Time-Bounded Verification*

Joël Ouaknine[1], Alexander Rabinovich[2], and James Worrell[1]

[1] Oxford University Computing Laboratory, UK
{joel,jbw}@comlab.ox.ac.uk
[2] School of Computer Science, Tel Aviv University, Israel
rabinoa@post.tau.ac.il

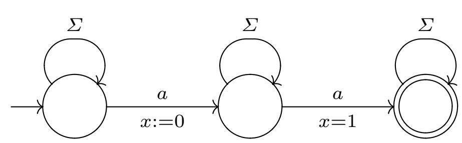
Abstract. We study the decidability and complexity of verification problems for timed automata over time intervals of fixed, bounded length. One of our main results is that time-bounded language inclusion for timed automata is 2EXPSPACE-Complete. We also investigate the satisfiability and model-checking problems for Metric Temporal Logic (MTL), as well as monadic first- and second-order logics over the reals with order and the +1 function (FO($<, +1$) and MSO($<, +1$) respectively). We show that, over bounded time intervals, MTL satisfiability and model checking are EXPSPACE-Complete, whereas these problems are decidable but non-elementary for the predicate logics. Nevertheless, we show that MTL and FO($<, +1$) are equally expressive over bounded intervals, which can be viewed as an extension of Kamp's well-known theorem to metric logics.

It is worth recalling that, over unbounded time intervals, the satisfiability and model-checking problems listed above are all well-known to be undecidable.

1 Introduction

Timed automata were introduced by Alur and Dill in [2] as a natural and versatile model for real-time systems. They have been widely studied ever since, both by practitioners and theoreticians. A celebrated result concerning timed automata, which originally appeared in [1] in a slightly different context, is the PSPACE decidability of the *language emptiness* (or *reachability*) problem.

Unfortunately, the *language inclusion* problem—given two timed automata $\mathcal{A}$ and $\mathcal{B}$, is every timed word accepted by $\mathcal{A}$ also accepted by $\mathcal{B}$?—is known to be undecidable. A related phenomenon is the fact that timed automata are not closed under complementation. For example, the automaton below accepts every timed word in which there are two a-events separated by exactly one time unit.



* This work was partially supported by the ESF Research Networking Programme Games and the UK's EPSRC.

The complement language consists of all timed words in which no two a-events are separated by precisely one time unit. Intuitively, this language is not expressible by a timed automaton, since such an automaton would need an unbounded number of clocks to keep track of the time delay from each a-event. (We refer the reader to [17] for a formal treatment of these considerations.)

The undecidability of language inclusion severely restricts the algorithmic analysis of timed automata, both from a practical and theoretical perspective, as many interesting questions can be phrased in terms of language inclusion. Over the past decade, several researchers have therefore attempted to circumvent this state of affairs by investigating language inclusion, or closely related concepts, under various assumptions and restrictions. Among others, we note the use of (i) topological restrictions and digitisation techniques: [14, 10, 31, 28]; (ii) fuzzy semantics: [13, 15, 30, 7]; (iii) determinisable subclasses of timed automata: [4, 37]; (iv) timed simulation relations and homomorphisms: [42, 26, 23]; and (v) restrictions on the number of clocks: [32, 11].

The undecidability of language inclusion, first established in [2], derives from the undecidability of an even more fundamental problem, that of *universality*: does a given timed automaton accept every timed word? The proof of undecidability of universality in [2] uses in a crucial way the unboundedness of the time domain. Roughly speaking, this allows one to encode arbitrarily long computations of a Turing machine. On the other hand, many verification questions are naturally stated over bounded time domains. For example, a run of a communication protocol might normally be expected to have an *a priori* time bound, even if the total number of messages exchanged is potentially unbounded. Thus numerous researchers have considered the problem of time-bounded verification in the context of real-time systems [39, 8, 22]. This leads us to the question of the decidability of the time-bounded version of the language inclusion problem for timed automata. This problem asks, given timed automata $\mathcal{A}$ and $\mathcal{B}$ and a time bound N, whether all finite timed words of duration at most N that are accepted by $\mathcal{A}$ are also accepted by $\mathcal{B}$. One of the main results of this paper is that the time-bounded language inclusion problem is 2EXPSPACE-Complete. It is worth noting that, since we are working with a dense model of time, time-bounded runs of a given automaton may contain arbitrarily many events. Moreover the restriction to time boundedness does not alter the fact that timed automata are not closed under complement, and hence classical techniques for language inclusion do not trivially apply.

A second line of investigation in this paper concerns the relative expressiveness of monadic second-order and first-order metric logics over the reals. This direction is motivated by the celebrated result of Kamp [21] that Linear Temporal Logic (LTL) has the same expressiveness over the structure $(\mathbb{N}, <)$ as monadic first-order logic ($\mathsf{FO}(<)$). An influential consequence of Kamp's result is that LTL has emerged as the canonical temporal logic over the naturals. While a version of Kamp's result holds over the structure $(\mathbb{R}_{\geq 0}, <)$, the correspondence between predicate logics and temporal logics becomes considerably more complicated with the introduction of *metric* specifications. In practice this has led to a veritable babel of metric temporal logics over the reals [5].

A natural predicate logic in which to formalise metric specifications over the reals is the first-order monadic logic over the structure $(\mathbb{R}_{\geq 0}, <, +1)$. Given a set of uninterpreted monadic predicates $\mathbf{P}$, a model over $(\mathbb{R}_{\geq 0}, <, +1)$ is nothing but a function $f : \mathbb{R}_{\geq 0} \to 2^{\mathbf{P}}$ mapping each $x \in \mathbb{R}_{\geq 0}$ to the set of predicates that hold at x. Such a model is called a *flow* or *signal*, and naturally corresponds to the trajectory of a real-time system.

On the side of temporal logics, an appealing extension of LTL, called Metric Temporal Logic (MTL), was proposed by Koymans [24] almost twenty years ago. While MTL has been widely studied, it is well-known that there are first-order formulas over $(\mathbb{R}_{\geq 0}, <, +1)$ that cannot be expressed in MTL [20].

Our second main result is that, over bounded time domains, MTL has the same expressiveness as monadic first-order logic. Specifically, we show that MTL is as expressive as first-order logic over the structure $([0, N), <, +1)$, for any fixed integer N. Thus, as with language inclusion for timed automata, the restriction to time-boundedness leads to a better-behaved theory.

