Reasoning with Metric Temporal Logic and Resettable Skewed Clocks

Alberto Bombardelli[1,2] [0000−0003−3385−3205] and
Stefano Tonetta[1] [0000−0001−9091−7899]

[1] Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Povo TN, Italy
{abombardelli,tonettas}@fbk.eu
[2] University of Trento, Via Sommarive, 9, 38123 Povo TN, Italy
alberto.bombardell-1@unitn.it

Abstract. The formal verification of distributed real-time systems is challenging due to the interaction between the local computation and the communication between the different nodes of a network. The overall correctness of the system relies on local properties and timing exchange of data between components. This requires to take into account the drift of local clocks and their synchronization. The reference time of local properties may be given by skewed and resettable (thus non-monotonic) clocks.
In this paper, we consider automated reasoning over MTLSK, a variant of MTL over Resettable Skewed Clocks. We focus on metric operators with lower and upper parametric bounds. We provide an encoding into temporal logic modulo the theory of reals and we solve satisfiability with SMT-based model checking techniques. We implemented and evaluated the approach on typical properties of real-time systems.

1 Introduction

Distributed Real-Time Systems (DRTS) consist of different real-time components connected by a communication network. Each component, therefore, responds to input data or events within a specified period of time or generates output data or events periodically. The correctness of the overall DRTS depends not only on the logic of the input/output functions but also on the timing constraints on the data/events. The formal verification of temporal properties of DRTS is thus very challenging due to the need for reasoning about both communication and timing constraints. Since local clocks of components are usually not perfect and drift from each other, DRTS employ various consensus algorithms to synchronize them.

In formal verification, the properties are typically specified in temporal logics such as Linear-time Temporal Logic (LTL) [19], which can specify temporal constraints on the succession of events or exchange of messages. When dealing with real-time systems, one of the most popular temporal logics is Metric Temporal Logics (MTL) [16], which enriches the temporal operators with bounds to constrain the time intervals in which formulas must be satisfied. Another variant,

the Event Clock Temporal Logic (ECTL) [20] uses event clock constraints to specify bounds on the time since the last time or until the next time a formula holds. Timed Propositional Temporal Logic (TPTL) [1], instead, uses freezing quantifiers to compare and constrain time at different points.

When these logics are used to formalize properties of DRTS, they must be extended in two directions. First, the local properties should use local clocks as the reference time for the timing constraints. Second, the local clocks should be possibly skewed and resettable to consider the potential drift and synchronization of clocks. Therefore, distributed variants of MTL and ECTL use local temporal operators that refer to local times (e.g., [18]). Moreover, in [4], the logic MTLSK was introduced where specific variants of the metric temporal operators overcome the issue of standard operators when the reference time is not monotonic as for resettable clocks.

In this paper, we address the problem of automated reasoning over MTLSK properties. We focus on metric operators with either lower or upper parametric bounds. We consider local properties of DRTS components that use local clocks that are occasionally reset for synchronization and, so, that may be not monotonic. We use assumptions on drift and synchronization mechanisms to entail global properties. This compositional reasoning is formalized into MTLSK.

The main contribution of the paper is a procedure that reduces the satisfiability of the new logic to Satisfiability Modulo Theories (SMT) of discrete-time First-Order LTL [9, 23]. The encoding takes into account that a clock c may be not monotonic, which implies that interval constraints such as $c \leq p$ may be true in disjoint time intervals. Thus, differently from the monotonic case, it is not sufficient to guess the first point at which a subformula is satisfied. Guess variables are introduced to predict minimum/maximum values of the local clocks, for which subformulas hold. The correctness of the encoding exploits the assumptions that local clocks, although resettable, are diverging (i.e. do not converge to a finite value).

We implemented the approach on top of the timed features of nuXmv [8] using bounded model checking for finding traces and the k-zeno algorithm to prove the validity of formulas. We evaluated them on typical patterns of real-time properties real-time systems also showing scalability on tautologies derived from compositional reasoning.

The remainder of the paper is organized as follows: in Section 2, we describe a motivating example to explain the kind of automated reasoning addressed in the paper; in Section 3, we compare our approach with other related work; in Section 4, we provide preliminary knowledge about first-order LTL and temporal satisfiability modulo theory; in Section 5, we recall the definition of MTLSK; in Section 6 we show the reduction to satisfiability modulo theories; in Section 7 we provide the results of an experimental evaluation of our work; finally, in Section 8, we draw the conclusions and some future directions.

2 Motivating example

2.1 Semantics with resettable skewed clocks

In this section, we describe the type of compositional reasoning that we address in this paper. In the following, we use standard LTL operators such as G (always in the future) and F (eventually in the future), as well as their distributed metric variants. We informally recall the semantics and refer the reader to the next sections for a formal definition.

The formula $F^c_{\leq p} b$ is true in a time point t whenever b holds in some future point within p time units. The superscript c indicates that the time constraint must be evaluated using a clock c. Thus b must hold in a point t' such that $c(t') - c(t) \leq p$. The dual formula $G^c_{\leq p} b$ requires b to be true in all points t' such that $c(t') - c(t) \leq p$.

As an example, consider the plot in Figure 1. Let us first refer to the perfect clock c (black line). The formula $F^c_{\leq 5}(y_1 \geq 2)$ is true in 0 because there is a point between time 4 and 5 where $y_1 \geq 2$. For the same reason, the formula $G^c_{\leq 5}(y_1 \leq 2)$ (which is equivalent to the negation of the first one) is false. If we consider instead y_2, the formula $G^c_{\leq 5}(y_2 \leq 2)$ is true.

Let us now consider the skewed clock sc (orange line), which runs too fast and is reset at points 2 and 5 to approximately the correct value (black line). If we consider again $F^{sc}_{\leq 5}(y_1 \geq 2)$, the formula is now false because of the drift as $y_1 \geq 2$ should hold before. Note however that thanks to the reset the points where $sc(t) \leq 5$ lie in disconnected intervals, namely $[0, 4]$ and $[5, 16/3]$. Thus, while $G^c_{\leq 5}(y_1 \leq 2)$ is true, $G^c_{\leq 5}(y_2 \leq 2)$ is false.

To better characterize the intended semantics over non-monotonic clocks, we consider the variant $\overline{F}^c_{\leq p}$ and $\overline{G}^c_{\leq p}$ introduced in [4], which consider only the first interval where $c(t') - c(t) \leq p$. Thus, in our example, $\overline{G}^{sc}_{\leq 5}(y_1 \leq 2)$ is true, $G^c_{\leq 5}(y_2 \leq 2)$ is true.

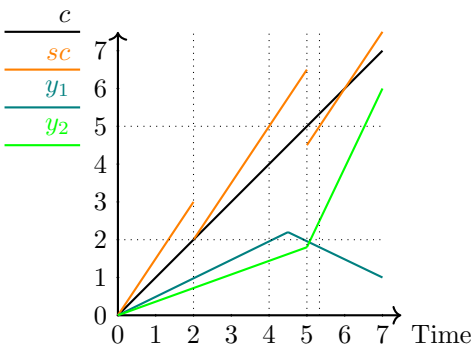


Fig. 1: Examples of real functions, including a perfect and a skewed resettable clock

Finally, note that we are dealing with a super-dense model of time, which augments the interpretation of time over real numbers with instantaneous discrete steps such as resets and discrete variable updates. Thus, in the example above, there are two-time points where the real-time is 2: in the first point, sc is equal to 3, while in the second point, due to a reset, sc is equal to 2 even if real-time does not change. We use the functional symbol $next$ to refer to next value after a discrete step. For example, the above-mentioned reset point satisfies $next(sc) = sc - 1$.

2.2 Compositional Reasoning Example

Consider a system composed of two components. The first component sends an alive signal (variable *alv*) to the second component unless there is a fault (variable *flt*). The second component monitors the alive signal and, if absent, raises an alarm (variable *alm*). Globally, we expect that if there is a fault, an alarm is triggered in due time. The components use clocks *cl1* and *cl2*, while the global clock is *cl*.

