On Real-time Monitoring with Imprecise Timestamps*

David Basin[1], Felix Klaedtke[2], Srdjan Marinovic[1], and Eugen Zălinescu[1]

[1] Institute of Information Security, ETH Zurich, Switzerland
[2] NEC Europe Ltd., Heidelberg, Germany

Abstract. Existing real-time monitoring approaches assume traces with precise timestamps. Their correctness is thus indefinite when monitoring the behavior of systems with imprecise clocks. We address this problem for a metric temporal logic: We identify classes of formulas for which we can leverage existing monitors to correctly reason about observed system traces.

1 Introduction

Existing runtime-verification approaches for real-time logics, e.g., [1,2,5,6], assume that the monitored system emits events with precise (i.e. exact) timestamps. This assumption however does not hold for real-world systems, and thus monitors may produce incorrect outputs. To account for the clocks' imprecision, an error may be associated with events' timestamps. For instance, Google's distributed database Spanner [3] associates a time interval with each event, and Spanner guarantees that each event happened at some point in its associated interval.

This paper poses and explores the problem of whether existing monitoring approaches for real-time logics can account for timestamp imprecision, and thereby provide correctness guarantees for the monitors' outputs. In our study, we focus on the real-time temporal logic MTL [4] over a continuous dense time domain, for which we propose a monitoring approach that accounts for imprecise timestamps. For monitoring, we (a) first modify the specification by syntactically rewriting the MTL formula and (b) use an existing monitor for precise timestamps on the modified specification over one precisely timestamped trace that is obtained from the given imprecisely timestamped one. We identify MTL formulas for which conformance with the modified specification implies conformance with the given specification of all possible precise traces corresponding to the given imprecise trace. We also identify formulas for which the approach provides a weaker—but still a useful—guarantee that there is some precise trace satisfying the specification.

In summary, our contributions are the following. (1) We raise the problem of imprecise timestamps in runtime verification with respect to specifications in

* This work was partially supported by the Zurich Information Security and Privacy Center (www.zisc.ethz.ch).

real-time logics. (2) We provide correctness guarantees for the use of existing monitors over imprecise traces for certain MTL fragments.

Related to this work are the results of Zhang et al. [8] and Wang et al. [7]. Zhang et al. [8] explore the issue of imprecise timestamps in data-stream processing. In contrast to our approach, their solution is for a more restrictive specification language, relies on a discrete time domain, and outputs probabilistic verdicts. In runtime verification, Wang et al. [7] explore trace imprecision due to an unknown ordering between events. Events do not have explicit timestamps and thus only linear time properties (in LTL) are considered. In contrast, we monitor real-time properties (expressed in MTL). Furthermore, they propose a specialized monitoring algorithm, while we leverage existing monitoring algorithms.

2 Preliminaries

Let $\mathbb{T} := \mathbb{R}_{\geq 0}$ be the time domain and let P be a nonempty finite set of atomic propositions. A *timeline* is a function $\pi : \mathbb{T} \to 2^P$ in which values do not change infinitely often over bounded intervals. That is, for any bounded nonempty interval $I \subseteq \mathbb{T}$, there is a partition of I into nonempty intervals $I_1, \dots, I_n$ for some $n \geq 1$ such that π is constant on each I_i.

MTL formulas are given by the grammar

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathbin{\mathsf{S}_I} \varphi \mid \varphi \mathbin{\mathsf{U}_I} \varphi\,,$$

where p ranges over P and I over the intervals of $\mathbb{T}$ with rational endpoints or ∞ as a right endpoint. Given a timeline π, a time $t \in \mathbb{T}$, and a formula φ, the satisfaction relation $\models$ is defined as follows.

$$\begin{array}{lll}
\pi, t \models p & \text{iff} & p \in \pi(t) \\
\pi, t \models \neg\varphi & \text{iff} & \pi, t \not\models \varphi \\
\pi, t \models \varphi \wedge \psi & \text{iff} & \pi, t \models \varphi \text{ and } \pi, t \models \psi \\
\pi, t \models \varphi \mathbin{\mathsf{S}_I} \psi & \text{iff} & \text{there is some } t' \in \mathbb{T} \text{ with } t - t' \in I \text{ such that} \\
 & & \pi, t' \models \psi \text{ and } \pi, t'' \models \varphi, \text{ for all } t'' \in \mathbb{T} \text{ with } t' < t'' \leq t \\
\pi, t \models \varphi \mathbin{\mathsf{U}_I} \psi & \text{iff} & \text{there is some } t' \in \mathbb{T} \text{ with } t' - t \in I \text{ such that} \\
 & & \pi, t' \models \psi \text{ and } \pi, t'' \models \varphi, \text{ for all } t'' \in \mathbb{T} \text{ with } t \leq t'' < t'
\end{array}$$