Finally, we relate automata and logics by showing decidability of the model-checking problem for timed automata against specifications expressed in MTL, first-order, and second-order monadic logics over $([0, N), <, +1)$. We also show decidability of the satisfiability problems for first-order and second-order logics over $([0, N), <, +1)$. In contrast to the case of language inclusion between timed automata, the model-checking and satisfiability problems for monadic predicate logics all have non-elementary complexity, whereas these problems are EXPSPACE-Complete in the case of MTL.

2 Timed Automata

Let X be a finite set of clocks, denoted x, y, z, etc. We define the set Φ_X of clock constraints over X via the following grammar, where $k \in \mathbb{N}$ stands for any non-negative integer, and $\bowtie \in \{=, \neq, <, >, \leq, \geq\}$ is a comparison operator:

$$\phi ::= \mathbf{true} \mid x \bowtie k \mid x - y \bowtie k \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \,.$$

A ***timed automaton*** $\mathcal{A}$ is a six-tuple $(\Sigma, S, S_0, S_F, X, \Delta)$, where

- Σ is a finite set (alphabet) of events,
- S is a finite set of states,
- $S_0 \subseteq S$ is a set of initial states,
- $S_F \subseteq S$ is a set of accepting states,
- X is a finite set of clocks, and
- $\Delta \subseteq S \times S \times \Sigma \times \Phi_X \times 2^X$ is a finite set of transitions. A transition (s, s', a, ϕ, R) allows a jump from state s to s', consuming event $a \in \Sigma$ in the process, provided the constraint ϕ on clocks is met. Afterwards, the clocks in R are reset to zero, while all other clocks remain unchanged.

Given a timed automaton $\mathcal{A}$ as above, a *clock valuation* is a function $\nu : X \to \mathbb{R}_{\geq 0}$. If $t \in \mathbb{R}_{\geq 0}$, we let $\nu + t$ be the clock valuation such that $(\nu + t)(x) = \nu(x) + t$ for all $x \in X$.

A *configuration* of $\mathcal{A}$ is a pair (s, ν), where $s \in S$ is a state and ν is a clock valuation.

An *accepting run* of $\mathcal{A}$ is a finite alternating sequence of configurations and delayed transitions $\pi = (s_0, \nu_0) \xrightarrow{d_1, \theta_1} (s_1, \nu_1) \xrightarrow{d_2, \theta_2} \ldots \xrightarrow{d_n, \theta_n} (s_n, \nu_n)$, where each $d_i \in \mathbb{R}_{>0}$ and each $\theta_i = (s_{i-1}, s_i, a_i, \phi_i, R_i) \in \Delta$, subject to the following conditions:

1. $s_0 \in S_0$, and for all $x \in X$, $\nu_0(x) = 0$.
2. For all $0 \leq i \leq n - 1$, $\nu_i + d_{i+1}$ satisfies ϕ_{i+1}.
3. For all $0 \leq i \leq n - 1$, $\nu_{i+1}(x) = \nu_i(x) + d_{i+1}$ for all $x \in X \setminus R_{i+1}$, and $\nu_{i+1}(x) = 0$ for all $x \in R_{i+1}$.
4. $s_n \in S_F$.

Each d_i is interpreted as the (strictly positive[3]) time delay between the firing of transitions, and each configuration (s_i, ν_i), for $i \geq 1$, records the data immediately following transition θ_i. Abusing notation, we also write runs in the form $(s_0, \nu_0) \xrightarrow{d_1, a_1} (s_1, \nu_1) \xrightarrow{d_2, a_2} \ldots \xrightarrow{d_n, a_n} (s_n, \nu_n)$ to highlight the run's events.

A ***timed word*** is a pair (σ, τ), where $\sigma = \langle a_1 a_2 \ldots a_n \rangle \in \Sigma^*$ is a word and $\tau = \langle t_1 t_2 \ldots t_n \rangle \in (\mathbb{R}_{>0})^*$ is a strictly increasing sequence of real-valued timestamps of the same length.

Such a timed word is *accepted* by $\mathcal{A}$ if $\mathcal{A}$ has some accepting run of the form $\pi = (s_0, \nu_0) \xrightarrow{d_1, a_1} (s_1, \nu_1) \xrightarrow{d_2, a_2} \ldots \xrightarrow{d_n, a_n} (s_n, \nu_n)$ where, for each $1 \leq i \leq n$, $t_i = d_1 + d_2 + \ldots + d_i$.

In this paper, we are mainly concerned with behaviours over time domains of the form $[0, N)$, where $N \in \mathbb{N}$ is a positive integer. Let us in general write $\mathbb{T}$ to denote either $[0, N)$ or $\mathbb{R}_{\geq 0}$. We then define $L_{\mathbb{T}}(\mathcal{A})$ to be the set of timed words accepted by $\mathcal{A}$ all of whose timestamps belong to $\mathbb{T}$.

Remark 1. Our timed automata have transitions that are labelled with instantaneous events; this is by far the most common model found in the literature. Alternatives include automata in which states are labelled with atomic propositions [3], or even mixed models in which states carry atomic propositions and transitions carry events. Other variants allow for silent transitions (invisible events), invariants on states, and combinations thereof. All the results presented in this paper carry over without difficulty to these more expressive models.

Note that we are focussing on *finite* words. Timed automata can be defined to accept infinite words (for example, by using Büchi acceptance conditions [2]), although over bounded time infinite words are automatically *Zeno* (and *ipso*

[3] This gives rise to the *strongly monotonic* semantics for timed automata; in contrast, the *weakly monotonic* semantics allows multiple events to happen 'simultaneously' (or, more precisely, with null-duration delays between them). The main results of this paper remain substantively the same under either semantics, although the weakly monotonic semantics causes some slight complications.

facto ruled out from the accepted language by most researchers). Theorem 4 could nonetheless be extended to such infinite words, if desired.

3 Metric Logics

3.1 Syntax

Let **Var** be a set of *first-order variables*, denoted x, y, z, etc., ranging over non-negative real numbers. Let **MP** be a set of *monadic predicates*, denoted P, Q, R, etc. Monadic predicates will alternately be viewed as second-order variables over $\mathbb{R}_{\geq 0}$, i.e., ranging over sets of non-negative real numbers, and also as atomic propositions holding at various points in time.