The compositional reasoning is formalized with the following formula:

$$(G(flt \rightarrow G^{cl1}_{\leq p}\neg alv) \wedge G(G^{cl2}_{\leq p}\neg alv \rightarrow (F^{cl2}_{\leq p} alarm))) \rightarrow G(flt \rightarrow F^{cl}_{\leq p} alm)$$

The formula is valid if the clocks are not skewed. If instead the clocks diverge, the formulas $G^{cl1}_{\leq p}\neg alv$ and $G^{cl2}_{\leq p}\neg alv$ are not equivalent anymore (similarly for $F^{cl2}_{\leq p} alm$ and $F^{cl}_{\leq p} alm$). In this case, we need to consider safety margins in the guarantees of the two components. Let us assume that the error on the clock derivatives is bounded by ϵ. For simplicity, let us also consider for the moment no reset. In this case, given=$\tilde{\epsilon} = 1 + 2\epsilon/(1-\epsilon)$, we can prove that the following formula is valid:

$$(G(flt \rightarrow G^{cl1}_{\leq p\tilde{\epsilon}}\neg alv) \wedge G(G^{cl2}_{\leq p}\neg alv \rightarrow (F^{cl2}_{\leq p} alm))) \rightarrow G(flt \rightarrow F^{cl}_{\leq p\tilde{\epsilon}} alm)$$

Let us now consider resets. Intuitively, assuming that the clocks are reset with period q and that q is much smaller than p, we can reduce the safety margins. In particular, the resulting formula is:

$$\begin{aligned}&(G((Reset(cl1) \rightarrow next(cl1) = cl) \wedge (Reset(cl2) \rightarrow next(cl2) = cl) \wedge (\neg Reset(cl)\\ &)) \wedge GF^{cl}_{\leq q}(next(cl) = cl1) \wedge GF^{cl}_{\leq q}(next(cl) = cl2) \wedge r \geq q(1 + 2\epsilon/(1-\epsilon)) \wedge\\ &G(flt \rightarrow \overline{G}^{cl1}_{\leq p}\neg alv) \wedge G(\overline{G}^{cl2}_{\leq p-4r}\neg alv \rightarrow (\overline{F}^{cl2}_{\leq p} alm))) \rightarrow G(flt \rightarrow \overline{F}^{cl}_{\leq p+2r} alm)\end{aligned}$$

First, note that we use the "first-interval" variants. In fact, we require that after a fault the alive signal of the first component is down for an interval without discontinuity; similarly, when the second component sends an alarm it cannot rely on a future reset and must send it within the first interval in which the local clock is below the given bound.

Second, let us consider the assumptions on the resets. The first row says that whenever the clocks $cl1$ and $cl2$ are reset, they are set to the value of cl, and cl is never reset. The second row requires that at most every q time units the clocks are reset.

Summarizing, the logic addressed by the paper considers metric operators with parametric upper (or lower) bounds, super-dense semantics, resettable skewed clocks, and first-order constraints on the clocks and parameters.

3 Related work

Various works customized the modal operators of temporal logics to better suit the specification of DRTS. TPTL was extended in [24] by using explicitly multi-

ple local clocks and supporting inequalities to express constraints on the precedence between local clock readings. In [18], a distributed variant of ECTL is proposed. Similarly, [17] defines a distributed modal logic where the time varies independently in each component of the system, represented by a network of timed automata. In all these works, local times are assumed strictly increasing, thus, not addressing the semantic issues of the temporal operators when the time is not monotonic.

The problem of modelling DRTS with drifting and synchronized clocks was considered in [21], where specific patterns of timed automata were proposed and verified. Similarly, [5] considers the problem of parametrized verification of 8N1 and Biphase Mark protocols with skewed clocks using the SAL model checker [14]. These works focus on the modelling of clock drifts and synchronizations, but do not consider the specification of timed properties that refer to skewed synchronized clocks.

The problem of validating the correctness of drifted clocks synchronization algorithms have been studied in other works using theorem provers, e.g., [2, 22].

The work closer to the problem addressed in this paper is focused on the satisfiability of MTL and TPTL over non-monotonic time, which has been studied in [7] in the more general context of data words. Timed words are considered a special case, although [7] considers only discrete time. A decision procedure is given for the fragment without negation and only temporal operators X and F. Instead, we address an undecidable fragment of MTLSK with SMT-based model checking.

Last, we mention [15], which focuses on runtime verification of MTL formulas in a distributed system. Here, the authors address the problem of monitoring global properties on all traces that are compatible with a given sequence of local observations with timestamps taking into account the possible drift of local clocks. Thus, the metric operators are not, as in our case, used in local properties and related to local clocks.

4 First-order LTL over discrete or super-dense time

4.1 Discrete and super-dense time

A time model is a structure $\tau = \langle T, <, \mathbf{0}, v\rangle$ with a temporal domain T, a total order $<$ over T, a minimum element $\mathbf{0} \in T$, and a function $v : T \to \mathbb{R}_0^+$ that represents the real-time of a time point in T. A *time point* is an element of T. A time interval sequence is a sequence $I_0, I_1, I_2, \ldots$ of intervals of reals such that, for all $i \geq 0$, I_i and I_{i+1} are almost adjacent (subsequent intervals can overlap in at most one point) and $\bigcup_{i\geq 0} I_i = \mathbb{R}_0^+$.

- In discrete time models, $T = \mathbb{N}$, $\mathbf{0}$ and $<$ are the standard zero and order over natural numbers, $v(0) = 0$ and $v(0), v(1), v(2), \ldots$ is a non-decreasing divergent sequence.
- In super-dense time models, 1) $T \subset \mathbb{N} \times \mathbb{R}_0^+$ such that the sequence of sets $I_0, I_1, I_2, \ldots$ where, for all $i \geq 0$, the set $I_i := \{r \mid \langle i, r\rangle \in T\}$, is a time

interval sequence, 2) $\langle i, r\rangle < \langle i', r'\rangle$ iff $i < i'$ or $i = i'$ and $r < r'$, 3) $\mathbf{0} = \langle 0, 0\rangle \in \mathbb{N} \times \mathbb{R}_0^+$, and 4) $v(\langle i, r\rangle) = r$.

Intuitively, super-dense time models consist of the union of discrete sets of points in the form $\langle i, r\rangle, \langle i+1, r\rangle, \ldots$ (with the same timestamp r) and dense sets (containing an uncountable number of time points with different timestamps) of the form $\langle i, r\rangle, \langle i, r'\rangle, \langle i, r''\rangle, \ldots$ (with the same counter i). If $\langle i, r\rangle, \langle i+1, r\rangle \in T$, we say that there is a discrete step in $\langle i, r\rangle$ and we define a partial function *succ* as $succ(\langle i, r\rangle) = \langle i + 1, r\rangle$ if there is a discrete step in $\langle i, r\rangle$.

4.2 Linear Temporal Logic

We consider First-Order Linear-time Temporal Logic that we denote as LTL for short. LTL formulas are interpreted over discrete and super-dense time models. Following [9], the "tomorrow"[3] operator X is generalized to the super-dense time case and we include its "dense counterpart" $\tilde{X}$. Intuitively, $X\phi$ holds in t whenever there is a discrete step in t and ϕ holds in $succ(t)$, while $\tilde{X}\phi$ holds in t whenever there is no discrete step in t and ϕ holds in a right neighborhood of t (formally defined below). It should be noted that $X\varphi$ is equivalent to $\bot\tilde{U}\varphi$ where $\tilde{U}$ is until with the "strict" semantics (in which $t' \geq t$ is replaced by $t' > t$ in the semantics). $\bot\tilde{U}\varphi$ disallow t' to have a point t'' such that $t < t'' < t'$; thus, t' must be $succ(t)$. Moreover, $\tilde{X}\varphi$ is equivalent to $\neg X\top \wedge \varphi\tilde{U}\top$: there is no discrete step and all points t'' preceding t' satisfy φ.

First-order formulae are composed of Boolean logic connectives, a given set of variables V and a first-order signature Σ.

Definition 1. *Given a signature Σ and a set of variables V, LTL formulas φ are built with the following grammar:*

$$\phi := pred(u, \ldots, u) \mid \phi \wedge \phi \mid \neg\phi \mid \phi U \phi \mid X\phi \mid \tilde{X}\phi$$
$$u := c \mid x \mid next(x) \mid f(u, \ldots, u) \mid ite(\varphi, u, u)$$

where c, f, and $pred$ are respectively a constant, a function, and a predicate of the signature Σ and x is a variable in V.

