On Expressiveness and Complexity in Real-Time Model Checking

Patricia Bouyer[1,2], Nicolas Markey[2], Joël Ouaknine[1], and James Worrell[1]

[1] Oxford University Computing Laboratory
{First.Last}@comlab.ox.ac.uk
[2] Laboratoire Spécification & Vérification
{First.Last}@lsv.ens-cachan.fr

Abstract. Metric Interval Temporal Logic (MITL) is a popular formalism for expressing real-time specifications. This logic achieves decidability by restricting the precision of timing constraints, in particular, by banning so-called *punctual* specifications. In this paper we introduce a significantly more expressive logic that can express a wide variety of punctual specifications, but whose model-checking problem has the same complexity as that of MITL. We conclude that for model checking the most commonly occurring specifications, such as invariance and bounded response, punctuality can be accommodated at no cost.

1 Introduction

One of the most successful approaches to verification is *model checking*: given a representation S of a system together with a specification φ, determine whether S satisfies φ. In the world of real time, a prominent modelling framework is to use timed automata to represent systems and Metric Temporal Logic (MTL) as the specification formalism.

MTL was proposed nearly twenty years ago by Koymans [12] and has since been extensively studied. MTL is an extension of Linear Temporal Logic (LTL) which allows one to specify a wide range of timed behaviours. The formula $\Box(p \rightarrow \Diamond_{\{1\}} q)$, for example, asserts that whenever the system finds itself in a p-state, then it will be in a q-state precisely one time unit later.

Unfortunately, the model-checking and satisfiability problems for MTL over dense time are undecidable [3,16]. In fact, it was widely held until quite recently that any formalism in which 'punctual' (exact) timing constraints could be expressed would automatically be undecidable—see [3,4,9], among others. The formula given in the previous paragraph is a typical example of a punctual specification.

Many researchers were thus led to consider relaxations and variations of the original MTL formalism in search of decidability and tractability. The identification of *Metric Interval Temporal Logic* (MITL) as a decidable fragment of MTL is a classic result in real-time verification. MITL consists of those formulas in which every constraining interval is non-singular. This syntactic restriction directly removes the problem of punctuality, but correspondingly loses considerable expressiveness. Satisfiability and model checking for MITL were shown to be EXPSPACE-complete in [2] via a translation of formulas into equivalent timed automata; see also [11,15].

L. Aceto et al. (Eds.): ICALP 2008, Part II, LNCS 5126, pp. 124–135, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The starting point of this paper is to identify a new decidable fragment of MTL, which we call Bounded-MTL. This is the subset of MTL in which the constraining intervals appearing in any formula have finite length. For instance $\Box_{[0,25)}(p \rightarrow \Diamond_{\{1\}} q)$ is a Bounded-MTL formula. Note that, unlike in MITL, punctual formulas are permitted. We show that Bounded-MTL is decidable in EXPSPACE if the time constraints in formulas are encoded in binary, and in PSPACE if time constraints are encoded in unary. Notwithstanding these bounds, we provide examples of Bounded-MTL formulas that can only be satisfied by signals whose variability is doubly exponential in the size of the formula. Moreover we observe that there exist Bounded-MTL formulas for which there is no equivalent timed automata, unlike the situation for MITL formulas.

Bounded-MTL shows that, at least in the time-bounded setting, punctuality need not be fatal for the complexity of model checking. However the restriction to time-bounded modalities in Bounded-MTL is severe, for example prohibiting the expression of basic safety properties such as invariance. This leads us to isolate the notion of *flatness*, which generalises boundedness. We introduce coFlat-MTL, a natural extension of both MITL and Bounded-MTL, which is closed under the *always* operator $\Box$ and the bounded until operator U_I. In particular, if φ is a Bounded-MTL formula, expressing some time-bounded property, then the invariance specification $\Box\varphi$ is in coFlat-MTL.

