Monitoring for a decidable fragment of MTL-$\int$

André de Matos Pedro[1], David Pereira, Luís Miguel Pinho, and Jorge Sousa Pinto[2]

[1] CISTER/INESC TEC, ISEP, Polytechnic Institute of Porto, Portugal
[2] HASLab/INESC TEC & Universidade do Minho, Portugal

Abstract. Temporal logics targeting real-time systems are traditionally undecidable. Based on a restricted fragment of MTL-$\int$, we propose a new approach for the runtime verification of hard real-time systems. The novelty of our technique is that it is based on incremental evaluation, allowing us to effectively treat duration properties (which play a crucial role in real-time systems). We describe the two levels of operation of our approach: offline simplification by quantifier removal techniques; and online evaluation of a three-valued interpretation for formulas of our fragment. Our experiments show the applicability of this mechanism as well as the validity of the provided complexity results.

1 Introduction

Temporal logics are widely used formalisms in the field of specification and verification of reactive systems [19], since they provide a natural and abstract technique for the analysis of safety and liveness properties. Linear Temporal Logic (LTL) describes properties concerning the temporal order of the input model, and is well studied in terms of expressiveness, decidability and complexity. *Timed temporal logics* are extensions of temporal logics with quantitative constraints to handle temporal logic specifications [2]. Metric Temporal Logic (MTL) [12,15] is an undecidable real-time extension of LTL, describing the temporal order constrained by quantitative intervals on the temporal operators.

These formalisms have been used for formal verification, either by deductive or by algorithmic methods [11]. However, real-time logics are notably less well-behaved than traditional temporal logics. In particular, the model checking problem for MTL is known to be undecidable [15]. Decidable real-time formalisms that can be used as alternatives are currently the focus of much attention.

A diversity of MTL fragments reveal that the undecidable results of MTL are due to the excessive precision of the timing constraints (i.e., *punctuality* [1]), the presence of unbounded temporal operators (*unboundedness*), the presence of *unsafe formulas*, and the excessive richness of the *semantic model* [15]. Metric Interval Temporal Logic (MITL) is a fragment that avoids punctuality by constraining any interval on the temporal operators to be non-singular; Bounded MTL (BMTL) is another fragment that, instead of avoiding punctual intervals, bounds intervals that are infinitely large. Both are decidable fragments. Syntactic restrictions on temporal logic operators of MTL may also result in decidable

fragments. Ouaknine and Worrell [16] describe a fragment of MTL named Safety MTL (SMTL), that does not allow expressing invariant formulas, and Bouyer *et al.* [5] have introduced the term *flatness* for MTL.

In addition to being undecidable, the previous logics also fail to capture the notion of *duration*. This notion, however, is of paramount importance when specifying and developing real-time systems, mainly because the fundamental results about the reliability of this class of systems are related to ensuring that the execution time of the involved components does not miss some predetermined deadline. Lakhneche and Hooman [13] came up with *Metric temporal logic with durations* (MTL-$\int$) and Chaochen and colleagues [9] with Duration Calculus, which provide expressive power to specify and reason about durations within *real intervals*. By applying syntactic and semantic restrictions it is possible to derive decidable fragments for duration properties.

The motivation for this work is that of providing an expressive formal language that fits the timing requirements of real-time systems, from the point of view of *runtime verification* (RV). RV is concerned with the problem of generating monitors from formal specifications, and adding these monitors into the target code as a safety-net that is able to detect abnormal behaviors and, possibly, respond to them via the release of counter-measures. As such, RV methods can be applied to systems where the source code is not available due to intellectual property, or in those cases where we have access to the code but the complexity of the system's requirements is too high to be addressed via any of the known static verification approaches.

The major contribution of this paper is a new mechanism for runtime verification of hard real-time systems regarding duration properties, based on a decidable fragment of MTL-$\int$ and a three-valued abstraction of this fragment. The fragment allows for expressing quantified formulae, and is adequate for quantifier elimination: we give an algorithm for the simplification of formulas containing quantifiers and free logic variables. Intuitively, we abstract our fragment into *first order logic of real numbers* ($\text{FOL}_{\mathbb{R}}$) to obtain quantifier-free formulas.

One particular application scenario for RV is in scheduling theory of hard real-time systems. Rigorous calculation of the *worst case execution time* (WCET) is commonly difficult, and the known approximation methods based on statistical abstractions degrade the dependability of the systems, since the available schedulability theory tends to assume the WCET. Application of monitors in this case will make the system more reliable. We will show through an application example (based on *resource models*, which are mechanisms that ensure time isolation for execution units) the interest of allowing formal specifications to express existential quantification over durations, for real applications.

The paper is organized as follows: in Section 2 we introduce suitable restrictions over MTL-$\int$; Section 3 describes the three-valued semantics of restricted MTL-$\int$, and Section 4 describes an algorithm for inequality abstraction. In Section 5 we then introduce an evaluation algorithm for the restricted MTL-$\int$ with three-valued semantics. Section 6 describes our experimental work and finally Section 7 discusses related work and concludes the paper.

2 Specification Language RMTL-$\int$

MTL-$\int$ is more expressive than DC [13], but is undecidable since the relation over terms or the term function may themselves be undecidable. Let us begin by briefly reviewing MTL-$\int$.

Definition 1. *Let $\mathcal{P}$ be a set of propositions and $\mathcal{V}$ a set of logic variables. The syntax of MTL-$\int$ terms η and formulas φ is defined inductively as follows:*

$$\eta ::= \alpha \mid x \mid f(\eta_1, \ldots, \eta_n) \mid \int^{\eta} \varphi$$
$$\varphi ::= p \mid R(\eta_1, \ldots, \eta_n) \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \varphi_1 \, U_{\sim\gamma} \, \varphi_2 \mid \varphi_1 \, S_{\sim\gamma} \, \varphi_2 \mid \exists x \, \varphi$$

where $\alpha \in \mathbb{R}$, $x \in \mathcal{V}$ is a logic variable, f a function symbol of arity n, $\int^{\eta} \varphi$ is the duration of the formula φ in the interval $[0, \eta]$, $p \in \mathcal{P}$ is an atomic proposition, U and S are temporal operators with $\sim \in \{<, =\}$, $\gamma \in \mathbb{R}_{\geq 0}$, and the meaning of $R(\eta_1, \ldots, \eta_n), \varphi_1 \vee \varphi_2, \neg\varphi, \exists x \, \varphi$ is defined as usual.

We will use the following abbreviations: $\varphi \wedge \psi$ for $\neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi$ for $\neg\varphi \vee \psi$, $\mathtt{tt}$ for $\varphi \vee \neg\varphi$, $\mathtt{ff}$ for $\varphi \wedge \neg\varphi$, $\Diamond_{\sim\gamma} \, \varphi$ for $\mathtt{tt} \, U_{\sim\gamma} \, \varphi$, and $\Box_{\sim\gamma} \, \varphi$ for $\neg(\mathtt{tt} \, U_{\sim\gamma} \, \neg\varphi)$.

An observation function σ of length $\delta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ over $\mathcal{P}$ is a function from $\mathcal{P}$ into the set of functions from interval $[0, \delta)$ into $\{\mathtt{tt}, \mathtt{ff}\}$. The length of σ is denoted by $\#\sigma$. A *logical environment* is any function $\upsilon : \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$. For any such υ, $x \in \mathcal{V}$ and $r \in \mathbb{R}$, we will denote by $\upsilon[x \mapsto r]$ the logical environment that maps x to r and every other variable y to $\upsilon(y)$. The following auxiliary definition will be used in the interpretation of the duration of a formula.