Note that MTL's time domain is dense and its semantics is continuous. We use standard syntactic sugar. For instance, we define $\varphi \mathbin{\mathsf{T}_I} \psi := \neg(\neg\varphi \mathbin{\mathsf{S}_I} \neg\psi)$, $\varphi \mathbin{\mathsf{R}_I} \psi := \neg(\neg\varphi \mathbin{\mathsf{U}_I} \neg\psi)$, $\blacklozenge_I \varphi := \mathsf{true} \mathbin{\mathsf{S}_I} \varphi$, $\blacksquare_I \varphi := \mathsf{false} \mathbin{\mathsf{T}_I} \varphi$, $\Diamond_I \varphi := \mathsf{true} \mathbin{\mathsf{U}_I} \varphi$, and $\Box_I \varphi := \mathsf{false} \mathbin{\mathsf{R}_I} \varphi$, with $\mathsf{true} := p \vee \neg p$ and $\mathsf{false} := p \wedge \neg p$, for some $p \in P$.

A *timed word* is a sequence $(a_i, \tau_i)_{i \in \mathbb{N}}$ of tuples with $a_i \in 2^P$ and $\tau_i \in \mathbb{T}$, for any $i \in \mathbb{N}$, such that the sequence $(\tau_i)_{i \in \mathbb{N}}$ is non-strictly ascending and progressing. Intuitively, a timed word represents the observed, imprecisely timestamped trace, while a timeline represents the real system behavior. In the following, we assume a timestamp imprecision of $\delta \geq 0$, which we fix for the rest of the paper. For an "observed" timed word $(a_i, \tau_i)_{i \in \mathbb{N}}$, it would be natural to additionally assume that the τ_is are from a discrete infinite subset of $\mathbb{T}$, in which all elements have a finite representation. However, our results are valid without this additional assumption.

Given a timed word $\bar{\sigma} = (\bar{a}, \bar{\tau})$, the set of *possible timelines* of $\bar{\sigma}$, denoted $TL(\bar{\sigma})$, is the set of functions $\pi : \mathbb{T} \to 2^P$ with

$$\pi(t) := \begin{cases} a_i & \text{if } ts^{-1}(t) = \{i\} \text{ for some } i \in \mathbb{N}, \\ \emptyset & \text{otherwise,} \end{cases}$$

for any $t \in \mathbb{T}$, where $ts : \mathbb{N} \to \mathbb{T}$ is an injective function such that $ts(i) \in [\tau_i - \delta, \tau_i + \delta]$, for any $i \in \mathbb{N}$. We remark that the progress condition on $(\tau_i)_{i\in\mathbb{N}}$ ensures that the elements of $TL(\bar{\sigma})$ are indeed timelines. Furthermore, note that the requirement that ts is injective corresponds to the assumption that, in reality, no two events happen at the same point in time.

Example 1. Given $\delta := 1$ and the time word $\bar{\sigma} := (\{p\}, 1)(\{q\}, 1)(\{r\}, 2)(\{s\}, 5) \dots$, one of the timelines in $TL(\bar{\sigma})$ is π where $\pi(0.6) = \{q\}$, $\pi(1.2) = \{r\}$, $\pi(1.3) = \{p\}$, and $\pi(t) = \emptyset$ for $t \in [0, 4) \setminus \{0.6, 1.2, 1.3\}$. Note that the ordering of events in $\bar{\sigma}$ differs from that in π.

3 MTL Monitoring of Imprecisely Timestamped Traces

Informally, we are interested in what can be said about the conformance of the possible timelines of an observed timed word $\bar{\sigma}$ with respect to a given formula φ, where $\bar{\sigma}$ is observed incrementally. Formally, we focus on the following problems, where a problem instance consists of a formula φ, a timed word $\bar{\sigma}$, and a time $t \in \mathbb{T}$. For $\ell \in \{\exists, \forall\}$, the question is whether $\bar{\sigma}, t \models_\ell \varphi$ holds, where we write (i) $\bar{\sigma}, t \models_\exists \varphi$ if $\pi, t \models \varphi$, for *some* $\pi \in TL(\bar{\sigma})$, and (ii) $\bar{\sigma}, t \models_\forall \varphi$ if $\pi, t \models \varphi$, for *all* $\pi \in TL(\bar{\sigma})$. We focus on answering these questions online, using monitoring.