Our main result is that the model checking problem for coFlat-MTL on timed automata is EXPSPACE-complete, that is, in the same complexity class as MITL model checking. This substantiates the main thesis of this paper—that in model checking the most common specifications, including invariance and bounded response, punctuality can be accommodated for free. However we note that coFlat-MTL is not closed under negation, and its satisfiability problem is undecidable. In this respect coFlat-MTL is similar to the branching-time logic TCTL for which model checking is PSPACE-complete but satisfiability is undecidable (again due to the problem of punctuality).

This paper adopts the standard semantics for MTL in which a model of a formula is a *signal*: a function from the positive reals into a finite set, indicating which propositions hold at every instant in time. An alternative approach, used in our earlier work [6], is the so-called *point-based* semantics, which represents models as countable sequences of timestamped snapshots. The signal semantics can be shown to generalise the pointwise semantics. To accommodate this extra generality we had to move from the automata-based proof techniques used in [6] to model-theoretic ones. As a side benefit, this shift has allowed us to lift our previous restriction to finitely-variable models. Finally, the logic which we term coFlat-MTL in the present paper strictly generalises the logic by the same name in [6]; in particular, MITL is now a fragment of coFlat-MTL, so that our results also extend the original EXPSPACE model checking of MITL [2].

2 Metric Temporal Logic

Given a set P of atomic propositions, the formulas of MTL are built from P using Boolean connectives, and time-constrained versions of the *until* operator U as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, U_I \, \varphi \,,$$

where $I \subseteq (0, \infty)$ is an interval of reals with endpoints in $\mathbb{N} \cup \{\infty\}$. We sometimes abbreviate $U_{(0,\infty)}$ to U, calling this the *unconstrained* until operator. We assume a dag

representation of formulas, and define the size of a formula φ, denoted $|\varphi|$, to be the number of distinct subformulas of φ. We also write M_φ for the maximum finite integer occurring as an endpoint of a constraining interval in φ.

We denote by $\mathbb{R}_+$ the set of nonnegative real numbers. Given a set X, a *signal* is a function $f\colon \mathbb{R}_+ \to X$. We say that f has *finite variability* if its set of discontinuities has no limit points. We say that f has *variability* $n \in \mathbb{N}$ if it has at most n discontinuities in any open unit-length subinterval $(k, k+1)$ of its domain, where $k \in \mathbb{N}$. Given an MTL formula φ over the set of propositional variables P, and a signal $f\colon \mathbb{R}_+ \to 2^P$, the satisfaction relation $f \models \varphi$ is defined inductively, with the classical rules for atomic propositions and Boolean operators, and with the following rule for the "until" modality, where f^t denotes the signal $f^t(s) = f(t+s)$:

$$f \models \varphi_1\ U_I\ \varphi_2 \text{ iff for some } t \in I,\ f^t \models \varphi_2 \text{ and } f^u \models \varphi_1 \text{ for all } u \in (0,t).$$

Note that we adopt a *strict* semantics for U_I, in which the judgement $f \models \varphi_1\ U_I\ \varphi_2$ is independent of $f(0)$ (recall that $0 \notin I$ by assumption). In the following we write $\varphi_1\ U\ \varphi_2$ for $\varphi_1\ U_{(0,\infty)}\ \varphi_2$.

In general we do not assume that signals are finitely variable. Indeed there are formulas that are satisfiable only by *infinitely* variable signals: *e.g.* $\neg(p\ U\ p) \wedge \neg(\neg p\ U\ \neg p)$.

Further connectives can be defined following the usual conventions. In addition to propositions $\top$ (true) and $\bot$ (false), and to disjunction $\vee$, we have the *constrained eventually* operator $\Diamond_I \varphi \equiv \top\ U_I\ \varphi$, the *constrained always* operator $\Box_I \varphi \equiv \neg\Diamond_I \neg\varphi$, and the *constrained dual until* operator $\varphi_1\ \widetilde{U}_I\ \varphi_2 \equiv \neg((\neg\varphi_1)\ U_I\ (\neg\varphi_2))$.

Admitting only $\widetilde{U}_I$ as an extra connective one can transform any MTL formula into an equivalent *negation normal form*, in which negation is only applied to propositional variables.