Definition 2 (MTL-$\int$ semantics). *The truth value of a formula φ will be defined relative to a model (κ, υ, t) consisting of a timed state sequence k, a logical environment υ, and a time instant $t \in \mathbb{R}_{\geq 0}$. We will write $(\kappa, \upsilon, t) \models \varphi$ when φ is interpreted as true in the model (κ, υ, t). Terms and formulas will be interpreted in a mutual recursive way. First of all, for each formula φ, timed state sequence k and logical environment υ, the auxiliary indicator function $1_{\varphi(\kappa,\upsilon)} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is defined as follows, making use of the satisfaction relation:*

$$1_{\varphi(\kappa,\upsilon)}(t) = \begin{cases} 1 & \text{if } (\kappa, \upsilon, t) \models \varphi, \\ 0 & \text{otherwise.} \end{cases}$$

The value $\mathscr{T}[\![\eta]\!](\kappa, \upsilon) \, t$ of a term η relative to a model is then defined using a Riemann integral [8] *of the function $1_{\varphi(\kappa,\upsilon)}$ for the case of a duration $\int^{\eta} \varphi$:*

$$\begin{aligned}
\mathscr{T}[\![\alpha]\!](\kappa, \upsilon) \, t &= \alpha \\
\mathscr{T}[\![x]\!](\kappa, \upsilon) \, t &= \upsilon(x) \\
\mathscr{T}[\![f(\eta_1, \ldots, \eta_n)]\!](\kappa, \upsilon) \, t &= f\left(\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon) \, t, \ldots, \mathscr{T}[\![\eta_n]\!](\kappa, \upsilon) \, t\right) \\
\mathscr{T}\left[\!\!\left[\int^{\eta} \varphi\right]\!\!\right](\kappa, \upsilon) \, t &= \begin{cases} \int_t^{t + \mathscr{T}[\![\eta]\!](\kappa,\upsilon)t} 1_{\varphi(\kappa,\upsilon)}(t_*) \; dt_* & \text{if } (*) \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

where $(*)$ *means that* $1_{\varphi(\kappa,\upsilon)}$ *satisfies the Dirichlet condition [13, p.7] and the sub-term* $\mathscr{T}[\![\eta]\!](\kappa,\upsilon)\,t$ *is non-negative. The satisfaction relation in turn is defined as follows:*

$$
\begin{array}{lll}
(\kappa,\upsilon,t) \models p & \textit{iff} & \sigma(p)(t) = \mathtt{tt} \textit{ and } t < \#\sigma \\
(\kappa,\upsilon,t) \models R(\eta_1,\ldots,\eta_n) & \textit{iff} & R(\mathscr{T}[\![\eta_1]\!](\kappa,\upsilon)\,t,\ldots,\mathscr{T}[\![\eta_n]\!](\kappa,\upsilon)\,t) \\
(\kappa,\upsilon,t) \models \varphi_1 \vee \varphi_2 & \textit{iff} & (\kappa,\upsilon,t) \models \varphi_1 \textit{ or } (\kappa,\upsilon,t) \models \varphi_2 \\
(\kappa,\upsilon,t) \models \neg\varphi & \textit{iff} & (\kappa,\upsilon,t) \not\models \varphi \\
(\kappa,\upsilon,t) \models \varphi_1\, U_{\sim\gamma}\, \varphi_2 & \textit{iff} & \textit{there exists } t' \textit{ such that } t < t' \sim t+\gamma,\ (\kappa,\upsilon,t') \models \varphi_2, \\
 & & \quad \textit{and for all } t'',\ t < t'' < t',\ (\kappa,\upsilon,t'') \models \varphi_1 \\
(\kappa,\upsilon,t) \models \varphi_1\, S_{\sim\gamma}\, \varphi_2 & \textit{iff} & \textit{there exists } t' \textit{ such that } t-\gamma \sim t' < t,\ (\kappa,\upsilon,t') \models \varphi_2, \\
 & & \quad \textit{and for all } t'',\ t' < t'' < t,\ (\kappa,\upsilon,t'') \models \varphi_1 \\
(\kappa,\upsilon,t) \models \exists x\,\varphi & \textit{iff} & \textit{there exists an } r \in \mathbb{R} \textit{ such that } (\kappa,\upsilon[x \mapsto r],t) \models \varphi
\end{array}
$$

Note that the semantics of the until operator is strict and non-matching [4].

To overcome the undecidability results of MTL-$\int$, we apply restrictions over MTL-$\int$. *Restricted metric temporal logic with durations* (RMTL-$\int$) is a syntactically and semantically restricted fragment of MTL-$\int$; the syntactic restrictions over MTL-$\int$ include the use of *bounded formulas*, of a single relation $<$ over the real numbers, the restriction of the n-ary function terms to use one of the $+$ or $\times$ operators, and a restriction of α constants to the set or rationals $\mathbb{Q}$. Tarski's theorem [21] states that the first-order theory of reals with $+$, $\times$, and $<$ allows for quantifiers to be eliminated. Algorithmic quantifier elimination leads to decidability, assuming that the truth values of sentences involving only constants can be computed. We will denote by Φ the set of RMTL-$\int$ formulas.

The semantic restrictions on the other hand include the conversion of the *continuous* semantics of MTL-$\int$ into an *interval-based* semantics, where models are timed state sequences and formulas are evaluated in a given logical environment at a time $t \in \mathbb{R}_{\geq 0}$. A timed state sequence κ is an infinite sequence of the form $\kappa = (p_0, [i_0, i'_0[), (p_1, [i_1, i'_1[) \ldots$, where $p_j \in \mathcal{P}$, $i'_j = i_{j+1}$ and $i_j, i'_j \in \mathbb{R}_{\geq 0}$ such that $i_j < i'_j$ and $j \geq 0$. The replacement rule for propositions is $(\kappa,\upsilon,t) \models p$ iff $p \in \kappa(t)$.

Real-time systems operate in continuous time, and state changes may be performed at any real-numbered time point; the semantics used in MTL-$\int$ is thus appropriate for reasoning about such systems. However, the verification of digital systems does not require the expressive power of continuous ($\mathbb{R}$) semantics. Instead one may use RMTL-$\int$, where the input model is restricted to the observation of a set of step functions, which are indeed *timed state sequences*. Many verification methods are based on the assumption that states are observed at integer points only [10] applying the notion of *digitization*, a technique that allows for the encoding of dense-time traces.

An important property of our restriction is that RMTL-$\int$ satisfies by construction the Dirichlet conditions implying the Riemann property:

Lemma 1. *For any formula* φ *in RMTL-*$\int$*, timed state sequence* κ*, and logical environment* υ*, the indicator function* $1_{\varphi(\kappa,\upsilon)}$ *is Riemann integrable.*

3 Three-valued Abstraction of RMTL-$\int$

The three-valued logic abstraction of RMTL-$\int$, which we will call *three-valued restricted metric temporal logic with durations* (RMTL-$\int_3$), is syntactically defined as before, but contains two new terms. These terms allow variables to be maximized and minimized in a certain interval, subject to a constraint given as a formula. The terms must be introduced here due to the situation in which no minimum or maximum exists (the formula is not satisfied in the interval), since we need to define an infeasible value instead of assigning a real number to these terms. The language of terms of RMTL-$\int_3$ is defined as follows:

$$\eta \ ::= \ \alpha \mid x \mid \underset{x \in I}{min}\ \varphi \mid \underset{x \in I}{max}\ \varphi \mid \eta_1 \circ \eta_2 \mid \int^{\eta} \varphi$$

where $\underset{x \in I}{min}\ \varphi$ and $\underset{x \in I}{max}\ \varphi$, with $I = [I_{min}, I_{max}]$ and $I_{min}, I_{max} \in \mathbb{R}$, and $\circ \in \{+, \times\}$. All other formulas and terms are as in RMTL-$\int$. We will denote by Φ^3 the set of RMTL-$\int_3$ formulas, and by Γ the set of RMTL-$\int_3$ terms.

