2013 20th International Symposium on Temporal Representation and Reasoning

A New Metric Temporal Logic for Hybrid Systems

Mark Reynolds
School of Computer Science and Software Engineering
The University of Western Australia, Perth
Email: mark.reynolds@uwa.edu.au

***Abstract*—We introduce a new way of defining metric temporal logic over the continuous real model of time. The semantics refer to a single universal clock in order to impose metric constraints to any desired precision. Furthermore, the expression of any non-metric aspects can correctly utilise the full power of continuous time temporal logic.**

Syntactic constructs afford the convenient succinct expression of many useful and typical constraints while other, more intricate properties are able to be captured but may require more lengthy formulation.

A decision procedure is provided via a simple translation into an existing non-metric temporal logic and this gives a workable complexity and the possibility of automated reasoning.

There are advantages in expressiveness, naturalness, generality and amenability to reasoning techniques over the existing metric temporal logics.

Combining purely continuous with adequate metric aspects in one language makes the logic very suitable for dealing with hybrid systems.

I. Introduction

Metric temporal logic is used for applications involving specification and verification of real-time and hybrid systems [10]. A dense, or specifically real-numbers, model of time is used and a formal logical language is employed with the ability to express metric, or quantitative timing requirements as well as the relative order and overlap of propositional states and events. Early approaches include [24] and [26].

Several closely related popular current metric temporal logics are called MTL for *metric temporal logic* [24], [2]. They allow convenient expression of metric or quantitative temporal constraints. For example, we might want to say that every time a button is pressed, p, then it will be disabled, q, until the dialogue disappears, r, and that will happen within 3 seconds. In MTL, our requirement is rendered as $G_{(0,\infty)}(p \to (qU_{(0,3)}r))$ where $G_{(0,\infty)}$ is a temporal operator quantifying a formula as always holding from now on, and $qU_{(0,3)}r$ indicates that formula q will hold until formula r does sometime within the next 3 seconds. Other metric temporal logics such as TPTL [4], MITL [1] or the Duration Calculus [9] have since developed and comparisons are not always straightforward but most subsequent work on metric temporal logics has been with MTL or fragments of MTL.

The current situation with metric temporal logics is unsatisfactory for several reasons. There are two main versions of MTL with different semantics. Unfortunately, the version that is less natural, and less expressive is the version most amenable to automated reasoning tasks.

The most highly developed version of MTL is based on what is often called *point-wise* or discrete semantics [28]. This means that we evaluate formulas over countably infinite discrete sequences of events at which a system may change state. This logic may be amenable to automated techniques essentially by conversion to discrete reasoning on the sequences. Unfortunately, the formulas in this logic have what can only be seen to be un-intuitive meanings with sub-formulas needing to be evaluated at state change points only. For example, $F_{(0,5)}F_{(0,5)}p$ may not be true even if p holds within ten seconds (because there is no intervening change of state). See [29], [19].

The other, less common, semantics for MTL is called the *continuous* semantics. It allows more natural understanding of temporal operators [28] and allows more properties to be expressed [14]. Unfortunately, MTL is highly undecidable over continuous time semantics [1]. See section II.

In this paper we introduce another seemingly similar but actually quite different metric temporal logic 1CMTL, *one clock metric temporal logic*, with a continuous time semantics. The main difference between 1CMTL and MTL is that 1CMTL is based on the idea of there being one universal clock available against which all quantitative measurements are evaluated. The 1CMTL clock is universal in the sense that the clock readings mentioned in one subformula of a 1CMTL specification are from the same clock as readings mentioned in any other subformula. This allows fairly straightforward expression of many typical metric temporal constraints. We will see how to translate 1CMTL formulas into roughly equivalent MTL ones.

For example, $G^{0.25}_{(0,8)}(p \to qU^{0.25}_{(0,4.25)}r)$ means that, as measured on a digital clock which changes its display only every 0.25 seconds, for the next 8 seconds, every p event is followed by q remaining continuously true until r holds, and that will happen between 0 and 4.25 seconds after the p event.

There are, of course, precursor suggestions allowing imprecision in metric constraints when formalising requirements. Notable are the imprecise recording of times of state changes in [4], the approximate properties in [17], the robust timed automata of [16], the weakening functions of [22] and the metric interval temporal logic MITL introduced under the idea of relaxing punctuality [1]. However, our one clock approach to imprecision seems to be novel and we will see that it has some advantages.

This paper also tackles the need to reason with 1CMTL. To that end we show that 1CMTL formulas can actually be expressed in a simple, somewhat low-level, metric temporal

1530-1311/13 $26.00 © 2013 IEEE
DOI 10.1109/TIME.2013.17

language called MRTL which was recently introduced in [32]. MRTL itself is built on RTL [31] which is a traditional non-metric temporal logic. RTL is just the propositional temporal logic of the Until and Since connectives over real-numbers time. This was developed in [23], shown to be sufficiently expressive [23], shown to be decidable in [8], axiomatized in [15], [30] and recently shown to be in PSPACE [31]. RTL has a pure continuous semantics with no ad hoc constraints such as finite variability or non-zenoness imposed on the behaviours of the propositions. We give a few more details about RTL in section IV.

It is important to note that the purely continuous RTL is a sublanguage of both MRTL and 1CMTL. Thus its pure continuous version of Until and Since are available for specifications. 1CMTL is thus not like older translations of continuous properties into discrete ones [17]. So, for example, density of a proposition is required by just a simple $G(\neg((\neg p)U\top))$ conjunct and this means we do not have intervals *of any length* without p holding ($\top$ is just truth). Such truly continuous properties are needed for hybrid systems and faithful capture of natural language specifications.

MRTL builds on top of RTL simply by using some propositions to mark the ticking of a single universal clock matching the standard metric on the reals. These metric, or *ticking*, propositions are fixed, or pre-defined in any MRTL structure. Having a hierarchy of nested ticking propositions allows simple reference to arbitrarily accurate and arbitrarily extensive constraints. Again, the other non-metric propositions are completely unconstrained. Interestingly, some simple useful properties can be easily expressed in MRTL even though they are known to be beyond any MTL style metric temporal logic [32].