3 Decidable Sublogics

It is well known that both model checking and satisfiability for MTL are highly undecidable (Σ_1^1-complete) [2]. Here we consider syntactic restrictions yielding sublogics with decidable model checking problem.

One approach, due to Alur, Feder and Henzinger [2], involves placing restrictions on *punctuality*. We say that a formula φ is *punctual* if its outermost connective is a temporal modality with a singular constraining interval, e.g., $\Diamond_{\{1\}} p$. Intuitively, a punctual formula specifies an exact timing constraint. *Metric Interval Temporal Logic* (MITL) is the subset of MTL in which all constraining intervals are non-singular, that is, in which punctual formulas are banned. The satisfiability and model checking problems for MITL are EXPSPACE-complete.

In this paper our starting point is, in some sense, dual to that of [2]. Rather than ban constraining intervals that are *too small*, we ban constraining intervals that are *too big*. We define Bounded-MTL to be the subset of MTL in which all constraining intervals have finite length, and we show that the satisfiability and model checking problems for Bounded-MTL are EXPSPACE-complete (or PSPACE-complete if timing constraints are encoded in unary), matching the complexity of MITL. However the following example illustrates the fundamentally different character of MITL and Bounded-MTL.

Example 1. Consider the Bounded-MTL formula $\varphi \equiv \Box_{(0,1)}(p \leftrightarrow \Diamond_{\{1\}} p)$. A variation on a well-known result tells us that the set of signals satisfying φ is not realisable as the language of a timed automaton [1,5]. Therefore φ defines a property that is not expressible in MITL since MITL formulas can be transformed into equivalent timed automata [2].

MITL and Bounded-MTL represent two different approaches to defining decidable metric temporal logics, and they have incomparable expressive power. In particular, Bounded-MTL is not capable of expressing invariance—one of the most basic safety specifications. To repair this deficiency we introduce *flatness* as a generalisation of boundedness. Our use of this term is motivated by similarities with logics introduced in [7,8].

We say that an MTL formula in negation normal form is *flat* if *(i)* in any subformula of the form $\varphi_1 \; U_I \; \varphi_2$, either I is bounded or φ_1 is in MITL, and *(ii)* in any subformula of the form $\varphi_1 \; \widetilde{U}_I \; \varphi_2$, either I is bounded or φ_2 is in MITL. For example $\Box\varphi \equiv \bot \; \widetilde{U} \; \varphi$ is flat if φ is in MITL. The intuition behind flatness is that *potentially persistent* subformulas must be in MITL. We write Flat-MTL for the fragment of MTL composed of all flat formulas.

Flatness is a key technical notion in this paper, however our main results are most naturally understood in terms of the dual notion, *coflatness*. A formula is coflat if it is the negation of a flat formula. More explicitly we say that a formula is *coflat* if *(i)* in any subformula of the form $\varphi_1 \; U_I \; \varphi_2$, either I is bounded or φ_2 is in MITL, and *(ii)* in any subformula of the form $\varphi_1 \; \widetilde{U}_I \; \varphi_2$, either I is bounded or φ_1 is in MITL. If we write coFlat-MTL for the sublogic of coflat formulas then coFlat-MTL includes both Bounded-MTL and MITL, is closed under $\Box_I$ for arbitrary I (invariance), and is closed under U_I for bounded I (bounded liveness). Thus, for specifications, coflatness is a much less restrictive property than flatness. While this generality renders the satisfiability problem for coFlat-MTL undecidable (the undecidability proof of [2] for the satisfiability of MTL makes use only of formulas in coFlat-MTL), we show that the model checking problem is no harder than for MITL.

Example 2. The formula $(\Box\Diamond_{(0,1)} \bigvee_{m\in M} in_m) \wedge (\Box \bigwedge_{m\in M} (in_m \rightarrow \Diamond_{\{1\}} out_m))$ specifies the behaviour of a perfect buffer which processes each message in one time unit, operating in an environment where at least one message arrives every time unit. This formula is in coFlat-MTL, but is not in Bounded-MTL (due to the unconstrained $\Box$) and is not in MITL (due to the punctual $\Diamond_{\{1\}}$).