Second-order monadic formulas are obtained from the following grammar:

$$\varphi ::= \mathbf{true} \mid x < y \mid +1(x,y) \mid P(x) \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \forall x\, \varphi \mid \forall P\, \varphi\,,$$

where $+1$ is a binary relation symbol, with the intuitive interpretation of $+1(x,y)$ as '$x+1 = y$'.[4] We refer to $\forall x$ and $\forall P$ as *first-order* and *second-order* quantifiers respectively.

The ***monadic second-order metric logic of order***, written $\mathsf{MSO}(<, +1)$, comprises all second-order monadic formulas. Its first-order fragment, the ***(monadic) first-order metric logic of order***, written $\mathsf{FO}(<, +1)$, comprises all $\mathsf{MSO}(<, +1)$ formulas that do not contain any second-order quantifier; note that these formulas are however allowed free monadic predicates.

We also define two further purely order-theoretic sublogics, which are peripheral to our main concerns but necessary to express some key related results. The *monadic second-order logic of order*, $\mathsf{MSO}(<)$, comprises all second-order monadic formulas that do not make use of the $+1$ relation. Likewise, the *(monadic) first-order logic of order*, $\mathsf{FO}(<)$, comprises those $\mathsf{MSO}(<)$ formulas that eschew second-order quantification.

Metric Temporal Logic, abbreviated MTL, comprises the following *temporal formulas*:

$$\theta ::= \mathbf{true} \mid P \mid \theta_1 \wedge \theta_2 \mid \theta_1 \vee \theta_2 \mid \neg\theta \mid \Diamond_I \theta \mid \Box_I \theta \mid \theta_1\, \mathcal{U}_I\, \theta_2\,,$$

where $P \in \mathbf{MP}$ is a monadic predicate, and $I \subseteq \mathbb{R}_{\geq 0}$ is an open, closed, or half-open interval with endpoints in $\mathbb{N} \cup \{\infty\}$. If $I = [0, \infty)$, then we omit the annotation I in the corresponding temporal operator. We also use pseudo-arithmetic expressions to denote intervals. For example, the expression '≥ 1' denotes $[1, \infty)$ and '$=1$' denotes the singleton $\{1\}$.

Note that our version of MTL includes only *forwards* temporal operators, in keeping with the most common definition found in the literature. All our results

[4] The usual approach is of course to define $+1$ as a unary function symbol; this however necessitates an awkward treatment over bounded domains, as considered in this paper. We shall nonetheless abuse notation later on and invoke $+1$ as if it were a function, in the interest of clarity.

extend straightforwardly to variants of MTL that make use of both forwards and backwards operators. It is also worth pointing out that the $\Diamond_I$ and $\Box_I$ operators are derivable from $\mathcal{U}_I$.

Finally, *Linear Temporal Logic*, written LTL, consists of those MTL formulas in which every indexing interval I on temporal operators is $[0, \infty)$ (and hence omitted).

Figure 3.1 pictorially summarises the syntactic inclusions and relative expressive powers of these various logics.

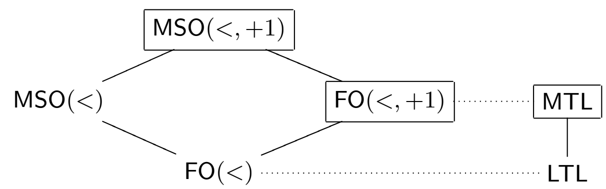


Fig. 1. Relative expressiveness among the various logics. Metric logics are enclosed in boxes. Straight lines denote syntactical inclusion, whereas dotted lines express semantic equivalence over bounded time domains (cf. Section 5).

3.2 Semantics

Let $\mathbf{P} \subseteq \mathbf{MP}$ be a finite set of monadic predicates, and let us again write $\mathbb{T}$ to denote either $[0, N)$ (for some fixed $N \in \mathbb{N}$) or $\mathbb{R}_{\geq 0}$. A ***flow*** (or *signal*) over $\mathbf{P}$ is a function $f : \mathbb{T} \to 2^{\mathbf{P}}$ that is *finitely variable*. Finite variability means that the restriction of f to any finite subinterval of $\mathbb{T}$ has only finitely many discontinuities.[5] Note that we do not place any bound on the variability, other than requiring that it be finite.

Observe that a timed word $(\langle a_1 \ldots a_n \rangle, \langle t_1 \ldots t_n \rangle)$ over alphabet Σ can be viewed as a (particular type of) flow, as follows. Let $\mathbb{T}$ be either $[0, N)$ (for some $N > t_n$) or $\mathbb{R}_{\geq 0}$, and let $\mathbf{P} = \Sigma$. Set $f(t_i) = \{a_i\}$, for $1 \leq i \leq n$, and $f(t) = \emptyset$ for all other values of $t \in \mathbb{T}$.

Fix a time domain $\mathbb{T}$, equipped with the standard order relation $<$ and the obvious binary relation $+1$, i.e., $+1(a, b)$ iff $a, b \in \mathbb{T}$ and $a+1 = b$. Given a formula φ of MSO$(<, +1)$ or one of its sublogics, let $\mathbf{P}$ and $\{x_1, \ldots, x_n\}$ respectively be the sets of free monadic predicates and free first-order variables appearing in φ. For any flow $f : \mathbb{T} \to 2^{\mathbf{P}}$ and real numbers $a_1, \ldots, a_n \in \mathbb{T}$, the satisfaction relation $(f, a_1, \ldots, a_n) \models \varphi$ is defined inductively on the structure of φ in the standard way. We shall particularly be interested in the special case in which φ

[5] The restriction to finitely-variable flows can be partially lifted, as we discuss in the full version of this paper [29]. Note however that from a computer science perspective, infinitely-variable flows do not correspond to feasible computations, hence the widespread adoption of the present restriction in the literature.

is a *sentence*, i.e., a formula with no free first-order variable. In such instances, we simply write the satisfaction relation as $f \models \varphi$.