In [32] we were also able to to show the decidability of the validity problem for MRTL in a surprising way, namely by transforming each MRTL formula into a non-metric RTL formula.

Neither ICMTL nor MRTL can be translated into (traditional discrete time) PLTL. It is true that in any given 1CMTL formula, there is a *most fine precision* mentioned and all metric durations can be assumed to be provided in multiples of that. Many traditional simple approaches to continuous time properties rely on such an assumption of *finite variability* for all properties and this allows a trivial translation into discrete time temporal logics. However, 1CMTL is designed to allow reasoning about truly continuous aspects of systems as well as metric constraints and the interrelationships between aspects of both sorts. Properties related to density, zenoness, accumulations of oscillations, separability, dedekind completeness etc are able to be truly stated in MRTL, because pure continuous RTL is a sub-language. There is no apparent way to preserve them while making a translation to a discrete model of time. We do not translate 1CMTL reasoning to PLTL reasoning.

The expressivity of the pure continuous semantics of 1CMTL, MRTL and RTL, and the consequent inability to translate trivially to PLTL, is shared by some other recent approaches to metric temporal reasoning. However, they seem to have their own complications. We could mention [20] with its infinite number of counting modalities and [7] with its complicated syntactical restrictions and undecidability of satisfiability.

As a fairly simple example of the usefulness of having a pure continuous semantics, consider one of the plausible (but not actually physically realisable) models of a repeatedly bouncing ball described in [12]. The ball bounces an infinite number of times after being dropped but comes to rest within a finite period because of the convergence of the sum of shorter and shorter bounce times (as it loses energy). In reasoning about a detector capable of sensing when the ball is touching the floor, say f, we may want an alarm a to sound within 5 seconds of the ball coming to rest after an episode of such infinite bouncing. In MTL, MITL, RTL, MRTL and 1CMTL we can use the *Since* S operator, the past-time mirror to U, to describe a bouncing accumulation point via $\beta = \beta_1 \wedge \beta_2 \wedge \beta_3$ as follows:

$$\begin{aligned} \beta_1 = &\quad \neg((\neg f)S\top) \\ \beta_2 = &\quad ((\neg f)Sf)S\top \\ \beta_3 = &\quad fU\top \end{aligned}$$

In MITL (and so in MTL), the overall specification for the alarm system can be given as $G(\beta \rightarrow F_{\leq 5}a)$ while in 1CMTL, we just have to also specify a clock precision and say, for example, $G(\beta \rightarrow F^{0.125}_{[0,5]}a)$. In both languages, the semantics allows truly continuous behaviour, and the specifications do mean what we say they mean: an alarm sounds shortly after the end of an accumulation of bounces. The advantage of the 1CMTL language here, over MITL, is that 1CMTL is decidable: as the bouncing property contravenes finite variability, there is no known decision procedure for handling such specifications in MITL (or MTL). There are many more sophisticated and practical examples from the world of mechanical, electrical and quantum devices which we hope to illustrate in a future paper.

We note that in other related work [33], [31], [34], [35] tableau techniques are being developed for reasoning with RTL. These will carry over directly to MRTL via the above-mentioned translation and then thus be available for 1CMTL.

We also show that the two translations give an EXPSPACE complexity result for 1CMTL in section VII.

Thus 1CMTL is a new metric temporal logic which is very expressive, can truly capture continuous-time properties, is natural to use, can be applied in very general situations, affords a wide range of useful operators, has a decision procedure and has tableau-based implementable decision procedures under development. Recall that, in contrast, continuous-time MTL is undecidable, MITL without finite variability has no existing decision procedure and MITL with finite variability can not express many truly continuous-time properties.

We will see below that there are also philosophical advantages to using 1CMTL, in addition to that of having a fairly simple metric language based faithfully, "conservatively", on a purely continuous model of time. In section II we explain how the one clock approach has advantages in physical plausibility over specification languages such as MTL and MITL which "involve" an infinite number of infinitely precise stop-clocks.

More generally, in section II we give a short account of existing metric temporal logics. In section III we describe our new metric temporal logic 1CMTL. In section IV we remind the reader of the non-metric RTL and the low-level metric MRTL. In section VI we show that 1CMTL formulas can be expressed in MRTL. This allows us to establish the decidability and complexity of 1CMTL in the next section. We conclude with a summary and discussion of future work.

II. Existing MTL

We have mentioned that there are a wide variety of metric temporal logics. There are choices in the fundamentals of the semantics, discrete or continuous, and also in the expressiveness of the language. As we have seen, and as already noted in [19], the situation is messy with general results about expressiveness, decidability and complexity being confused by ad hoc circumstances and these properties being sensitive to slight differences in the semantics or choice of operators [13]. Thus we only give a brief overview.

Most of the timing work over the last two decades have been using the discrete (also called pointwise) semantics and the reasoning algorithms have been built on discrete temporal techniques such as automata or translations to PLTL. However, such approaches give rise to an un-natural, un-intuitive, expression of specifications exemplified by the $F_{(0,5)}F_{(0,5)}\alpha$ situation which we mentioned above. The discrete logics are also less generally applicable especially as they make strict assumptions on the behaviour of propositions. Expressiveness of the approaches is compared in [11].

We will concentrate on the more natural continuous semantics (also called interval-based).

The models are based on *boolean signals*, i.e. maps which determine the truth or falsity of propositions at any real-numbers time. We consider signals over the whole real numbers flow, allowing behaviours to have been going on infinitely into the past. However, it is also common to see the positive or non-negative reals being used as a frame.

Definition. [R-structure] Fix a countable set $\mathcal{A}$ of atoms. An $\mathbb{R}$-*structure* $\mathbf{R} = (\mathbb{R}, <, h)$ has a frame $(\mathbb{R}, <)$, the reals under the usual irreflexive ordering, and a valuation, or *boolean signal*, h for the atoms, i.e. for each atom $p \in \mathcal{A}$, $h(p) \subseteq \mathbb{R}$.

In what follows, (metric) intervals I will be interval subsets of $(0, \infty) \subset \mathbb{R}$ with end points in $\mathbb{Q} \cup \{\infty\}$. Again, there are variants in which the intervals only have natural number end-points. The language for MTL is generated by the 2-place connectives U_I and S_I for each interval I along with classical $\neg$ and $\wedge$. So we define the set of formulas recursively to contain the atoms and for formulas α and β we include $\neg\alpha$, $\alpha \wedge \beta$, $\beta U_I \alpha$ and $\beta S_I \alpha$.