Model Checking. The model checking problem for coFlat-MTL asks, given a timed automaton $\mathcal{A}$ and a coFlat-MTL formula φ, whether all (finitely variable) signals accepted by $\mathcal{A}$ also satisfy φ. Rather than formally introducing timed automata we rely on a result of [10,17] that for each timed automaton $\mathcal{A}$ there is an MITL formula $\varphi_{\mathcal{A}}$, of size polynomial in $\mathcal{A}$, such that the language of $\mathcal{A}$ is a projection of the language of $\varphi_{\mathcal{A}}$. Since Flat-MTL subsumes MITL, using this result we can reduce the model checking problem for coFlat-MTL to the satisfiability problem for the dual logic Flat-MTL. The main result of this paper is that the latter problem is EXPSPACE-complete, as is the same problem for MITL [2].

Theorem 1. *The model-checking problem for* coFlat-MTL *is EXPSPACE-complete.*

The proof of Theorem 1 occupies Sections 5 and 6. The decision procedure involves a satisfiability-respecting translation of Flat-MTL into Linear Temporal Logic over the reals. In this translation the non-punctual connectives in Flat-MTL are handled using similar techniques to [11]. Dealing with the punctual connectives, however, requires completely new ideas.

4 Hardness

Proposition 1. *The satisfiability problem for* Bounded-MTL *is EXPSPACE-hard.*

Proof. Given a 2^n-space-bounded Turing machine $\mathcal{M}$ with input X, we construct in logarithmic space a Bounded-MTL formula $\varphi_{\mathcal{M},X}$ that is satisfiable if and only if $\mathcal{M}$ accepts X. This reduction bears some similarities with the undecidability proof for MTL [2], but it also differs in important respects. Indeed, directly applying the latter proof to Bounded-MTL would only yield EXPTIME-hardness.

We now sketch the main ideas behind the definition of $\varphi_{\mathcal{M},X}$. Suppose that $\mathcal{M}$ has set of control states S and tape alphabet Σ. The set of atomic propositions used by $\varphi_{\mathcal{M},X}$ is $P \cup \dot{P}$, where $P = \{p_\sigma, p_{\sigma,s} : \sigma \in \Sigma, s \in S\}$ and $\dot{P} = \{\dot{p} : p \in P\}$. Intuitively, proposition p_σ represents a tape cell that currently contains σ, whereas $p_{\sigma,s}$ represents a tape cell that currently contains σ and is pointed to by the head of $\mathcal{M}$, while $\mathcal{M}$ is in control state s. The dot is used as a pointer to aid in simulating $\mathcal{M}$: an entire computation of $\mathcal{M}$ is encoded in each time unit, and each step of the computation is checked using the distinguished dotted propositions in two consecutive unit intervals.

$\varphi_{\mathcal{M},X}$ is written as the conjunction of three components

$$\varphi_{\mathcal{M},X} \equiv \varphi_{UNIQUE} \wedge \varphi_{COPY} \wedge \varphi_{CHECK} \,.$$

The formula φ_{UNIQUE}, which is straightforward to formalise, ensures that any signal satisfying $\varphi_{\mathcal{M},X}$ defines a left-continuous function $f\colon [0, 2^n] \to P \cup \dot{P}$, that is, only one proposition holds at each moment, and propositions do not hold instantaneously.

The purpose of φ_{COPY} and φ_{CHECK} is to ensure that in any signal satisfying $\varphi_{\mathcal{M},X}$ the sequence of propositions holding in the time interval $[0, 1)$ encodes the computation history of $\mathcal{M}$ on X. Within this, the job of φ_{COPY} is to copy the sequence of propositions holding in each unit-duration time interval into the subsequent time interval, at the same time moving the dot superscript 'one place to the right'. Formally we have

$$\begin{aligned}\varphi_{COPY} = \bigwedge_{p \in P} \Box_{[0,2^n]}(p \to \Diamond_{\{1\}}(p \vee \dot{p})) \\ \wedge \bigwedge_{p,q \in P} \Box_{[0,2^n]}((\dot{p}\, U_{(0,1)}\, q) \leftrightarrow \Diamond_{\{1\}}(p\, U_{(0,1)}\, \dot{q}))\,,\end{aligned}$$

where $\Box_{[0,2^n]}\psi$ is a shorthand for $\psi \wedge \Box_{(0,2^n]}\psi$.