Definition 3 (RMTL-$\int_3$ Semantics). *The truth value of a formula φ will again be defined relative to a model (κ, υ, t) consisting of a timed state sequence k, a logical environment υ, and a time instant $t \in \mathbb{R}_{\geq 0}$. The interpretation of the term η will be given by $\mathscr{T}[\![\eta]\!](\kappa, \upsilon)\, t \in \mathbb{R} \cup \{\bot_{\mathbb{R}}\}$, as defined by the following rules. Whenever $\mathscr{T}[\![\eta]\!](\kappa, \upsilon)\, t = \bot_{\mathbb{R}}$, this means that the term η is infeasible.*

Rigid terms:

- *$\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon)\, t$ is defined as α if $\eta_1 = \alpha$, and as $\upsilon(x)$ if $\eta_1 = x$*

Minimum and Maximum terms:

- *If $\eta_1 = \underset{x \in I}{min}\ \varphi$, then $\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon)\, t =$ is defined as:*

$$\begin{cases} \mathfrak{I} = \min\{r \mid r \in I \text{ and } (\kappa, \upsilon[x \mapsto r], t) \models_3 \varphi\} & \text{if } \mathfrak{I} \neq \emptyset \\ \bot_{\mathbb{R}} & \text{otherwise} \end{cases}$$

- *If $\eta_1 = \underset{x \in I}{max}\ \varphi$, then $\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon)\, t$ is defined as:*

$$\begin{cases} \mathfrak{Z} = \max\{r \mid r \in I \text{ and } (\kappa, \upsilon[x \mapsto r], t) \models_3 \varphi\} & \text{if } \mathfrak{Z} \neq \emptyset \\ \bot_{\mathbb{R}} & \text{otherwise} \end{cases}$$

Duration term:

- *If $\eta_1 = \int^{\eta_2} \phi$, then $\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon)\, t$ is defined as:*

$$\begin{cases} \int_t^{t + \mathscr{T}[\![\eta_2]\!](\kappa, \upsilon)\, t} 1_{\phi(\kappa, \upsilon)}(t')\ dt' & \text{if } \mathscr{T}[\![\eta_2]\!](\kappa, \upsilon)\, t \geq 0 \\ \bot_{\mathbb{R}} & \text{otherwise} \end{cases}$$

Turning to the interpretation of formulas, we define $[\![\varphi]\!](\kappa, \upsilon, t)$ to be one of the three values in $\{\mathsf{tt}, \mathsf{ff}, \bot\}$, according to the following rules.

Basic formulae:

- *If $\phi = p$, then $[\![\phi]\!](\kappa, \upsilon, t)$ is tt if $p \in \kappa(t)$, ff if $p \notin \kappa(t)$, and $\bot$ if $\kappa(t)$ is undefined.*

Relation operator:

- *If* $\phi = \eta_1 < \eta_2$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is defined as:*

$$\begin{cases} \mathsf{tt} & \text{if } \mathscr{T}[\![\eta_1]\!](\kappa,\upsilon)\,t < \mathscr{T}[\![\eta_2]\!](\kappa,\upsilon)\,t, \text{ and} \\ & \mathscr{T}[\![\eta_1]\!](\kappa,\upsilon)\,t, \mathscr{T}[\![\eta_2]\!](\kappa,\upsilon)\,t \in \mathbb{R} \\ \mathsf{ff} & \text{if } \mathscr{T}[\![\eta_1]\!](\kappa,\upsilon)\,t \geq \mathscr{T}[\![\eta_2]\!](\kappa,\upsilon)\,t, \text{ and} \\ & \mathscr{T}[\![\eta_1]\!](\kappa,\upsilon)\,t, \mathscr{T}[\![\eta_2]\!](\kappa,\upsilon)\,t \in \mathbb{R} \\ \bot & \text{otherwise} \end{cases}$$

Boolean operators:

- *If* $\phi = \neg\varphi$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is* tt *if* $[\![\varphi]\!](\kappa, \upsilon, t) = \mathsf{ff}$*,* ff *if* $[\![\varphi]\!](\kappa, \upsilon, t) = \mathsf{tt}$*, and* $\bot$ *otherwise.*
- *If* $\phi = \varphi_1 \vee \varphi_2$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is* tt *if* $[\![\varphi_1]\!](\kappa, \upsilon, t) = \mathsf{tt} \vee [\![\varphi_2]\!](\kappa, \upsilon, t) = \mathsf{tt}$*,* ff *if* $[\![\varphi_1]\!](\kappa, \upsilon, t) = \mathsf{ff} \wedge [\![\varphi_2]\!](\kappa, \upsilon, t) = \mathsf{ff}$*, and* $\bot$*otherwise.*

Temporal Operators:

- *If* $\phi = \varphi_1 \, U_{\sim\gamma} \, \varphi_2$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is defined as:*

$$\begin{cases} \mathsf{tt} & \text{if } \exists t',\, t < t' \sim t + \gamma \text{ such that } [\![\varphi_2]\!](\kappa,\upsilon,t') = \mathsf{tt}, \text{ and} \\ & \quad \forall t'',\, t < t'' < t', [\![\varphi_1]\!](\kappa,\upsilon,t'') = \mathsf{tt} \\ \mathsf{ff} & \text{if } \forall t',\, t < t' \sim t + \gamma \text{ such that} \\ & \quad [\![\varphi_1]\!](\kappa,\upsilon,t') = \mathsf{ff} \rightarrow \exists t'',\, t < t'' < t', [\![\varphi_1]\!](\kappa,\upsilon,t'') = \mathsf{ff} \\ \bot & \text{otherwise} \end{cases}$$

- *If* $\phi = \varphi_1 \, S_{\sim\gamma} \, \varphi_2$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is defined as:*

$$\begin{cases} \mathsf{tt} & \text{if } \exists t',\, t - \gamma \sim t' < t \text{ such that } [\![\varphi_2]\!](\kappa,\upsilon,t') = \mathsf{tt}, \text{ and} \\ & \quad \forall t'',\, t' < t'' < t, [\![\varphi_1]\!](\kappa,\upsilon,t'') = \mathsf{tt} \\ \mathsf{ff} & \text{if } \forall t',\, t - \gamma \sim t' < t \text{ such that} \\ & \quad [\![\varphi_1]\!](\kappa,\upsilon,t') = \mathsf{ff} \rightarrow \exists t'',\, t' < t'' < t, [\![\varphi_1]\!](\kappa,\upsilon,t'') = \mathsf{ff} \\ \bot & \text{otherwise} \end{cases}$$

Existential operator:

- *If* $\phi = \exists x \ \varphi$*, then* $[\![\phi]\!](\kappa, \upsilon, t)$ *is defined as:*

$$\begin{cases} \mathsf{tt} & \text{if there exists a value } r \in \mathbb{R} \text{ such that } [\![\varphi]\!](\kappa, \upsilon[x \mapsto r], t) = \mathsf{tt} \\ \mathsf{ff} & \text{if for all } r \in \mathbb{R} \text{ such that } [\![\varphi]\!](\kappa, \upsilon[x \mapsto r], t) = \mathsf{ff} \\ \bot & \text{otherwise} \end{cases}$$

We will write $(\kappa, \upsilon, t) \models_3 \varphi$ when $[\![\varphi]\!](\kappa, \upsilon, t) = \mathsf{tt}$, and $(\kappa, \upsilon, t) \not\models_3 \varphi$ when $[\![\varphi]\!](\kappa, \upsilon, t) = \mathsf{ff}$. In what follows we will often write $x \in I$ as an abbreviated form for $I_{min} < x \wedge x < I_{max}$, and $\eta_1 = \eta_2$ for $\neg(\eta_1 < \eta_2) \wedge \neg(\eta_1 > \eta_2)$.

Preservation of RMTL-$\int$ Semantics. An immediate motivation for the choice of defining a three-valued semantics for our logic fragment comes from the nature of runtime verification, which evaluates timed sequences where it is not possible to determine a definitive true or false value without analyzing the complete trace. For instance, considering a prefix $\varkappa_p$ of a timed sequence $\varkappa$, we have that the evaluation of the same formula in the models $(\varkappa, \upsilon, t)$ and $(\varkappa_p, \upsilon, t)$ produces

different truth values. Classic semantics cannot provide a common truth value to make consistent incremental evaluations of the model, which is an important feature for RV.

The semantic preservation of both truth and falsity for the three-valued logic is defined using the following two relations: a partial relation $\prec$ on $\{\mathsf{tt}, \mathsf{ff}, \bot\}$ defined by $\bot \prec \mathsf{tt}$, $\bot \prec \mathsf{ff}$, $\bot \prec \bot$, $\mathsf{tt} \prec \mathsf{tt}$, and $\mathsf{ff} \prec \mathsf{ff}$; and a partial relation $\lhd : \mathbb{R}\times\mathbb{R}\cup\{\bot_{\mathbb{R}}\}$ defined by $0 \lhd \bot_{\mathbb{R}}$, and $n \lhd m$, with $n, m \in \mathbb{R}$, which gives a distinct treatment to duration terms that evaluate to 0 in the standard semantics.