MTL formulas are evaluated at points in structures $\mathbf{R} = (\mathbb{R}, <, h)$. We write $\mathbf{R}, x \models \alpha$ when α is true at the point $x \in \mathbb{R}$. This is defined recursively as in Figure 1.

Suppose that we have defined the truth of formulas α and β at all points of $\mathbf{R}$. Then for all points x:

$\mathbf{R}, x \models p$	iff	$x \in h(p)$, for p atomic;
$\mathbf{R}, x \models \neg\alpha$	iff	$\mathbf{R}, x \not\models \alpha$;
$\mathbf{R}, x \models \alpha \wedge \beta$	iff	both $\mathbf{R}, x \models \alpha$ and $\mathbf{R}, x \models \beta$;
$\mathbf{R}, x \models \beta U_I \alpha$	iff	there is $y > x$ in $\mathbb{R}$ such that $y - x \in I$ and $\mathbf{R}, y \models \alpha$ and for all $z \in \mathbb{R}$ such that $x < z < y$, we have $\mathbf{R}, z \models \beta$; and
$\mathbf{R}, x \models \beta S_I \alpha$	iff	there is $y < x$ in $\mathbb{R}$ such that $x - y \in I$ and $\mathbf{R}, y \models \alpha$ and for all $z \in \mathbb{R}$ such that $y < z < x$, we have $\mathbf{R}, z \models \beta$.

Fig. 1. Semantics for MTL

Abbreviations include $F_I\alpha \equiv \top I U_\alpha$ and $G_I\alpha \equiv \neg F_I \neg\alpha$. The unrestricted Until of RTL is also an abbreviation $\alpha U \beta \equiv \alpha U_{(0,\infty)}\beta$. Similarly, Since.

Finite variability of the boolean signals is often assumed (explicitly) in order to obtain technical results about expressibility, decidability or complexity of MTL-like languages. There are some different ways of defining these restrictions but essentially we require that every proposition changes truth value only a finite number of times in any bounded interval of time [19].

One aspect of MTL (and MTL-like languages) which is not often discussed but is relevant for us is the way that actual observed behaviours may be evaluated against specifications. This is important for practical applications, but it also determines whether specifications in the language can correspond to some desired property in theory.

MTL has formulas such as $G(r \to pU_{[2.5,3.5]}q))$ saying that every r event is followed after a period of between 2.5 and 3.5 units (inclusive), by a q event and p holds continuously in between. Checking whether an actual behaviour of a system satisfies an MTL specification may thus be physically impossible. It would seem to need an indefinite number of stop-watches, one being set off whenever an r event occurs. Why do we need a lot of stop-watches? Because, the time limits are strictly defined. There is a non-zero range or interval in which an event can occur (this means the formula is in the sub-language MITL that we meet below), but the range has very precise end points that have to be measured exactly from when the r event occurs. It is no good trying to use a global clock unless you are capable of recording the time of every r event with infinite accuracy. This is because having a q event 2.49999 units after r and none until the next 1.00002 units after that is not enough to satisfy the specification. MTL (and MITL) languages rely on measuring durations to infinite accuracy in order to determine whether formulas are satisfied, i.e. whether a particular observed duration lies within an interval or not. Furthermore, even a short formula (such as $G(r \to pU_{[2.5,3.5]}q))$ may require an infinite number of such measurements to be made (as an Until formula may need to be evaluated at an infinite number of starting points).

Thus we claim in this paper that MTL (and MITL) requires infinitely many, infinitely accurate clocks.

MTL-like languages can express a reasonable range of metric temporal constraints. However, Amir Pnueli suggested that the modalities of MTL, and similar languages, are not completely adequate. He presented the following example specification: p and then q will hold within the coming unit of time. He conjectured that such specifications can not be expressed in metric languages like MTL with finite numbers of connectives [2], [36]. This is sometimes known as Pnueli's conjecture. In [6] it was shown that this example can be expressed in MTL, but other useful properties can not be and in [18], [21], Hirschfield and Rabinovich proved a stronger lack of expressiveness result for MITL. It seems that MTL style languages are not able to express these simple and useful properties. See also [27].

It has long been known that deciding valid formulas in MTL over dense time is highly undecidable [3]: there can be no procedure for determining validity. With some restrictions on behaviour, the (restricted) logics can be decided [14] and [1] but the procedures are so complicated that no implementations exists. Over the much simpler discrete model of time MTL is decidable with an EXPSPACE complexity [3] and tools do exist [5]. Over continuous time semantics the best results are probably for the following two MTL-like languages.

Metric Interval Temporal Logic (MITL) was introduced in [1] to be the fragment of MTL in which the intervals I on the operators U_I can not be singleton intervals. They showed that deciding validity MITL (in pointwise time semantics with finite variability) is EXPSPACE-complete. MITL was introduced in a paper on "relaxing punctuality" but as we saw in the example above, the allowed intervals still have infinitely precise limits as measured from each and every one trigger event. Thus checking that a behaviour satisfies a MITL formula in general can involve an infinite number of infinitely precise stop clocks.

QTL, the quantified temporal logic, was introduced in [19]. It has ordinary non-metric until and since along with just two new operators $\Diamond_1\alpha$ meaning that α is true within the next one unit of time (plus mirror image operator). They showed that QTL is exactly as expressive as MITL over unrestricted continuous time semantics provided that the intervals in the MITL syntax only have integer end points. They show that deciding QTL is PSPACE-complete but note that the language is not succinct as the expression of any long term constraint requires repeated nesting of $\Diamond_1$.

III. 1CMTL

In this section we introduce the new metric temporal logic 1CMTL which is the main contribution of the paper.

Semantics is over $\mathbb{R}$-structures just as for the version of MTL which we introduced above.

Temporal formulas involve a clock granularity, or precision, which is assumed to be $e = 2^{-m}$ for some $m \geq 0$. Following MTL, (metric) intervals I will be interval subsets of $[0, \infty) \subset \mathbb{R}$ with end points in $\mathbb{Q} \cup \{\infty\}$. Note that 0 is allowed to be in the interval (unlike in MTL).