Note that atomic formulas can constrain the "current" and "next" value of variables. We call a predicate with no occurrence of *next* a state predicate, otherwise an event predicate. Event predicates are evaluated between two states during discrete transitions, next symbols are evaluated in the current state, while current symbols are evaluated in the current state.

We use standard abbreviations:
$\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \mathcal{R} \varphi_2 := \neg(\neg\varphi_1 U \neg\varphi_2)$, $F\varphi := (\top U \varphi)$, $G\varphi := \neg F \neg\varphi$.

A state $s = \langle M, \mu\rangle$ is given by a first-order structure M and an assignment μ of variables of V into the domain of M. Given a state $s = \langle M, \mu\rangle$ and a symbol c in Σ or variable $x \in V$, we use $s(c)$ to denote the interpretation of c in M and $s(x)$ to denote the value $\mu(x)$ assigned by μ to x. Given M, let V^M be the set of

[3] Note:"next" is used to refer to *next* instead of X

states with first-order structure M. A trace $\pi = \langle M, \tau, \overline{\mu} \rangle$ is given by a first-order structure M, a time model τ, and a mapping $\overline{\mu}$ from the domain of τ to V^M. Given a trace $\pi = \langle M, \tau, \overline{\mu} \rangle$ and $t \in \tau$, we denote by $\pi(t)$ the state $\langle M, \overline{\mu}(t) \rangle$. Note that the first-order structure M is shared by all time points, meaning that the interpretation of the symbols in the signature Σ is rigid (it does not vary with time). Terms that do not contain variables are called *parameters* and the interpretation does not depend on the time point.

We assume a Σ first-order theory $\mathcal{T}$ to be given. Given a Σ first-order structure M, an assignment μ to variables of V, and a Σ first-order formula ϕ over V, we use the standard notion of $\langle M, \mu \rangle \models_{\mathcal{T}} \phi$. In the rest of the paper, we omit the first-order signature Σ and theory $\mathcal{T}$ for simplicity.

Given a trace $\pi = \langle M, \tau, \overline{\mu} \rangle$, a time point t of τ, and a Σ formula ϕ, we define $\pi, t \models \phi$ as follows:

- if α is a state predicate, $\pi, t \models \alpha$ iff $\pi(t) \models \alpha$
- if α is an event predicate, $\pi, t \models \alpha$ iff there is a discrete step in t and $\pi(t) \cdot \pi(succ(t)) \models \alpha$
- $\pi, t \models \phi_1 \wedge \phi_2$ iff $\pi, t \models \phi_1$ and $\pi, t \models \phi_2$
- $\pi, t \models \neg\phi$ iff $\pi, t \not\models \phi$
- $\pi, t \models \phi_1 U \phi_2$ iff there exists $t' \geq t$ such that $\pi, t' \models \phi_2$ and for all $t'', t \leq t'' < t', \pi, t'' \models \phi_1$
- $\pi, t \models X\phi$ iff there is discrete step in t and $\pi, succ(t) \models \phi$
- $\pi, t \models \tilde{X}\phi$ iff there is not a discrete step in t and there exists $t' > t$, for all t'', $t < t'' < t'$, $\pi, t'' \models \phi$

Finally, $\pi \models \phi$ iff $\pi, \mathbf{0} \models \phi$. We say that ϕ is satisfiable iff there exists π such that $\pi \models \phi$. We say that ϕ is valid iff, for all π, $\pi \models \phi$.

5 MTL with skewed clocks and resets

In this section, we define a variant of the logic MTLSK introduced in [4], where time constraints of metric operators use parametrized lower and upper bounds, as described in Section 2. Moreover, we consider only skewed clocks and super-dense time, and we consider other standard logics as fragments (dense time and standard metric operators over perfect clocks).

We first formally define some assumptions on the clocks to ensure that, despite the resets, they are diverging. We assume to be given two constants ϵ and λ that are used as bounds for the drift and resets, respectively.

Definition 2. *A "resettable skewed clock" (henceforward, simply, "clock") is a real variable $c \in V$ such that, for every trace $\pi = \langle M, \tau, \overline{\mu} \rangle$ and $t = \langle i, r \rangle \in \tau$,*

1. *$\pi(t)(c)$ is differentiable in I_i with $\frac{d\pi(t)(c)}{dt} \in [1 - \epsilon, 1 + \epsilon]$.*
2. *If t is a discrete step $|\pi(succ(t))(c) - \nu(t)| \leq \lambda$.*

In particular, the clocks are piecewise continuous and strictly monotonic, with at most countably many resets where the clock can decrease.

Definition 3. *Given a signature Σ, a set of variables V, and a set of clock variables $C \subseteq V$, MTLSK formulas are built with the following grammar:*

$$\phi := pred(u, \ldots, u) \mid \phi \wedge \phi \mid \neg\phi \mid X\phi \mid \tilde{X}\phi \mid \phi U^c_{\lhd u}\phi \mid \phi U^c_{\rhd u}\phi \mid \phi \overline{U}^c_{\lhd u}\phi$$

where $c \in C$, $\lhd \in \{<, \leq\}$,$\rhd \in \{>, \geq\}$, and u is defined as in LTL but clock variables can occur only in event predicates.

Abbreviations are defined as in the standard case.

Note that metric operators use a clock variable in C as its reference for the timing constraints instead of the real-time, as done in [18] for ECTL.

We just define the semantics of metric operators, while the other cases are defined as for LTL. We assume here that the background first-order theory contains the theory of reals and that the clock variables and each parametric bound has real type.

- $\pi, t \models \varphi U^c_{\lhd p}\psi$ iff there exists $t' \geq t$ such that $\pi(t')(c) - \pi(t)(c) \lhd \pi(p)$ and $\pi, t' \models \psi$ and for all $t'', t \leq t'' < t', \pi, t'' \models \varphi$
- $\pi, t \models \varphi U^c_{\rhd p}\psi$ iff there exists $t' \geq t$ such that $\pi(t')(c) - \pi(t)(c) \rhd \pi(p)$ and $\pi, t' \models \psi$ and for all $t'', t \leq t'' < t', \pi, t'' \models \varphi$
- $\pi, t \models \varphi \overline{U}^c_{\lhd p}\psi$ iff there exists $t' \geq t$ such that $\pi(t')(c) - \pi(t)(c) \lhd \pi(p)$, $\pi, t' \models \psi$, and for all t'', $t \leq t'' < t'$, $\pi, t'' \models \varphi$ and $\pi(t'')(c) - \pi(t)(c) \lhd \pi(p)$

We recall the theorem proved in [4], which holds also for the parametric variant proposed here.

Theorem 1. *For all trace π, formulas φ, ψ: $\pi \models \varphi \overline{U}^c_{\lhd p}\psi \Rightarrow \pi \models \varphi U^c_{\lhd p}\psi$. If there is no reset (weakly monotonic case),*

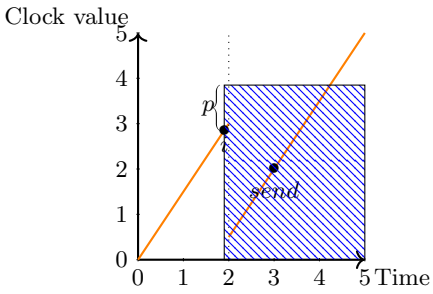
$$\pi \models \varphi \overline{U}^c_{\lhd p}\psi \Leftrightarrow \pi \models \varphi U^c_{\lhd p}\psi$$

Also, if there is no drift and no reset, i.e., $\epsilon = 0 \wedge \lambda = 0$ (perfect clocks), then

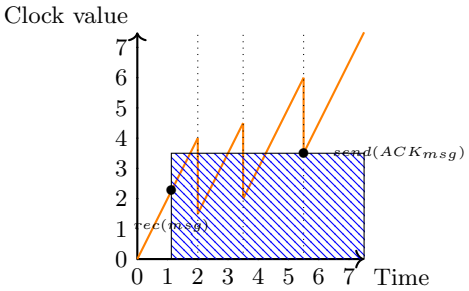
$$\pi \models \varphi \overline{U}^c_{\lhd p}\psi \Leftrightarrow \pi \models \varphi U_{\lhd p}\psi \Leftrightarrow \pi \models \varphi U^c_{\lhd p}\psi$$

Remark. As we discussed in Section 2, the non-monotonicity of the resettable clocks create some challenges as the time constraints defined by the metric operators are not simple intervals of $\mathbb{R}^+_0$. In fact, if we consider the formula $F^c_{\leq p}send$, Fig. 2a shows an execution in which *send* occurs in the near future but the clock difference is negative.