Definition 4. *Let (κ, υ, t) be a model. The three-valued semantics is said to* preserve the two-valued semantics *iff the following rules hold:*

1. *For basic formulas containing the relation operator, for all terms $\eta_1 \in$ RMTL-$\int$ and $\eta_2 \in$ RMTL-$\int_3$ excluding minimum and maximum terms, $\mathscr{T}[\![\eta_1]\!](\kappa, \upsilon)\, t \lhd \mathscr{T}[\![\eta_2]\!](\kappa, \upsilon)\, t$ holds and it implies that $0 \lhd \bot_{\mathbb{R}}$ if η_1 has the form $\int^{\eta_3} \phi$ and $\mathscr{T}[\![\eta_3]\!](\kappa, \upsilon)\, t < 0$; and $0 \lhd 0$ otherwise.*
2. *For each basic formula ϕ containing Boolean, temporal, and existential operators, $[(\kappa, \upsilon, t) \models_3 \gamma] \prec [(\kappa, \upsilon, t) \models \gamma]$ holds.*

We will now formulate two auxiliary results required to prove the semantic preservation of RMTL-$\int$ in RMTL-$\int_3$. From a close examination of the minimum and maximum term semantics, we have that these terms are indeed quantified formulas, interpreted as a minimum or a maximum value that satisfies the quantification, or as $\bot_{\mathbb{R}}$ when this minimum or maximum is nonexistent. First of all we observe that the following axioms [21, p. 205] extend to our present setting:

A 1 *$\eta_1 \circ \min_{x \in I} \phi \sim \eta_2$ implies that there exists an x such that $\eta_1 \circ x \sim \eta_2$, $x \in I$, and ϕ implies that for all y, $y < x$ and $\neg\phi$.*

A 2 *$\eta_1 \circ \max_{x \in I} \phi \sim \eta_2$ implies that there exists an x such that $\eta_1 \circ x \sim \eta_2$, $x \in I$, and ϕ implies that for all y, $y > x$ and $\neg\phi$.*

Theorem 1. *Let (κ, υ, t) be a model, and ϕ^3 be a formula in RMTL-$\int_3$. Then $[(\kappa, \upsilon, t) \models_3 \phi^3] \prec [(\kappa, \upsilon, t) \models f_t(\phi^3)]$.*

Example 1 (Application of Durations). Let us now consider a system whose evolution depends entirely on the occurrence of events, on evaluating propositions over these events, and that all of its tasks have an associated fixed set of events. Let ϕ_m be a formula that specifies the occurrence of the periodic release of resource events for a task in the system, and let ψ_m be a formula specifying every event triggered by that task. To monitor utilization and the release of timed resources, we employ the formula,

$$\phi_m \rightarrow \int^t \psi_m \leq \beta,$$

where t is the period of the periodic release of events, that is, their priority. Using the two-valued setting, the incremental evaluation is not allowed. This is not the case in our three-valued setting, since it allows the monitor to execute even when an incomplete trace is provided and incrementally evaluates the formula until a tt or ff verdict is obtained.

4 Inequality Abstraction Using a Theory of Reals

A close examination of the semantics of RMTL-$\int_3$ reveals that the timed state sequence κ and the logic environment υ are independent parameters for evaluating the truth value of formulas. This allows us to define a mechanism for introducing isolation by splitting formulas in two parts using a Boolean connective, and then analyzing one statically and the other at execution time.

The axiom system for the arithmetic of real numbers provided by Tarski [21] can be used as an abstraction of inequalities in RMTL-$\int_3$. Several properties provided by this well-known fragment will be used to facilitate the removal of existential quantifiers, when properties expressed as formulas containing them are monitored at execution time. From the Tarski–Seidenberg theorem [21] we have that for any formula in first order logic with *real* arithmetic $(\mathbb{R}, <, +, \times)$ containing an existential quantifier, there is an equivalent one without the quantifier. Thus there exists a decision procedure for quantifier elimination over $\mathrm{FOL}_{\mathbb{R}}$.

One of the most efficient algorithms, with complexity 2-EXPTIME, is the *cylindrical algebraic decomposition* later proposed by Collins [6,3]. To use it, we require a mechanism to solve quantified formulas as well as formulas without quantification, by applying several transformations on their Boolean connectives. Let us now describe the constraints required for an RMTL-$\int_3$ formula to be interpreted as a formula of $\mathrm{FOL}_{\mathbb{R}}$, and describe the notion of isolated formula.

Definition 5 (Inequality Abstraction Constraint). *Let ϕ_3 be a RMTL-$\int_3$. ϕ_3 is a formula in FOL$_{\mathbb{R}}$ if it is free of duration terms, minimum/maximum terms, temporal operators, and propositions.*

Definition 6 (Quantifier Isolation). *Let ϕ^3 be a formula in RMTL-$\int_3$. We say that ϕ^3 is an* isolated formula *if all propositions and temporal operators occur outside the scope of quantifications.*

Now, we require some axioms for isolating temporal operators and introducing logic variables in duration terms. Axioms A 3 and A 4 below describe how a temporal formula can be split into two: a quantified formula not containing occurrences of temporal operators, and another formula not containing quantifiers. Let $\phi_<$ be a formula in $\mathrm{FOL}_{\mathbb{R}}$, and $op \in \{\wedge, \vee\}$. Axiom 5 states that a formula containing a duration constrained in an interval can be transformed so the duration is constrained by a logic variable with appropriate coupled constraints. Intuitively, it reduces a duration term $\int^{\eta} \phi$ into $\int^{x} \phi$ with $x = \eta$.

A 3 $\left((\phi^1_< \mathit{op}_1 \phi_1)\ U\ (\phi^2_< \mathit{op}_2 \phi_2)\right) \rightarrow \left(\phi^2_< \mathit{op}_2\ \left(\neg(\phi^2_<) \rightarrow \left((\phi^1_< \mathit{op}_1 \phi_1)\ U\ \phi_2\right)\right)\right)$
A 4 $((\phi_< \mathit{op}_1 \phi_1)\ U\ \phi_2) \rightarrow ((\phi_< \rightarrow \mathit{true}\ U\ \phi_2)\ \mathit{op}_1\ \phi_1\ U\ \phi_2)$
A 5 $\int^{\eta_x} \phi_1 \circ \eta_1 \sim \eta_2 \rightarrow \exists x \left(x = \eta_x \wedge \neg(x < 0) \wedge \int^{x} \phi_1 \circ \eta_1 \sim \eta_2\right)$

In addition, for a formula to be compliant with Definition 5 we also require a technique for isolating propositions. We will not describe here axioms or strategies for their application, and simply consider that some mechanism for formula isolation is available, allowing a formula to be transformed into a disjunction or conjunction of a quantified part on the left and a propositional part on the right.

Let us now see a practical application of these axioms.

