \tau = \infty$, i.e., time diverges. The set of finite timed words is denoted by $T\Sigma^*$ and the set of infinite timed words by $T\Sigma^\omega$. We also represent a timed word as a sequence of pairs $(\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$. If $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots, (\sigma_n, \tau_n)$ is a finite timed word, we denote by $\tau(\rho)$ the total time duration of ρ, i.e., τ_n.

If $\rho_1 = (\sigma_1^1, \tau_1^1), \ldots, (\sigma_n^1, \tau_n^1)$ is a finite timed word, $\rho_2 = (\sigma_1^2, \tau_1^2), (\sigma_2^2, \tau_2^2), \ldots$ is a finite or infinite timed word, and $t \in \mathbb{Q}_{\geq 0}$ then the timed word concatenation $\rho_1 \cdot_t \rho_2$ is defined iff $\tau(\rho_1) \leq t$. Then, $\rho_1 \cdot_t \rho_2 = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$ such that

$$\sigma_i = \begin{cases} \sigma_i^1 & \text{iff } i \leq n \\ \sigma_{i-n}^2 & \text{else} \end{cases} \quad \text{and} \quad \tau_i = \begin{cases} \tau_i^1 & \text{iff } i \leq n \\ \tau_{i-n}^2 + t & \text{else.} \end{cases}$$

Timed Automata. A timed Büchi automaton (TBA) $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$ consists of a finite alphabet Σ, a finite set Q of locations, a set $Q_0 \subseteq Q$ of initial locations, a finite set C of clocks, a finite set $\Delta \subseteq Q \times Q \times \Sigma \times 2^C \times G(C)$ of transitions with $G(C)$ being the set of clock constraints over C, and a set $\mathcal{F} \subseteq Q$ of accepting locations. A transition (q, q', a, λ, g) is an edge from q to q' on input symbol a, where λ is the set of clocks to reset and g is a clock constraint over C. A clock constraint is a conjunction of atomic constraints of the form $x \sim n$, where x is a clock, $n \in \mathbb{N}_0$, and $\sim \in \{<, \leq, =, \geq, >\}$. A state of $\mathcal{A}$ is a pair (q, v) where q is a location in Q and $v \colon C \to \mathbb{R}_{\geq 0}$ is a valuation mapping clocks to their values. For any $d \in \mathbb{R}_{\geq 0}$, $v + d$ is the valuation $x \mapsto v(x) + d$.

A run of $\mathcal{A}$ from a state (q_0, v_0) over a timed word $(\sigma_1, \tau_1)(\sigma_2, \tau_2)\cdots$ is a sequence of steps $(q_0, v_0) \xrightarrow{(\sigma_1,\tau_1)} (q_1, v_1) \xrightarrow{(\sigma_2,\tau_2)} (q_2, v_2) \xrightarrow{(\sigma_3,\tau_3)} \cdots$ where for all $i \geq 1$ there is a transition $(q_{i-1}, q_i, \sigma_i, \lambda_i, g_i)$ such that $v_i(x) = 0$ for all x in λ_i and $v_i(x) = v_{i-1}(x) + (\tau_i - \tau_{i-1})$ otherwise, and g_i is satisfied by the valuation $v_{i-1} + (\tau_i - \tau_{i-1})$. Here, we use $\tau_0 = 0$. Given a run r, we denote the set of locations visited infinitely many times by r as $\mathrm{Inf}(r)$. A run r of $\mathcal{A}$ is accepting if $\mathrm{Inf}(r) \cap \mathcal{F} \neq \emptyset$. The language of $\mathcal{A}$ from a starting state (q, v), denoted $L(\mathcal{A}, (q, v))$, is the set of all infinite timed words with an accepting run in $\mathcal{A}$ starting from (q, v). We define the language of $\mathcal{A}$, written $L(\mathcal{A})$, to be $\bigcup_q L(\mathcal{A}, (q, v_0))$, where q ranges over Q_0 and where $v_0(x) = 0$ for all $x \in C$.

Logic. We use Metric Interval Temporal Logic (MITL) to express properties to be monitored; these are subsequently translated into equivalent TBA which we use in our monitoring algorithm. The syntax of MITL formulas over a finite alphabet Σ is defined as

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid X_I\varphi \mid \varphi\, U_I\varphi$$

where $p \in \Sigma$ and I ranges over non-singular intervals over $\mathbb{R}_{\geq 0}$ with endpoints in $\mathbb{N}_0 \cup \{\infty\}$. We write $\sim n$ for $I = \{d \in \mathbb{R}_{\geq 0} \mid d{\sim}n\}$ for $\sim \in \{<, \leq, \geq, >\}$ and $n \in \mathbb{N}$. We also define the standard syntactic sugar: $\texttt{true} = p \vee \neg p$, $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, $F_I\varphi = \texttt{true}\ U_I\varphi$, and $G_I\varphi = \neg F_I\neg\varphi$.

The satisfaction relation $\rho, i \models \varphi$ is defined for infinite timed words $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$, positions $i \geq 1$, and an MITL formulas φ:

- $\rho, i \models p$ iff $p = \sigma_i$.
- $\rho, i \models \neg\varphi$ iff $\rho, i \not\models \varphi$.
- $\rho, i \models \varphi \vee \psi$ if $\rho, i \models \varphi$ or $\rho, i \models \psi$.
- $\rho, i \models X_I\varphi$ iff $\rho, (i+1) \models \varphi$ and $\tau_{i+1} - \tau_i \in I$.
- $\rho, i \models \varphi\, U_I\psi$ iff there exists $k \geq i$ s.t. $\rho, k \models \psi$, $\tau_k - \tau_i \in I$, and $\rho, j \models \varphi$ for all $i \leq j < k$.

We write $\rho \models \varphi$ whenever $\rho, 1 \models \varphi$. The language $L(\varphi)$ of an MITL formula φ is the set of all $\rho \in T\Sigma^\omega$ such that $\rho \models \varphi$.

Theorem 1 ([2,10]). *For each MITL formula φ there exists a TBA $\mathcal{A}$ with $L(\varphi) = L(\mathcal{A})$.*

Fig. 2 illustrates Theorem 1 by providing TBA's for the formula $F_{[0,10]}a \wedge G_{[0,20]}\neg b$ from the introduction and its negation.

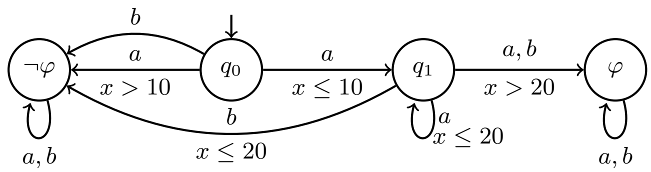

Fig. 2: An automaton for the language of the property $\varphi = F_{[0,10]}a \wedge G_{[0,20]}\neg b$ and its negation: If location φ is accepting then it accepts $L(\varphi)$, if location $\neg\varphi$ is accepting then it accepts $L(\neg\varphi)$.

3 Monitoring under Delayed Observation

According to McGraw-Hill's Encyclopedia of Networking and Telecommunications [24], a communication delay consists of a constant part (latency) and varying part (jitter). We describe the delay as a pair $(\delta, \varepsilon) \in \mathbb{R}^2_{\geq 0}$ where δ is the constant latency for all signals and ε is the bound on the jitter. Thus, all signals from the system are delayed within $[\delta, \delta + \varepsilon]$ before they arrive at the monitor.

In the simplest case, our obligation is to monitor violation of an MITL specification φ by a system while observing the events through a channel *Chan* featuring a constant, yet unknown (up to a given, but maybe trivial, lower bound $l \in \mathbb{R}_{\geq 0}$ and upper bound $u \in \mathbb{R}_{\geq 0}$) transportation latency $\delta \in [l, u]$ and a varying jitter bounded by $\varepsilon \in \mathbb{R}_{\geq 0}$. Fig. 1 shows an example of a property and an observation that conclusively violates the specification at time 27.1, even if the channel latency $\delta \in [0, \infty[$ is unknown, as long as the jitter is bounded by 0.2.

Thus, we need to distinguish between observations (the timed word corresponding to the events as they are observed by the monitoring device, subject to delay) and the possible ground-truths, as they may have been emitted by the the monitored system. We begin by formalizing the concept of observation, where the occurrence of observed events is constrained by a set $\mathcal{D}$ capturing known bounds on the delay. Obviously, under latency δ (and arbitrary jitter), the first observation can only be made after at least δ units of time.

Definition 1. *A delay set $\mathcal{D}$ is a nonempty subset of $\mathbb{R}^2_{\geq 0}$ containing pairs of latencies and jitters. A $\mathcal{D}$-observation, i.e. an observation that can in principle be made under delay in $\mathcal{D}$, is a finite timed word $\rho^* = (\sigma_1^*, \tau_1^*), \ldots, (\sigma_m^*, \tau_m^*)$ with $\tau_1^* \geq \delta$ for some $(\delta, \varepsilon) \in \mathcal{D}$.*

As the ground-truth occurrence times of events in the system cannot be determined exactly from their delayed copies that the monitor receives through the communication channel, we have to consider all ground-truth timed words that the particular observation is consistent with, as follows.[3]

[3] Note that we simplify our definitions by assuming that jitter does not change the order of observations. Under the additional assumption that only a (uniformly) bounded number of events can be generated by the system in each unit of time,

Definition 2 (Consistency). *Let $\rho^* = (\sigma_1^*, \tau_1^*), \ldots, (\sigma_m^*, \tau_m^*)$ be a $\{(\delta, \varepsilon)\}$-observation and let $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ be a finite timed word. We say that ρ is* consistent *with ρ^* at observation time $t \in \mathbb{R}_{\geq 0}$ under latency δ and jitter ε iff*

1. *$\tau_n \leq t$ and $\tau_m^* \leq t$,*
2. *$n \geq m$, and $\sigma_i = \sigma_i^*$ and $\tau_i + \delta - \tau_i^* \in [0, \varepsilon]$ for all $i \in \{1, \ldots, m\}$, and*
3. *if $n > m$ then $\tau_{m+1} \geq t - (\delta + \varepsilon)$.*

We denote the set of timed words ρ that are consistent with a $\{(\delta, \varepsilon)\}$-observation ρ^* at observation time t under latency δ and jitter ε by $GT_{\delta,\varepsilon}(\rho^*, t)$. Then, we define $GT_{\mathcal{D}}(\rho^*, t) = \bigcup_{(\delta,\varepsilon) \in \mathcal{D}} GT_{\delta,\varepsilon}(\rho^*, t)$.