Given a formula φ and an iteratively presented timed word $\bar{\sigma}$, our monitoring approach is the following, where formal definitions are given below:

1. Transform the formula φ into the formula $\mathsf{tf}(\varphi)$.
2. Transform at runtime the timed word $\bar{\sigma}$ into the timeline $\rho_{\bar{\sigma}}$.
3. Monitor the timeline $\rho_{\bar{\sigma}}$ with respect to the formula $\mathsf{tf}(\varphi)$.

The transformed formula $\mathsf{tf}(\varphi)$ accounts for timestamp imprecision by relaxing the implicit temporal constraints on atoms, that is, relaxing "atom p holds now" to "atom p holds within a $\pm\delta$ interval". Formally, for $p \in P$, we define $\mathsf{tf}(p) := (\blacklozenge_{[0,\delta]}\, p) \vee (\lozenge_{[0,\delta]}\, p)$ and extend tf homomorphically to non-atomic formulas.

The timeline $\rho_{\bar{\sigma}}$ is obtained by simply ignoring timestamp imprecision. For the timed word $\bar{\sigma} = (\bar{a}, \bar{\tau})$, we define the *monitored timeline* $\rho_{\bar{\sigma}}$ as $\rho_{\bar{\sigma}}(t) := \bigcup_{i\in\mathbb{N}}\{a_i \mid \tau_i = t\}$, for any $t \in \mathbb{T}$. Note that the timeline $\rho_{\bar{\sigma}}$ is easily built at runtime from the timed word $\bar{\sigma}$. In fact, if $t \in \mathbb{T}$ is the current time, then the value of $\rho_{\bar{\sigma}}$ at t can be obtained as soon as a tuple (a_i, τ_i) of elements of the timed word $\bar{\sigma}$ with $\tau_i > t$ arrives.

The following theorem states the guarantees provided by our monitoring approach. Concretely, for each of the two posed questions, we identify two classes of formulas for which the approach provides correct answers. We define these formula classes syntactically using the rules in Figure 1. We say that a formula φ in negation normal form is *labeled by* (ℓ) with $\ell \in \{\exists, \forall\}$ if $\varphi : (\ell)$ is derivable

$$\frac{}{\mathsf{true} : (\forall)} \quad \frac{}{\mathsf{false} : (\forall)} \quad \frac{}{p : (\exists)} \quad \frac{}{\neg p : (\forall)} \quad \frac{\varphi : (\forall) \quad \psi : (\forall)}{\varphi \mathbin{\mathsf{op}} \psi : (\forall)}\ \mathsf{op} \in \{\wedge, \vee, \mathsf{S}_I, \mathsf{T}_I, \mathsf{U}_I, \mathsf{R}_I\}$$

$$\frac{\varphi : (\exists) \quad \psi : (\forall)}{\varphi \wedge \psi : (\exists)} \quad \frac{\varphi : (\exists) \quad \psi : (\exists)}{\varphi \vee \psi : (\exists)} \quad \frac{\varphi : (\forall) \quad \psi : (\exists)}{\varphi \mathbin{\mathsf{op}_I} \psi : (\exists)}\ \mathsf{op} \in \{\mathsf{S}, \mathsf{T}, \mathsf{U}, \mathsf{R}\} \qquad \frac{\varphi : (\forall)}{\varphi : (\exists)}$$

Fig. 1. Labeling Rules.

using the rules in Figure 1. For the negation normal form, we assume that the formulas true and false, and the connectives $\vee$, T, and R are language primitives, while the connectives $\blacklozenge$, $\blacksquare$, $\Diamond$, and $\Box$ are still syntactic sugar. We denote by $\mathit{nnf}(\varphi)$ the negation normal form of φ.