Require: a formula ϕ in RMTL-$\int$
Ensure : a simplified formula ϕ, a formula $\phi_{\not<}$ in RMTL-$\int$ without logic variables, and op a Boolean operator

1 **Function** `simplify_inequalities` (ϕ) **is**
begin
2 **If not** `Is_Variable_Free`(ϕ) **then** return ϕ, tt, $\wedge$;
3 $\phi_r, E, f_E \leftarrow$ `Replace_Non_Rigid_Terms`(`Simplify_minmax_Terms`(ϕ));
4 $\phi_<, \phi_{\not<}, op \leftarrow$`Isolate_Quantifiers`$(\phi_r)$;
5 $\phi_{smp} \leftarrow$ `Minimum_Assignment`(`Cylindrical_Decomposition`$(\phi_<)$);
6 **for** $\eta_x \in E$ **do**
   begin
7 $\phi \leftarrow$ get ϕ of $\int^{\eta} \phi = f_E(\eta_x)$;
8 $\phi_{<s}, \phi_{\not<s}, op \leftarrow$ `simplify_inequalities` (ϕ);
9 replace $f_E(\eta_x)$ in ϕ_{smp} with η if $op = \vee$ and $\phi_{<s} =$ tt or with $\int^{\eta} \phi_{\not<s}$ otherwise
10 $\phi_{smp} \leftarrow \phi_{smp}\ op\ \phi_{<s}$
   end
11 return ϕ_{smp}, $\phi_{\not<}$, op
end

Algorithm 1: Simplification of RMTL-$\int_3$ Inequalities

Example 2. Consider the formula $\exists x \int_{x \in I}^{min\ \phi_2} \phi_1 \sim \eta$, where I is some positive real interval. By Axiom 5, we know that $\exists x\ x = \underset{x \in I}{min}\ \phi_2 \wedge \int^x \phi_1 \sim \eta$ holds. Then we get $\exists x\ \exists y \left(x = y \wedge y \in I \wedge (\phi_2 \rightarrow (\forall z\ z < y \wedge \neg\phi_2)) \wedge \int^x \phi_1 \sim \eta\right)$ by applying Axiom 1. We proceed by removing the logic variables x, y and z via cylindrical decomposition and substituting the duration terms accordingly, wich leads us to the formula scheme $\int^{\{\alpha_n\}} \phi_1 \sim \eta \coloneqq \int^{\alpha_1} \phi_1 \sim \eta \vee \cdots \vee \int^{\alpha_n} \phi_1 \sim \eta$, where $\{\alpha_n\}$ is the set of all solutions that satisfy constrained formulas resulting from the decomposition process.

It is clear from Example 2 that the formula can be as long as the cardinality of $\{\alpha_n\}$. To overcome this limitation some optimization techniques can be applied for checking the formula at runtime. One of them begins by reducing the search space of the logical variables, constraining only the variable x with the upper values of the sub-intervals when x has finite many discontinuities, or with the maximum value that the variable x allows. Since we are using a real arithmetic theory that admits existential quantification removal, a general method to compute minimum satisfying assignments [7] can be employed. The order of the quantifiers is extremely important for faster results when computing at execution time the set of values. Intuitively, the method begins by reorganizing the universal quantifiers considering the same cost for each quantifier, in order to find the minimum satisfying assignment for a certain formula.

Formulas and terms are of course defined in a mutually inductive way, so any algorithm for formula simplification will have to follow this mutual recursion structure. An algorithm can start from the leafs of a formula's parse tree and proceed up to the root (bottom-up), or conversely implement a top-down analysis. The bottom-up approach is inefficient, since each node will necessarily be analyzed. To avoid this, we propose Algorithm 1, a branch-and-cut algorithm for simplification of formulas containing quantified inequalities.

The algorithm begins by testing if a formula contains free logic variables or existential quantifiers. If the formula can be simplified we proceed, otherwise

we return the input formula $\phi \wedge \mathtt{tt}$ (Line 2). Next, the duration terms are replaced by new fresh variables in υ, and minimum and maximum terms are transformed into quantified inequalities. The function `Simplify_minmax_Terms` applies min/max term substitutions as provided by axioms 1, 2, and 5. The function `Replace_Non_Rigid_Terms` returns a triple composed of a formula ϕ_r containing the abstracted inequality, the set E of fresh logical variables, and a one-to-one mapping f_E from elements of E into formulas. The function `Isolate_Quantifiers` isolates propositions and temporal operators from the scopes of quantifiers. The function `Cylindrical_Decomposition` applies the algebraic method for cylindrical decomposition of polynomials with a certain order over the logical variables, and the function `Minimum_Assignment` introduces the minimum satisfying assignment to reorder and discard quantifiers when possible.

For each term, the algorithm recursively calls the simplification function for the formula given by the terms η_x and applies some substitutions (Lines 7 and 9). The simplified formula ϕ_{smp} may contain terms of the form $\int^x \phi$; for deciding these we require an external procedure for incremental evaluation of durations, where the bounds of x are known prior to evaluation, as in Example 2.

5 Computation of RMTL-$\int_3$ formulae

Given the definition of RMTL-$\int_3$, we can derive an evaluation algorithm for monitor synthesis. In what follows we will present the algorithm and study the time complexity of the computation with respect to both trace and formula size.

We begin with a set of preliminary definitions. The set of timed sequences is denoted by $\mathbf{K}$, the duration of the timed state sequence $\kappa \in \mathbf{K}$ is denoted by $d^{(\kappa)}$, and the set of logic environments is denoted by Υ. Let $\mathbf{B}_4$ be the set $\{\mathtt{tt}_4, \mathtt{ff}_4, \bot_4\} \cup \{\mathfrak{r}\}$ where $\mathfrak{r}$ is a new symbol that will be used only for purposes of formulae evaluation, and $\mathbf{D}$ the set $\mathbb{R}_{\geq 0} \cup \{\bot_{\mathbb{R}}\}$. The function $sub_{\mathbf{K}} : (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \mathbb{R}_{\geq 0} \to \mathbf{K}$ defines a timed sub-sequence constrained by the interval $]t, t+\gamma]$, where t and γ are real numbers to be used as parameters in $sub_{\mathbf{K}}$. The function $map^{\mathbf{B}_4} : \mathbb{B}_3 \to \mathbf{B}_4$ maps $\mathtt{tt}$ to $\mathtt{tt}_4$, $\mathtt{ff}$ to $\mathtt{ff}_4$ and $\bot$ to $\bot_4$; $map^{\mathbb{B}_3} : \mathbb{B} \times \mathbf{B}_4 \to \mathbb{B}_3$ maps $(\mathtt{tt}, \mathfrak{r})$ to $\bot$; $(\mathtt{ff}, \mathfrak{r})$, $(\mathtt{ff}, \mathtt{ff}_4)$, and $(\mathtt{tt}, \mathtt{ff}_4)$ to $\mathtt{ff}$; and $(\mathtt{ff}, \mathtt{tt}_4)$ and $(\mathtt{tt}, \mathtt{tt}_4)$ to $\mathtt{tt}$. We will employ a left $fold$ function defined in the usual way.

From close examination of the operators, we derive the corresponding Compute$_{(\neg)}$ and Compute$_{(\vee)}$ evaluation functions with time complexity constant in the number of timed sequence symbols, and linear in the size of the formula. Let us consider the functions Compute$_{(\eta)} :: (\mathbf{K} \times \Upsilon) \to \mathbb{R} \to \Gamma \to \mathbf{D}$ and Compute$_{\varphi} :: (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \Phi^3 \to \mathbb{B}_3$ for the evaluation of $U_<$ and $<$, and the term $\int$.

Operator $U_<$. Given formulas ϕ_1, ϕ_2 and $\gamma \in \mathbb{R}_{\geq 0}$, the formula $\phi_1 \, U_{<\gamma} \, \phi_2$ is evaluated in a model (κ, υ, t) by the function Compute$_{(U_<)}$: $(\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \mathbb{R}_{\geq 0} \to \Phi^3 \to \Phi^3 \to \mathbb{B}_3$, defined in Figure 1. We report here only on the computation function Compute$_{(U_<)}$; the remaining functions would be Compute$_{(U_=)}$ for punctual until, Compute$_{(S_<)}$ for the non-punctual dual operator, and Compute$_{(S_=)}$ for the punctual dual operator. These operators have at