For θ an MTL or LTL formula, let $\mathbf{P}$ be the set of monadic predicates appearing in θ. Given a flow $f : \mathbb{T} \to 2^{\mathbf{P}}$ and $t \in \mathbb{T}$, the satisfaction relation $(f, t) \models \theta$ is defined inductively on the structure of θ, as follows:

- $(f, t) \models \mathbf{true}$.
- $(f, t) \models P$ iff $P \in f(t)$.
- $(f, t) \models \theta_1 \wedge \theta_2$ iff $(f, t) \models \theta_1$ and $(f, t) \models \theta_2$.
- $(f, t) \models \theta_1 \vee \theta_2$ iff $(f, t) \models \theta_1$ or $(f, t) \models \theta_2$.
- $(f, t) \models \neg\theta$ iff $(f, t) \not\models \theta$.
- $(f, t) \models \Diamond_I \theta$ iff there exists $u \in \mathbb{T}$ with $u > t$, $u - t \in I$, and $(f, u) \models \theta$.
- $(f, t) \models \Box_I \theta$ iff for all $u \in \mathbb{T}$ with $u > t$ and $u - t \in I$, $(f, u) \models \theta$.
- $(f, t) \models \theta_1 \,\mathcal{U}_I\, \theta_2$ iff there exists $u \in \mathbb{T}$ with $u > t$, $u - t \in I$, $(f, u) \models \theta_2$, and for all $v \in (t, u)$, $(f, v) \models \theta_1$.

Finally, we write $f \models \theta$ iff $(f, 0) \models \theta$. This is sometimes referred to as the *initial semantics*.

Note that we have adopted a *strict* semantics, in which the present time t has no influence on the truth values of future temporal subformulas. Strictness is required for Theorem 2, but our other results hold under both the strict and non-strict semantics.

An important point concerning our semantics is that it is *continuous*, rather than *pointwise*: more precisely, the temporal operators quantify over all time points of the domain, as opposed to merely those time points at which a discontinuity occurs. Positive decidability results for satisfiability and model checking of MTL over unbounded time intervals have been obtained in the pointwise semantics [33, 34, 35]; it is worth noting that none of these results hold in the continuous semantics.

4 Satisfiability

The canonical time domain for interpreting the metric logics $\mathsf{MSO}(<, +1)$, $\mathsf{FO}(<, +1)$, and MTL is the non-negative real line $\mathbb{R}_{\geq 0}$. Unfortunately, none of these logics are decidable over $\mathbb{R}_{\geq 0}$ [5, 6, 19].

Our main focus in this paper is therefore on satisfiability over bounded time domains of the form $[0, N)$, for $N \in \mathbb{N}$. For each of the logics introduced in Section 3.1, one can consider the corresponding ***time-bounded satisfiability problem***: given a sentence φ over a set $\mathbf{P}$ of free monadic predicates, together with a time bound $N \in \mathbb{N}$, does there exist a flow $f : [0, N) \to 2^{\mathbf{P}}$ such that $f \models \varphi$?

One of our main results is the following:

Theorem 1. *The time-bounded satisfiability problems for the metric logics* $\mathsf{MSO}(<, +1)$*,* $\mathsf{FO}(<, +1)$*, and* MTL *are all decidable, with the following complexities:*[6]

[6] All the complexity results in this paper assume that the time bound N is provided in binary.

$\mathsf{MSO}(<, +1)$	Non-elementary
$\mathsf{FO}(<, +1)$	Non-elementary
MTL	EXPSPACE-Complete

Remark 2. It is worth noting that if one allows second-order quantification over flows of *arbitrary* variability, then $\mathsf{MSO}(<)$ is undecidable over $\mathbb{R}_{\geq 0}$ [40]. Since any non-trivial interval of the form $[0, N)$ is order-isomorphic to $\mathbb{R}_{\geq 0}$, the same result holds over bounded time domains, and also clearly carries over to $\mathsf{MSO}(<, +1)$. $\mathsf{FO}(<, +1)$ and MTL however remain decidable over bounded time domains regardless of the variability of flows—see [29].

The proofs of all theorems in this paper are given in Appendix A.

5 Expressiveness

Fix a time domain $\mathbb{T}$ to be either $[0, N)$ (for some $N \in \mathbb{N}$) or $\mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ and $\mathcal{J}$ be two logics. We say that $\mathcal{L}$ is ***semantically at least as expressive as*** $\mathcal{J}$ *(with respect to the initial semantics of* $\mathbb{T}$*)* if, for any sentence θ of $\mathcal{J}$, there exists a sentence φ of $\mathcal{L}$ such that θ and φ are satisfied by precisely the same set of flows over $\mathbb{T}$.

Two logics are then said to be ***semantically equally expressive*** *(with respect to the initial semantics of* $\mathbb{T}$*)* if each is at least as expressive as the other.

The following result can be viewed as an extension of Kamp's celebrated theorem [21, 12] to metric logics over bounded time domains:

Theorem 2. *For any fixed bounded time domain of the form* $[0, N)$*, with* $N \in \mathbb{N}$*, the metric logics* $\mathsf{FO}(<, +1)$ *and* MTL *are semantically equally expressive. Moreover, this equivalence is effective.*

Remark 3. Note that semantic expressiveness here is relative to a *single* structure $\mathbb{T}$, rather than to a *class* of structures. In particular, although $\mathsf{FO}(<, +1)$ and MTL are equally expressive over any bounded time domain of the form $[0, N)$, the correspondence and witnessing formulas may very well vary according to the time domain.

It is interesting to note that $\mathsf{FO}(<, +1)$ is strictly more expressive than MTL over $\mathbb{R}_{\geq 0}$ [20]. For example, MTL is incapable of expressing the following formula (in slightly abusive but readable notation)

$$\exists x \, \exists y \, \exists z \, (x < y < z < x + 1 \land P(x) \land P(y) \land P(z))$$

over the non-negative reals. This formula asserts that, sometime in the future, P will hold at three distinct time points within a single time unit.

It is also worth noting that $\mathsf{MSO}(<, +1)$ is strictly more expressive than $\mathsf{FO}(<, +1)$—and hence MTL—over any time domain; see [29].

Finally, we point out that, in contrast to Kamp's theorem [21], but similarly to [12], Theorem 2 does not require backwards temporal operators for MTL (although adding these would be harmless).

6 Model Checking and Language Inclusion

We now turn to questions concerning the time-bounded behaviours of timed automata. Recall from Section 3.2 that timed words over an alphabet Σ can be viewed as flows from a sufficiently large time domain over the set of monadic predicates $\mathbf{P} = \Sigma$. The ***model checking problem*** takes as inputs a timed automaton $\mathcal{A}$ with alphabet Σ, a sentence φ with set of free monadic predicates $\mathbf{P} = \Sigma$, and a time domain $\mathbb{T}$ (taken to be either $[0, N)$, for some $N \in \mathbb{N}$, or $\mathbb{R}_{\geq 0}$). The question is then whether every timed word (flow) in $L_{\mathbb{T}}(\mathcal{A})$ satisfies φ.