Theorem 2. *Let $\bar{\sigma}$ be a timed word, $\ell \in \{\exists, \forall\}$, and φ a formula with $\mathit{nnf}(\varphi)$ labeled by (ℓ). For any $t \in \mathbb{T}$, if $\rho_{\bar{\sigma}}, t \models \mathsf{tf}(\varphi)$, then $\bar{\sigma}, t \models_\ell \varphi$.*

Due to space limitations, we omit the theorem's proof, which is by induction over the formula structure, and give instead the intuition behind the theorem and some of the rules in Figure 1. The true and false formulas can be labeled by $(\forall)$ as their satisfaction does not depend on the trace. Positive literals p can only be labeled by $(\exists)$. If $\mathsf{tf}(p)$ is satisfied at t, then p is satisfied at some t' within the interval $[t-\delta, t+\delta]$, and thus there is a possible timeline for which p is satisfied at t. However, in general the other possible timelines do not satisfy p at t. In contrast, negative literals $\neg p$ can be labeled by $(\forall)$. If p is not satisfied on the interval $[t-\delta, t+\delta]$ on the monitored timeline, then there is no possible timeline satisfying p at t. Any formula of the form $\varphi \mathbin{\mathsf{op}} \psi$ can be labeled by $(\forall)$, as long as φ and ψ can both be labeled by $(\forall)$. That is, the $(\forall)$ fragment consists of those formulas in which atomic propositions occur only negatively. The last rule expresses that if all possible timelines satisfy φ at t then there is a possible timeline satisfying φ at t. Thus the $(\forall)$ fragment is included in the $(\exists)$ fragment.

By monitoring $\rho_{\bar{\sigma}}$ with respect to $\mathsf{tf}(\varphi)$ and using Theorem 2, we may obtain correctness guarantees about whether some or all timelines in $\mathit{TL}(\bar{\sigma})$ satisfy φ. This depends on whether the negation normal form of φ or $\neg\varphi$ can be labeled, and on the monitoring result for $\mathsf{tf}(\varphi)$ on $\rho_{\bar{\sigma}}$ at t. To clarify when guarantees are obtained, we consider the following cases.

- *Neither $\mathit{nnf}(\varphi)$ nor $\mathit{nnf}(\neg\varphi)$ can be labeled.* Then we cannot apply Theorem 2 to obtain the guarantees.
- *Only $\mathit{nnf}(\varphi)$ is labeled.* If the monitoring result is positive, i.e. $\rho_{\bar{\sigma}}, t \models \mathsf{tf}(\varphi)$, then we simply apply Theorem 2 to obtain the guarantees. If however $\rho_{\bar{\sigma}}, t \not\models \mathsf{tf}(\varphi)$, then nothing can be concluded about the system's conformance with respect to φ.
- *Only $\mathit{nnf}(\neg\varphi)$ is labeled.* This case is similar to the previous one, and we only obtain the guarantees if the monitoring result is negative. That is, when $\rho_{\bar{\sigma}}, t \not\models \mathsf{tf}(\varphi)$, we can apply Theorem 2 to $\neg\varphi$. This is because $\mathsf{tf}(\neg\varphi) \equiv \neg\mathsf{tf}(\varphi)$, and thus $\rho_{\bar{\sigma}}, t \not\models \mathsf{tf}(\varphi)$ iff $\rho_{\bar{\sigma}}, t \models \mathsf{tf}(\neg\varphi)$.
- *Both $\mathit{nnf}(\varphi)$ and $\mathit{nnf}(\neg\varphi)$ are labeled.* We obtain the guarantees regardless of the monitoring result. If $\rho_{\bar{\sigma}}, t \models \mathsf{tf}(\varphi)$ then we apply Theorem 2 to φ; otherwise, we apply it to $\neg\varphi$.

The last case is illustrated through the following example.

Example 3. Let $\varphi := \neg p \rightarrow \Diamond_I q$. We have that $\mathit{nnf}(\varphi) = p \vee (\mathsf{true}\,\mathsf{S}_I\, q) : (\exists)$ and $\mathit{nnf}(\neg\varphi) = \neg p \wedge (\mathsf{false}\,\mathsf{T}_I\, \neg q) : (\forall)$. According to Theorem 2, the guarantees that we obtain by monitoring $\rho_{\bar{\sigma}}$ with respect to $\mathsf{tf}(\varphi)$ are as follows. For any $t \in \mathbb{T}$, (1) if $\rho_{\bar{\sigma}}, t \models \mathsf{tf}(\varphi)$, then there is a $\pi \in \mathit{TL}(\bar{\sigma})$ with $\pi, t \models \varphi$, and (2) if $\rho_{\bar{\sigma}}, t \not\models \mathsf{tf}(\varphi)$, then $\pi, t \not\models \varphi$, for all $\pi \in \mathit{TL}(\bar{\sigma})$.