Most important, as shown in section 2, the time intervals may be disconnected. Thus, it is not sufficient to look at the first occurrence of the subformulas, but we have to consider the minimum value of clocks. Consider for example the liveness property $\phi_{BR} := G(receive(msg) \rightarrow F^c_{\leq 1}send(ACK_{msg}))$ representing the bounded response of a component when it receives a message. Fig. 2b shows an execution where the component receives the message at instant $t = 1$ while the local clock exits the time interval bounded by 1 but is later reset to a value lower than $c(t) + 1$. The component then sends the acknowledgement for the message *msg* within the time constraint and the property holds.



(a) This figure shows a run that satisfies $F_{\leq p} send$ with a negative difference of the clock value



(b) Example trace satisfying ϕ_{BR} with an ack after a reset

Fig. 2

The key observation here is that it is sufficient to track the minimum value of c after t where the acknowledgement is sent and check if that is within the bound. With the operator $\overline{F}^{c}_{\leq p}$ instead we can express the property that the ack is sent within the *first* interval before $t + p$. In this case, we can therefore track the first point at which the acknowledgement is sent making sure that the time constraint was not violated before.

6 Encoding and verification

In this section, we define an encoding from MTLSK to first-order LTL. The encoding is divided into 3 parts: the first part considers a novel intermediate logic LTL-min-max that can express MTLSK properties through a straightforward rewriting Υ; the second part defines a discretization process of LTL-min-max based on the discretization proposed in [23]; the last part defines an encoding from LTL-min-max to First-order LTL.

6.1 LTL-min-max definition and rewriting

We construct a new intermediate logic LTL-min-max extending First-order LTL with minimum and maximum operators. The intuition behind this logic follows from the remark of Section 5: while the satisfaction of bounded operators in $MTL_{0,+\infty}$ can be achieved by looking at the value of time in the next occurrence of the formula, in MTLSK due to non-monotonicity of time we have to analyze the minimum and maximum time in which the property holds.

Definition 4. *Given a signature Σ, a set of variables V, and a set of clock variables $C \subseteq V$, LTL-min-max formulas are built with the following grammar:*

$$\phi := p(u, \ldots, u) \mid tu \bowtie cu \mid \phi \wedge \phi \mid \neg\phi \mid \phi U \phi \mid X\phi \mid \tilde{X}\phi$$

$$u := cu \mid x \mid f(u,\dots,u) \mid ite(\phi,u,u) \mid next(u)$$
$$tu := time \mid c \mid min\Delta^c_{\phi U\phi} \mid f(tu,\dots,tu) \mid next(c) \mid max\Delta^c_{\phi U\phi} \mid maxbef\,\Delta^c_\phi$$
$$cu := p \mid f(cu,\dots,cu)$$

where $\bowtie \in \{\leq,\geq,<,>\}$ *and clock variables can occur only in event predicates and in min-max operator superscripts i.e.* $min\Delta^c_{\phi U\psi}$, $max\Delta^c_{\varphi U\psi}$ *and* $maxbef\,\Delta^c_\varphi$.

Definition 5. *The semantics of LTL-min-max new operators is defined as follows:*

If $\pi,t \models \varphi U\psi$ *then* $\pi(t)(min\Delta^c_{\varphi U\psi}) = \min(\pi(t)(U^c_{\varphi U\psi})) - \pi(t)(c)$

If $\pi,t \models (\varphi U\psi) \wedge \Phi_{finU}$ *then* $\pi(t)(max\Delta^c_{\varphi U\psi}) = \max(\pi(t)(U^c_{\varphi U\psi})) - \pi(t)(c)$

If $\pi,t \models F\varphi$ *then* $\pi(t)(maxbef\,\Delta^c_\varphi) = \max(Bef^c_\pi(t,\varphi)) - \pi(t)(c)$

where: $\Phi_{finU} := F(\neg\varphi \vee G\neg\psi)$,

$\pi(t)(U^c_{\varphi U\psi}) := \{\pi(t')(c) | t' \geq t : \pi,t' \models \psi$ *and for all* $t \leq t'' < t' : \pi,t'' \models \varphi\}$,

$\pi(t)(Bef^c_\varphi) := \{\pi(t')(c) | t' \geq t :$ *for all* $t < t'' < t' : \pi,t'' \not\models \varphi\}$.

$\pi(t)(U^c_{\varphi U\psi})$ represents the value of each $\pi(t')(c)$ such that ψ holds at point t' and each point between t and t' (excluded) satisfies φ. This set takes all the witnesses of π satisfying $\varphi U\psi$ from point t; then, it extracts the value of c at point t' i.e. when ψ holds. $min\Delta^c_{\varphi U\psi}$ and $max\Delta^c_{\varphi U\psi}$ represent respectively the minimum and the maximum of that set. The existence of the minimum of $\pi(t)(U^c_{\varphi U\psi})$ is guaranteed if $\varphi U\psi$ holds. $\pi,t \models \varphi U\psi$ is a sufficient condition because clocks diverge (see Definition 2) and ψ contains clocks only inside event predicates. Similarly, the assumptions on $max\Delta$ guarantee that the maximum exists if the property holds and it holds only finitely many times. If Φ_{finU} does not hold, then the maximum does not exist because the clock diverges.

$\pi(t)(Bef^c_\psi)$ is the set containing all clock values from point t to the first point such that ψ holds after t. $maxbef\,\Delta^c_\psi$ represents the maximum in that set. $F\varphi$ guarantees the existence of the maximum of that set.

Definition 6. *We define the rewriting* Υ *from MTLSK to LTL-min-max as follows:*

$$\Upsilon(\varphi U^c_{\lhd p}\psi) := \Upsilon(\varphi U\psi) \wedge min\Delta_{\Upsilon(\varphi U\psi)} \lhd p$$
$$\Upsilon(\varphi U^c_{\rhd p}\psi) := \Upsilon(G(\varphi \wedge F\psi)) \vee \Upsilon(\varphi U\psi) \wedge max\Delta_{\Upsilon(\varphi U\psi)} \rhd p$$
$$\Upsilon(\varphi \overline{U}^c_{\lhd p}\psi) := \Upsilon(\varphi U\psi) \wedge maxbef\,\Delta^c_{\Upsilon(\psi)} \lhd p$$

Theorem 2. φ *and* $\Upsilon(\varphi)$ *are equisatisfiable*

Proof (Sketch). $\varphi U^c_{\bowtie p}\psi$ is true at point t only if there exist a point t' such that ψ hold, $\pi(t')(c) - \pi(t)(c) \bowtie \pi(p)$ and all points from t to t' satisfy φ. Since $min\Delta^c_{\varphi U\psi}$ and $max\Delta^c_{\varphi U\psi}$ are respectively the minimum/maximum of the set containing the evaluation of c in each t' point minus $\pi(t)(c)$, then the translation holds. The same applies to $\varphi\overline{U}^c_{\lhd p}\psi$. In that case, the property holds iff $\varphi U\psi$ holds and each value of $\pi(t'')(c)$ with $t \leq t'' \leq t'$ is not greater than the upper bound $\pi(t)(c) + \pi(p)$. Thus, it is sufficient to verify that the $max(\pi(t)(Bef)) - \pi(t)(c) \lhd \pi(p)$.

6.2 LTL-min-max discretization

We apply our discretization process based on the one defined in [23]. The discretization process defines time evolution as a sequence of open or singular intervals. The discretization process assumes that the satisfiability of each subformula of ϕ does not vary inside intervals. That assumption is ensured because MTLSK allows clocks only as part of event predicates.

Singular intervals are encoded by the fresh variable ι while the time elapsed in each interval is encoded by δ. The real variable ζ encodes an arbitrary accumulation of sums of δ. ζ is used to guarantee the divergence of the global time. For each skewed clock c, we introduce two fresh variables: δ_c and diff_c. δ_c encodes the clock time elapse in each transition while diff_c represents the difference between the global time and the current clock value c. We encode c using diff_c to guarantee the divergence of c relying on the divergence of global time.