$GT_{\mathcal{D}}(\rho^*, t)$ thus collects the possible ground-truths that are consistent with the observation ρ^* when the time elapsed since the system has started is t, and the delay (δ, ε) is within the set $\mathcal{D}$. Note that $GT_{\mathcal{D}}(\rho^*, t)$ is always nonempty, if ρ^* is a $\mathcal{D}$-observation and $t \geq \tau(\rho^*)$ (recall that $\tau(\rho^*)$ denotes the last time point of ρ^*).

Example 1. Fig. 3 shows a $\{(\delta, \varepsilon)\}$-observation and a consistent ground-truth and illustrates how the delay shifts the timestamps of the events. The length of ρ is $n = 9$ and the length of ρ^* is $m = 4$. Recall that t is the time of observation.

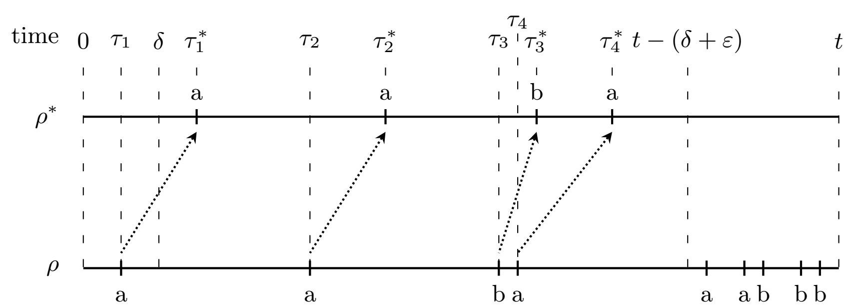

Fig. 3: A $\{(\delta, \varepsilon)\}$-observation ρ^* and a consistent ground-truth ρ.

In particular, notice the following:

– No event can occur in the observation ρ^* with a timestamp smaller than δ, as it takes at least δ units of time for an event to reach to be send from the system through the communication channel to the monitor. Obviously, at the system side (i.e., in the ground-truth ρ) events can happen at any timestamp, also before δ (e.g., the first a).

it is possible to take "overtaking" of events into account, by looking at all consistent permutations. However, this would lead to a severe overhead in the implementation.

- The difference $\tau_i^* - \tau_i$ for $i \leq 4$ (i.e., the difference between the observation time and the time the event was emitted) must be in the interval $[\delta, \delta + \varepsilon]$.
- The time elapsed between the b and the last a in the observation ρ^* is larger than the time elapsed between the corresponding events in the ground-truth ρ. This means the jitter for the a is larger than the jitter for the b.
- The last five events in ρ have not yet been observed in ρ^*. Such events can only have timestamps in the interval $[t - (\delta + \varepsilon), t]$, as all earlier events must necessarily have been observed. Said differently, there cannot be any events between timestamp τ_4 (corresponding to the last observed event in ρ^* with timestamp τ_4^*) and timestamp $t - (\delta + \varepsilon)$, as any such event would have arrived at the monitor, even under the maximal possible delay of $\delta + \varepsilon$. However, there can be an arbitrary number of events in ρ between timestamps $t - (\delta + \varepsilon)$ and t.

A monitor obviously ought to supply a verdict iff that verdict applies across *all possible* ground-truth timed words that the observed word explains. To define our definition of monitor, we use the set $\mathbb{B}_3 = \{\top, \textbf{?}, \bot\}$ of verdicts, as usual.

Definition 3 (Monitor verdicts under delay). *Given a language $L \subseteq T\Sigma^\omega$, a set of possible observation delays $\mathcal{D}$, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and an observation time $t \geq \tau(\rho^*)$, the function $\mathcal{V}_\mathcal{D} \colon 2^{T\Sigma^\omega} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ evaluates to the verdict*

$$\mathcal{V}_\mathcal{D}(L)(\rho^*, t) = \begin{cases} \top & \text{if } \rho \cdot_t \mu \in L \text{ for all } \rho \in GT_\mathcal{D}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega, \\ \bot & \text{if } \rho \cdot_t \mu \notin L \text{ for all } \rho \in GT_\mathcal{D}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega, \\ \textbf{?} & \text{otherwise.} \end{cases}$$

$\mathcal{V}_\mathcal{D}(L)(\rho^*, t)$ is undefined when $t < \tau(\rho^*)$.

Example 2. Consider the property $\varphi = F_{[0,10]} a \wedge G_{[0,20]} \neg b$ and observed word $\rho^* = (a, 17.3), (b, 27.1)$ shown in Fig. 1, time point $t = 27.1$, and set of delays $\mathcal{D} = \{(\delta, \varepsilon) \mid \varepsilon = 0.2\}$. As the jitter is bounded by 0.2, in all ground truths either a occurred after time point 10, or b occurred before time point 20. Thus, all extensions of all possible ground truths satisfy $\neg\varphi$, i.e., $\mathcal{V}_\mathcal{D}(L(\varphi)))(\rho^*, t) = \bot$.

Note that for the special case of $\mathcal{D} = \{(0, 0)\}$ we cover classical (i.e., delay-free) monitoring [15]. Before we turn our attention to computing $\mathcal{V}$ we study some properties of our definition. First, let us note that the ability to make firm verdicts increases with increased certainty of the observation channel delay.

Lemma 1. *Let $L \subseteq T\Sigma^\omega$, $\rho^* \in T\Sigma^*$, let $\mathcal{D} \subseteq \mathcal{D}'$ be delay sets, let ρ^* be a $\mathcal{D}$-observation, and let $t \geq \tau(\rho^*)$. Then, $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \top$ implies $\mathcal{V}_\mathcal{D}(L)(\rho^*, t) = \top$ and $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \bot$ implies $\mathcal{V}_\mathcal{D}(L)(\rho^*, t) = \bot$.*

As a refinement of the verdict function in Definition 3, one may provide information about the delay parameters (δ, ε) that can explain an observation.

Given $L \subseteq T\Sigma^\omega$, a finite timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, the set of delays $\Delta(L, \rho^*, t)$ that are consistent with the observation ρ^* at t is defined as

$$\Delta(L, \rho^*, t) = \{(\delta, \varepsilon) \mid \exists \rho \in GT_{\delta,\varepsilon}(\rho^*, t)\, \exists \mu \in T\Sigma^\omega \text{ s.t. } \rho \cdot_t \mu \in L\}.$$

We denote by $\Delta_{\mathcal{D}}(L, \rho^*, t)$ the set $\Delta(L, \rho^*, t) \cap \mathcal{D}$. Now we can characterize the conclusive monitoring verdicts via these delay sets.

Lemma 2. *Given $L \subseteq T\Sigma^\omega$, a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we have*

1. *$\Delta_{\mathcal{D}}(L, \rho^*, t) = \emptyset$ iff $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$, and*
2. *$\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) = \emptyset$ iff $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$.*

But even in the case when both delay-sets are nonempty (i.e., the verdict is *?*), we can still provide useful information in terms of the sets $\Delta(L, \rho^*, t)$ and $\Delta(\overline{L}, \rho^*, t)$ of consistent delays. In particular, the set of consistent delays is non-increasing during observations. To explain this, we first define a notion of extensions for finite timed words (w.r.t. current time instants t, t'). Formally, let $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ and $\rho' = (\sigma'_1, \tau'_1), \ldots, (\sigma'_{n'}, \tau'_{n'})$ be two finite timed words. Also let $t, t' \in \mathbb{R}_{\geq 0}$. Then, we define $(\rho, t) \sqsubseteq (\rho', t')$, if $n \leq n'$, $\sigma_i = \sigma'_i$ and $\tau_i = \tau'_i$ for all $i \leq n$, and either $n = n'$ and $t \leq t'$ or $n < n'$ and $t \leq \tau'_{n+1}$. By extending the observations, we (potentially) reduce the set of consistent delays.

Lemma 3. *Let $(\rho^*_1, t_1) \sqsubseteq (\rho^*_2, t_2)$ for finite timed words ρ^*_1, ρ^*_2 and $t_1 \geq \tau(\rho^*_1)$ and $t_2 \geq \tau(\rho^*_2)$. Then, $\Delta(L, \rho^*_1, t_1) \supseteq \Delta(L, \rho^*_2, t_2)$.*

Another interesting point is that in some cases, no extension of the observed word will provide a definitive verdict.

Example 3. Consider the language $L(F_{\leq 10} a)$, the observation $\rho^* = (a, 15)$, and the set $\mathcal{D} = \{(\delta, 0) \mid \delta \in [0, 10]\}$ of delays. For any given $t \geq \tau(\rho^*)$ the sets of consistent delays are $\Delta_{\mathcal{D}}(L, \rho^*, t) = \{(\delta, 0) \mid \delta \in [5, 10]\}$ and $\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) = \{(\delta, 0) \mid \delta \in [0, 5)\}$, i.e., both sets of consistent delays are a strict subset of $\mathcal{D}$. Further, due to Lemma 3, this will be the case, no matter what observations occur in the future, as the set of consistent delays can only shrink when further observations are made. So, the verdict is *?*, even if additional observations occur.

The following lemma formalizes this: as soon as the set of consistent delays w.r.t. L ($\overline{L}$) is no longer equal to $\mathcal{D}$, then the verdict can never become $\top$ ($\bot$).

Lemma 4. *Let $L \subseteq T\Sigma^\omega$, $\mathcal{D}$ be a set of delays, and $\rho^* = (\sigma^*_1, \tau^*_1), \ldots, (\sigma^*_m, \tau^*_m)$ a nonempty $\mathcal{D}$-observation. Then, for all $t > \tau(\rho^*)$*

1. *$\Delta_{\mathcal{D}}(L, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau^*_1\}$ implies there is no $\rho^*_1 \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho^*_1, t') = \top$ for any $t' \geq t + \tau(\rho^*_1)$, and*
2. *$\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau^*_1\}$ implies there is no $\rho^*_1 \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho^*_1, t') = \bot$ for any $t' \geq t + \tau(\rho^*_1)$.*

Note that $\Delta_{\mathcal{D}}(L, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau^*_1\}$ and $\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau^*_1\}$ can both be true simultaneously (as in Example 3). In this situation, we will under no future observation reach a conclusive verdict.