$\mathbf{R}, x \models p$	iff	$x \in h(p)$, for $p \in \mathcal{A}$;
$\mathbf{R}, x \models \neg\alpha$	iff	$\mathbf{R}, x \not\models \alpha$;
$\mathbf{R}, x \models \alpha \wedge \beta$	iff	both $\mathbf{R}, x \models \alpha$ and $\mathbf{R}, x \models \beta$;
$\mathbf{R}, x \models \beta U_I^e \alpha$	iff	there is $y > x$ in $\mathbb{R}$ such that $[y]_e - [x]_e \in I$ and $\mathbf{R}, y \models \alpha$ and for all $z \in \mathbb{R}$ s.t. $x < z < y$, we have $\mathbf{R}, z \models \beta$; and
$\mathbf{R}, x \models \beta S_I^e \alpha$	iff	there is $y < x$ in $\mathbb{R}$ such that $[x]_e - [y]_e \in I$ and $\mathbf{R}, y \models \alpha$ and for all $z \in \mathbb{R}$ s.t. $y < z < x$, we have $\mathbf{R}, z \models \beta$.

Fig. 2. Semantic clauses for 1CMTL

The language for 1CMTL is generated by the 2-place connectives U_I^e and S_I^e for each interval I and each clock granularity e, along with classical $\neg$ and $\wedge$. So we define the set of formulas recursively to contain the atoms and for formulas α and β we include $\neg\alpha$, $\alpha \wedge \beta$, $\beta U_I^e \alpha$ and $\beta S_I^e \alpha$.

Suppose $e = 2^{-m}$ for some $m \in \mathbb{Z}_{\geq 0}$. This value will determine the granularity of clock that is used to assess the metric information. With respect to a clock of granularity e, we say that the e-clock time of an event which occurs at real-time $t \in \mathbb{R}$ is the rational value $[t]_e = e\lfloor t/e \rfloor$. For example, $[0.367]_{0.25} = 0.25$. Thus we imagine a digital 0.25-clock showing 0.25 on the display at every time $t \in [0.25, 0.5)$.

The termination interval is determined by two non-negative rational numbers, $a, b \in \mathbb{Q}$ such that $0 \leq a \leq b$. In fact, we will see that we can also suppose that $a = [a]_e$ and $b = [b]_e$ as specifying an interval to a greater accuracy is ignored in our semantics. Thus, expressing a and b in binary needs at most m places after the binary point.

We now give the semantics for the new two-place connective $U_{[a,b]}^e$. The formula $pU_{[a,b]}^e q$ means that p holds from now, at real-time t say, until some future real-time s, such that q holds at real-time s, and the difference $[s]_e - [t]_e$ in e-clock time between now and real-time s lies in the range $[a]_e$ to $[b]_e$, i.e. $[a]_e \leq [s]_e - [t]_e \leq [b]_e$. Figure 2 gives the semantic clauses.

Abbreviations include $F_I^e \alpha \equiv \top U_I^e \alpha$ and $G_I^e \alpha \equiv \neg F_I^e \neg\alpha$. The unrestricted Until of RTL is also an abbreviation $\alpha U \beta \equiv \alpha U_{[0,\infty)}^1 \beta$ and the reader can check that this is indeed standard Until despite the imprecision of our timing. Unrestricted future occurrence F, and constant truth into the future G are then defined in the usual way from U. Similarly, Since is used to define $P_I^e \alpha \equiv \top S_I^e \alpha$, $H_I^e \alpha \equiv \neg P_I^e \neg\alpha$ and unrestricted S, P and H. The combination $GH\alpha$ thus means that α holds at all times in the past, present and future.

As an example, suppose that $e = 0.25$, q holds at time 1.2 and p holds constantly over the open interval $(0.1, 1.2)$. Then $pU_{[0.5,1.0]}^{0.25} q$ is true at time 0.1. This is despite the fact that q was not true up to 1 unit after the evaluation time and stayed false a little longer. That is because the clock concerned showed time 0 when we started the evaluation and still showed time 1 when q became true.

In the same situation the formula $pU_{[0.5,1.0]}^{0.125} q$ is not true at time 0.1. This is because the (new more accurate clock) shows time 0 when we start and it shows time 1.125 (and not still 1.0) before q is true.

Note that, as mentioned above, the formula $\alpha U^e_{[a,b]}\beta$ has the same truth conditions as $\alpha U^e_{[a',b']}\beta$ whenever $[a]_e = [a']_e$ and $[b]_e = [b']_e$. Thus, we assume for now that we always choose a and b so that $a = [a]_e$ and $b = [b]_e$.

On a similar note, it is recommended, but not required, that properties are usually formalised using one minimum precision value in all superscripts. This is intuitively sensible if we are thinking about timing behaviours against one global clock. Asking whether p is true within 5 seconds, as measured on the clock using 1 second precision, but not true within 5 seconds if using a 0.5 second precision is possible but not a very natural query. It is better to ask the equivalent query: is p true next within 4.5 to 5 seconds, as measured on the clock using 0.5 second precision.

For a more complicated example consider the MTL formula $G(pU_{[2.5,3.5]}q)$. Again, we can not say this exactly in 1CMTL. However, we can something arbitrarily close to this. Consider the 1CMTL version of this property as rendered with a clock display updating every 0.25 seconds: it is $G(pU^{0.25}_{[2.5,3.5]}q)$. The task of assessing whether a particular behaviour satisfies this formula just involves running one global clock with the far from infinite precision of 0.25 seconds. We do have to note down the time of each r event, but we only have to note what the clock says– accurate to 0.25 seconds. For example if there was an r event at time 1.75921 and at time 1.9203 then we just look at the clock each time and note that it says 1.75 both times. We then watch p staying true until perhaps we see q true when the clock says 5.25 and we are done. We do *not* have to start off a new infinitely precise stop-watch every time that an r event occurs as one does in MTL or MITL. In the same situation for MITL, in contrast, you have to start one infinitely precise stop-watch when you see r true at time 1.75921 and another one at time 1.9203. You need to watch very carefully to see that the q event occurs before 5.25921 seconds after the 3.5 duration elapses from the first r. Because infinite precision matters in MITL —yes, even though the timing intervals have non-zero duration— every trigger event needs its own stop watch in principle.