The rewritten formula ϕ_D is composed of four parts: the constraints defining time and forcing discrete variables to stutter when time elapses (ψ_{time}), the constraints defining each clock c according to Definition 2 (ψ^c_{clock}), the constraint to define when an interval is open or when it is singular (ψ_ι) and the discretized formula ($\mathcal{D}(\phi)$). The whole transformation is as follows:

$$\phi_D := \psi_{time} \wedge \bigwedge_{c \in C} \psi^c_{clock} \wedge \psi_\iota \wedge \mathcal{D}(\phi)$$

$$\psi_{time} := time = 0 \wedge G(next(time) - time = \delta) \wedge G(\delta > 0 \rightarrow \bigwedge_{v \in V} (next(v) = v))$$

$$\psi^c_{clock} := \mathit{diff}_c = 0 \wedge G(next(\mathit{diff}_c) - \mathit{diff}_c = \delta_c - \delta) \wedge G((\delta > 0 \rightarrow \delta_c \in [\delta(1-\epsilon), \delta(1+\epsilon)]) \wedge (\delta = 0 \rightarrow |\mathit{diff}_c| \leq \lambda))$$

$$\psi_\iota := \iota \wedge G((\iota \wedge \delta = 0 \wedge X\iota) \vee (\iota \wedge \delta > 0 \wedge X\neg\iota) \vee (\neg\iota \wedge \delta > 0 \wedge X\iota)) \wedge G((next(\zeta) - \zeta = \delta) \vee (\zeta \geq 1 \wedge \zeta = 0)) \wedge GF(\zeta \geq 1 \wedge next(\zeta) = 0)$$

Definition 7. *The discretization rewriting $\mathcal{D}$ is defined as follows:*

$$\mathcal{D}(X\varphi) := \iota \wedge X(\iota \wedge \mathcal{D}(\varphi)) \quad \mathcal{D}(\tilde{X}\varphi) := (\neg\iota \wedge \mathcal{D}(\phi)) \vee X(\neg\iota \wedge \mathcal{D}(\varphi))$$

$$\mathcal{D}(\varphi U \psi) := \mathcal{D}(\psi) \vee (\mathcal{D}(\varphi) U \tilde{\psi}) \quad \mathcal{D}(\mathit{maxbef}\, \Delta^c_\varphi) := \mathit{maxbef}\, \Delta^c_{\mathcal{D}(\varphi)}$$

$$\mathcal{D}(\mathit{min}\Delta^c_{\varphi U \psi}) := \mathit{ite}(\mathcal{D}(\psi) \wedge 0 \leq \mathit{min}\Delta^c_{\mathcal{D}(\varphi) U \tilde{\psi}}, 0, \mathit{min}\Delta^c_{\mathcal{D}(\varphi) U \tilde{\psi}})$$

$$\mathcal{D}(\mathit{max}\Delta^c_{\varphi U \psi}) := \mathit{ite}(\mathcal{D}(\psi) \wedge 0 \geq \mathit{max}\Delta^c_{\mathcal{D}(\varphi) U \tilde{\psi}}, 0, \mathit{max}\Delta^c_{\mathcal{D}(\varphi) U \tilde{\psi}})$$

where $\tilde{\psi} = \mathcal{D}(\psi) \wedge (\iota \vee \mathcal{D}(\varphi))$.

$X\varphi$ is discretized forcing a discrete transition i.e. $\iota \wedge X\iota$. $\tilde{X}\varphi$ requires that either φ holds now in an open interval or it holds in the next state in an open interval. Until forces either ψ to hold now or φ has to hold until ψ holds; moreover, if ψ holds in an open interval also φ must hold in that point too. The discretization of $\mathit{min}\Delta^c_{\varphi U \psi}$ and $\mathit{max}\Delta^c_{\varphi U \psi}$ is similar to the one of until. It considers the current point in the minimum/maximum computation as a candidate min/max and then applies the discrete operator on the discretized until.

Theorem 3. *ϕ and ϕ_D are equisatisfiable*

6.3 LTL-min-max discrete encoding

The discrete setting enables recursive definitions of minimum and maximum. Consider for instance the set $S^v_\top$ which stores the values of variable v at each time point. $\pi(t)(S^v_\top)$ can be defined by the following recursive definition: $\pi(t)(S^v_\top) = \{\pi(t)(v)\} \cup \pi(t+1)(S^v_\top)$. Our discrete encoding of LTL-min-max exploits this inductive structure of $U^c_{\varphi U \psi}$ and Bef^c_φ to provide a sound translation to first-order LTL. Our encoding introduces new monitor variables representing $min\Delta^c_{\varphi U \psi}$, $max\Delta^c_{\varphi U \psi}$ and $maxbef\,\Delta^c_\varphi$. Each operator is replaced with the monitor in the formula and the formula is implied by the monitor constraints.

In the remainder of this section, we denote $\rho' := next(\rho)$, $\delta_c := c' - c$ and $\tilde{U}$ as the "strict" version of U operator (also expressible as $\varphi \tilde{U} \psi := \varphi \wedge X(\varphi U \psi)$ in the discrete setting).

$$\begin{aligned}\mathcal{R}epl(\Psi, min\Delta^c_{\varphi U \psi}) := G(\varphi U \psi \rightarrow \rho_{min\Delta^c_{\varphi U \psi}} = \\ ite(\psi \wedge (\neg(\varphi \tilde{U} \psi) \vee 0 \leq \rho'_{min\Delta^c_{\varphi U \psi}} + \delta_c), 0, \rho'_{min\Delta^c_{\varphi U \psi}} + \delta_c) \wedge \\ (F(\psi \wedge \rho_{min\Delta^c_{\varphi U \psi}} = 0))) \rightarrow \Psi\lceil min\Delta^c_{\varphi U \psi} / \rho_{min\Delta^c_{\varphi U \psi}} \rfloor\end{aligned}$$

The value of $\rho_{min\Delta^c_{\varphi U \psi}}$ is evaluated as the minimum only when $\varphi U \psi$ holds; otherwise, $min(U^c_{\varphi U \psi})$ would be undefined. The *ite* expression evaluates the minimum between 0 and $\rho'_{min\Delta^c_{\varphi U \psi}} + \delta_c$ ($min(\pi(succ(t))(U^c_{\varphi U \psi}))$. Finally, $F(\psi \wedge \rho_{min\Delta^c_{\varphi U \psi}} = 0)$ guarantees that a minimum exists.

Theorem 4. *Ψ and $\mathcal{R}epl(\Psi, min\Delta^c_{\varphi U \psi})$ are equisatisfiable*

$$\begin{aligned}\mathcal{R}epl(\Psi, max\Delta^c_{\varphi U \psi}) := G((\varphi U \psi) \wedge \Phi_{finU} \rightarrow \rho_{max\Delta^c_{\varphi U \psi}} = \\ ite(\psi \wedge (\neg(\varphi \tilde{U} \psi) \vee 0 \geq \rho'_{max\Delta^c_{\varphi U \psi}} + \delta_c), 0, \rho'_{max\Delta^c_{\varphi U \psi}} + \delta_c) \wedge \\ (F(\psi \wedge \rho_{max\Delta^c_{\varphi U \psi}} = 0))) \rightarrow \Psi\lceil max\Delta^c_{\varphi U \psi} / \rho_{max\Delta^c_{\varphi U \psi}} \rfloor\end{aligned}$$

The encoding of $max\Delta$ is the same as the one of $min\Delta$ only flipping the sign and introducing the constraint Φ_{finU} from definition 5 to evaluate the monitor only if the maximum exists i.e. the formula holds finitely often.

Theorem 5. *Ψ and $\mathcal{R}epl(\Psi, max\Delta^c_{\varphi U \psi})$ are equisatisfiable*

$$\begin{aligned}&\mathcal{R}epl(\Psi, maxbef\,\Delta^c_\varphi) := G(F\varphi \rightarrow \rho_{maxbef\,\Delta^c_\varphi} = \\ &ite(\varphi \vee 0 \geq \rho'_{maxbef\,\Delta^c_\varphi}, 0, \rho'_{maxbef\,\Delta^c_\varphi} + \delta_c)) \rightarrow \Psi\lceil maxbef\,\Delta^c_\varphi / \rho_{maxbef\,\Delta^c_\varphi} \rfloor\end{aligned}$$

The encoding of $maxbef\,\Delta$ evaluates the maximum of c before and including the first point in which φ holds.