$$
\begin{array}{ll}
\mathrm{ev}^{\mathrm{i}}_{\mathrm{al}} & :: \mathbb{B}_3 \to \mathbb{B}_3 \to \mathbf{B}_4 \\
\mathrm{ev}^{\mathrm{i}}_{\mathrm{al}}\ b_1\ b_2 & \triangleq \begin{cases} map^{\mathbf{B}_4}\ b_2 & \text{if } b_2 \neq \mathsf{ff} \\ map^{\mathbf{B}_4}\ b_1 & \text{if } b_1 \neq \mathsf{tt} \text{ and } b_2 = \mathsf{ff} \\ \mathfrak{r} & \text{otherwise} \end{cases} \\
\mathrm{ev}^{\mathrm{b}}_{\mathrm{al}} & :: (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \Phi^3 \to \Phi^3 \to \mathbf{B}_4 \to \mathbf{B}_4 \\
\mathrm{ev}^{\mathrm{b}}_{\mathrm{al}}\ m\ \phi_1\ \phi_2\ v & \triangleq \begin{cases} \mathrm{ev}^{\mathrm{i}}_{\mathrm{al}}\ (\mathrm{Compute}_{\varphi}\ m\ \phi_1)\ (\mathrm{Compute}_{\varphi}\ m\ \phi_2) & \text{if } v = \mathfrak{r} \\ v & \text{otherwise} \end{cases} \\
\mathrm{ev}^{\mathrm{fold}}_{\mathrm{al}} & :: (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \Phi^3 \to \Phi^3 \to \mathbf{K} \to \mathbf{B}_4 \\
\mathrm{ev}^{\mathrm{fold}}_{\mathrm{al}}\ (\kappa, \upsilon, t)\ \phi_1\ \phi_2\ \varkappa & \triangleq fold\ \left(\lambda v\ (p, (i, t')) \to \mathrm{ev}^{\mathrm{b}}_{\mathrm{al}}\ (\kappa, \upsilon, t' - \epsilon)\ \phi_1\ \phi_2\ v\right)\ \mathfrak{r}\ \varkappa \\
ev^{C}_{al} & :: (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \mathbb{R}_{\geq 0} \to \Phi^3 \to \Phi^3 \to \mathbf{K} \to (\mathbb{B} \times \mathbf{B}_4) \\
ev^{C}_{al}\ (\kappa, \upsilon, t)\ \gamma\ \phi_1\ \phi_2\ \varkappa & \triangleq \left(d^{(\kappa)} \leq t + \gamma,\ \mathrm{ev}^{\mathrm{fold}}_{\mathrm{al}}\ (\kappa, \upsilon, t)\ \phi_1\ \phi_2\ \varkappa\right) \\
\mathbf{Compute}_{(U_<)}\ m\ \gamma\ \phi_1\ \phi_2 & \triangleq \begin{cases} map^{\mathbb{B}_3}\ \left(ev^{C}_{al}\ m\ \gamma\ \phi_1\ \phi_2\ (sub_{\mathbf{K}}\ m\ \gamma)\right) & \text{if } \gamma \geq 0 \\ \mathsf{ff} & \text{otherwise} \end{cases} \\
\hline
ev^{<}_{al} & :: \mathbb{R} \to \mathbb{R} \to \mathbb{R} \\
ev^{<}_{al}\ val_1\ val_2 & \triangleq \begin{cases} val_1 < val_2 & \text{if } val_1 \in \mathbb{R} \text{ and } val_2 \in \mathbb{R} \\ \bot & \text{otherwise} \end{cases} \\
\mathbf{Compute}_{(<)}\ m\ h_1\ h_2 & \triangleq ev^{<}_{al}\ \left(\mathrm{Compute}_{(\eta)}\ m\ h_1\right)\ \left(\mathrm{Compute}_{(\eta)}\ m\ h_2\right) \\
\hline
1_{\varphi(\kappa,\upsilon)} & :: (\mathbf{K} \times \Upsilon) \to \mathbb{R}_{\geq 0} \to \Phi^3 \to \{0, 1\} \\
1_{\varphi(\kappa,\upsilon)}\ (\kappa, \upsilon)\ t\ \phi & \triangleq \begin{cases} 1 & \text{if } \mathrm{Compute}_{\varphi}\ (\kappa, \upsilon, t)\ \phi = \mathsf{tt} \\ 0 & \text{otherwise} \end{cases} \\
ev^{\eta}_{al} & :: (\mathbf{K} \times \Upsilon) \to \Phi^3 \to \mathbf{K} \to \mathbb{R}_{\geq 0} \\
ev^{\eta}_{al}\ (\kappa, \upsilon)\ \phi\ \varkappa & \triangleq fold\ \left(\lambda s, (p, (i, t')) \to t' \cdot \left(1_{\varphi(\kappa,\upsilon)}\ (\kappa, \upsilon)\ t'\ \phi\right) + s\right)\ 0\ \varkappa \\
\mathbf{Compute}_{(\int)}\ (\kappa, \upsilon)\ t\ a\ \phi & \triangleq \begin{cases} ev^{\eta}_{al}\ (\kappa, \upsilon)\ \phi\ (sub_{\mathbf{K}}\ (\kappa, \upsilon, t)\ a) & \text{if } a \geq 0 \\ \bot_{\mathbb{R}} & \text{otherwise} \end{cases}
\end{array}
$$

Fig. 1: Evaluation of the operators $U_<$ and $<$, and of duration terms

most two new branches. Given an input κ with size n_κ, and m a measure of the number of temporal operators in φ, we obtain from the structure of the computation the lower bound of time complexity $2(n_\kappa)^2 \cdot m(\varphi) - 4(n_\kappa)^2 + n_\kappa \cdot m(\varphi) - 2(n_\kappa)$.

Operator $<$. Given two terms $\eta_1, \eta_2 \in \Gamma$, the formula $\eta_1 < \eta_2$ is evaluated relative to a model (κ, υ, t) by the function $\mathrm{Compute}_{(<)} : (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \Gamma \to \Gamma \to \mathbb{B}_3$, also shown in Figure 1. The time complexity of this computation is constant, since any formula containing only the relation operator $<$ cannot have the size of the formula greater than one or consume any input symbols.

Term $\int$. The evaluation of a duration term $\int^a \phi$ in the model (κ, υ, t) is performed by the function $\mathrm{Compute}_{(\int)} : (\mathbf{K} \times \Upsilon) \to \mathbb{R}_{\geq 0} \to \mathbb{R} \to \Phi^3 \to \mathbf{D}$, again defined in Figure 1. It has linear time complexity in the size of the timed sequence, and constant time complexity in the formula size. $+$ and $\times$ terms are directly mapped into their respective computational operations. The complexity of those operations is directly related to the number of terms. Given a formula φ and a measure m_η describing the number of operators $+$ and $\times$ occurring in a formula φ, we have a linear lower bound of time complexity in $m_\eta(\varphi)$.

Function $\text{Compute}_{(\eta)}\ (\kappa, \upsilon)\ t\ h :: (\mathbf{K} \times \Upsilon) \to \mathbb{R} \to \Gamma \to \mathbf{D}$ **is**
  case h **of**
    α : $eval_\alpha\ \alpha$
    $h_1 + h_2$: $\left(\text{Compute}_{(\eta)}\ m\ h_1\right) + \left(\text{Compute}_{(\eta)}\ m\ h_2\right)$
    $h_1 \times h_2$: $\left(\text{Compute}_{(\eta)}\ m\ h_1\right) \times \left(\text{Compute}_{(\eta)}\ m\ h_2\right)$
    $\int^{h_1} \phi$: $\text{Compute}_{(\int)}\ (\kappa, \upsilon)\ t\ \left(\text{Compute}_{(\eta)}\ (\kappa, \upsilon)\ t\ h_1\right)\ \phi$
  end
end

Function $\text{Compute}_{\varphi}\ m\ \phi :: (\mathbf{K} \times \Upsilon \times \mathbb{R}_{\geq 0}) \to \Phi^3 \to \mathbb{B}_3$ **is**
  case ϕ **of**
    p : $eval_p\ m\ p$ – base case
    $\neg\phi$: $\text{Compute}_{(\neg)}\ m\ \phi$ – Boolean operators
    $\phi_1 \vee \phi_2$: $\text{Compute}_{(\vee)}\ m\ \phi_1\ \phi_2$
    $\phi_1\ U_{<\gamma}\ \phi_2$: $\text{Compute}_{(U_<)}\ m\ \gamma\ \phi_1\ \phi_2$ – temporal operators
    $\phi_1\ S_{<\gamma}\ \phi_2$: $\text{Compute}_{(S_<)}\ m\ \gamma\ \phi_1\ \phi_2$
    $\eta_1 < \eta_2$: $\text{Compute}_{(<)}\ m\ \eta_1\ \eta_2$ – relational operator
  end
end

Algorithm 2: Computation of RMTL-$\int_3$ formulas (Compute_{φ})

Time complexity of the evaluation algorithm. We are now in a position to present the recursive top-level evaluation Algorithm 2 excluding *punctual* temporal operators, using the previous definitions for auxiliary computations. Let m be a measure for $\vee$, $<$, temporal operators, and non-rigid terms. Given the complexity of these formulas and term operators, and knowing that all temporal operators have the same complexity as the until operator, we have by semantic definition that any combination of formulas has higher complexity. As such, the complexity of Algorithm 2 is polynomial in the input size of the formula and the timed state sequence, as given by the lower bound identified above.