Unfortunately, the model checking problem for timed automata and any of the metric logics introduced in this paper is undecidable over the non-negative real line $\mathbb{R}_{\geq 0}$ [5, 6]; this in fact also follows easily from the undecidability of the satisfiability problem for these logics over flows. We therefore focus on the ***time-bounded model checking problem***, in which the time domain is required to be bounded. We have:

Theorem 3. *The time-bounded model-checking problems for timed automata against the metric logics* $\mathsf{MSO}(<, +1)$*,* $\mathsf{FO}(<, +1)$*, and* MTL *are all decidable, with the same complexities as the corresponding time-bounded satisfiability problems (cf. Theorem 1): non-elementary for* $\mathsf{MSO}(<, +1)$ *and* $\mathsf{FO}(<, +1)$*, and EXPSPACE-Complete for* MTL*.*

The ***language inclusion problem*** takes as inputs two timed automata, $\mathcal{A}$ and $\mathcal{B}$, sharing a common alphabet, together with a time domain $\mathbb{T}$ (of the form $[0, N)$, for some $N \in \mathbb{N}$, or $\mathbb{R}_{\geq 0}$). The question is then whether every timed word accepted by $\mathcal{A}$ over $\mathbb{T}$ is also accepted by $\mathcal{B}$.

As for model checking, language inclusion is unfortunately undecidable over $\mathbb{R}_{\geq 0}$ [2]. The ***time-bounded language inclusion problem*** circumvents this by restricting to bounded time domains. This leads us to our final main result, as follows:

Theorem 4. *The time-bounded language inclusion problem for timed automata is decidable and 2EXPSPACE-Complete.*

A Proofs of Theorems

Theorem 1. *The time-bounded satisfiability problems for* $\mathsf{MSO}(<, +1)$ *and* $\mathsf{FO}(<, +1)$ *are decidable and non-elementary, whereas the time-bounded satisfiability problem for* MTL *is EXPSPACE-Complete.*

Proof. Throughout this proof, let $N \in \mathbb{N}$ be fixed.

An easy first observation is that any MTL formula can be translated into an equivalent $\mathsf{FO}(<, +1)$ formula over $[0, N)$. For decidability, it therefore suffices to handle the case of $\mathsf{MSO}(<, +1)$.

Let $\mathbf{P} \subseteq \mathbf{MP}$ be a finite set of monadic predicates. With each $P \in \mathbf{P}$, we associate a collection $P_0, \ldots, P_{N-1}$ of N fresh monadic predicates. We then let $\overline{\mathbf{P}} = \{P_i \mid P \in \mathbf{P}, 0 \leq i \leq N-1\}$.

Intuitively, each monadic predicate P_i represents P over the subinterval $[i, i+1)$. Indeed, there is an obvious bijection (indicated by overlining) between the set of flows $\{f : [0, N) \to 2^{\mathbf{P}}\}$ and the set of flows $\{\overline{f} : [0, 1) \to 2^{\overline{\mathbf{P}}}\}$.

Let φ be an $\mathsf{MSO}(<, +1)$ sentence with set of free monadic predicates $\mathbf{P}$. We will define an $\mathsf{MSO}(<)$ sentence $\overline{\varphi}$ such that, for any flow $f : [0, N) \to 2^{\mathbf{P}}$, $f \models \varphi$ iff $\overline{f} \models \overline{\varphi}$.

We can assume that φ does not contain any (first- or second-order) existential quantifiers, by replacing the latter with combinations of universal quantifiers and negations if need be. It is also convenient to rewrite φ into a formula that makes use of the integer constants $0, 1, \ldots, N$ as well as a family of unary functions $+k$ (for $k \in \mathbb{N}$) instead of the $+1$ relation. To this end, replace every occurrence of $+1(x, y)$ in φ by $(x < N - 1 \land x + 1 = y)$.

Next, replace every instance of $\forall x\, \psi$ in φ by the formula

$$\forall x\, (\psi[x/x] \land \psi[x + 1/x] \land \ldots \land \psi[x + (N-1)/x])\,,$$

where $\psi[t/x]$ denotes the formula resulting from substituting every free occurrence of the variable x in ψ by the term t. Intuitively, this transformation is legitimate since first-order variables in our target formula will range over $[0, 1)$ rather than $[0, N)$.

Having carried out these substitutions, rewrite every term in φ in the form $x + k$, where x is a variable and $k \in \mathbb{N}$ is a non-negative integer constant.

Every inequality occurring in φ is now of the form $x + k_1 < y + k_2$. Replace every such inequality by (i) $x < y$, if $k_1 = k_2$; (ii) **true**, if $k_1 < k_2$; and (iii) $\neg$**true** otherwise.

Every occurrence of a monadic predicate in φ now has the form $P(x + k)$. Replace every such predicate by (i) $P_k(x)$, if $k \leq N-1$, and (ii) $\neg$**true** otherwise.

Finally, replace every occurrence of $\forall P\, \psi$ in φ by $\forall P_0\, \forall P_1\, \ldots \forall P_{N-1}\, \psi$. The resulting formula is the desired $\overline{\varphi}$.

It is now straightforward to prove by induction that the set of $[0, N)$-flows satisfying the original φ are indeed in one-to-one correspondence with the set of $[0, 1)$-flows satisfying $\overline{\varphi}$.

Note that $\overline{\varphi}$ does not use any $+1$ or $+k$ functions or relations, and is therefore indeed a non-metric (i.e., purely order-theoretic) sentence in $\mathsf{MSO}(<)$. Moreover, the satisfiability problem for $\mathsf{MSO}(<)$ by finitely-variable flows over $\mathbb{R}_{\geq 0}$ is decidable [36]. However, although $\mathbb{R}_{\geq 0}$ and $[0, 1)$ are order-isomorphic, finitely-variable flows over $\mathbb{R}_{\geq 0}$ do not necessarily translate to finitely-variable flows over $[0, 1)$. In fact, it is easily seen that the finitely-variable flows over $\mathbb{R}_{\geq 0}$ whose counterparts in $[0, 1)$ are finitely-variable are precisely those flows that are ultimately constant: every monadic predicate P is such that there is a time point beyond which P is either always true or always false.