We remark that one can build the monitored timeline $\rho_{\bar{\sigma}}$ in different manners. Instead of taking the middle of the "uncertainty" intervals $[\tau_i - \delta, \tau_i + \delta]$ as the representative point in the monitored timeline, one could take another point as representative, provided that subsequent points have the same offset to the middle of the corresponding interval. The formula transformation must then be adjusted accordingly. However, monitoring such other timelines does not result in new conformance (with respect to the given property) guarantees as the following proposition demonstrates. In other words, it is sufficient to monitor the timeline considered in Theorem 2.

We first generalize the formula transformation. Given $\epsilon \in [0, \delta]$ and $* \in \{+, -\}$, let $\mathsf{tf}_{*\epsilon}(p) := (\blacklozenge_{[0,\delta\,*\,\epsilon]}\, p) \vee (\Diamond_{[0,\delta\,\bar{*}\,\epsilon]}\, p)$, for any $p \in P$, where $\bar{*}$ switches $*$ to its dual value. For instance, $\mathsf{tf}_0(p) = \mathsf{tf}(p)$ and $\mathsf{tf}_{-\delta}(p) = (\blacklozenge_{[0,0]}\, p) \vee (\Diamond_{[0,2\delta]}\, p)$. As before, $\mathsf{tf}_{*\epsilon}(\cdot)$ is extended homomorphically to non-atomic formulas.

Proposition 4. *Let $\delta \in \mathbb{T}$, $\epsilon_1, \epsilon_2 \in [0, \delta]$, $*_1, *_2 \in \{+, -\}$, a timed word $\bar{\sigma} = (a_i, \tau_i)_{i \in \mathbb{N}}$, and the timelines ρ_1 and ρ_2 be given with $\rho_j(t) := \bigcup_{i \in \mathbb{N}} \{a_i \mid \tau_i = t *_j \epsilon_j\}$, for any $t \in \mathbb{T}$ and $j \in \{1, 2\}$. For any formula φ and any $t \in \mathbb{T}$, we have that $\rho_1, t \models \mathsf{tf}_{*_1\epsilon_1}(\varphi)$ iff $\rho_2, t \models \mathsf{tf}_{*_2\epsilon_2}(\varphi)$.*

4 Discussion

Fragments. The $(\exists)$ fragment is practically relevant because the negation normal form of various common specifications patterns are included in it. For instance, consider the common specification pattern $\Box\,\varphi$ with $\varphi = (p \wedge \alpha) \rightarrow \blacklozenge_I (q \wedge \beta)$, for some $p, q \in P$ and some formulas α and β. When $\mathit{nnf}(\neg\alpha)$ is labeled by $(\exists)$ and $\mathit{nnf}(\beta)$ is labeled by $(\forall)$, then $\mathit{nnf}(\varphi)$ is labeled by $(\exists)$. Similarly, when $\mathit{nnf}(\alpha)$ is labeled by $(\forall)$ and $\mathit{nnf}(\neg\beta)$ is labeled by $(\forall)$, then $\mathit{nnf}(\neg\varphi)$ is labeled by $(\exists)$. Observe that $\mathit{nnf}(\varphi)$ and $\mathit{nnf}(\neg\varphi)$ can both be labeled only in some special cases, for instance, when both $\mathit{nnf}(\alpha)$ and $\mathit{nnf}(\neg\alpha)$ can be labeled and when $\beta = \mathsf{true}$. Furthermore, the $(\exists)$ fragment is limited in that conformance guarantees are given for only one possible timeline. In contrast, the $(\forall)$ fragment offers strong conformance guarantees; however, it is practically less relevant. Note that a formula in the $(\forall)$ fragment requires that all propositions occur negatively in φ. This is a strong restriction on the form of φ.

We do not, however, see how to extend the fragments in any significant way. For instance, the given rules cannot be strengthened by using stronger labels. This is illustrated by the following example, which shows that a rule that labels $\varphi \wedge \psi$ by $(\exists)$ whenever φ and ψ are labeled by $(\exists)$ is not sound.