Theorem 6. *Ψ and $\mathcal{R}epl(\Psi, maxbef\,\Delta^c_\psi)$ are equisatisfiable*

7 Results

In this section, we present the implementation and experimental analysis of the procedure we described in the previous section to prove the validity and satisfiability of MTLSK formulas.

Formula	Time in sec.	λ	ϵ	alg	valid
$G(\overline{F}^c_{\leq p} a \rightarrow F^c_{\leq p} a)$	2.81	any	any	ic3	True
$F(c = p) \rightarrow F(((\overline{G}^c_{\leq p} a) \wedge (\overline{G}^c_{>p} \neg a)) \rightarrow \bot)$	3.62	any	0.4	kzeno	True
$(q \geq p) \rightarrow G((\overline{G}^c_{\leq q} a) \rightarrow (\overline{G}^c_{\leq p} a))$	0.38	any	any	ic3	True
$G^c_{\leq p} a \rightarrow G(a \vee c > p)$	9.03	any	any	kzeno	True
$G^c_{\leq p} a \rightarrow G(a \vee c > p)$	1.09	any	any	ic3	True
$(q \geq p) \rightarrow G((G^c_{>p} a) \rightarrow (G^c_{\geq q} a))$	2.22	any	any	ic3	True
$\Phi_{exp} := q = p(2 + \epsilon) + 2\lambda \wedge (G(fault \rightarrow G\neg alive) \wedge G(\overline{G}^{cl}_{\leq p} \neg alive \rightarrow (\overline{F}^{cl}_{\leq p} alarm))) \rightarrow G(fault \rightarrow F_{[0,q]} alarm)$	94.26	14.0	0.1	ic3	True
$(G((Reset(cl1) \rightarrow next(cl1) = cl) \wedge (\neg Reset(cl))) \wedge GF^{cl}_{\leq q}(next(cl) = cl1)) \rightarrow G(cl - cl1 \leq q * (1 + 2\epsilon/(1 - \epsilon)))$	7.05	any	any	kzeno	True
$G(f \rightarrow \overline{G}^{cl1}_{\leq p} \neg alv) \wedge G(\overline{G}^{cl2}_{\leq p-4r} \neg alv \rightarrow (\overline{F}^{cl2}_{\leq p} alm))) \rightarrow G(f \rightarrow \overline{F}^{cl}_{\leq p+2r} alm)$	19.86	any	any	ic3	True
$G(F^c_{\leq p} a \rightarrow \overline{F}^c_{\leq p} a)$	0.27	any	any	bmc	False
$G((a \vee Xa) \rightarrow (F^c_{\leq 0} a \wedge F^c_{>0} a))$	0.18	any	any	bmc	False
Bounded Response invalid with 11 clocks	1.36	any	any	bmc	False

Table 1: Some MTLSK properties and their verification results.

7.1 Implementation

We implemented MTLSK verification as an extension of timed nuXmv [8]. We used the following model-checking algorithms to verify the validity and satisfiability of the formulas: kzeno [11] in lockstep with bmc, ic3-ia [10], bmc [8] (with diverging variables). The algorithms used inside nuXmv are constructed on top of MathSAT5 [12] SMT-solver, which supports combinations of theories such as $\mathcal{LIA}$, $\mathcal{LRA}$ and $\mathcal{EUF}$. Unlike the logic definition of Section 5, our implementation permits the usage of formulas containing clocks in state predicates (e.g., $G(c - c1 \leq p)$ is allowed). Indeed, we introduced continuity constraints of [23] inside the discretization process to relax this syntactic restriction. Our implementation considers λ and ϵ constants from Section 5 to instantiate bounds for resettable clocks. These parameters are defined inside the SMV model either as scalar values (DEFINE) or as a parameter (FROZENVAR).

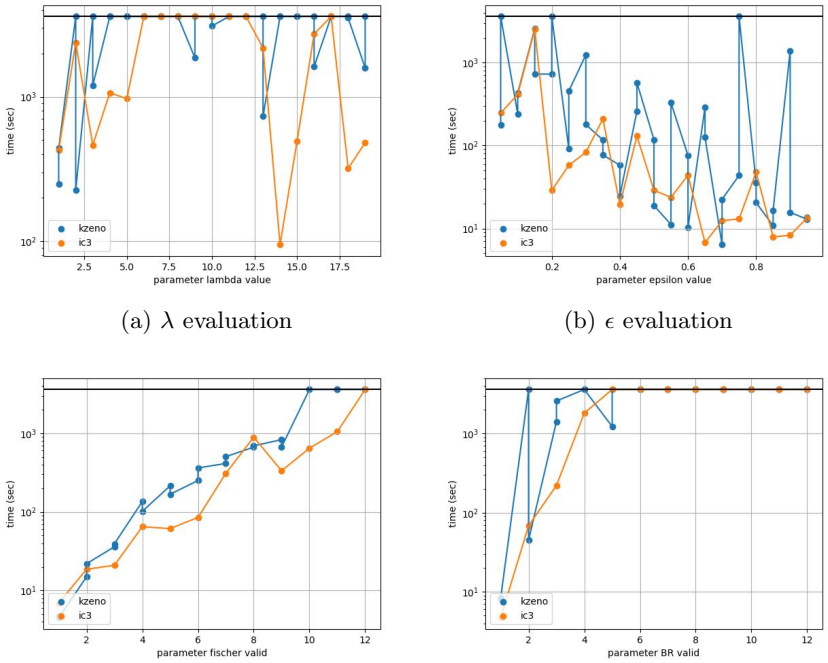
7.2 Experimental evaluation

We are not aware of other tools supporting MTLSK or other variations of MTL over resettable skewed clock; therefore, our experimental evaluation is performed only on our implementation. The experiments[4] were run in parallel on a cluster with nodes with Intel Xeon CPU running at 2.40GHz with 12CPU, 64GB. The timeout for each run was one hour and the memory cap was set to 1GB. The experimental evaluation considers the scalability of the tool concerning λ, ϵ, formulas size and until bound.

Benchmarks. Our experiment is divided into the following groups:

The first group of formulas is composed of a chain of *bounded response* (BR) where each local component i sends to its successor message a_i in at most p

[4] The results of the experimental evaluation can be found at https://es-static.fbk.eu/people/bombardelli/papers/nfm23/nfm23.tar.gz



(a) λ evaluation

(b) ϵ evaluation

(c) Fischer experimental evaluation

(d) BR experimental evaluation

Fig. 3: Parametric experimental evaluations

time unit where p is a parameter and time is interpreted as the local clock c_i. Finally, given a parametrized maximum bound r between local clocks and the global clock, the bounded response chain guarantees that $q = 2n(p + r)$ is the right bound for the global bounded response.
$G(\bigwedge_{0 \leq i < n} ((a_i \rightarrow \overline{F}^{c_i}_{\leq p} a_{i+1}) \wedge (c - c_i \leq r \wedge c_i - c \leq r))) \rightarrow G(a_0 \rightarrow \overline{F}^{c}_{\leq q} a_n)$

The second group of formulas is a Fischer algorithm benchmark taken from [8] MTL experimental evaluation and adapted for MTLSK. The pool of formulas is parametrized over the number of components. Each component has its skewed clock and a local formula representing its behaviour, the conjunction of all the formulas is used to imply a property of the first component.

The third group of formulas are a variation of the example proposed in Section 2 that instantiates the parameters p, λ and ϵ to study their impact on the validity checking performance. We also considered a timed and simplified version of a Wheel Brake System model of [13]. Given a redundant signal of the brake pedal, a redundant braking component, the property states that the hydraulic system should brake before a specified time threshold if the redundant components do not fail at the same time. This property has been translated to MTLSK and each component has been augmented with a local clock.

The last group is a collection of roughly 100 MTLSK specifications, 60 valid and 40 invalid, defined to validate the semantics and the implementation.

Experimental results. Table 1 shows the results on a subset of formulas. The solver proves most of the tautologies of group 4 in less than 10 seconds and 38 properties were proved in less than 2 seconds. All the invalid properties of this group were disproved in less than 2 seconds with the bmc algorithm. Moreover, we performed an experimental evaluation using the example defined in Section 2. We were able to prove the example by splitting the property into two parts. First, we proved that q is an upper bound for the maximum distances between c and $c1$ if $c1$ is synchronized to c every r time unit. Second, we proved that assuming a maximum distance between c and the other clocks the property holds.