Thus, we have sketched an expressiveness result along the lines of the following. For each formula of the form $\alpha U_I \beta$ of existing MTL, there is a sense in which we can approximate the semantics, the meaning, to any desired level of accuracy with a 1CMTL formula. We leave the rigorous formulation and proof of a result along these lines for the future.

IV. RTL

As background we outline recent work on a traditional non-metric temporal logic over real-numbers time. RTL, the propositional temporal logic over real-numbers time uses the Until and Since connectives introduced in [23]. We know from [23] that this logic is sufficiently expressive for many applications: technically it is expressively complete and so at least as expressive as any other usual temporal logic which could be defined over real-numbers time and as expressive as the first-order monadic logic of the real numbers. Later in the section we will see that there are reasoning techniques for RTL.

The language $L(U,S)$ is generated by the 2-place connectives U and S along with classical $\neg$ and $\wedge$.

Formulas are again evaluated at points in $\mathbb{R}$-structures. The semantic clauses are as for 1CMTL except that $\mathbf{R}, x \models \beta U \alpha$ iff there is $y > x$ in $\mathbb{R}$ such that $\mathbf{R}, y \models \alpha$ and for all $z \in \mathbb{R}$ such that $x < z < y$, we have $\mathbf{R}, z \models \beta$ (and $\beta S \alpha$ is the mirror image).

In [31], we show that, as far as determining validity is concerned, RTL is just as easy to reason with as PLTL. In particular, the complexity of the decision problem is PSPACE-complete. The proof in that paper uses intricate reasoning with the mosaic techniques in temporal logic. We decide whether a finite set of small pieces of models is sufficient to be used to build a real-numbers model of a given formula.

Mosaic reasoning techniques can often be the foundation of tableau implementations [25]. The mosaic proof in [31] suggests a tableau based method for determining RTL validity but details and subsequent development were left for future work. In [35] we make further progress in this direction but there is still much to do.

V. MRTL

In this section we remind the reader of our recently introduced low-level metric temporal logic, MRTL [32].

To define MRTL we work in RTL but split the set of propositional atoms $\mathcal{L}$ into two disjoint infinite sets and reserve one of the countable sets of atoms for special metric purposes leaving the other countable set of atoms for normal propositions. Suppose $\mathcal{L} = \mathcal{A} \cup \mathcal{T}$ where $\mathcal{A}$ and $\mathcal{T}$ are disjoint countably infinite sets of atoms.

Suppose further that $\mathcal{T} = \{..., !^{-2}, !^{-1}, !^0, !^1, !^2, ...\}$. These metric propositions are going to represent the ticking of a clock over time. One element, $! \in \mathcal{T}$, also called $!^0$, will hold for an instant on the event of the regular ticking of the clock every one unit of time. The other propositions in $\mathcal{T}$ represent finer and coarser rates of ticking allowing us to easily refer to arbitrarily small and arbitrarily large durations of time. We will informally call $! = !^0$ a *tick*, $!^{-1}$ a *sub-tick*, $!^1$ a *super-tick*, $!^{-2}$ a *sub-sub-tick*, $!^2$ a *super-super-tick*, etc. These are a range of levels of granularity of ticking. See Figure 3.

The various levels of ticks are propositions indicating the ticking of one single universal clock. Ticks will occur regularly across time. Sub-ticks will happen mid-way between ticks as well as coinciding with each tick: so they are twice as frequent. Super-ticks happen only at alternative ticks so they are half as frequent. Sub-sub-ticks will occur on every sub-tick and mid-way between each adjacent pair of sub-ticks. Super-super-ticks will only occur once every four ticks. And so on. There is thus a two-way infinite linear hierarchy of ticking propositions related to each other by factors of powers of two. Base 10, or other bases, could equally be used instead.

As we mentioned earlier, we call the clock *universal* because there is just one clock, albeit with a hierarchy of layers of ticking, in the semantics. All references to ticks within subformulas of an MRTL formula are references to the ticks in that one hierarchy. We see that this is in contrast to the semantics of MTL-like languages, where every truth evaluation of a subformula at every time point sets off its own stopwatch.

So ! will be true in our MRTL structures at the integer points $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\} \subset \mathbb{R}$. In general $!^n$ will be true at exactly the points $2^n\mathbb{Z} = \{m \cdot 2^n | m \in \mathbb{Z}\}$. So at time point $t = -2$ we have all the following propositions true $\{..., !^{-2}, !^{-1}, !, !^1\}$ in all MRTL structures. See the Figure 1.

To be more rigorous, *MRTL-structures* $\mathbf{R} = (\mathbb{R}, <, h)$ will have the reals as frame and a valuation h for the atoms which is restricted, i.e. pre-defined, for the atoms in $\mathcal{T}$ as follows: for each $n \in \mathbb{Z}$, $h(!^n) = 2^n\mathbb{Z}$.

The language is just $L(U, S)$ as for RTL and formulas are evaluated at points in structures $\mathbf{R} = (\mathbb{R}, <, h)$ just as for RTL. Thus we do not set out the semantic clauses yet again here.

The surprising result from [32] is that reasoning about validity (or equally satisfiability) in the metric logic MRTL can be be accomplished by reasoning (about a slightly different formula) in the non-metric RTL. This gives a PSPACE decision procedure for MRTL via what is quite a simple translation to then use the PSPACE decision procedure for RTL [31]. See [32] for full details.

VI. 1CMTL INTO MRTL

In this section we show how to translate 1CMTL formulas into equ-satisfiable ones in MRTL. Similar ideas allow translation of 1CMTL directly into equivalent MRTL ones but, for the purposes of deciding satisfiability, it is better to translate to shorter equi-satisfiable formulas of MRTL (which are not necessarily equivalent).