6 Experiments

Our approach uses an offline algorithm for formula simplification, and an online evaluation procedure that can be directly applied for the synthesis of runtime monitors. We will now show an example of application of Algorithm 1 for monitoring the budget of a set of *resource model*s (RMs); then we will present the empirical validation of the complexity results for Algorithm 2.

RMs are mechanisms to ensure time isolation between tasks. In the case of periodic RMs [20], they are defined by a replenishment period and a budget supply. The budget supply is available as time passes, and is replenished at each period by the resource model. Elastic periodic RMs are resource models containing *elastic coefficients* (similar to spring coefficients in physics), describing how a task can be compressed when the system is overloaded, allowing RV of imprecise computation. Naturally, the coefficients need to be constrained (linearly or non-linearly) before execution. Intuitively, the idea is to check the coefficients according to the polynomial constraints using our static phase, and provide the simplified formulas for the further runtime evaluation phase.

Let us now extend Example 1 for multiple RMs, considering without loss of generality the case of two RMs. We will use indexed formulas ϕ_{m_i}, ψ_{m_i} with $0 \leq i < 2$, and let α_i, α_{ai} be pre-defined constants. For measuring their budgets we could use the following invariant:

$$\bigwedge_{i=0}^{n-1} \phi_{m_i} \wedge \Box_{<\infty^*} \left(\left(\bigwedge_{i=0}^{n-1} \phi_{m_i} \right) \rightarrow \left(0 \leq \sum_{i=0}^{n-1} c_i \times \int^{\alpha_i} \psi_{m_i} < \alpha_b \wedge r_m \wedge \Diamond_{=\pi} \bigwedge_{i=0}^{n-1} \phi_{m_i} \right) \right),$$

where c_i are coefficients that have different weights for each RM, compliant with the restrictions r_m constrained in the interval $[0, \alpha_b[$, $\alpha_b \in \mathbb{R}_{\geq 0}$, and $\bigwedge_{i=0}^{n-1} \phi_{m_i}$ corresponds to the periodic release of the RMs with period π. A more detailed description can be found in [17]. The problem is then to find values for c_1, c_2 satisfying the constraints $r_1 := \frac{1}{250}(245 - 444x + 200{c_1}^2) = c_2$, $r_2 := 1 - c_1 = c_2$, or $r_3 := 1 - {c_1}^2 = c_2$, as shown in Figure 2, based on two duration observations over ψ_{m_i} formulas.

We will use Algorithm 1 for discarding possible inconsistencies, and decompose the formulas into sub-formulas that are free of quantifiers. Let us simplify the previously defined invariant for two resource models where the coefficient c_0 is existentially quantified and constrained by r_2. After some transformations on the formula we obtain

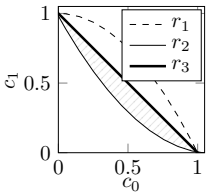
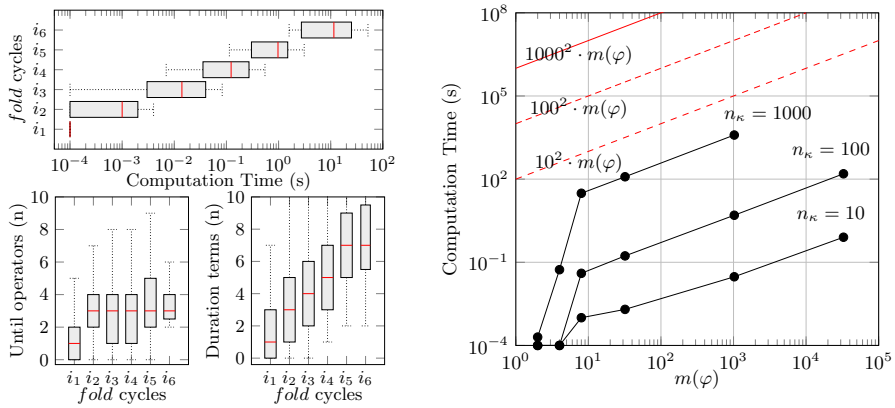


Fig. 2

$$\phi^1_{\not<} := \phi_{m_0} \wedge \phi_{m_1} \wedge \neg(\mathsf{tt}\, U_{<\infty^*} ((\phi_{m_0} \wedge \phi_{m_1} \wedge \neg \Diamond_{=\pi} (\phi_{m_0} \wedge \phi_{m_1})) \vee (\phi_{m_0} \wedge \phi_{m_1} \wedge \neg \phi^1_<))),$$

such that $\phi^1_< := \exists c_0\ 0 \leq c_0 \times a + c_1 \times b < \alpha_b \wedge 1 - c_0 = c_1 \wedge c_0 \geq 0 \wedge c_1 \geq 0$ holds. Duration terms have been replaced by the logic variables a and b. Since Axioms 3 and 4 cannot be used here for isolation purposes, we have to substitute the inequality formula by a constant Θ. We will then have an isolated formula, and apply cylindrical algebraic decomposition to determine if $\phi^1_<$ is satisfied. If it is, then we directly replace Θ by tt, otherwise we have the bounds that satisfy $\phi^1_<$. For this case, we obtain $(a = 0 \wedge b \geq 10 \wedge 0 \leq c_1 < \frac{10}{b}) \vee (a = 0 \wedge 0 \leq b < 10 \wedge 0 \leq c_1 \leq 1) \vee (a \geq 10 \wedge \frac{a-10}{a-b} < c_1 \leq 1 \wedge 0 \leq b < 10) \vee (b \geq 10 \wedge 0 < a \wedge a < 10 \wedge 0 \leq c_1 < \frac{a-10}{a-b}) \vee (0 < a < 10 \wedge 0 \leq b < 10 \wedge 0 \leq c_1 \leq 1)$. This is applied recursively for all the terms that have been substituted by fresh logic variables. In this particular case there are no subsequent iterations. After these steps the simplified bounds are ready to be evaluated by the online method.

Let us now discuss the complexity of Algorithm 2 and establish an empirical comparison with the lower bounds presented previously. We observe that the generation of nested durations is more critical on average than the nesting of temporal operators. This result matches the semantics of both terms and formulas, since the duration terms can integrate any indicative function provided for any trace, unlike the until operator that requires a successful trace to maximize its search. Consider Figure 3a, where the boxes i_1 to i_6 are respectively the intervals $]10^j, 10^{j+1}]$ for all $j \in [1, 7[$. They represent the number of cycles performed by folding functions. The results confirm that as the number of until operators stabilizes and the number of duration operators increases, the computation time also increases at a higher rate due to the presence of durations. This occurs for *generated* uniform formulas and traces; deep nesting of until operators and nested durations is unlikely to occur in hand-written specifications



(a) computation time vs. execution cycles of fold functions, $m(\varphi) = 2^5 - 1$ and $n_\kappa = 1000$

(b) computation time vs. formula size constructed with nested Until operators

Fig. 3: Experimental validation of the complexity results

(it has not been clearly confirmed whether they are useful for real-life applications). The experiments confirm the theoretical complexity bounds obtained earlier (Figure 3b). We have performed the experiments on an Intel Core i3-3110M at 2.40GHz CPU, and 8 GB RAM running Fedora 21 X86'64; the source code is available in `http://webpages.cister.isep.ipp.pt/~anmap/rv15/`.

7 Discussion and Future Work

We have developed a new approach for the RV of hard real-time systems, where duration properties play an important role, and incremental evaluation is required. The closest approaches to ours are that of Nickovic and colleagues [14], who provide synthesis algorithms for MTL specifications, and the work of Pike and colleagues [18], who have developed a framework based on a formal stream language embedded into Haskell, together with a synthesis mechanism that generates monitors to run in a distributed way. However, none of these previous approaches is sufficiently expressive to allow reasoning about duration properties, which is the novelty of our work.