Let $\varphi := p \wedge \blacklozenge_{[1,1]}\, q$ and $\psi := p \wedge \Diamond_{[1,1]}\, q$. Let $\delta := 2$ and consider the timed word $\bar{\sigma} := (\{p\},2)(\{q\},3)(\{r\},10)\ldots$. We have $\rho_{\bar{\sigma}}(2) = \{p\}$, $\rho_{\bar{\sigma}}(3) = \{q\}$, and $\rho_{\bar{\sigma}}(t) = \emptyset$, for any $t \in [0,5] \setminus \{2,3\}$, and $\mathsf{tf}(\varphi \wedge \psi) \equiv (\blacklozenge_{[0,2]} \Diamond_{[0,2]}\, p) \wedge (\blacklozenge_{[0,3]} \Diamond_{[0,1]}\, q) \wedge (\blacklozenge_{[0,1]} \Diamond_{[0,3]}\, q)$. Clearly $\rho_{\bar{\sigma}}, 2 \models \mathsf{tf}(\varphi \wedge \psi)$ but $\pi, 2 \not\models \varphi \wedge \psi$, for any $\pi \in TL(\bar{\sigma})$.

Point-based Monitoring. It is appealing to monitor directly the observed timed word $\bar{\sigma}$ using a monitor for the more prevalent point-wise semantics of MTL. See [1] for a comparison of the two semantics with respect to monitoring. However, it is harder to obtain correctness guarantees for such a setting because one must use two different MTL semantics, the point-wise one for the monitored traces and the continuous one for the possible timelines. Note that monitoring precise traces with respect to a point-wise semantics is inappropriate as there is no reference evaluation point for comparing the evaluation of the observed trace with the evaluation of the precise traces. Recall that, under a point-wise semantics, evaluation points are event indices and these depend on the events' occurrence times.

Conclusions. The previous discussion motivates the need for alternative approaches. We are investigating a quantitative MTL monitoring approach along the lines explored in [8]. However, the raised problem may require not only new algorithmic solutions, but also specification languages that allow for the explicit reasoning about timestamp imprecision.

References

1. D. Basin, F. Klaedtke, and E. Zălinescu. Algorithms for monitoring real-time properties. In *Proceedings of the 2nd International Conference on Runtime Verification*, volume 7186 of *LNCS*, pages 260–275. Springer, 2012.
2. A. Bauer, M. Leucker, and C. Schallhart. Runtime verification for LTL and TLTL. *ACM Transactions on Software Engineering and Methodology*, 20(4), 2011.
3. J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. C. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford. Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems*, 31(3):8, 2013.
4. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
5. O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proceedings of the Joint International Conferences on Formal Modelling and Analysis of Timed Systems and on Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 3253 of *LNCS*, pages 152–166. Springer, 2004.
6. P. Thati and G. Roşu. Monitoring algorithms for metric temporal logic specifications. In *Proceedings of the 4th Workshop on Runtime Verification*, volume 113 of *ENTCS*, pages 145–162. Elsevier, 2005.
7. S. Wang, A. Ayoub, O. Sokolsky, and I. Lee. Runtime verification of traces under recording uncertainty. In *Proceedings of the 2nd International Conference on Runtime Verification*, volume 7186 of *LNCS*, pages 442–456. Springer, 2012.
8. H. Zhang, Y. Diao, and N. Immerman. Recognizing patterns in streams with imprecise timestamps. *Proceedings of the VLDB Endowment*, 3(1–2):244–255, 2010.

A Proof Details

A.1 Proof of Theorem 2

We prove the theorem by induction on the height of the derivation tree. We distinguish cases based on the rule applied at the root of the derivation tree. Assume that $\rho, t \models \mathsf{tf}(\varphi)$.