4 Towards an Algorithm

Typically, monitoring algorithms rely on automata-based techniques. To this end, first the specification and its complement are translated into suitable automata. Then one computes the set of states reachable by processing the observation and then checks whether from one of these states the automaton can still accept an infinite continuation. If this is the case for both automata, then the verdict is **?**, if it is only the case for the automaton for the specification, then the verdict is $\top$, and vice versa for the complement automaton and $\bot$.

We want to follow the same blueprint, but we need to make adjustments to handle delay. Intuitively, we need to compute all states that are reachable by possible ground-truths of a given observation. However, a ground-truth may contain more events than the observation, as some events may not yet have been observed due to delay. This complicates the construction of the set of reachable states, as an unbounded number of events may not yet have been observed.

In the definition of $GT_{\mathcal{D}}$ (Definition 2) there is an implicit universal quantification over all possible sequences of such events that have not yet been observed (e.g., the last five events in ρ in Fig. 3). We exploit the fact that the verdicts are defined with respect to all possible extensions μ of a possible ground-truth (i.e., also a universal quantification over the μ's) to "merge" the universal quantification over events that have not yet been observed into the universal quantification of the extension μ. Then, a possible ground-truth has exactly the same number of events as the observation (i.e., ground-truth and observation have Equal Length). We begin defining this restricted notion of possible ground-truth by strengthening Definition 2.

Definition 4 (EL-Consistency). *Let $\rho^* = (\sigma_1^*, \tau_1^*), \ldots, (\sigma_m^*, \tau_m^*)$ be a $\{(\delta, \varepsilon)\}$-observation and let $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ be a finite timed word. We say that ρ is* EL-consistent *with ρ^* at observation time $t \in \mathbb{R}_{\geq 0}$ under latency δ and jitter ε iff ρ is consistent with ρ^* at t under δ and ε and $m = n$. We denote the set of timed words ρ that are EL-consistent with an $\{(\delta, \varepsilon)\}$-observation ρ^* at observation time t under latency δ and jitter ε by $GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ and define $GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*, t) = \bigcup_{(\delta,\varepsilon)\in\mathcal{D}} GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$.*

Example 4. Continuing Example 1, an EL-consistent ground-truth of the observation ρ^* in Fig. 3 has exactly four events corresponding to the four events in the observation. Thus, there cannot be any unobserved events between $t-(\delta+\varepsilon)$ and t in an EL-consistent ground-truth (e.g., the last five events of ρ in Fig. 3).

Now, we present the revised verdict function using only EL ground-truths. Note that merging the unobserved events from the possible ground-truth ρ into the extension μ requires changing the time instant at which we concatenate the ground-truth and the extension: $t - (\delta + \varepsilon)$ is the earliest time point at which an event can occur that may not yet have been observed at time t. Due to jitter however, there might also be events after $t - (\delta + \varepsilon)$ that have been observed, which are in the possible ground-truth ρ: the last such event happened at time $\tau(\rho)$. Hence, we need to concatenate at time point $\max(\tau(\rho), t - (\delta + \varepsilon))$.

Definition 5 (Monitor verdicts under delay – EL version). *Given $L \subseteq T\Sigma^\omega$, a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, the function $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}} \colon 2^{T\Sigma^\omega} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ evaluates to the verdict*

$$\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \begin{cases} \top & \text{if } \rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \mu \in L \text{ for all } (\delta, \varepsilon) \in \mathcal{D}, \\ & \text{all } \rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega, \\ \bot & \text{if } \rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \mu \notin L \text{ for all } (\delta, \varepsilon) \in \mathcal{D}, \\ & \text{all } \rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega, \\ ? & \text{otherwise.} \end{cases}$$

$\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t)$ is undefined when $t < \tau(\rho^*)$.

Next, we show that both verdict functions coincide.

Lemma 5. *$\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \mathcal{V}_{\mathcal{D}}(L)(\rho^*, t)$ for all $L \subseteq T\Sigma^\omega$, all sets $\mathcal{D}$ of delays, all $\mathcal{D}$-observations ρ^*, and all $t \geq \tau(\rho^*)$.*

Next, we show that we can indeed make the definition of $\mathcal{V}^{\mathrm{el}}$ effective using automata-theoretic constructions. First, we formally capture the set of states that can be reached by processing the possible EL ground-truths of an observation. Let $\mathcal{A}$ be a TBA. We write $(q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q_n, v_n)$ for a finite timed word $\rho = (\sigma, \tau) \in T\Sigma^*$ to denote the existence of a finite sequence of states $(q_0, v_0) \xrightarrow{(\sigma_1,\tau_1)} (q_1, v_1) \xrightarrow{(\sigma_2,\tau_2)} \cdots \xrightarrow{(\sigma_n,\tau_n)} (q_n, v_n)$ of $\mathcal{A}$ where for all $1 \leq i \leq n$ there is a transition $(q_{i-1}, q_i, \sigma_i, \lambda_i, g_i)$ of $\mathcal{A}$ such that $v_i(x) = 0$ for all x in λ_i and $v_{i-1}(x) + (t_i - t_{i-1})$ otherwise, and g is satisfied by the valuation $v_{i-1} + (t_i - t_{i-1})$, where we use $t_0 = 0$. Given a TBA $\mathcal{A}$, a set $\mathcal{D}$ of delays, a finite observed timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we define

$$\begin{aligned} \mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q, v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))) \mid{}& (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where} \\ & (q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } x \in C, \text{ and} \\ & \rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ for some } (\delta, \varepsilon) \in \mathcal{D}\}. \end{aligned}$$

We call this the reach-set of ρ in $\mathcal{A}$ at t w.r.t. $\mathcal{D}$.

Next, we define the set of states of a TBA from where it is possible to reach an accepting location infinitely many times in the future, i.e., those states from which an accepting run is possible. This is useful, because if processing a finite timed word leads to such a state, then the timed word can be extended to an infinite one in the language of the automaton, a notion that underlies the definitions of the verdict functions. Given a TBA $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$, the set of states with nonempty language is

$$S^{ne}_{\mathcal{A}} = \{(q, v) \mid q \in Q, v \in C \to \mathbb{R}_{\geq 0} \text{ s.t. } L(\mathcal{A}, (q, v)) \neq \emptyset\}.$$

The set $S^{ne}_{\mathcal{A}}$ can be computed using a zone-based fixpoint algorithm [15]. Using these definitions, we can give an *effective* definition of the verdict functions, which we show to be equivalent to the previous definitions and implementable.

In the following definition, $\mathbb{A}$ denotes the set of all TBA.

Definition 6 (Monitoring TBA). *Given a TBA $\mathcal{A}$, a complement automaton $\overline{\mathcal{A}}$ (i.e., with $L(\overline{\mathcal{A}}) = T\Sigma^\omega \setminus L(\mathcal{A})$), a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho)$, $\mathcal{M}_\mathcal{D} \colon \mathbb{A} \times \mathbb{A} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ computes the verdict*

$$\mathcal{M}_\mathcal{D}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \begin{cases} \top & \text{if } \mathcal{R}^\mathcal{D}_{\overline{\mathcal{A}}}(\rho^*, t) \cap S^{ne}_{\overline{\mathcal{A}}} = \emptyset, \\ \bot & \text{if } \mathcal{R}^\mathcal{D}_{\mathcal{A}}(\rho^*, t) \cap S^{ne}_{\mathcal{A}} = \emptyset, \\ ? & \text{otherwise.} \end{cases}$$

$\mathcal{M}_\mathcal{D}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t)$ is undefined if $t < \tau(\rho)$.

Next we show that this automata-based definition of monitoring is equal to the verdict functions defined above.

Theorem 2. *$\mathcal{M}_\mathcal{D}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \mathcal{V}^{\mathrm{el}}_\mathcal{D}(L(\mathcal{A}))(\rho^*, t)$ for all sets $\mathcal{D}$ of delays, all TBA $\mathcal{A}$ (and complement automata $\overline{\mathcal{A}}$), all $\mathcal{D}$-observations ρ^*, and all $t \geq \tau(\rho^*)$.*

Recall that $S^{ne}_\mathcal{A}$ can be computed for any given TBA $\mathcal{A}$. Therefore, in the next section, we show how to calculate $\mathcal{R}^\mathcal{D}_\mathcal{A}(\rho^*, t)$ for a given TBA $\mathcal{A}$, set $\mathcal{D}$ of delays, observation ρ^*, and time point t using a zone-based algorithm. This will then allow us to compute verdicts effectively.

5 A Zone-Based Online Monitoring Algorithm

In this section, we demonstrate how to compute the reach-set of ρ^* in $\mathcal{A}$ at t w.r.t. $\mathcal{D}$. So far we have developed the theory with observations, latency, and jitter being reals. Now, we are concerned with algorithms and thus assume all these quantities to be rationals. For the monitoring algorithm, we use – as standard in analysing timed automata models – symbolic states being pairs (q, Z) of locations and zones. A zone is a finite conjunction of constraints of the form $x \sim t$ and $x - x' \sim t$ for clocks x, x', constants $t \in \mathbb{Q}_{\geq 0}$, and $\sim \in \{<, \leq, =, \geq, >\}$. Given two zones Z and Z' over a set C of clocks, and a set of clocks $\lambda \subseteq C$, we define the following operations on zones (which can be efficiently implemented using the DBM data-structure [9]):

- $Z[\lambda] = \{v \mid \exists v' \models Z \text{ s.t. } v(x) = 0 \text{ if } x \in \lambda, \text{ otherwise } v(x) = v'(x)\}$
- $Z^\nearrow = \{v \mid \exists v' \models Z \text{ s.t. } v = v' + d \text{ for some } d \in \mathbb{R}_{\geq 0}\}$
- $Z \wedge Z' = \{v \mid v \models Z \textit{ and } v \models Z'\}$

We can use these functions to compute the successor states after an input. Given a TBA $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$, a symbolic state (q, Z), and a letter $a \in \Sigma$, we define

$$\mathrm{Post}((q, Z), a) = \{(q', Z') \mid (q, q', a, \lambda, g) \in \Delta, Z' = (Z^\nearrow \wedge g)[\lambda]\},$$

as the set of states one can reach by taking an a-transition at some point in the future from (q, Z). Using Post we can compute the successor states of a timed input $(a, \tau) \in \Sigma \times \mathbb{Q}_{\geq 0}$ by extending the zones with an additional clock *time* just recording time since system start. The successors of a symbolic state is

$$\mathrm{Succ}((q, Z), (a, \tau)) = \{(q', Z') \mid (q', Z'') \in \mathrm{Post}((q, Z), a), Z' = Z'' \wedge \mathit{time} = \tau\}$$

and the successors of a set of symbolic states S is

$$\mathrm{Succ}(S, (a, \tau)) = \bigcup\nolimits_{(q', Z') \in S} \mathrm{Succ}((q', Z'), (a, \tau)).$$

In handling delayed observations, we assume that the delay set $\mathcal{D}$ consists of pairs (δ, ε) where the latency δ is bounded by an interval $[l, u] \subseteq \mathbb{R}_{\geq 0}$ for given $l, u \in \mathbb{Q}_{\geq 0}$, and that the jitter is bounded by a given $\varepsilon \in \mathbb{Q}_{\geq 0}$.