The experiments on λ and ϵ pointed out the instability brought by instantiating these parameters to an arbitrary value. In particular, figure 3a and figure 3b show the impact of the two constants increasing their values. The plots show a strong instability with jumps in execution time with both λ and ϵ.

Figure 3c shows the experimental evaluation of the Fischer algorithm formula. The IC3 algorithm proves the correctness of the formula with 11 components in less than one hour. Figure 3d shows the Bounded Response experimental evaluation. No algorithm can prove the formula with 6 components. Indeed, this model is challenging because to prove that the global component sees a_n before q time units we need to ensure that each component skewed does not delay too much the response. Moreover, this model is parametrized over λ, ϵ, p and r; thus, the solver needs to explore a larger state space.

8 Conclusions

In this paper, we addressed the problem of automated reasoning with MTLSK properties, a variant of MTL for DRTS with local clocks that are skewed and resettable. To cope with the non-monotonicity of timing constraints in the presence of resets, we introduce min/max operators that guess the min/max value of clocks when subformulas are satisfied. We described and implemented an encoding into LTL satisfiability modulo theory. The results show that the approach is feasible and can be used to assess the validity of formulas for different configurations of the skewed resettable clocks. In the future, we would like to investigate more efficient techniques to find counterexamples based on bounded model checking with different clocks as in [6]; we would study the impact of non-monotonic resets on the compositional reasoning with input/output data [3]; we would like to cover a deeper case study concerning real life examples such as 8N1 protocol and Biphase Mark protocol; we would like to relax the constraints of resettable clocks to match more realistic assumptions on the system; finally, we would extend distributed runtime verification as in [15] to MTLSK.

References

1. R. Alur and T. A. Henzinger. A Really Temporal Logic. *J. ACM*, 41(1):181–204, 1994.
2. D. Barsotti, L. Prensa Nieto, and A. Tiu. Verification of clock synchronization algorithms: Experiments on a combination of deductive tools. *Electronic Notes in Theoretical Computer Science*, 145:63–78, 2006. Proceedings of the 5th International Workshop on Automated Verification of Critical Systems (AVoCS 2005).
3. A. Bombardelli and S. Tonetta. Asynchronous Composition of Local Interface LTL Properties. In *NFM*, pages 508–526, 2022.
4. A. Bombardelli and S. Tonetta. Metric Temporal Logic with Resettable Skewed Clocks - version with proofs. In *DATE*, 2023. To appear, preproceeding version available at https://es-static.fbk.eu/people/bombardelli/papers/date23/extended_abstract.pdf.
5. G. Brown and L. Pike. Easy Parameterized Verification of Biphase Mark and 8N1 Protocols. volume 3920, pages 58–72, 03 2006.
6. L. Bu, A. Cimatti, X. Li, S. Mover, and S. Tonetta. Model Checking of Hybrid Systems Using Shallow Synchronization. In *FMOODS/FORTE*, volume 6117 of *LNCS*, pages 155–169, 2010.
7. C. Carapelle, S. Feng, O. F. Gil, and K. Quaas. Satisfiability for MTL and TPTL over Non-monotonic Data Words. In *LATA*, volume 8370 of *LNCS*, pages 248–259, 2014.
8. A. Cimatti, A. Griggio, E. Magnago, M. Roveri, and S. Tonetta. Extending nuXmv with Timed Transition Systems and Timed Temporal Properties. In *Computer Aided Verification*, pages 376–386, Cham, 2019.
9. A. Cimatti, A. Griggio, E. Magnago, M. Roveri, and S. Tonetta. SMT-based satisfiability of first-order LTL with event freezing functions and metric operators. *Information and Computation*, 272:104502, 12 2019.
10. A. Cimatti, A. Griggio, S. Mover, and S. Tonetta. IC3 Modulo Theories via Implicit Predicate Abstraction. In *TACAS*, volume 8413 of *Lecture Notes in Computer Science*, pages 46–61. Springer, 2014.
11. A. Cimatti, A. Griggio, S. Mover, and S. Tonetta. Verifying LTL Properties of Hybrid Systems with K-Liveness. In A. Biere and R. Bloem, editors, *Computer Aided Verification*, pages 424–440, 2014.
12. A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani. The MathSAT5 SMT Solver. In N. Piterman and S. A. Smolka, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 93–107, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
13. W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand. Using contract-based component specifications for virtual integration testing and architecture design. pages 1 – 6, 04 2011.
14. L. de Moura, S. Owre, H. Ruess, J. Rushby, N. Shankar, M. Sorea, and A. Tiwari. Sal 2. volume 3114, pages 496–500, 07 2004.
15. R. Ganguly, Y. Xue, A. Jonckheere, P. Ljungy, B. Schornsteiny, B. Bonakdarpour, and M. Herlihy. Distributed Runtime Verification of Metric Temporal Properties for Cross-Chain Protocols. *CoRR*, abs/2204.09796, 2022.
16. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4):255–299, oct 1990.
17. J. J. Ortiz, M. Amrani, and P. Schobbens. MLν: A Distributed Real-Time Modal Logic. In *NFM*, pages 19–35, 2019.

18. J. J. Ortiz, A. Legay, and P. Schobbens. Distributed Event Clock Automata - Extended Abstract. In *CIAA*, pages 250–263, 2011.
19. A. Pnueli. The temporal logic of programs. pages 46–57, 09 1977.
20. J.-F. Raskin and P.-Y. Schobbens. The Logic of Event Clocks - Decidability, Complexity and Expressiveness. *Journal of Automata, Languages and Combinatorics*, 4(3):247–286, 1999.
21. G. Rodríguez-Navas and J. Proenza. Using Timed Automata for Modeling Distributed Systems with Clocks: Challenges and Solutions. *IEEE Trans. Software Eng.*, 39(6):857–868, 2013.
22. J. M. Rushby and F. W. von Henke. Formal verification of the interactive convergence clock synchronization algorithm using ehdm. 1989.
23. S. Tonetta. Linear-time temporal logic with event freezing functions. *Electronic Proceedings in Theoretical Computer Science*, 256, 09 2017.
24. F. Wang, A. K. Mok, and E. A. Emerson. Distributed Real-Time System Specification and Verification in APTL. *TOSEM*, 2(4):346–378, 1993.

A Proofs

A.1 Proof of theorem 2

Proof. We prove the theorem inductively on the length of the formula.

Base case: Base case is trivial since the formulae are equal, and thus, they are also equisatisfiable.

Inductive case: We ignore homomorphic transformations since they trivially holds.