For P a monadic predicate, let us therefore write σ_P to denote the first-order sentence

$$\exists x \, (\forall y \, (x < y \rightarrow P(y)) \vee \forall y \, (x < y \rightarrow \neg P(y))) \, .$$

We now perform one last transformation on $\overline{\varphi}$, by replacing every instance of $\forall P_i \, \psi$ in $\overline{\varphi}$ by $\forall P_i(\sigma_{P_i} \rightarrow \psi)$, yielding a new sentence $\overline{\varphi}'$. Recall that $\overline{\mathbf{P}}$ is the set of monadic predicates appearing freely in $\overline{\varphi}$ (and hence also $\overline{\varphi}'$). We now have that our original sentence φ is satisfiable by finitely-variable flows over $[0, N)$ iff the $\mathsf{MSO}(<)$ sentence

$$\overline{\varphi}' \wedge \bigwedge \{\sigma_{P_i} \mid P_i \in \overline{\mathbf{P}}\}$$

is satisfiable by finitely-variable flows over $\mathbb{R}_{\geq 0}$, which is decidable.

We now tackle the complexity. Satisfiability of $\mathsf{FO}(<)$ over $\mathbb{R}_{\geq 0}$ is known to be non-elementary [41, 27]. An examination of the proof shows that this result in fact also holds over (finitely-variable) flows that are ultimately constant. Since $\mathbb{R}_{\geq 0}$ is order-isomorphic to $[0, N)$, $\mathsf{FO}(<)$, and *a fortiori* $\mathsf{FO}(<, +1)$ and $\mathsf{MSO}(<, +1)$, also have non-elementary time-bounded satisfiability problems.

Let θ now be an MTL formula. It would be possible to show, following an approach somewhat similar to the reduction above of $\mathsf{MSO}(<, +1)$ over $[0, N)$ to $\mathsf{MSO}(<)$, that one can manufacture an equi-satisfiable but exponentially larger formula of *Linear Temporal Logic with past operators* ($\mathsf{LTL{+}Past}$). Since $\mathsf{LTL{+}Past}$ satisfiability is PSPACE-Complete, satisfiability for θ over $[0, N)$ can be decided in exponential space.

In the interest of brevity, we shall however proceed differently, and translate instead the satisfiability problem for θ over $[0, N)$ to that of a related *Flat Metric Temporal Logic* ($\mathsf{Flat\text{-}MTL}$) formula over $\mathbb{R}_{\geq 0}$. The result will follow by the known EXPSPACE-Completeness of satisfiability for $\mathsf{Flat\text{-}MTL}$ [9].

Note first that we can assume that all intervals I appearing as subscripts to temporal operators in θ are subsets of $[0, N)$—indeed, if this is not the case, replace any offending interval I by $I \cap [0, N)$ to obtain a semantically equivalent formula over $[0, N)$. θ can therefore be taken to be a *Bounded Metric Temporal Logic* ($\mathsf{Bounded\text{-}MTL}$) formula.

Next, let us postulate a fresh predicate T which will be required to hold precisely within the time domain $[0, N)$. We now modify θ by relativising all temporal operators to quantify over the 'absolute' time domain $[0, N)$, as follows: (i) replace every subformula in θ of the form $\Diamond_I \rho$ by $\Diamond_I (T \wedge \rho)$; (ii) replace every subformula in θ of the form $\Box_I \rho$ by $\Box_I (T \rightarrow \rho)$; and (iii) replace every subformula in θ of the form $\rho_1 \, \mathcal{U}_I \, \rho_2$ by $\rho_1 \, \mathcal{U}_I \, (T \wedge \rho_2)$. Let us call the resulting formula θ'.

Consider now the formula

$$\theta' \wedge \Box_{[0,N)} T \wedge \Box_{[N,\infty)} \neg T$$

which belongs to $\mathsf{Flat\text{-}MTL}$ (cf. [9]) since θ' is in $\mathsf{Bounded\text{-}MTL}$. This formula is clearly satisfiable by finitely-variable flows over $\mathbb{R}_{\geq 0}$ iff θ is satisfiable by finitely-variable flows over $[0, N)$. Since the former can be decided in EXPSPACE, then so can the latter.

Finally, the EXPSPACE-Hardness proof of Bounded-MTL satisfiability in [9] easily carries over to MTL satisfiability over $[0, N)$. This therefore establishes EXPSPACE-Completeness of time-bounded satisfiability for MTL. □

Before proving Theorem 2, we first state a preliminary lemma, easily proven by induction.

Lemma 1. *Let φ be an FO($<$) or LTL formula with set of monadic predicates $\mathbf{P}$. Assume that φ is only satisfiable by ultimately-constant flows. Let β be any bijection from $[0, 1)$ to $\mathbb{R}_{\geq 0}$. Then β extends to a bijection between $[0, 1)$-flows and $\mathbb{R}_{\geq 0}$-flows that are ultimately constant. Moreover, this bijection preserves and reflect the flows that satisfy φ; in other words, for any flow $f : [0, 1) \to 2^{\mathbf{P}}$, $f \models \varphi$ iff $\beta(f) \models \varphi$.*

Theorem 2. *For any fixed bounded time domain of the form $[0, N)$, with $N \in \mathbb{N}$, the metric logics FO($<, +1$) and MTL are semantically equally expressive. Moreover, this equivalence is effective.*

Proof. Throughout this proof, let $N \in \mathbb{N}$ be fixed.

The reduction from MTL to FO($<, +1$) is straightforward, and therefore omitted.

For the other direction, let φ be an FO($<, +1$) sentence with set of free monadic predicates $\mathbf{P} \subseteq \mathbf{MP}$. As in the proof of Theorem 1, let $\overline{\mathbf{P}} = \{P_i \mid P \in \mathbf{P}, 0 \leq i \leq N-1\}$ be a set of fresh monadic predicates. The construction used in Theorem 1 yields an FO($<$) sentence $\overline{\varphi}$ with set of free monadic predicates $\overline{\mathbf{P}}$, such that there is a bijection (indicated by overlining) from the set of $[0, N)$-flows over $\mathbf{P}$ satisfying φ to the set of $[0, 1)$-flows over $\overline{\mathbf{P}}$ satisfying $\overline{\varphi}$. Moreover, we can ensure that, when interpreted over $\mathbb{R}_{\geq 0}$, $\overline{\varphi}$ is only satisfied by flows that are ultimately constant.