First, look briefly at an example. How do we translate $pU^{0.25}_{[2.5,3.5]}q$? Given that we are considering a duration of nearly 4 seconds here and that we want to work at a granularity of 0.25 seconds one way of defining this property in MRTL is to use $4/0.25 = 16$ different conjuncts for the 16 different intervals between sub-sub-ticks lying between adjacent super-super-ticks. Each of these conjuncts is to capture the situation if the starting time point lies in that interval. For example, one conjunct might say (roughly) that the starting point (modulo 4) is in $[1.25, 1.5)$ and q holds from then until an end point where p holds at a time (modulo 4) in either the next interval $[3.75, 4)$ or the next interval $[0, 1.0)$. Saying that a time modulo 4 lies in a particular interval such as $[1.25, 1.5)$ can be accomplished by checking the truth of the various ticks at the end point. For example, $(\neg!^{-2})U(!^{-2} \wedge !^{-1} \wedge \neg!^0 \wedge (\neg!^0)U(!^1 \wedge \neg!^2))$ says that a time (modulo 4) lies within the interval $[1.25, 1.5)$.

To return to the proof now, our first, and most complicated, task is to show how simple 1CMTL formulas can be translated into MRTL. Later we will show how to work on more complicated formulas by breaking them down. We will just consider the case of a formula of the form $pU^e_{[a,b]}q$ for $0 < a < b$. Other formulas such as $pU^e_{(a,b]}q$, $pU^e_{(0,b]}q$, $pU^e_{[a,\infty)}q$ and $pS^e_{[a,b]}q$ are similar and we do not present the details.

So we sketch the proof that the formula $pU^e_{[a,b]}q$ can be expressed in MRTL.

Recall that we can assume that $e = 2^{-m}$ for some $m \geq 0$ and that $a = [a]_e$ and $b = [b]_e$.

Say that $n = \lfloor \log_2 b \rfloor + 1$. Our rendering of the formula will be a rather long MRTL formula using only the atoms p and q and those in $\mathcal{T}^n_{-m} = \{!^{-m}, !^{-m+1}, ..., !^n\}$.

The definitions below assume that n and m are now fixed in context.

Consider $pU^e_{[a,b]}q$ being true at some time $t_0 \in \mathbb{R}$ in some MRTL model. The atoms in $\mathcal{T}^n_{-m}$ switch on and off in a certain pattern spread over a length of 2^n which repeats infinitely in both directions. For each $j \in \mathbb{Z}$, call the interval $[j.2^n, (j + 1).2^n)$ the jth *repeat* of the pattern. There are $2^{n+m} + 1$ distinct time points during that interval of repeated pattern of length 2^n at which $!^{-m}$ (and possibly some of the other ticking propositions) are true, including the start and end point. The jth repeat (which does not include the final end point) is thus divided into 2^{n+m} sub-intervals of the form $\pi^j_k = [j \cdot 2^n + k \cdot 2^{-m}, j \cdot 2^n + (k + 1) \cdot 2^{-m})$ for each $k = 0, 1, ..., 2^{n+m} - 1$.

Thus every time point in $\mathbb{R}$ belongs to one such sub-interval. For given n and m, we say that a time point t is in the kth sector and the jth repeat iff $k = ([t]_e - [t]_{2^n})/e$ and $j = [t]_{2^n}$ or equivalently, $t \in \pi^j_k$ for some j. There are 2^{n+m} sectors of each repeat indexed by $k = 0, 1, ..., 2^{n+m} - 1$.

We will break up the meaning of $pU^e_{[a,b]}q$ into 2^{n+m} different disjuncts depending on the sector of the starting position. For each $k = 0, 1, ..., 2^{n+m} - 1$ we will below make a formula τ_k. This says that now we are in the kth sector (of some repeat) and $\alpha U^e_{[a,b]}\beta$ holds. Thus $\alpha U^e_{[a,b]}\beta \equiv \bigvee_{k=0}^{2^{n+m}-1} \tau_k$.

Now consider some fixed k. We will put $\tau_k = \theta(\eta)$ where η is a MRTL formula and $\theta(r)$ is a MRTL formula using a fresh atom r which we later substitute by η. Let $l = ((k + \lfloor a/e \rfloor) mod 2^{n+m})$ which is the sector number at a duration of a after the start of the kth sector and let $l' = ((k + \lfloor b/e \rfloor) mod 2^{n+m})$ which is the sector number at a duration of b after the start of the kth sector. We will arrange that $\theta(r)$ says that p holds from now (during the kth sector) until the very start of the next lth sector when r holds. And η says that p holds from now at the very start of a lth sector until some time at which q holds and lying at the latest in the next l'th sector.

Let us represent a sector number $k = 0, 1, ..., 2^{n+m} - 1$ in binary as $k = \Sigma_{j=0}^{j<n+m} b_j(k).2^j$ where each $b_j(k) \in \{0, 1\}$. We can say that now we are in a kth sector via $\rho_k = (\neg!^{-m})U(\bigwedge_{j=-m}^{n} \beta_j(k + 1))$ where $\beta_j(k + 1) = !^j$ if $b_j(k + 1) = 1$ and $\beta_j(k + 1) = \neg!^j$ if $b_j(k + 1) = 0$. We thus look ahead to the start of the $k + 1$th sector to see what metric atoms hold then. Note that we could as well have made an alternative formulation instead looking back to the start of the kth sector using Since.

Lemma. $t \in \mathbb{R}$ is in a kth sector iff $R, t \models \rho_k$.

Now, between the kth sector and the next lth sector there should be no $!^n$ true if $k < l < 2^{n+m}$ or exactly one place where $!^n$ is true if $l \leq k$. We need to say that in order to limit

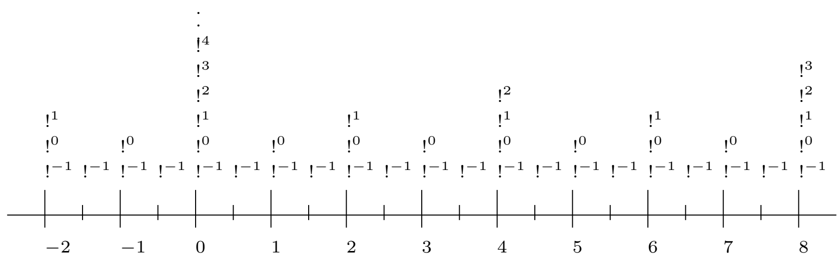

Fig. 3. MRTL pre-defined atoms

the range of the until operator to just the next lth sector (not any later ones).