- $\varphi : (\exists)$ is derived from $\varphi : (\forall)$.
  From the induction hypothesis, we get $\bar{\sigma}, t \models_\forall \varphi$. That is, for all $\pi \in TL(\bar{\sigma})$ we have $\pi, t \models \varphi$. As $TL(\bar{\sigma})$ is a non-empty set, there is a $\pi \in TL(\bar{\sigma})$ such that $\pi, t \models \varphi$. That is $\bar{\sigma}, t \models_\exists \varphi$.
- $\varphi = \mathsf{false}$ or $\varphi = \mathsf{true}$. These cases are trivial.
- $\varphi = p$. Then $p : (\exists)$ is derivable.
  We have that $\rho, t' \models p$, for some $t' \in [t - \delta, t + \delta]$. Thus $p \in \rho(t')$. As $\rho(t')$ is non-empty, it follows that there is an $i \in \mathbb{N}$ such that $t' = \tau_i$ and $p \in a_i$. Since $t \in [\tau_i - \delta, \tau_i + \delta]$, there is a $\pi \in TL(\bar{\sigma})$ such that $\pi, t \models p$. Hence $\bar{\sigma}, t \models_\exists p$.
- $\varphi = \neg p$. Then $\neg p : (\forall)$ is derivable.
  Suppose, by absurdity, that $\bar{\sigma}, t \not\models_\forall \neg p$. Then there is a $\pi \in TL(\bar{\sigma})$ such that $\pi, t \models p$. Therefore, $p \in \pi(t)$ and there is an $i \in \mathbb{N}$ such that $p \in a_i$ and $t \in [\tau_i - \delta, \tau_i + \delta]$. It follows that $p \in \rho(\tau_i)$. Hence $\rho, \tau_i \models p$ and thus $\rho, t \models \mathsf{tf}(p)$. This is a contradiction.
- $\varphi = \varphi_1 \wedge \varphi_2$ and $\varphi : (\exists)$ is derivable from $\varphi_1 : (\exists)$ and $\varphi_2 : (\forall)$.
  We have that $\rho, t \models \mathsf{tf}(\varphi_1) \wedge \mathsf{tf}(\varphi_2)$. From the induction hypothesis, we get that $\bar{\sigma}, t \models_\exists \varphi_1$ and $\bar{\sigma}, t \models_\forall \varphi_2$. That is, there is a $\pi_1 \in TL(\bar{\sigma})$ such that $\pi_1, t \models \varphi_1$ and for all $\pi_2 \in TL(\bar{\sigma})$ we have $\pi_2, t \models \varphi_2$. In particular $\pi_1, t \models \varphi_2$. Hence $\pi_1, t \models \varphi$, that is $\bar{\sigma}, t \models_\exists \varphi$.
- $\varphi = \varphi_1 \wedge \varphi_2$ and $\varphi : (\forall)$ is derivable from $\varphi_1 : (\forall)$ and $\varphi_2 : (\forall)$.
  From the induction hypothesis, we get that $\bar{\sigma}, t \models_\forall \varphi_1$ and $\bar{\sigma}, t \models_\forall \varphi_2$. That is, $\pi, t \models \varphi_1$ for all $\pi \in TL(\bar{\sigma})$, and $\pi, t \models \varphi_2$ for all $\pi \in TL(\bar{\sigma})$. It follows that $\pi, t \models \varphi$ for all $\pi \in TL(\bar{\sigma})$. That is, $\bar{\sigma}, t \models_\forall \varphi$.
- $\varphi = \varphi_1 \vee \varphi_2$ and $\varphi : (\exists)$ is derivable from $\varphi_1 : (\exists)$ and $\varphi_2 : (\exists)$.
  We have that, say, $\rho, t \models \mathsf{tf}(\varphi_1)$. By the induction hypothesis, $\bar{\sigma}, t \models_\exists \varphi_1$. It follows that $\bar{\sigma}, t \models_\exists \varphi$.
- $\varphi = \varphi_1 \vee \varphi_2$ and $\varphi : (\forall)$ is derivable from $\varphi_1 : (\forall)$ and $\varphi_2 : (\forall)$.
  We have that, say, $\rho, t \models \mathsf{tf}(\varphi_1)$. By the induction hypothesis, $\pi, t \models \varphi_1$ for all $\pi \in TL(\bar{\sigma})$. It follows that $\pi, t \models \varphi$ for all $\pi \in TL(\bar{\sigma})$. That is, $\bar{\sigma}, t \models_\forall \varphi$.
- $\varphi = \varphi_1 \,\mathsf{S}_I\, \varphi_2$ and $\varphi : (\exists)$ is derivable from $\varphi_1 : (\forall)$ and $\varphi_2 : (\exists)$.
  As $\rho, t \models \mathsf{tf}(\varphi_1) \,\mathsf{S}_I\, \mathsf{tf}(\varphi_2)$, we have that there is a $t_2 \leq t$ such that $t - t_2 \in I$, $\rho, t_2 \models \mathsf{tf}(\varphi_2)$, and $\rho, t_1 \models \mathsf{tf}(\varphi_1)$ for all $t_1 \in (t_2, t]$. By the induction hypothesis, there is a $\pi \in TL(\bar{\sigma})$ such that $\pi, t_2 \models \varphi_2$. Furthermore, for every $\pi' \in TL(\bar{\sigma})$ and $t_1 \in (t_2, t]$, we have $\pi', t_1 \models \varphi_1$. In particular, we have $\pi, t_1 \models \varphi_1$, for every $t_1 \in (t_2, t]$. Hence, $\pi, t \models \varphi_1 \,\mathsf{S}_I\, \varphi_2$. That is, $\bar{\sigma}, t \models_\exists \varphi$.
- $\varphi = \varphi_1 \,\mathsf{S}_I\, \varphi_2$ and $\varphi : (\forall)$ is derivable from $\varphi_1 : (\forall)$ and $\varphi_2 : (\forall)$. This case is similar with the previous one.