To represent the latency and thereby be able to reason about and indirectly store the latency bounds, we add a clock *etime* representing the "expected" real time that an event generated just now could be observed by the monitor after having been delayed according to the latency. This allows us

1. to represent the actual latency as $\mathit{etime} - \mathit{time}$,
2. to represent the initial knowledge about latencies by initializing $\mathit{etime} - \mathit{time}$ to the initially known bounds on latency, namely $\mathit{etime} - \mathit{time} \in [l, u]$ by setting *time* to 0 and constraining *etime* to $[l, u]$, and
3. to refine our knowledge about the actual latency after having observed an event (σ^*, τ^*) by then setting *etime* to a value in $[\tau^* - \varepsilon, \tau^*]$.

Consequently, we change the initial zones to include the latency bounds l and u as the differences between the clocks *etime* and *time*. This way, *etime* represents the expected time an event is observed at the monitor, given l and u, and *time* represents the actual time the event happened (at the system). The aforementioned refinement (see Item 3 above and Fig. 4) then permits to deduce actual latency ranges consistent with the specification (or its negation) from observation times of events.

In detail, this refinement of the $\mathit{etime} - \mathit{time}$ relation works as follows. Given a TBA $\mathcal{A}$ extended with the clocks *time* and *etime*, and an observation $(\sigma, \tau^*) \in \Sigma \times \mathbb{Q}_{\geq 0}$, the successors of (q, Z) are

$$\begin{aligned} \mathrm{Succ}_d((q, Z), (\sigma, \tau^*)) = \{(q', Z') \mid (q', Z'') \in \mathrm{Post}((q, Z), \sigma), \\ Z' = Z'' \wedge \mathit{etime} \leq \tau^* \wedge \mathit{etime} \geq \tau^* - \varepsilon\} \end{aligned}$$

and the successors $\mathrm{Succ}_d(S, (\sigma, \tau^*))$ of a set of symbolic states S is equal to $\cup_{(q,Z) \in S} \mathrm{Succ}_d((q, Z), (\sigma, \tau^*))$.

The online monitoring algorithm will essentially apply Succ_d repeatedly to update the reach-set, once for each new observation. Note that there is a slight mismatch, as Succ_d is computed with the two auxiliary clocks *time* and *etime*, which are not clocks of $\mathcal{A}$.

The initial reach-set is given by the following zone Z_0^d requiring all ordinary clocks of the TBA $\mathcal{A}$ to be zero and with *time* and *etime* satisfying $\mathit{etime} - \mathit{time} \in [l, u]$. That is

$$Z_0^d \equiv \underbrace{etime - time \leq u \wedge time - etime \leq -l}_{etime - time \in [l,u]} \wedge \underbrace{\bigwedge\nolimits_{x \in C \cup \{time\}} x = 0}_{x_1, \ldots, x_{|C|} = 0,\ time = 0} .$$

Given a fixed jitter bound ε, we can now compute the reach-set after a sequence of observations under delay, where the latency is bounded in $[l, u]$.

Theorem 3. *Given a TBA $\mathcal{A}$, a delay set $\mathcal{D} = \{(\delta, \varepsilon) \mid \delta \in [l, u]\}$ with $l, u, \varepsilon \in \mathbb{Q}_{\geq 0}$, a $\mathcal{D}$-observation $\rho^* = (\sigma_1, \tau_1^*), \ldots, (\sigma_n, \tau_n^*)$, and $t \in \mathbb{Q}_{\geq 0}$ with $t \geq \tau_n^*$, let $S_0 = \{(q_0, Z_0^d) \mid q_0 \in Q_0\}$ and $S_i = \mathrm{Succ}_d(S_{i-1}, (\sigma_i, \tau_i^*))$ for $i \in \{1, \ldots, n\}$. Then, the reach-set $\mathcal{R}_{\mathcal{A}}^{\mathcal{D}}(\rho^*, t)$ is the projection of*

$$\{(q', Z') \mid (q', Z'') \in S_n, Z' = Z''^{\nearrow} \wedge etime = t - \varepsilon\}$$

to the clocks of $\mathcal{A}$ (obtained by removing all constraints on time and etime).

This theorem allows us to implement a monitoring algorithm by computing the reach-sets and intersecting them with the set of nonempty language states.

The observation of events may lead to refinement of the difference between *time* and *etime* as depicted in Fig. 4.

Lemma 6. *Given $\mathcal{A}$, $\mathcal{D}$, ρ^*, t, and S_n as in Theorem 3, we can compute the set of consistent delays by looking at the bounds on etime − time:*
$\Delta_{\mathcal{D}}(L(\mathcal{A}), \rho^*, t) = \{(\delta, \varepsilon) \in \mathcal{D} \mid S_n \models etime - time = \delta\}$.

This information can be used to decorate the ***?*** verdict, so that we can report a set of bounds on the latency for which we would provide a $\top$ or $\bot$ verdict.

Example 5. Let us show an example of our algorithm for monitoring under delayed observation. Note that, for the sake of readability, we use sets of clock constraints instead of conjunction of clock constraints when specifying zones.

Consider the property $\varphi = F_{[0,10]} a \wedge G_{[0,20]} \neg b$ from Fig. 1. The TBA's accepting $L(\varphi)$ and $L(\neg\varphi)$ are shown in Fig. 2. The nonempty language states for $\mathcal{A}_\varphi$ and $\mathcal{A}_{\neg\varphi}$ are $S_{\mathcal{A}_\varphi}^{ne} = \{(q_0, \{x \leq 10\}), (q_1, \mathtt{true}), (\varphi, \mathtt{true})\}$ and $S_{\mathcal{A}_{\neg\varphi}}^{ne} = \{(q_0, \mathtt{true}), (q_1, \{x \leq 20\}), (\neg\varphi, \mathtt{true})\}$. Let us assume the latency is between 0 and 10, and the jitter is bounded by 0.2. Now we compute the reach-sets S_0 (initial), S_1 (after $(a, 17.3)$), and S_2 (after $(b, 27.5)$) as

$$\begin{aligned}
S_0 &= \{(q_0, \{x = 0,\ etime \leq 10,\ (etime - x) \in [0, 10]\})\}, \\
S_1 &= \{(q_1, \{x \in [7.1, 10],\ etime \in [17.1, 17.3],\ (etime - x) \in [7.1, 10]\}), \\
&\qquad (\neg\varphi, \{x \in [10, 17.3],\ etime \in [17.1, 17.3],\ (etime - x) \leq 7.3\})\}, \text{ and} \\
S_2 &= \{(\varphi, \{x \in]20, 20.4], etime \in [27.3, 27.5[,\ (etime - x) \in [7.1, 7.5]\}), \\
&\qquad (\neg\varphi, \{x \in [17.3, 27.5],\ etime \in [27.3, 27.5],\ (etime - x) \in [0, 10]\})\}.
\end{aligned}$$

Note that we omit the clock *time* and only look at x and *etime* since *time* and x always have the same constraints.

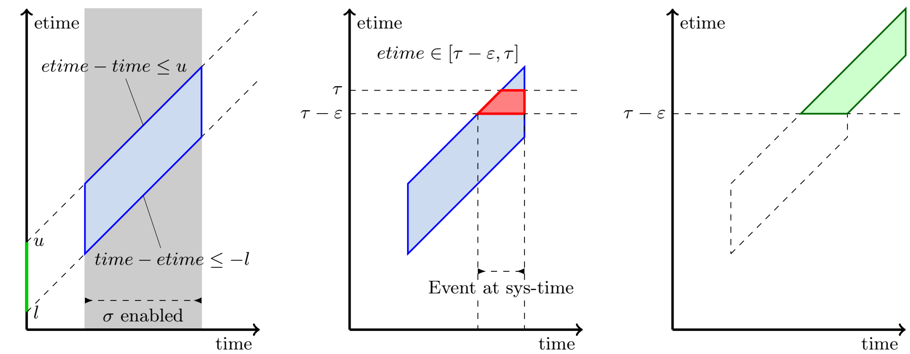

Fig. 4: Illustration of a single zone in the Succ_d computation (only *time*-*etime* plane depicted). Left: initial zone (in green) is diagonally extrapolated for time passage and then intersected with the guard of an edge. Middle: observing event σ at time τ. By restricting *etime* to $[\tau - \varepsilon, \tau]$, the clock *time* is restricted to when the event could have occurred at the system. Right: computing the future zone we see that the bound on $time - etime$ is now stricter and thus the bounds for the consistent latencies are refined.

All reach-sets intersect with both sets of nonempty language states; thus, the verdict is *?*. However, we can refine this verdict with knowledge about the consistent delays that change after each observation. The jitter bound is fixed at 0.2, but the bounds on the latency can be found in the clock constraints on the difference between *etime* and x. For $\bot$, the latency range remains $[0, 10]$ in all reach-sets. For $\top$, the consistent latency range is $[0, 10]$ in S_0, $[7.1, 10]$ in S_1, and it is $[7.1, 7.5[$ in S_2. This means that if the latency is outside $[7.1, 7.5[$, then the verdict is $\bot$.