- $\varphi U^c_{\lhd p}\psi$:
  $\pi, t \models \varphi U^c_{\lhd p}\psi \Leftrightarrow \exists t' \geq t.\pi, t' \models \psi$ and $\pi(t')(c) - \pi(t)(c) \lhd p$ and $\forall t < t'' < t' : \pi, t'' \models \varphi$. By U set definition $\pi(t)(U^c_{\varphi U\psi}) \neq \emptyset \Leftrightarrow \pi, t \models \varphi U\psi$.
  If $\pi, t \not\models \varphi U\psi$ then both the formulae are not satisfied. If $\pi, t \models \varphi U\psi$ then $\pi, t \models \varphi U^c_{\lhd p}\psi \Leftrightarrow \exists c_v \in U^c_\pi(t, \varphi, \psi)$ s.t. $c_v - \pi(t)(c) \lhd p$. Since c diverges (i.e. $\forall t, \exists t' \geq t$ s.t. $\pi(t')(c) > \pi(t)(c)$) and by assumption λ bounds the reset lower value that could be reached, $min(U^c_\pi(t, \varphi, \psi)$ is defined. Since min is the smallest value of the set, then $\exists c_v \in U^c_\pi(t, \varphi, \psi)$ s.t. $c_v - c \lhd p \Leftrightarrow \min(U^c_\pi(t, \varphi, \psi)) - c \lhd p \Leftrightarrow \pi, t \models min\Delta^c_{\varphi U\psi} \lhd p$. By induction hypothesis φ and $\Upsilon(\varphi)$ are equisatisfiable, ψ and $\Upsilon(\psi)$ are equisatisfiable; thus, $\cdots \Leftrightarrow \pi, t \models min\Delta^c_{\Upsilon(\varphi U\psi)} \lhd p$.
- $\varphi U^c_{\rhd p}\psi$: If $\pi, t \models G(\varphi \wedge F\psi)$ then there are infinitely many $t' > t$ s.t. $\pi, t' \models \psi$ and $\pi(t')(c) - \pi(t)(c) \rhd p$ and $\forall t \leq t'' < t : \pi, t'' \models \varphi$ (because clock diverges). Thus, if $\pi, t \models G(\varphi \wedge F\psi)$ both properties holds. As for $\lhd$ case, if $\varphi U\psi$ does not hold, then both properties are falsified. The remain of this case is equal to the $\lhd$ case (just replace min with max).
- $\varphi \overline{U}^c_{\lhd p}\psi$: If $\pi, t \not\models \varphi U\psi$ then neither $\varphi \overline{U}^c_{\lhd p}\psi$ nor $\varphi U\psi \wedge maxbef\,\Delta^c_{v(\varphi)} \lhd p$ holds. If $\pi, t \models \varphi U\psi$, then $\pi(t)(maxbef\,\Delta^c_\psi) = \max(\pi(t)(Bef^c_\psi)) - \pi(t)(c)$. By $\overline{U}^c_{\lhd p}$ definition, $\pi, t \models \varphi \overline{U}^c_{\lhd p}\psi \Leftrightarrow \pi, t \models \varphi U\psi$ and $\forall c_v \in \pi(t)(Bef^c_\psi) c_v - c \lhd p \Leftrightarrow \max(Bef^c_\psi) - c \lhd p \Leftrightarrow \pi, t \models maxbef\,\Delta^c_\psi \lhd p$. (Similarly to the previous cases, all the points are below the threshold iff their maximum is below that threshold).

A.2 Proof of theorem 4

We need to prove that for all π, for all t: $\pi(t)(min\Delta^c_{\varphi U\psi}) = \pi(t)(\rho_{min\Delta^c_{\varphi U\psi}})$. We need to prove that $\pi(t)(\rho_{\min^c_{\varphi U\psi}}) = min(\pi(t)(U^c_{\varphi U\psi})) - \pi(t)(c)$ (min is granted to exists because the set is not empty and definition 2 prevents the clocks from resetting below time -λ).

Let $m_t = min(\pi(t)(U^c_{\varphi U\psi}))$ and $m\delta_t = m_t - \pi(t)(c)$
Either $m_t = \pi(t)(c)$ or $m_t = m_{t+1}$.

If $m_t = \pi(t)(c)$ then ψ holds and either $\pi, t \not\models \varphi \tilde{U} \psi$ or $m_t \leq m_{t+1}$.
If $m_t = m_{t+1}$ then either $\pi, t \not\models \psi$ or $\pi(t)(c) > m_{t+1}$
Thus, $\pi(t)(m_t) = \pi(t)(ite(\psi \wedge (\neg(\varphi \tilde{U} \psi) \vee \pi(t)(c) \leq m_{t+1}), c, m_{t+1})$.
We derive the formula in term of $m\delta_i$:
$m\delta_i = ite(\psi \wedge (\neg\varphi \tilde{U} \psi \vee c - c \leq m\delta_{i+1} +$
$next(c) - c, c - c, m\delta_{i+1} + next(c) - c) =$
$ite(\psi \wedge (\varphi \tilde{U} \psi \vee 0 \leq m\delta_{i+1} + \delta_c, 0, m\delta_{i+1} + \delta_c))$
By replacing $m\delta_i$ with $\rho_{min\Delta^c_{\varphi U \psi}}$ we obtain
$\rho_{min\Delta^c_{\varphi U \psi}} = ite(\psi \wedge (\neg(\varphi \tilde{U} \psi) \vee 0 \leq \rho'_{min\Delta^c_{\varphi U \psi}} + \delta_c, 0, \rho_{min\Delta^c_{\varphi U \psi}} + \delta_c))$
Finally, if $\pi, t \models F(\psi \wedge \rho_{min\Delta^c_{\varphi U \psi}} = 0)$ then $\pi(t)(min\Delta^c_{\varphi U \psi}) = \pi(t)(\rho_{min\Delta^c_{\varphi U \psi}})$

A.3 Proof of theorem 5

Same as theorem 4.

A.4 Proof of theorem 6

We need to prove that for all π, if $\pi, t \models F\varphi$, then $\pi(t)(maxbef\Delta^c_\varphi) = \pi(t)(\rho_{maxbef\Delta^c_\varphi})$.

- Since $F\varphi$ is satisfied then $Bef^c_\pi(t, \varphi)$ is a finite (π is a discrete trace) and not empty. We prove it by induction. $\pi(t)(\rho_{maxbef\Delta^c_\varphi} = max(Bef^c_\pi(t+1, \varphi) - c$
  Base: $\pi, t \models \varphi$. $Bef^c_\pi(t, \varphi) = \{\pi(t)(c)\}$; thus, $max(Bef^c_\pi(t, \varphi) = \pi(t)(c)$.
  Thus, $\pi(t)(maxbef\Delta^c_\varphi) = 0$. Since $\pi, \models \varphi$, then $\pi(t)(\rho_{\max\_bef^c_\varphi}) = 0$. (if then else).
  Inductive case: Since $\pi, t \not\models \varphi$ and $\pi, t \models F\varphi$, then $Bef^c_\pi(t, \varphi) = \{\pi(t)(c)\} \cup \pi(t+1)(Bef^c_\varphi)\}$. Thus, $max(Bef^c_\pi(t, \varphi)) = max(\pi(t)(c), max(\pi(t+1)(Bef^c_\varphi)))$.
  By induction hypothesis: $\pi(t+1)(\rho_{maxbef\Delta^c_\varphi}) = max(\pi(t+1)(Bef^c_\varphi))$. Thus, if $0 \geq \rho'_{maxbef\Delta^c_\varphi} + \delta_c$ then $\pi(t)(c)$ is the max. Otherwise $\pi(t+1)(c)$ is the maximum (as shown by ite expr.).

A.5 Proof of theorem 3

We use the proofs described in [9] and [23] with the difference that since in our case clocks can occur only inside event predicates, we do not need continuity constraints.

Therefore, we need to extend the inductive proof for $\mathcal{D}$ to $min\Delta, max\Delta$ and $maxbef\Delta$.

Subsequently, we consider the notation of [9] where σ represents the timed trace, σ_D the discrete trace, t_i represents the point in the timed trace represented by the interval t_i and i as its discrete point counterpart.

- $min\Delta$. Suppose that $\sigma, t_i \models \varphi U \psi$. $\sigma(t_i)(min\Delta^c_{\varphi U \psi}) = min(\sigma(t)(U^c_{\varphi U \psi}))$. $\sigma(t_i)(U^c_{\varphi U \psi}) = \{\sigma(t')(c) | t' \geq t$ and $\sigma, t' \models \psi$ and $\forall t \leq t'' < t' : \sigma, t'' \models \varphi\})$. $\cdots = \{\sigma(t_i)(c)$if $\sigma, t_i \models \psi\} \cup \{\sigma(t)(c) | t' > t_i \pi, t' \models \psi$ and $\forall t_i \leq t'' < t' \sigma, t'' \models \varphi\}$. By induction, $t' \in I_j$ for some j. Since σ is fine, $\sigma, t_j \models \psi$ and if t_j is open then $\sigma, t_j \models \varphi$. Thus, $\cdots = \{\sigma_D(i)(c)\} \cup \sigma_D(i)(U^c_{\mathcal{D}(\varphi)U(\mathcal{D}(\psi) \wedge (\iota \vee \mathcal{D}(\varphi)))})$ if $\sigma_D, i \models \mathcal{D}(\psi)$. Otherwise, $\cdots = U^c_{\mathcal{D}(\varphi)U(\mathcal{D}(\psi) \wedge (\iota \vee \mathcal{D}(\varphi)))}$. Thus, the discretization is correct (ite, then $\mathcal{D}(\varphi)U(\mathcal{D}(\psi) \wedge (\iota \vee \mathcal{D}(\varphi)))$).
- $max\Delta$ As $min\Delta$.
- $maxbef\Delta$. The discretization trivially holds because φ is fine, and thus, does not change value during the interval.