According to [18], one can now construct an LTL formula ψ, with set of monadic predicates $\overline{\mathbf{P}}$, that defines precisely the same set of finitely-variable $\mathbb{R}_{\geq 0}$-flows as $\overline{\varphi}$. By Lemma 1, $\overline{\varphi}$ and ψ therefore also define precisely the same set of finitely-variable $[0, 1)$-flows over $\overline{\mathbf{P}}$.

It therefore suffices to exhibit an MTL formula θ, over set of monadic predicates $\mathbf{P}$, such that, for any flow $f : [0, N) \to 2^{\mathbf{P}}$, $f \models \theta$ iff $\overline{f} \models \psi$.

To this end, write ι to denote the MTL formula $\Diamond_{=(N-1)}\mathbf{true}$. Note that, when interpreted within the time domain $[0, N)$, ι holds precisely over the time interval $[0, 1)$. Perform the following substitutions on ψ to obtain the desired θ: (i) for each $P \in \mathbf{P}$, replace every occurrence of P_0 in ψ by P, and every occurrence of P_i in ψ (for $i \geq 1$) by $\Diamond_{=i} P$; (ii) replace every occurrence of $\Diamond\gamma$ in ψ by $\Diamond(\iota \wedge \gamma)$; (iii) replace every occurrence of $\Box\gamma$ in ψ by $\Box(\iota \to \gamma)$; (iv) replace every occurrence of $\gamma_1 \,\mathcal{U}\, \gamma_2$ in ψ by $\gamma_1 \,\mathcal{U}\, (\iota \wedge \gamma_2)$.

Finally, show by induction on ψ that, for any flow $f : [0, N) \to 2^{\mathbf{P}}$ and any $t \in [0, 1)$, one has $(f, t) \models \theta$ iff $(\overline{f}, t) \models \psi$. The desired result follows by setting $t = 0$. □

Theorem 3. *The time-bounded model-checking problems for timed automata against the metric logics MSO($<, +1$), FO($<, +1$), and MTL are all decidable,* *with the same complexities as the corresponding time-bounded satisfiability problems: non-elementary for* $\mathsf{MSO}(<,+1)$ *and* $\mathsf{FO}(<,+1)$*, and EXPSPACE-Complete for* MTL*.*

Proof. Fix $N \in \mathbb{N}$, and let $\mathcal{A}$ be a timed automaton over alphabet Σ. In [16], it is shown how to construct (in polynomial time) an MTL formula $\theta_{\mathcal{A}}$, over a potentially larger set of monadic predicates $\mathbf{P} \supseteq \Sigma$, such that, for any flow $f : [0, N) \to 2^{\Sigma}$, $f \in L_{[0,N)}(\mathcal{A})$ iff there exists a flow $g : [0, N) \to 2^{\mathbf{P}}$ such that $g \models \theta_{\mathcal{A}}$ and $g\restriction_{\Sigma} = f$. Intuitively, the extra monadic predicates of $\theta_{\mathcal{A}}$ keep track of the (otherwise invisible) identity of transitions and clock resets that occur during runs of $\mathcal{A}$.

Of course, $\theta_{\mathcal{A}}$ can clearly instead be taken to be an $\mathsf{FO}(<,+1)$ or $\mathsf{MSO}(<,+1)$ formula, if desired. In all cases, given a metric formula φ, the model-checking problem for $\mathcal{A}$ and φ over $[0, N)$ boils down to whether $\theta_{\mathcal{A}} \wedge \neg\varphi$ is unsatisfiable over $[0, N)$ or not.

This shows that time-bounded model checking reduces to time-bounded satisfiability. For the converse, simply pick an automaton $\mathcal{A}$ that accepts every flow. □

Theorem 4. *The time-bounded language inclusion problem for timed automata is decidable and 2EXPSPACE-Complete.*

Proof. Fix $N \in \mathbb{N}$, and let $\mathcal{A}$ and $\mathcal{B}$ be timed automata over alphabet Σ. We give a procedure for deciding whether $L_{[0,N)}(\mathcal{A}) \subseteq L_{[0,N)}(\mathcal{B})$.

As in the proof of Theorem 3, let $\theta_{\mathcal{A}}$ be an MTL formula over set of monadic predicates $\mathbf{P} = \Sigma \cup \mathbf{U}$, with the property that each $[0, N)$-timed word over Σ accepted by $\mathcal{A}$ can be extended to a $[0, N)$-flow over $\mathbf{P}$ satisfying $\theta_{\mathcal{A}}$, and vice-versa. Likewise, let $\theta_{\mathcal{B}}$ be a similar MTL formula over set of monadic predicates $\mathbf{Q} = \Sigma \cup \mathbf{V}$ for the timed automaton $\mathcal{B}$. We assume that Σ, $\mathbf{U}$, and $\mathbf{V}$ are all pairwise disjoint.

Abusing notation, we see that $L_{[0,N)}(\mathcal{A}) \subseteq L_{[0,N)}(\mathcal{B})$ iff the following formula holds over $[0, N)$:

$$\forall\Sigma\,\forall\mathbf{U}\,\exists\mathbf{V}\,(\neg\theta_{\mathcal{A}}(\Sigma,\mathbf{U}) \vee \theta_{\mathcal{B}}(\Sigma,\mathbf{V}))\,. \qquad (1)$$

Observe, as argued in the proof of Theorem 1, that $\theta_{\mathcal{A}}$ and $\theta_{\mathcal{B}}$ can in fact be taken to be $\mathsf{Bounded\text{-}MTL}$ formulas. We can therefore invoke a result of [9] and transform $\neg\theta_{\mathcal{A}} \vee \theta_{\mathcal{B}}$ into an equivalent but exponentially larger formula ψ of $\mathsf{LTL{+}Past}$. More precisely, ψ has a different (and exponentially larger) set of monadic predicates $\mathbf{R} = \overline{\Sigma} \cup \overline{\mathbf{U}} \cup \overline{\mathbf{V}} \cup \mathbf{W}$, yet there is a one-to-one correspondence between the flows satisfying $\neg\theta_{\mathcal{A}} \vee \theta_{\mathcal{B}}$ and those satisfying ψ. We also adjust the outside quantifiers accordingly to transform Formula (1) into the equivalent formula

$$\forall\overline{\Sigma}\,\forall\overline{\mathbf{U}}\,\exists\overline{\mathbf{V}}\,\exists\mathbf{W}\psi(\overline{\Sigma},\overline{\mathbf{U}},\overline{\mathbf{V}},\mathbf{W})\,.$$

Next, following [25, 38], we transform ψ into an equivalent untimed finite-state automaton $\mathcal{C}$ whose transitions are labelled by subsets of $\overline{\Sigma} \cup \overline{\mathbf{U}} \cup \overline{\mathbf{V}} \cup \mathbf{W}$. This incurs a second exponential blowup.