On the other hand, for the observation $\rho^* = (a, 17.3), (b, 27.1)$ from Example 2 (and using the same latency and jitter bounds as above), we compute the reach-sets S_0, S_1, and S_2' where

$$S_2' = \{(\neg\varphi, \{x \in [16.9, 27.1], etime \in [26.9, 27.1], (etime - x) \in [0, 10]\})\}.$$

As S_2' has an empty intersection with $S^{ne}_{\mathcal{A}_\varphi}$, the verdict is $\bot$.

6 Prototype Implementation

We implemented the methods described in this paper in the tool MoniTAal[4] written in C++. This includes the difference-bounded matrix data structure to handle clock-zones, parsing property automata modelled in Uppaal, computing the set of nonempty language states, computing the reach-sets in an online fashion over an observed word based on latency and jitter bounds in $[0, \infty[$, providing verdicts $\top, \bot$ or *?* and latency bounds consistent with $\top$ and $\bot$.

[4] `https://github.com/DEIS-Tools/MoniTAal`

Table 1: Results for simultaneously monitoring six response properties over a trace generated by the gear controller model from [21].

# Observ.	$\tau(\rho)$	Time (ms)		Max. resp. time (μs)		# Symbolic States	
		Delay	Delay-free	Delay	Delay-free	Delay	Delay-free
1000	63112	76	42	182	143	39	12
2000	124028	223	84	228	188	57	12
3000	184743	418	116	382	86	78	12
4000	244717	691	154	410	81	109	12
5000	306015	1037	198	571	281	135	12
6000	366814	1463	237	680	163	167	12
7000	438799	1973	278	767	175	192	12
8000	501070	2554	314	986	193	215	12
9000	563296	3212	357	1159	175	238	12
10000	624530	3929	384	1099	109	266	12

We demonstrate MoniTAal on a trace generated by simulating the gear controller model from [21] in Uppaal [8]. The model, along with formal requirements, was created by the company Mecel. For the monitored properties, we replaced six error locations in the model with response properties on the form $G_{[0,\infty]}a \to F_{[0,b]}c$ expressing that some signal a is followed by a response c within some bound b. For such a property, it is not possible to give a $\top$ verdict, since it will always possible to violate it in the future. The six properties are all satisfied by the model, which means that we will never terminate with a verdict, no matter how long the trace observed. This allows us to test how arbitrarily long traces can affect the performance of our algorithm.

We show results with and without delay consisting of a latency in $[0, 100]$, and a jitter bounded by 5. The results in Table 1 show the number of observations, length (in time) of the observed word, the overall running time, the maximal response time (the time it takes to process a single observation and return the next verdict), and the maximal number of stored symbolic states. The running time, maximal response time and number of symbolic states over the number of observations are plotted in Fig. 5. Under delay we see that the state storage grows linearly with the number of observations, which in turn results in a growing response time. The reason for this is that the uncertainty of the delay increases the size of the reach-set. Nevertheless, the maximal response time is in all cases less than 1.2 ms. In the delay-free case, the memory usage is constant, thus the response time is also constant, although with tiny fluctuations. The total time in the delayed case has approximately a quadratic growth. This is not surprising, since it is the running sum of the response time which seems to grow linearly, but we do not have enough experiments here to conclude a precise relation between the state storage and response time. In general, the results show us that the tool in this case is able to handle multiple properties and long traces in terms of time horizon as well as number of observations.

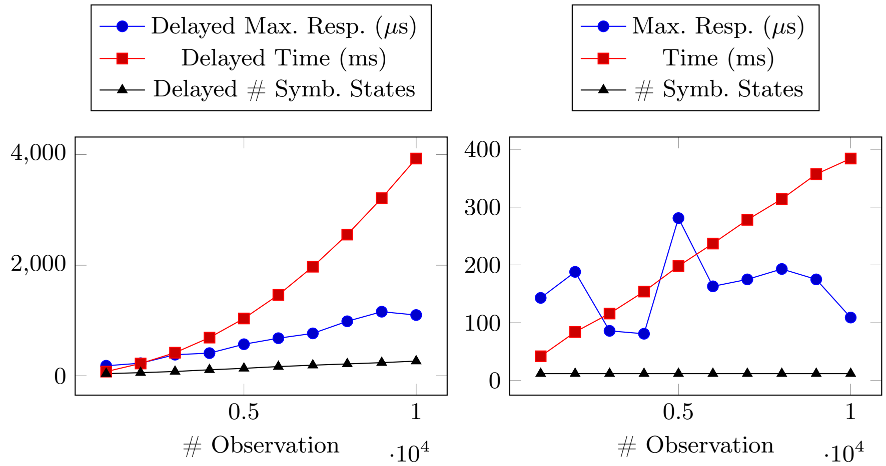

Fig. 5: Graph plotting the total time, maximal response time and number of symbolic states over the trace length (number of observations) from Table 1 of the delayed (left) and delay-free (right) case.

7 Conclusion

We have introduced a zone-based algorithm realizing optimal (in the sense of being anticipating [7]) online operational monitoring of embedded real-time systems when the communication between the monitor and the system is subject to unknown (up to bounds) delay. This situation is rather typical in practice as observations are mediated by sensors, may involve conversion between analog and digital, or pass communication networks and consequently are indirect in general, leading to delays and inexact time-stamping. Our constructions thus fill a gap in the pre-existing theories for monitoring hard real-time systems, which tend to assume full and exact temporal observability by immediate coupling or, equivalently, perfect synchrony between systems and their monitors.

A notable point of our construction is that it applies a reduction to simple timed automata and is purely zone-based despite the unknown communication delay being a timing parameter. The construction thus not only avoids the complexities of property analysis for parameterized timed automata [4], but also provides an instance of monitoring under uncertainty where the underlying arithmetic constraint systems remain of fixed dimensionality (namely the number of clocks in the property automata plus two for monitoring) despite their history dependence. This is in stark contrast to direct constraint encodings growing linearly over history length as in [13].

In further research, we study the question of monitorability [7]: some properties will never give definitive verdicts (e.g., "infinitely often a") and are therefore not useful for monitoring. We conjecture that our zone-based approach can be exploited to decide monitorability of real-time properties.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science **126**(2), 183–235 (1994). https://doi.org/https://doi.org/10.1016/0304-3975(94)90010-8
2. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. Journal of the ACM **43**(1) (01 1996)
3. Alur, R., Henzinger, T.A., Vardi, M.Y.: Parametric real-time reasoning. In: Kosaraju, S.R., Johnson, D.S., Aggarwal, A. (eds.) Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA. pp. 592–601. ACM (1993). https://doi.org/10.1145/167088.167242
4. André, É., Lime, D., Roux, O.H.: Reachability and liveness in parametric timed automata. Log. Methods Comput. Sci. **18**(1) (2022). https://doi.org/10.46298/lmcs-18(1:31)2022
5. Baldor, K., Niu, J.: Monitoring dense-time, continuous-semantics, metric temporal logic. In: Runtime Verification. pp. 245–259. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-35632-2_24
6. Basin, D., Klaedtke, F., Zălinescu, E.: Algorithms for monitoring real-time properties. In: RV. pp. 260–275. Springer (2012). https://doi.org/10.1007/978-3-642-29860-8_20
7. Bauer, A., Leucker, M., Schallhart, C.: Monitoring of real-time properties. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science. pp. 260–272. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11944836_25
8. Behrmann, G., David, A., Larsen, K.G.: A tutorial on uppaal. In: Bernardo, M., Corradini, F. (eds.) Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM-RT 2004, Bertinoro, Italy, September 13-18, 2004, Revised Lectures. LNCS, vol. 3185, pp. 200–236. Springer (2004). https://doi.org/10.1007/978-3-540-30080-9_7, https://doi.org/10.1007/978-3-540-30080-9_7
9. Bengtsson, J., Yi, W.: Timed automata: Semantics, algorithms and tools. In: Lectures on Concurrency and Petri Nets, Advances in Petri Nets. LNCS, vol. 3098, pp. 87–124. Springer (2003). https://doi.org/10.1007/978-3-540-27755-2_3
10. Brihaye, T., Geeraerts, G., Ho, H.M., Monmege, B.: MightyL: A compositional translation from MITL to timed automata. In: Computer Aided Verification. pp. 421–440. Springer (2017). https://doi.org/10.1007/978-3-319-63387-9_21
11. Cimatti, A., Grosen, T.M., Larsen, K.G., Tonetta, S., Zimmermann, M.: Exploiting assumptions for effective monitoring of real-time properties under partial observability. arXiv **2409.05456** (2024). https://doi.org/10.48550/arXiv.2409.05456
12. Donzé, A., Ferrère, T., Maler, O.: Efficient robust monitoring for STL. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings. LNCS, vol. 8044, pp. 264–279. Springer (2013). https://doi.org/10.1007/978-3-642-39799-8_19, https://doi.org/10.1007/978-3-642-39799-8_19
13. Finkbeiner, B., Fränzle, M., Kohn, F., Kröger, P.: A truly robust signal temporal logic: Monitoring safety properties of interacting cyber-physical systems under uncertain observation. Algorithms **15**(4), 126 (2022). https://doi.org/10.3390/a15040126