$\theta(r) = \rho_k \wedge (\neg !^n \wedge p)U(\rho_l \wedge r)$ if $k < l$ and $\theta(r) = \rho_k \wedge (\neg !^n \wedge p)U(!^n \wedge p \wedge (\neg !^n \wedge p)U\rho_l \wedge r)$ if $l \leq k$.

Lemma. p holds from now (during the kth sector) until the start of the next lth sector when r holds iff $R, t \models \theta(r)$.

Similar considerations allow us to define η to mean that p holds from now (at the start of an lth sector) until some time at which q holds before the start of the next $(l'+1)$th sector. Thus, pUq holds and we do not have $\neg q$ holding constantly until the start of the next $(l'+1)$th sector. So we put $\eta = (pUq) \wedge \neg(\neg q \wedge \neg !^n)U(\rho_{l'+1})$ if $l < l'$ and $\eta = (pUq) \wedge \neg(\neg q \wedge \neg !^n)U(\neg q \wedge !^n \wedge (\neg q \wedge \neg !^n)U\rho_{l'+1})$ if $l' \leq l$.

Lemma. p holds from now at the start of a lth sector until some time at which q holds and lying at the latest in the next l'th sector iff $R, t \models \eta$.

So finally we can define $\tau_k = \theta(\eta)$ and we get our result.

Lemma. p holds from now (during the kth sector) until the start of the next lth sector and further until some time at which q holds and lying at the latest in the next l'th sector iff $R, t \models \tau_k$.

Lemma. $R, t \models \bigvee_{k=0}^{2^{n+m}-1} \tau_k$ iff $R, t \models \alpha U^e_{[a,b]}\beta$.

As we have said, we are not going to go through the details for other variations on simple 1CMTL formulas as they are similar and similarly intricate.

It remains to show how to translate a more complicated formula, ϕ say, from 1CMTL into MRTL. This is accomplished by a very straightforward induction based on using fresh atoms for each of the direct subformulas. For example, to translate $\alpha U^e_{[a,b]}\beta$ we choose fresh atoms p_α and p_β and assuming the translation function is T, we put $T(\alpha U^e_{[a,b]}\beta)$

$$= T(p_\alpha U p_\beta) \wedge GH(p_\alpha \leftrightarrow T(\alpha)) \wedge GH(p_\beta \leftrightarrow T(\beta)).$$

Lemma. The length of the equi-satisfiable MRTL formula to 1CMTL formula α of length n is less than 2^{4n}.

VII. Complexity

In this section we consider the complexity of the decision procedure which we have just outlined.

The main work of the procedure is done by the RTL decision procedure which we know from [31] is in PSPACE. This means that (1) there is a Turing machine which can accept formulas of RTL as input, (2) for any formula it halts answering correctly whether the formula is valid in RTL or not, and (3) the space taken for machine to do its job is bound by a polynomial in terms of the size of the input formula.

It is important to be careful about what we mean by the size of the input formula when we consider complexity results for decision procedures for formal languages. In [31] we showed that for RTL we could consider the length of the input to be the same as the number of symbols in the formula. This is despite the fact that we could not use an input alphabet which has a separate symbol for each atom as there are infinitely many atoms.

We will see that things are different for MRTL and 1CMTL. In [32] we define the length of a formula of MRTL to be the number of symbols in the formula (counting repeated appearances) with the integer numbers in the superscripts of $!^{\pm n}$ being in unary notation. That is the length of $!^n$ is $2 + |n|$.

For example, $\psi_3 = !^3 \wedge (((\neg !^3)U!^3) \wedge ((\neg !^3)S!^3))$ has length 31.

The use of unary for the superscripts was justified on the basis that those superscripts will be used to support durations specified in binary as we will see below. In essence, the use of $!^5$, for example, is to encode a duration given as $32 = 2^5$: thus the length of the duration, input as a binary number is 5.

Using that definition of length we get a PSPACE decision procedure for MRTL via the translation into RTL [32].

Now consider 1CMTL.

We would want to assume that the binary rational numbers attached to connectives and specifying durations can be entered in binary. For example, in $pU^{0.5}_{(0,32)}q$, meaning that p will hold until q is true and that happens before a clock with precision 0.5 units reaches 32 units more than what it displays now. Consider the 32 unit duration. It will be convenient for such a measure to be entered in binary. Thus this formula would be input into an algorithm in binary as something like `p U^0.1_(0,100000)q` of length 18 symbols. When we translate such a formula into low-level MRTL there is an exponential expansion into a formula involving levels of ticks from $!^{-1}$ up to $!^5$. So, by writing the 32 in binary using six symbols we have effectively made use of a reference to $!^5$.

We have seen in the last section that a 1CMTL formula of length n translates into an equally-satisfiable formula of MRTL of length 2^n (and using only metric atoms $!^k$ for $-n \leq k \leq n$). Inputting such a formula into the PSPACE decision procedure for MRTL will answer our 1CMTL satisfiability query and the computation space will be bound by some exponential in n:

Lemma. 1CMTL is in EXPSPACE.

VIII. Conclusion

In this paper we have presented a new paradigm for metric temporal logics.

We have introduced the simple metric temporal logic 1CMTL which is based on the idea of referring temporal constraints to the reading on an arbitrarily, but not infinitely precise single universal clock. This contrasts with existing metric temporal logics which require specifications of infinite accuracy, or at least specifications which put infinitely accurate end points on ranges. Existing metric temporal logics also allow specifications which are unrealistic to check because they seem to require the setting off of an infinite number of infinitely accurate stop-watches.

1CMTL is a general metric temporal logic being able to handle arbitrary boolean signals over real-numbers time. There are no finite variability assumptions on the behaviour of the signals.

1CMTL is superficially similar to existing MTL but the semantics are quite different in several ways. Nevertheless, 1CMTL is an expressive language able to specify a wide range of metric temporal constraints. It captures adequately approximately all MTL formulas.

We have proved the decidability of 1CMTL by showing how it it can be translated via the low-level metric MRTL into the non-metric dense-time temporal logic RTL. This gives an EXPSPACE decision procedure. This contrasts with the undecidability of MTL and (more than) matches the complexity of common sub-languages of MTL used for metric specifications.