Note that the existential quantifications $\exists \mathbf{W}$ and $\exists \overline{\mathbf{V}}$ simply correspond to relabelling all $\mathbf{W}$- and $\overline{\mathbf{V}}$-labelled transitions in $\mathcal{C}$; this can be carried out in polynomial time. We are therefore asking whether the resulting automaton is universal over $\overline{\Sigma} \cup \overline{\mathbf{U}}$, i.e., accepts any string over this alphabet. Since universality is decidable in PSPACE, the overall procedure can be carried out in doubly-exponential space.

The proof of 2EXPSPACE-Hardness is fairly intricate and will appear in the full version of this paper [29]. $\square$

References

[1] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. In *Proceedings of LICS*. IEEE Computer Society Press, 1990.
[2] R. Alur and D. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126, 1994.
[3] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1), 1996.
[4] R. Alur, L. Fix, and T. A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211, 1999.
[5] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In *REX Workshop*, volume 600 of *Lecture Notes in Computer Science*. Springer, 1991.
[6] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. and Comput.*, 104(1), 1993.
[7] R. Alur, S. La Torre, and P. Madhusudan. Perturbed timed automata. In *Proceedings of HSCC*, volume 3414. Springer LNCS, 2005.
[8] C. Baier, H. Hermanns, J.-P. Katoen, and B. R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theor. Comput. Sci.*, 345(1), 2005.
[9] P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell. On expressiveness and complexity in real-time model checking. In *Proceedings of ICALP*, volume 5126 of *Lecture Notes in Computer Science*. Springer, 2008.
[10] D. Bošnački. Digitization of timed automata. In *Proceedings of FMICS*, 1999.
[11] M. Emmi and R. Majumdar. Decision problems for the verification of real-time software. In *Proceedings of HSCC*, volume 3927. Springer LNCS, 2006.
[12] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal basis of fairness. In *Proceedings of POPL*. ACM Press, 1980.
[13] V. Gupta, T. A. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proceedings of HART*, volume 1201. Springer LNCS, 1997.
[14] T. A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In *Proceedings of ICALP*, volume 623. Springer LNCS, 1992.
[15] T. A. Henzinger and J.-F. Raskin. Robust undecidability of timed and hybrid systems. In *Proceedings of HSCC*, volume 1790. Springer LNCS, 2000.
[16] T. A. Henzinger, J.-F. Raskin, and P.-Y. Schobbens. The regular real-time languages. In *Proceedings of ICALP*, volume 1443 of *Lecture Notes in Computer Science*. Springer, 1998.
[17] P. Herrmann. Timed automata and recognizability. *Inf. Process. Lett.*, 65, 1998.
[18] Y. Hirshfeld and A. Rabinovich. Future temporal logic needs infinitely many modalities. *Inf. Comput.*, 187(2), 2003.

[19] Y. Hirshfeld and A. Rabinovich. Logics for real time: Decidability and complexity. *Fundam. Inform.*, 62(1), 2004.

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

[21] H. Kamp. Tense logic and the theory of linear order. *Ph.D. Thesis*, 1968.

[22] J.-P. Katoen and I. S. Zapreev. Safe on-the-fly steady-state detection for time-bounded reachability. In *Proceedings of QEST*. IEEE Computer Society, 2006.

[23] D. K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager. Timed I/O Automata: A mathematical framework for modeling and analyzing real-time systems. In *Proceedings of RTSS*. IEEE Computer Society Press, 2003.

[24] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4), 1990.

[25] C. Lutz, D. Walther, and F. Wolter. Quantitative temporal logics over the reals: PSpace and below. *Inf. and Comput.*, 205, 2007.

[26] N. A. Lynch and H. Attiya. Using mappings to prove timing properties. *Distributed Computing*, 6(2), 1992.

[27] A. R. Meyer. Weak monadic second-order theory of successor is not elementary-recursive. In *Logic colloquium '72–73*, volume 453 of *LNM*. Springer, 1975.

[28] J. Ouaknine. Digitisation and full abstraction for dense-time model checking. In *Proceedings of TACAS*, volume 2280. Springer LNCS, 2002.

[29] J. Ouaknine, A. Rabinovich, and J. Worrell. Time-bounded verification (full version). In preparation, 2009.

[30] J. Ouaknine and J. Worrell. Revisiting digitization, robustness, and decidability for timed automata. In *Proceedings of LICS*. IEEE Computer Society Press, 2003.

[31] J. Ouaknine and J. Worrell. Universality and language inclusion for open and closed timed automata. In *Proceedings of HSCC*, volume 2623. Springer LNCS, 2003.

[32] J. Ouaknine and J. Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *Proceedings of LICS*. IEEE Computer Society Press, 2004.

[33] J. Ouaknine and J. Worrell. On the decidability of Metric Temporal Logic. In *Proceedings of LICS*. IEEE Computer Society Press, 2005.

[34] J. Ouaknine and J. Worrell. Safety Metric Temporal Logic is fully decidable. In *Proceedings of TACAS*, volume 3920. Springer LNCS, 2006.

[35] J. Ouaknine and J. Worrell. On the decidability and complexity of Metric Temporal Logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.

[36] A. Rabinovich. Finite variability interpretation of monadic logic of order. *Theor. Comput. Sci.*, 275(1-2), 2002.

[37] J.-F. Raskin. *Logics, Automata and Classical Theories for Deciding Real Time*. PhD thesis, University of Namur, 1999.

[38] M. Reynolds. The complexity of temporal logic over the reals. Submitted, 2004.

[39] Olivier Roux and Vlad Rusu. Verifying time-bounded properties for ELECTRE reactive programs with stopwatch automata. In *Hybrid Systems*, volume 999 of *Lecture Notes in Computer Science*. Springer, 1994.

[40] S. Shelah. The monadic theory of order. *Ann. Math.*, 102, 1975.

[41] L. J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, MIT, 1974.

[42] S. Taşiran, R. Alur, R. P. Kurshan, and R. K. Brayton. Verifying abstractions of timed systems. In *Proceedings of CONCUR 96*, volume 1119. Springer LNCS, 1996.