14. Fränzle, M., Grosen, T.M., Larsen, K.G., Zimmermann, M.: Monitoring real-time systems under parametric delay. arXiv **2404.18282** (2024). https://doi.org/10.48550/ARXIV.2404.18282
15. Grosen, T.M., Kauffman, S., Larsen, K.G., Zimmermann, M.: Monitoring timed properties (revisited). In: Bogomolov, S., Parker, D. (eds.) Formal Modeling and Analysis of Timed Systems - 20th International Conference, FORMATS 2022, Warsaw, Poland, September 13-15, 2022, Proceedings. LNCS, vol. 13465, pp. 43–62. Springer (2022). https://doi.org/10.1007/978-3-031-15839-1_3
16. Ho, H.M., Ouaknine, J., Worrell, J.: Online monitoring of metric temporal logic. In: Runtime Verification. pp. 178–192. Springer (2014). https://doi.org/10.1007/978-3-319-11164-3_15
17. Kallwies, H., Leucker, M., Sánchez, C.: Symbolic runtime verification for monitoring under uncertainties and assumptions. In: Bouajjani, A., Holík, L., Wu, Z. (eds.) Automated Technology for Verification and Analysis - 20th International Symposium, ATVA 2022, Virtual Event, October 25-28, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13505, pp. 117–134. Springer (2022). https://doi.org/10.1007/978-3-031-19992-9_8
18. Köhl, M.A., Hermanns, H.: Model-based diagnosis of real-time systems: Robustness against varying latency, clock drift, and out-of-order observations. ACM Trans. Embed. Comput. Syst. **22**(4), 68:1–68:48 (2023). https://doi.org/10.1145/3597209, https://doi.org/10.1145/3597209
19. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. Int. J. Softw. Tools Technol. Transf. **1**(1-2), 134–152 (1997). https://doi.org/10.1007/S100090050010, https://doi.org/10.1007/s100090050010
20. Lindahl, M., Pettersson, P., Yi, W.: Formal design and analysis of a gear controller. In: Steffen, B. (ed.) Tools and Algorithms for Construction and Analysis of Systems (TACAS). LNCS, vol. 1384, pp. 281–297. Springer (1998). https://doi.org/10.1007/BFb0054178
21. Lindahl, M., Pettersson, P., Yi, W.: Formal Design and Analysis of a Gearbox Controller. Springer International Journal of Software Tools for Technology Transfer (STTT) **3**(3), 353–368 (2001)
22. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Lakhnech, Y., Yovine, S. (eds.) Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings. LNCS, vol. 3253, pp. 152–166. Springer (2004). https://doi.org/10.1007/978-3-540-30206-3_12, https://doi.org/10.1007/978-3-540-30206-3_12
23. Ničković, D., Yamaguchi, T.: Rtamt: Online robustness monitors from STL. In: Hung, D.V., Sokolsky, O. (eds.) Automated Technology for Verification and Analysis. pp. 564–571. Springer International Publishing, Cham (2020)
24. Sheldon, T.: McGraw-Hill's Encyclopedia of Networking and Telecommunications. McGraw-Hill Professional (2001)
25. Thati, P., Rosu, G.: Monitoring algorithms for metric temporal logic specifications. In: Havelund, K., Rosu, G. (eds.) Proceedings of the Fourth Workshop on Runtime Verification, RV@ETAPS 2004, Barcelona, Spain, April 3, 2004. Electronic Notes in Theoretical Computer Science, vol. 113, pp. 145–162. Elsevier (2004). https://doi.org/10.1016/J.ENTCS.2004.01.029

26. Ulus, D., Ferrère, T., Asarin, E., Maler, O.: Timed pattern matching. In: Formal Modeling and Analysis of Timed Systems. pp. 222–236. Springer (2014). https://doi.org/10.1007/978-3-319-10512-3_16
27. Ulus, D., Ferrère, T., Asarin, E., Maler, O.: Online timed pattern matching using derivatives. In: Tools and Algorithms for the Construction and Analysis of Systems. pp. 736–751. Springer (2016). https://doi.org/10.1007/978-3-662-49674-9_47
28. Visconti, E., Bartocci, E., Loreti, M., Nenzi, L.: Online monitoring of spatio-temporal properties for imprecise signals. In: Arun-Kumar, S., Méry, D., Saha, I., Zhang, L. (eds.) MEMOCODE '21: 19th ACM-IEEE International Conference on Formal Methods and Models for System Design, Virtual Event, China, November 20 - 22, 2021. pp. 78–88. ACM (2021). https://doi.org/10.1145/3487212.3487344, https://doi.org/10.1145/3487212.3487344
29. Waga, M., André, É., Hasuo, I.: Model-bounded monitoring of hybrid systems. ACM Trans. Cyber Phys. Syst. **6**(4), 30:1–30:26 (2022). https://doi.org/10.1145/3529095, https://doi.org/10.1145/3529095

A Proofs Omitted in the Main Part

Throughout the appendix, we often need to "shift" a timed word in the sense that we add or subtract a $d \in \mathbb{R}_{\geq 0}$ to each time point of ρ. In the latter case, we need to be careful to ensure that the time points in the shifted word are still nonnegative. For the sake of readability, let us introduce some notation for these operations. Given a (finite or infinite) timed word $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$ and such a $d \in \mathbb{R}_{\geq 0}$,

- let $\rho + d$ denote the timed word $(\sigma_1, \tau_1 + d), (\sigma_2, \tau_2 + d), \ldots$, and
- if $d \leq \tau_1$, let $\rho - d$ denote the timed word $(\sigma_1, \tau_1 - d), (\sigma_2, \tau_2 - d), \ldots$. This is well-defined, as we require that τ_1 (and therefore each τ_i) is at least d, so we never obtain negative time points in $\rho - d$.

The following properties follow directly from the definition of timed concatenation and will be applied in the proofs below.

Remark 1. Let $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n) \in T\Sigma^*$, let $\mu = (\sigma'_1, \tau'_1), (\sigma'_2, \tau'_2), \ldots \in T\Sigma^\omega$, and let $t \leq \tau(\rho)$.

1. Let $t' \in \mathbb{R}_{\geq 0}$ be such that $t - t' \geq \tau(\rho)$. Then
$$\rho \cdot_t \mu = \rho \cdot_{t-t'} (\mu + t').$$
2. Let $0 \leq t' \leq t$, let $n' \in \{0, 1, \ldots, n\}$ be such that $\tau_{n'} \leq t' \leq \tau_{n'+1}$ (were we use $\tau_0 = -\infty$ to allow $n' = 0$ and $\tau_{n+1} = \infty$ to allow $n' = n$) and define $\rho_1 = (\sigma_1, \tau_1), \ldots, (\sigma_{n'}, \tau_{n'})$ as well as $\rho_2 = (\sigma_{n'+1}, \tau_{n'+1}), \ldots, (\sigma_n, \tau_n)$. Then
$$\rho \cdot_t \mu = \rho_1 \cdot_{t'} ((\rho_2 - t') \cdot_{t-t'} \mu).$$
3. Let $t' \geq 0$ be such that $\tau(\rho) \leq t - t'$, let $n' \geq 1$ be such that $\tau'_{n'} \leq t' \leq \tau'_{n'+1}$ (this is well-defined due to time-divergence), and define $\rho' = (\sigma'_1, \tau'_1), \ldots, (\sigma'_{n'}, \tau'_{n'})$ as well as $\mu' = (\sigma'_{n'+1}, \tau'_{n'+1}), (\sigma'_{n'+2}, \tau'_{n'+2}), \ldots$. Then
$$\rho \cdot_{t-t'} \mu = (\rho \cdot_{t-t'} \rho') \cdot_t (\mu' - t').$$

Furthermore, the following properties about consistent words will be useful in the proofs below.

Lemma 7.

1. *Let $(\rho_1^*, t_1) \sqsubseteq (\rho_2^*, t_2)$, let $\rho_2 = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n) \in GT_{\delta,\varepsilon}(\rho_2^*, t_2)$, let $n' \in \{0, 1, \ldots, n\}$ be such that $\tau_{n'} \leq t_1 \leq \tau_{n'+1}$ (were we use $t_0 = -\infty$ to allow $n' = 0$ and $\tau_{n+1} = \infty$ to allow $n' = n$), and define $\rho_1 = (\sigma_1, \tau_1), \ldots, (\sigma_{n'}, \tau_{n'})$. Then $\rho_1 \in GT_{\delta,\varepsilon}(\rho_1^*, t_1)$.*
2. *Let $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$, and let ρ' be a finite timed word with $\tau(\rho') \leq t - \max(\tau(\rho), t - (\delta + \varepsilon))$. Then, $\rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \rho' \in GT_{\delta,\varepsilon}(\rho^*, t)$.*
3. *Let ρ^* be a finite timed word, say with m letters. Let $\rho \in GT_{\delta,\varepsilon}(\rho^*, t)$ and let ρ' be the prefix of ρ with m letters. Then, $\rho' \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$.*

Proof. 1.) We need to show that ρ_1 is consistent with ρ_1^* at t_1 under δ and ε. The first requirement of the definition of consistency follows from $\tau(\rho_1^*) \leq t_1$ and the choice of n' (which implies $\tau(\rho_1) \leq t_1$). The second requirement follows from the fact that ρ_2 is consistent with ρ_2^* at t_2 under δ and ε and the fact that ρ_1 is a prefix of ρ_2 and ρ_1^* is a prefix of ρ_2^*. Finally, consider the third requirement and assume it is violated, i.e., let ρ_1^* have m letters and assume ρ_1 has at least $m+1$ letters such that the time point τ_{m+1} of the $(m+1)$-st letter of ρ_1 satisfies $\tau_{m+1} + \delta + \varepsilon < t_1$. Then, as ρ_2 is consistent with ρ_2^* at t_2 under δ and ε, we obtain a contradiction. Either ρ_2^* has also m letters. In this case, $\tau_{m+1} + \delta + \varepsilon < t_1 \leq t_2$ implies that the third requirement of the definition of consistency is violated for ρ_2 and ρ_2^*. Otherwise, ρ_2^* has at least $m+1$ letters. In this case $\tau_{m+1} + \delta + \varepsilon < t_1 \leq t_2$ implies that the second requirement of the definition of consistency is violated for $i = m+1$.

2.) We have to show that $\rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \rho'$ is consistent with ρ^* at t under δ and ε. This follows directly from the fact that all events in ρ' have time points (in $\rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \rho'$) in the interval $[t-(\delta+\varepsilon), t]$ and are therefore covered by the third requirement of the definition of consistency.

3.) We need to show that ρ' is EL-consistent with ρ^* at t under δ and ε. By definition, ρ' has the same length as ρ^* and the first two requirements of the definition of consistency are satisfied, as ρ is consistent with ρ^* at t under δ and ε and ρ' is a prefix of ρ. Hence, it is EL-consistent, as the third requirement only refers to ground-truth that have more letters than the observation. $\square$

Now we are ready to present the proofs omitted in the main part.

Proof of Lemma 1

Recall that we need to show $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \top$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$ and that $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \bot$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$ for $\mathcal{D} \subseteq \mathcal{D}'$.