The first level of operation of our approach consists of offline analysis for the simplification of formulas by means of quantifier removal techniques; the second is an online evaluation algorithm for RV purposes. This algorithm is polynomial in the sizes of the trace and of the formula, as confirmed by our experiments, which also reveal that the duration terms are on average computationally more demanding than the temporal operators. We restrict syntactically and semantically the two-valued MTL-$\int$ logic, with a three-valued interpretation. Incremental evaluation allows our technique to handle millions of samples, with formulas containing hundreds of operators. It remains to be seen whether extensions of LTL that are strictly more expressive than MTL, such as TPTL [4] could be used as an alternative for dealing with durations.

References

1. R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, January 1996.
2. R. Alur and T.A. Henzinger. Logics and models of real time: A survey. In *Proceedings of the Real-Time: Theory in Practice, REX Workshop*, pages 74–106, London, UK, UK, 1992. Springer-Verlag.
3. S. Basu, R. Pollack, and M.F. Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer, 2006.
4. P. Bouyer, F. Chevalier, and N. Markey. On the expressiveness of TPTL and MTL. *Information and Computation*, 208(2):97 – 116, 2010.
5. P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell. On expressiveness and complexity in real-time model checking. ICALP '08, pages 124–135, Berlin, Heidelberg, 2008. Springer-Verlag.
6. G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: A synopsis. *SIGSAM Bull.*, 10(1):10–12, February 1976.
7. I. Dillig, T. Dillig, K.L. McMillan, and A. Aiken. Minimum satisfying assignments for SMT. CAV'12, pages 394–409, Berlin, Heidelberg, 2012. Springer-Verlag.
8. R.A. Gordon. *The Integrals of Lebesgue, Denjoy, Perron, and Henstock*. Graduate studies in mathematics. American Mathematical Soc., 1994.
9. M.R. Hansen and D. Van Hung. Domain modeling and the duration calculus. chapter A Theory of Duration Calculus with Application, pages 119–176. Springer-Verlag, Berlin, Heidelberg, 2007.
10. T.A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? ICALP '92, pages 545–558, London, UK, UK, 1992. Springer-Verlag.
11. M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning About Systems*. Cambridge University Press, New York, NY, USA, 2004.
12. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4):255–299, October 1990.
13. Y. Lakhneche and J. Hooman. Metric temporal logic with durations. *Theor. Comput. Sci.*, 138(1):169–199, February 1995.
14. D. Ničković and N. Piterman. From MTL to deterministic timed automata. FORMATS'10, pages 152–167, Berlin, Heidelberg, 2010. Springer-Verlag.
15. J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. FORMATS '08, pages 1–13, Berlin, Heidelberg, 2008. Springer-Verlag.
16. Joël Ouaknine and James Worrell. Safety metric temporal logic is fully decidable. In H. Hermanns and J. Palsberg, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920 of *LNCS*, pages 411–425. Springer Berlin Heidelberg, 2006.
17. A. M. Pedro, D. Pereira, L. M. Pinho, and J. S. Pinto. Logic-based Schedulability Analysis for Compositional Hard Real-Time Embedded Systems. *SIGBED rev.*, 2015. to appear.
18. L. Pike, A. Goodloe, R. Morisset, and S. Niller. Copilot: A hard real-time runtime monitor. RV'10, pages 345–359, Berlin, Heidelberg, 2010. Springer-Verlag.
19. A. Pnueli. The temporal logic of programs. SFCS '77, pages 46–57, Washington, DC, USA, 1977. IEEE Computer Society.
20. I. Shin and I. Lee. Periodic resource model for compositional real-time guarantees. RTSS '03, pages 2–, Washington, DC, USA, 2003. IEEE Computer Society.
21. A. Tarski. *Introduction to Logic and to the Methodology of Deductive Sciences*. Dover Books on Mathematics Series. Dover Publications, 1995.

A Appendix

Proof of Lemma 1 (sketch). Let $(\kappa, \upsilon, t) \models p$ behave as a step function along t for any υ, and

$$\mathscr{T}\left[\!\left[\int^1 \phi\right]\!\right](\kappa, \upsilon)\, t = 1 - t.$$

We proceed by contradiction on the claim that the function $1_{\varphi(\kappa,\upsilon)}$ has finitely many discontinuities for any $t \in \mathbb{R}$. In the case when ϕ is a formula of the form

$$\neg(\int^1 \phi < 1 \circ a) \wedge \neg(\int^1 \phi > 1 \circ a)$$

then t is directly related to the variable a, when the timed state sequence κ has finite length 1. If a has infinitely many discontinuities along t then $1_{\varphi(\kappa,\upsilon)}$ also contains infinitely many discontinuities. Considering the above relation between t and the logic variable a, we can only introduce discontinuities along t by a. However, we can only introduce many discontinuities in $\mathbb{R}$ if we have an infinite formula. Consequently, from a close examination of the semantics of the logic, we can conclude that t can be constrained only by linear combination using the operators $+$ and $\times$, and that any formula to be evaluated needs to be finite, since the discontinuities in terms can only be provided by the relation $<$.

We skip the proof for the remaining cases, since no more relations between t and logic variables can be allowed semantically, other than those originating in duration terms in certain circumstances. To conclude the proof, we have that no infinitely many discontinuities exist, and the integral is always bounded in $[0, 1]$, thus $1_{\varphi(\kappa,\upsilon)}$ is a step function for φ and for all κ, υ, and t. Since, $1_{\varphi(\kappa,\upsilon)}$ is a step funtion, then any step function is Riemann integrable. □

Lemma 2. *Let $\phi^3_<$ be a formula in RMTL-$\int_3$ constructed with the relation operator. Moreover let $f_t : \Phi^3 \to \Phi$ be some surjective function. Then there exists a formula $\phi_<$ in RMTL-$\int$ such that $\phi_< = f_t(\phi^3_<)$.*

Proof of Lemma 2. We have to prove that there exists a function f that removes the minimum and maximum terms of the formulas $\phi^3_<$, and the result is a formula $\phi_< \in$ RMTL-$\int$. The proof follows by induction on the structure of the formulas $\phi_<$.

Case $\phi_< = \eta_1 \circ \underset{x \in I}{min}\ \phi \sim \eta_2$ with $\eta_1 = 0$ and $\circ = +$. By Axiom 1, there exists an x such that $0 + x \sim \eta_2$, $x \in I$, and ϕ implies that for all y, $y < x$ and $\neg\phi$. We skip the case when $\eta_1 \neq 0$, and also the case of $\phi_< = \eta_1 \circ \underset{x \in I}{max}\ \phi \sim \eta_2$, since the sketch is the same.

Case $\phi_< = \eta_1 \circ \underset{x \in I}{min}\ \phi \sim \eta_2 \circ \underset{x \in I}{min}\ \phi$. By the fourth axiom of the second axiomatization of Tarski [21, p. 205], we have that there exists a z such that $\eta_1 \circ \underset{x \in I}{min}\ \phi \sim z$ and $z \sim \eta_2 \circ \underset{x \in I}{min}\ \phi$. For the other cases where minimum terms

are commuted with maximum terms, the same reasoning is applied. Then, there exists a function f_t that converts a formula containing minimum and maximum terms into a formula without these terms, by adding at most three new quantifiers. □

Proof of Theorem 1. We proceed by induction over the structure of the formula ϕ^3. For Boolean, temporal and existential operators the proof follows from inductive hypotheses and from monotonicity of these operators with respect to the partial order $\prec$ on $\{\mathsf{tt}, \mathsf{ff}, \bot\}$ following Definition 4 (2). The remaining case is the relation operator. For formulas ϕ^3 containing $\alpha, x, \int^{\eta} \phi$, and $\circ$ terms, the claim follows from inductive hypotheses and by the monotonicity of these terms with respect to the partial order $\lhd$, following Definition 4 (1). If terms are of the form $\underset{x \in I}{min}\ \varphi$ or $\underset{x \in I}{max}\ \varphi$, then by Lemma 2 there exists an equivalent formula without minimum and maximum terms. Thus, the claim holds for all cases. □