- $\varphi = \varphi_1 \mathsf{T}_I \varphi_2$ and $\varphi : (\exists)$ is derivable from $\varphi_1 : (\exists)$ and $\varphi_2 : (\forall)$.
  We have that $\rho, t \models \mathsf{tf}(\varphi_1) \mathsf{T}_I \mathsf{tf}(\varphi_2)$. Then either $\rho, t_2 \models \mathsf{tf}(\varphi_2)$ for all $t_2 \leq t$, or there is a $t_1 \leq t$ such that $t - t_1 \in I$, $\rho, t_1 \models \mathsf{tf}(\varphi_1)$, and $\rho, t_2 \models \mathsf{tf}(\varphi_2)$ for all $t_2 \in [t_1, t]$.
  Consider the first case. By the induction hypothesis, we have that $\pi, t_2 \models \varphi_2$ for all $t_2 \leq t$ and $\pi \in \mathit{TL}(\bar{\sigma})$. That is, $\pi, t \models \varphi_1 \mathsf{T}_I \varphi_2$ for all $\pi \in \mathit{TL}(\bar{\sigma})$ and thus $\bar{\sigma}, t \models_\exists \varphi$.
  Consider now the second case. By the induction hypothesis, we have that there is a $\pi_1 \in \mathit{TL}(\bar{\sigma})$ such that $\pi_1, t_1 \models \varphi_1$. And we also have that $\pi_2, t_2 \models \varphi_2$ for all $t_2 \in [t_1, t]$ and all $\pi_2 \in \mathit{TL}(\bar{\sigma})$, hence in particular for π_1. That is, $\pi_1, t \models \varphi_1 \mathsf{T}_I \varphi_2$ and thus $\bar{\sigma}, t \models_\exists \varphi$ also in this case.
- $\varphi = \varphi_1 \mathsf{T}_I \varphi_2$ and $\varphi : (\forall)$ is derivable from $\varphi_1 : (\forall)$ and $\varphi_2 : (\forall)$. This case is similar with the previous one.
- $\varphi = \varphi_1 \mathsf{U}_I \varphi_2$ or $\varphi = \varphi_1 \mathsf{R}_I \varphi_2$, and $\varphi_2 : (\forall)$, and $\varphi_1 : (\exists)$ or $\varphi_2 : (\forall)$. These cases are similar to the analogous cases for the dual past operators.

A.2 Proof of Proposition 4

The proof is by structural induction on the form of φ. We only present the base case, as the inductive steps are straightforward.

Let $\varphi = p \in P$. Suppose that $*_1 = *_2 = +$. The other cases are similar. Let $t \in \mathbb{T}$ such that $\rho_1, t \models \mathsf{tf}_{\epsilon_1}(p)$. That is, there is $t_1 \in [t - (\delta + \epsilon_1), t + (\delta - \epsilon_1)]$ such that $p \in \rho_1(t_1)$. The time constraint on t_1 is equivalent with $|t - (t_1 + \epsilon_1)| \leq \delta$. From $p \in \rho_1(t_1)$ it follows that there is $i \in \mathbb{N}$ such that $p \in a_i$ and $\tau_i = t_1 + \epsilon_1$. Let $t_2 := t_1 + \epsilon_1 - \epsilon_2$. We have that $\tau_i = t_2 + \epsilon_2$, thus $p \in \rho_2(t_2)$. Furthermore, $t - (t_1 + \epsilon_1) = t - (t_2 + \epsilon_2)$, hence $t_2 \in [t - (\delta + \epsilon_2), t + (\delta - \epsilon_2)]$. It follows that $\rho_2, t \models \mathsf{tf}_{\epsilon_2}(p)$.