Proof. Note that $\mathcal{D} \subseteq \mathcal{D}'$ implies $GT_{\mathcal{D}}(\rho^*, t) \subseteq GT_{\mathcal{D}'}(\rho^*, t)$. Thus, the universal quantification over possible ground-truths ρ in the first two cases of the definition of $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t)$ ranges over a subset of the possible ground-truths that are considered for $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t)$. Hence, the result follows. $\square$

Proof of Lemma 2

Recall that we need to show

1. $\Delta_{\mathcal{D}}(L, \rho^*, t) = \emptyset$ iff $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$, and
2. $\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) = \emptyset$ iff $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$.

Proof. We have

$$\begin{aligned}
&\Delta_{\mathcal{D}}(L, \rho^*, t) = \emptyset \\
\Leftrightarrow\ &\rho \cdot_t \mu \in \overline{L} \text{ for all } (\delta, \varepsilon) \in \mathcal{D}, \text{ all } \rho \in GT_{\delta,\varepsilon}(\rho^*, t), \text{ and all } \mu \in T\Sigma^\omega \\
\Leftrightarrow\ &\rho \cdot_t \mu \in \overline{L} \text{ for all } \rho \in GT_{\mathcal{D}}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega \\
\Leftrightarrow\ &\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot.
\end{aligned}$$

The second claim is obtained by a dual argument (swapping $\bot$ with $\top$ and L with $\overline{L}$). $\square$

Proof of Lemma 3

Recall that we need to show $\Delta(L, \rho_1^*, t_1) \supseteq \Delta(L, \rho_2^*, t_2)$ for all $(\rho_1^*, t_1) \sqsubseteq (\rho_2^*, t_2)$.

Proof. Let $(\delta, \varepsilon) \in \Delta(L, \rho_2^*, t_2)$, i.e., there exists $\rho_2 \in GT_{\delta,\varepsilon}(\rho_2^*, t_2)$ and a $\mu_2 \in T\Sigma^\omega$ such that $\rho_2 \cdot_{t_2} \mu_2 \in L$.

Let $\rho_2 = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ and let n' be maximal with $\tau_{n'} \leq t_1$. Then, Lemma 7.1 yields

$$\rho_1 = (\sigma_1, \tau_1), \ldots, (\sigma_{n'}, \tau_{n'}) \in GT_{\delta,\varepsilon}(\rho_1^*, t_1)$$

and an application of Remark 1.2 yields

$$\rho_1 \cdot_{t_1} \Big(\big[((\sigma_{n'+1}, \tau_{n'+1}), \ldots, (\sigma_n, \tau_n)) - t_1 \big] \cdot_{t_2 - t_1} \mu \Big) = \rho_2 \cdot_{t_2} \mu_2 \in L.$$

This implies $(\delta, \varepsilon) \in \Delta(L, \rho_1^*, t_1)$. $\square$

Proof of Lemma 4

Recall that we need to show

1. $\Delta_{\mathcal{D}}(L, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau_1^*\}$ implies there is no $\rho_1^* \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*, t') = \top$ for any $t' \geq t + \tau(\rho_1^*)$, and
2. $\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau_1^*\}$ implies there is no $\rho_1^* \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*, t') = \bot$ for any $t' \geq t + \tau(\rho_1^*)$.

for all $t \geq \tau(\rho^*)$.

Proof. Let $\Delta_{\mathcal{D}}(L, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau_1^*\}$, i.e., there is a $(\delta, \varepsilon) \in \mathcal{D}$ with $\delta \leq \tau_1^*$ and $(\delta, \varepsilon) \notin \Delta_{\mathcal{D}}(L, \rho^*, t)$. Thus, by definition, for all $\rho \in GT_{\delta,\varepsilon}(\rho^*, t)$ and all $\mu \in T\Sigma^\omega$ we have $\rho \cdot_t \mu \in \overline{L}$ (†).

Now, let $\rho_1^* \in T\Sigma^*$ and assume, towards a contradiction, we have $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*, t') = \top$ for $t' \geq t + \tau(\rho_1^*)$, i.e., for all $\rho' \in GT_{\mathcal{D}}(\rho^* \cdot_t \rho_1^*, t')$ and all $\mu \in T\Sigma^\omega$ we have $\rho' \cdot_{t'} \mu \in L$ (††). As $GT_{\delta,\varepsilon}(\rho^* \cdot_t \rho_1^*, t')$ is nonempty, let us fix one such $\rho' \in GT_{\delta,\varepsilon}(\rho^* \cdot_t \rho_1^*, t')$. Also, let us fix a $\mu \in T\Sigma^\omega$.

Note that we have $(\rho^*, t) \sqsubseteq (\rho^* \cdot_t \rho_1^*, t')$. So, let $\rho' = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ and let n' be maximal with $\tau_{n'} \leq t$. Then, Lemma 7.1 yields

$$\rho_1' = (\sigma_1, \tau_1), \ldots, (\sigma_{n'}, \tau_{n'}) \in GT_{\delta,\varepsilon}(\rho^*, t)$$

and application of Remark 1.2 yields

$$\rho' \cdot_{t'} \mu = \rho_1' \cdot_t \underbrace{\Big(\big[((\sigma_{n'+1}, \tau_{n'+1}), \ldots, (\sigma_n, \tau_n)) - t \big] \cdot_{t'-t} \mu \Big)}_{=\mu'}.$$

This yields the desired contradiction, as $\rho' \cdot_{t'} \mu$ is in L (see ††) while $\rho_1' \cdot_t \mu'$ is not in L (see †).

The second claim is proven by a dual argument (swapping L with $\overline{L}$ and $\top$ with $\bot$). $\square$

Proof of Lemma 5

Recall that we need to show $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \mathcal{V}_{\mathcal{D}}(L)(\rho^*, t)$.

Proof. Let $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$. We show $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \top$ by proving that we have $\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu' \in L$ for all $\rho' \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ for some $(\delta, \varepsilon) \in \mathcal{D}$ and all $\mu' = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots \in T\Sigma^\omega$.

First, consider the case where $\tau(\rho') < t-(\delta+\varepsilon)$. Let n be maximal with $\tau_n \leq \delta + \varepsilon$ (this is well-defined due to time-divergence), let $\rho'_1 = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n)$ and $\mu'_2 = ((\sigma_{n+1}, \tau_{n+1}), (\sigma_{n+2}, \tau_{n+2}), \ldots) - (\delta + \varepsilon)$. Note that μ'_2 is well-defined as τ_{n+1} is, by the choice of n, greater than $\delta + \varepsilon$. Then, Lemma 7.2 yields $\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1$ is in $GT_{\delta,\varepsilon}(\rho^*, t)$ and Remark 1.3 yields

$$\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu' = \rho' \cdot_{t-(\delta+\varepsilon)} \mu' = (\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1) \cdot_t \mu'_2.$$

Therefore, $\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu'$ is the concatenation of the possible ground-truth $(\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1)$ of ρ^* and the suffix μ'_2. As we have $\rho \cdot_t \mu \in L$ for all $\rho \in GT_{\mathcal{D}}(\rho^*, t)$ and all $\mu \in T\Sigma^\omega$ (due to $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$), we conclude $\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu' \in L$ as required.

Now, consider the case where $\tau(\rho') \geq t - (\delta + \varepsilon)$. Note that we have $t - \tau(\rho') \geq 0$ due to $\rho' \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$. Hence, let n be maximal with $\tau_n \leq t - \tau(\rho')$ (again, this is well-defined due to time-divergence), let $\rho'_1 = (\sigma_1, \tau_1) \cdots (\sigma_n, \tau_n)$ and $\mu'_2 = ((\sigma_{n+1}, \tau_{n+1})(\sigma_{n+2}, \tau_{n+2}) \cdots) - (t - \tau(\rho'))$. Then, Lemma 7.2 yields $\rho' \cdot_{\tau(\rho')} \rho'_1$ is in $GT_{\delta,\varepsilon}(\rho^*, t)$ and Remark 1.3 yields

$$\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu' = \rho' \cdot_{\tau(\rho')} \mu' = (\rho' \cdot_{\tau(\rho')} \rho'_1) \cdot_t \mu'_2.$$

As $\rho' \cdot_{\max(\tau(\rho'), t-(\delta+\varepsilon))} \mu'$ is the concatenation of a possible ground-truth of ρ^* and an arbitrary suffix, it is again, as required, in L.

Using a dual argument (i.e., swapping $\top$ with $\bot$ and L with $\overline{L}$), we can show that $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$ implies $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \bot$.

Now, we show that $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \top$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$. A dual argument again shows that $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \bot$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$. This will then complete our proof, as both functions only have three elements in their codomain and we have shown that two of them have the same preimage w.r.t. both functions.

So, let $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \top$. We show $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$ by showing $\rho \cdot_t \mu \in L$ for all $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n) \in GT_{\mathcal{D}}(\rho^*, t)$ and all $\mu = (\sigma'_1, \tau'_1), (\sigma'_2, \tau'_2), \ldots \in T\Sigma^\omega$. By definition, there is a $(\delta, \varepsilon) \in \mathcal{D}$ such that $\rho \in GT_{\delta,\varepsilon}(\rho^*, t)$.

Let ρ^* have m letters. If $m = n$, then we also have $\rho \in GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*, t)$. We consider two cases: If $\tau(\rho) < t-(\delta+\varepsilon)$, then an application of Remark 1.1 yields

$$\rho \cdot_t \mu = \rho \cdot_{t-(\delta+\varepsilon)} (\mu + (\delta + \varepsilon)) = \rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} (\mu + (\delta + \varepsilon)),$$

and if $\tau(\rho) \geq t - (\delta + \varepsilon)$, then an application of Remark 1.1 yields

$$\rho \cdot_t \mu = \rho \cdot_{\tau(\rho)} (\mu + (t - \tau(\rho))) = \rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} (\mu + (t - \tau(\rho))),$$

where $t-\tau(\rho)$ is nonnegative by definition of consistency. Hence, in both cases, $\rho \cdot_t \mu$ is the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix. Due to $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*, t) = \top$, all concatenations $\rho \cdot_{\max(\tau(\rho), t-(\delta+\varepsilon))} \mu$ for $(\delta, \varepsilon) \in \mathcal{D}$, $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$, and $\mu \in T\Sigma^\omega$ are in L, which yields $\rho \cdot_t \mu \in L$.