Future work will concentrate on reasoning tasks. Work is underway [35] on developing tableau techniques for languages like RTL. Hopefully, that can be extended to MRTL and 1CMTL. There are currently no implementations for decision procedures for MTL-like languages.

References

[1] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

[2] Rajeev Alur and Thomas A. Henzinger. Logics and models of real time: A survey. In J. W. de Bakker, Cornelis Huizing, Willem P. de Roever, and Grzegorz Rozenberg, editors, *REX Workshop*, volume 600 of *Lecture Notes in Computer Science*, pages 74–106. Springer, 1991.

[3] Rajeev Alur and Thomas A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[4] R. Alur and T. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–203, 1994.

[5] Domenico Bianculli, Paola Spoletini, Angelo Morzenti, Matteo Pradella, and Pierluigi San Pietro. Model checking temporal metric specifications with trio2promela. In *In FSEN'07*, 2007.

[6] P. Bouyer, F. Chevalier, and N. Markey. On the expressiveness of TPTL and MTL. In *FSTTCS 2005*, pages 432–443. Springer, 2005.

[7] P. Bouyer, N. Markey, J. Ouaknine, J. Worrell. On Expressiveness and Complexity in Real-Time Model Checking. ICALP (2), pages 124–135, 2008.

[8] J. P. Burgess and Y. Gurevich. The decision problem for linear temporal logic. *Notre Dame J. Formal Logic*, 26(2):115–128, 1985.

[9] Zhou Chaochen. Duration calculus, a logical approach to real-time systems. In Armando Martin Haeberer, editor, *AMAST*, volume 1548 of *Lecture Notes in Computer Science*, pages 1–7. Springer, 1998.

[10] J. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88(7):985–1010, 2000.

[11] Deepak D'Souza and Pavithra Prabhakar. On the expressiveness of mtl in the pointwise and continuous semantics. *STTT*, 9(1):1–4, 2007.

[12] E. Falcon, C. Laroche, S. Fauve, and C. Coste. Behaviour of one inelastic ball bouncing repeatedly off the ground. *The European Physical Journal B*, B(3):45–57, 1998.

[13] Carlo A. Furia and Matteo Rossi. On the expressiveness of mtl variants over dense time. In Jean-François Raskin and P. S. Thiagarajan, editors, *FORMATS*, volume 4763 of *LNCS*, pages 163–178. Springer, 2007.

[14] Carlo A. Furia and Matteo Rossi. Mtl with bounded variability: Decidability and complexity. In Franck Cassez and Claude Jard, editors, *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2008.

[15] D. M. Gabbay and I. M. Hodkinson. An axiomatisation of the temporal logic with until and since over the real numbers. *Journal of Logic and Computation*, 1(2):229 – 260, 1990.

[16] V. Gupta, T. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proceedings of the International Workshop on Hybrid and Real-Time Systems (HART)*, volume 1201 of *LNCS*, pages 331–345. Springer, 1997.

[17] T. A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In *the 19th International Colloquium on Automata, Languages, and Prog. (ICALP)*, vol. 623 of *LNCS*, pages 545–558. Springer, 1992.

[18] Yoram Hirshfeld and Alexander Moshe Rabinovich. A framework for decidable metrical logics. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 422–432. Springer, 1999.

[19] Yoram Hirshfeld and Alexander Moshe Rabinovich. Logics for real time: Decidability and complexity. *Fundam. Inform.*, 62(1):1–28, 2004.

[20] Y. Hirshfeld and A. Rabinovich. An expressive temporal logic for real time. *MFCS*, 2006.

[21] Y. Hirshfeld and A. Rabinovich. Expressiveness of metric modalities for continuous time. *Logical Methods in Computer Science*, 3(1), 2007.

[22] J. Huang, J. Voeten and M. Geilen. Real-time property preservation in approximation of timed systems. 1st ACM and IEEE International Conference on Formal Methods and Models for Co-Design, 2003.

[23] H. Kamp. *Tense logic and the theory of linear order*. PhD thesis, University of California, Los Angeles, 1968.

[24] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4):255–299, 1990.

[25] M. Marx, S. Mikulas, and M. Reynolds. The mosaic method for temporal logics. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Proceedings of International Conference, TABLEAUX 2000, Saint Andrews, Scotland, July 2000*, LNAI 1847, pages 324–340. Springer, 2000.

[26] J. S. Ostroff. *Temporal Logic for Real-Time Systems*. Advanced Software Development Series. John Wiley and Sons, 1989.

[27] J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *Proceedings of the 20th Annual Symposium on Logic in Computer Science (LICS'05)*, pages 188–197. IEEE Comp. Soc. Press, July 2005.

[28] Joël Ouaknine and James Worrell. Some recent results in metric temporal logic. In *FORMATS '08: Proceedings of the 6th international conference on Formal Modeling and Analysis of Timed Systems*, pages 1–13, Berlin, Heidelberg, 2008. Springer-Verlag.

[29] J. Raskin. *Logics, Automata and Classical Theories for Deciding Real Time*. PhD thesis, Université de Namur, Belgium, 1999.

[30] M. Reynolds. An axiomatization for Until and Since over the reals without the IRR rule. *Studia Logica*, 51:165–193, May 1992.

[31] M. Reynolds. The complexity of the temporal logic over the reals. *Annals of Pure and Applied Logic*, 161(8):1063–1096, 2010.

[32] M. Reynolds. Metric temporal reasoning with less than 2 clocks. *Journal of Applied Non-classical Logics*, 20:437–455, 2010.

[33] M. Reynolds. Dense time reasoning via mosaics. In *TIME '09*, pages 3–10, Washington, DC, USA, 2009. IEEE Comp. Soc.

[34] Mark Reynolds. A tableau for until and since over linear time. In *TIME 2011*, pages 41–48, 2011.

[35] Mark Reynolds. A tableau for general linear temporal logic. *J of Logic and Computation*, 2013.

[36] Thomas Wilke. Specifying timed state sequences in powerful decidable logics and timed automata (extended abstract). In *LNCS 863*, pages 694–715. Springer-Verlag, 1994.