It remains to consider the case where $n > m$, which we again split into two subcases. But first let us define $\rho_1 = (\sigma_1, \tau_1) \cdots (\sigma_m, \tau_m)$ as well as $\rho_2 = (\sigma_{m+1}, \tau_{m+1}) \cdots (\sigma_n, \tau_n)$. Lemma 7.3 yields $\rho_1 \in GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*, t)$.

First, consider the subcase where $\tau(\rho_1) < t - (\delta + \varepsilon)$. By definition of consistency, $n > m$ implies $\tau_{m+1} + \delta + \varepsilon \geq t$, and thus $\tau_{m+1} \geq t - (\delta + \varepsilon)$ (†). Hence, an application of Remark 1.2 yields

$$\begin{aligned} \rho \cdot_t \mu &= \rho_1 \cdot_{t-(\delta+\varepsilon)} [(\rho_2 - (t - (\delta + \varepsilon))) \cdot_{\delta+\varepsilon} \mu] \\ &= \rho_1 \cdot_{\max(\tau(\rho_1), t-(\delta+\varepsilon))} [(\rho_2 - (t - (\delta + \varepsilon))) \cdot_{\delta+\varepsilon} \mu] \end{aligned}$$

Note that the first time point of ρ_2, τ_{m+1}, is at least $(t - (\delta + \varepsilon))$ as required, as $\tau_{m+1} \geq t - (\delta + \varepsilon)$ (see †). Hence, $\rho \cdot_t \mu$ is the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix and therefore in L.

Finally, consider the subcase where $\tau(\rho_1) \geq t - (\delta + \varepsilon)$. Then, an application of Remark 1.2 yields

$$\begin{aligned} \rho \cdot_t \mu &= \rho_1 \cdot_{\tau(\rho_1)} [(\rho_2 - \tau(\rho_1)) \cdot_{t-\tau(\rho_1)} \mu] \\ &= \rho_1 \cdot_{\max(\tau(\rho_1), t-(\delta+\varepsilon))} [(\rho_2 - \tau(\rho_1)) \cdot_{t-\tau(\rho_1)} \mu]. \end{aligned}$$

Again, this is well-defined as we have $\tau_{m+1} \geq \tau(\rho_1)$ (as τ_{m+1} is the next time instant after $\tau_m = \tau(\rho_1)$ in ρ) and as $t \geq \tau(\rho_1)$ by the definition of consistency. Hence, $\rho \cdot_t \mu$ is again the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix and therefore in L. □

Proof of Theorem 2

Recall that we need to show $\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t)$.

Proof. We will show that $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'}$ is nonempty iff there exists a $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ and a $\mu \in T\Sigma^\omega$ with $\rho \cdot_{\max(\tau(\rho), (t-(\delta+\varepsilon)))} \mu \in L(\mathcal{A}')$ for any TBA $\mathcal{A}'$. Then we obtain

- $\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \top$ iff $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t) = \top$ by instantiating the equivalence for $\mathcal{A}' = \overline{\mathcal{A}}$, and
- $\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \bot$ iff $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t) = \bot$ by instantiating the equivalence for $\mathcal{A}' = \mathcal{A}$.

This completes the proof, as both functions only have three elements in their codomain and we have shown that two of them have the same preimage w.r.t. both functions.

So, let $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'} \neq \emptyset$. Then, by definition, there is a state (q, v') of $\mathcal{A}'$ such that

– $(q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}'} (q, v)$ for some initial state (q_0, v_0) of $\mathcal{A}'$, some $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ for some $(\delta, \varepsilon) \in \mathcal{D}$, and $v' = v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))$, and
– there is an accepting infinite run of $\mathcal{A}'$ starting in (q, v') that processes some $\mu \in T\Sigma^\omega$.

These two runs can be combined into an accepting run of $\mathcal{A}'$ that starts in (q_0, v_0) and processes

$$\rho \cdot_{\tau(\rho)+\max(0,(t-(\tau(\rho)+\delta+\varepsilon)))} \mu = \rho \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu,$$

which implies that it is in $L(\mathcal{A}')$ as required.

Conversely, let there be a $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ and a $\mu \in T\Sigma^\omega$ with

$$\mu \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu \in L(\mathcal{A}').$$

Then, there exists an accepting run of $\mathcal{A}'$ starting in some initial state (q_0, v_0) that processes $\rho \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu$. This run can be split into

– $(q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}'} (q, v)$ for some state (q, v) of $\mathcal{A}'$ and
– an accepting infinite run of $\mathcal{A}'$ starting in (q, v') that processes μ, where

$$v' = v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon))).$$

Hence, $(q, v') \in \mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'}$, which is therefore, as required, nonempty. □

Proof of Theorem 3

Let $\rho^* = (\sigma^*_1, \tau^*_1), \ldots, (\sigma^*_n, \tau^*_n)$ be an observed timed word. We want to show that

$$\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q', Z') \mid (q', Z'') \in S_n, Z' = Z''^{\nearrow} \wedge \mathit{etime} = t - \varepsilon\} \tag{1}$$

where $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t)$ is defined by

$$\begin{aligned}\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q, v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))) \mid {} & (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where} \\ & (q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } c \in C, \text{ and} \\ & \rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ for some } (\delta, \varepsilon) \in \mathcal{D}\}.\end{aligned}$$

in the setting where $\mathcal{D} = \{(\delta, \varepsilon) \mid \delta \in [l, u]\}$ for given $l, u, \varepsilon \in \mathbb{Q}_{\geq 0}$. Also, recall that we have defined $S_0 = \{(q_0, Z^d_0) \mid q_0 \in Q_0\}$ and $S_i = \mathrm{Succ}_d(S_{i-1}, (\sigma_i, \tau^*_i))$ for $i \in \{1, \ldots, n\}$.

Proof. Given the form of $\mathcal{D}$ we can rewrite the definition of the reach-set to

$$\begin{aligned}\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q, v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))) \mid {} & (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where} \\ & (q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } c \in C, \text{ and} \\ & \rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ for some } \delta \in [l, u]\}.\end{aligned}$$

Now, extending the transition relation $\xrightarrow{\rho}_{\mathcal{A}}$ to clock valuations over the extended set of clocks $C \cup \{time, etime\}$, we may further reformulate the reach-set as follows:

$$\begin{aligned}\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q_n, v^* + \max(0, (t - (v^*_n(etime) + \varepsilon)))) \mid{} & (q_0, v^*_0) \xrightarrow{\rho}_{\mathcal{A}} (q_n, v^*_n) \text{ where} \\ & (q_0, v^*_0) \text{ with } q_0 \in Q_0,\ v^*_0(x) = 0 \text{ for all } c \in C, \text{ and} \\ & v^*_0(etime) - v^*_0(time) \in [l, u] \text{ and} \\ & v^*_i(etime) \leq \tau^*_i \wedge v^*_i(etime) \geq \tau^*_i - \varepsilon \wedge \sigma_i = \sigma^*_i \text{ for } i \in \{0, \ldots, n\}\}.\end{aligned}$$

A key observation for the correctness of the above reformulation, is that the extended clocks *etime* and *time* are *not* modified by the TBA $\mathcal{A}$. That is $v^*_i(etime) - v^*_i(time) = v^*_0(etime) - v^*_0(time) \in [l, u]$ for all $i = \{0, \ldots, n\}$.

Now let $\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*, t)$ for $j \in \{0, \ldots, n\}$ be defined as follows:

$$\begin{aligned}\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*, t) = \{(q_j, v^*_j) \mid{} & (q_0, v^*_0) \xrightarrow{\rho}_{\mathcal{A}} (q_j, v^*_j) \text{ where} \\ & (q_0, v^*_0) \text{ with } q_0 \in Q_0,\ v^*_0(x) = 0 \text{ for all } c \in C, \text{ and} \\ & v^*_0(etime) - v^*_0(time) \in [l, u] \text{ and} \\ & v^*_i(etime) \leq \tau^*_i \wedge v^*_i(etime) \geq \tau^*_i - \varepsilon \wedge \sigma_i = \sigma^*_i \text{ for } i \in \{0, \ldots, j\}\}.\end{aligned}$$

Then clearly $\mathcal{R}^{\mathcal{D},0}_{\mathcal{A}}(\rho^*, t) = \{(q_0, Z^d_0) \mid q_0 \in Q_0\} = S_0$ and $\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*, t) = \mathrm{Succ}_d(\mathcal{R}^{\mathcal{D},j-1}_{\mathcal{A}}(\rho^*, t), (\sigma_i, \tau_i))$ for $j \in \{1, \ldots, n\}$, using standard arguments for the correctness of symbolic exploration of timed automata, e.g. [9]. Finally, as $(t - (v^*_n(etime) + \varepsilon))) = ((t - \varepsilon) - v^*_n(etime))$ it follows that $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \mathcal{R}^{\mathcal{D},n}_{\mathcal{A}}(\rho^*, t)^{\nearrow} \wedge etime = t - \varepsilon$. $\square$

B More Details on the Implementation

MoniTAal requires, as input, two automata accepting the complement language of the other, a series of observations, and optionally bounds on latency and jitter. MoniTAal can be run as a binary where the automata are parsed from a UPPAAL xml file while the observations are parsed as as text from a file or standard input. MoniTAal can also be used as a C++ library.

In Listing 1.1 we give a short demonstration going through Example 5 using MoniTAal. In the output we see that the consistent latency for the $\top$ verdict (`POSITIVE`) verdict tightens to $[71, 100]$ and then $[71, 75[$ while the final verdict is ***?*** (`INCONCLUSIVE`). Note that we multiply all values (in observations and automata) by 10 in order to use integer, rather than rational, time points.

```
$ ./MoniTAal-bin -p positive delay.xml -n negative delay.xml -v --latencyl 0
    --latencyu 10 --jitter 2
Input: @[173, 173] a

Verdict: INCONCLUSIVE
Positive:
Consistent latencies: {[71,100]}
Jitter bound: 2
Negative:
Consistent latencies: {[0,100]}
Jitter bound: 2

Input: @[271, 271] b

Verdict: INCONCLUSIVE
Positive:
Consistent latencies: {[71,75)}
Jitter bound: 2
Negative:
Consistent latencies: {[0,100]}
Jitter bound: 2
```

Listing 1.1: Demonstration of MoniTAal over Example 5.