Thus the sequence of propositions holding in each subsequent time interval $[k, k+1)$, $k = 1, \ldots, n-1$, should also represent the computation history of $\mathcal{M}$ on X. The only

difference is that in the interval $[k, k+1)$ the dot should decorate exactly those propositions encoding the contents of the k-th tape cell in each configuration in the computation history.

The role of φ_{CHECK} is to verify that the sequence of propositions holding in each subsequent unit-length interval does indeed encode the computation history of $\mathcal{M}$ on X. As it 'reads' the segment of the input signal defined over the time interval $[k, k+1)$, φ_{CHECK} uses the dots as pointers to check the correctness of the k-th tape cell in each configuration. Thus, in 2^n time units the whole computation is checked. We omit the details of φ_{CHECK}, but point out that it is equivalent to an LTL formula. (In fact each modality is decorated with the constraining interval $(0, 2^n)$ merely to ensure that φ_{CHECK} is in Bounded-MTL). □

The proof of Proposition 1 assumes that constants are encoded in binary (in order to concisely write $\Box_{[0,2^n]}$). It can be proved that model checking Bounded-MTL drops to PSPACE when constants are encoded in unary. However, for the more expressive logic Flat-MTL we can adapt the above encoding to show EXPSPACE-hardness assuming only unary encoding of constants. Thus Theorem 1 holds irrespective of whether constants are encoded in unary or binary.

5 Closure Labellings

It is well-known that the constrained until and dual-until operators U_I and $\widetilde{U}_I$ can be expressed in terms of the unconstrained operators U and $\widetilde{U}$ and the unary operators $\Box_I$ and $\Diamond_I$ [11,15]. Unfortunately, adopting this simplification makes it impossible to express flatness as a syntactic property, hence we prefer to retain a bit more flexibility in our basic syntax. To this end we say that an MTL formula is in *constraint normal form* if it is generated by the grammar

$$\varphi ::= p \mid \neg p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \, U_I \, \varphi_2 \mid \varphi_1 \, \widetilde{U}_I \, \varphi_2 \mid \Diamond_J \varphi \mid \Box_J \varphi \,,$$

where I is a left-open, initial (*i.e.*, with left end-point 0) interval, while J is arbitrary.

Any MTL formula can be transformed into an equivalent constraint normal form using equivalences such as

$$\varphi_1 \, U_{(\ell,r]} \, \varphi_2 \quad \leftrightarrow \quad \Box_{(0,\ell]}(\varphi_1 \, U_{(0,r]} \, \varphi_2) \, \wedge \, \Diamond_{(\ell,r]} \varphi_2 \,.$$

This transformation is linear with respect to the DAG-size of formulas and it preserves both MITL and Flat-MTL. Henceforth, without loss of generality, we assume that all formulas are in constraint normal form.

Given $I \subseteq \mathbb{R}_+$ and $n \in \mathbb{N}$, write $I - n = \{x \in (0, \infty) : x + n \in I\}$. Define the closure $cl(\varphi)$ of a formula φ to be the smallest set such that the following hold (where we adopt the identifications $\Box_\emptyset \varphi \equiv \top$ and $\Diamond_\emptyset \varphi \equiv \bot$).

C1. $cl(\varphi)$ contains all subformulas of φ
C2. $\varphi_1 \, U_I \, \varphi_2 \in cl(\varphi)$ implies $\varphi_1 \, U \, \varphi_2, \Diamond_I \varphi_2 \in cl(\varphi)$
C3. $\varphi_1 \, \widetilde{U}_I \, \varphi_2 \in cl(\varphi)$ implies $\varphi_1 \, \widetilde{U} \, \varphi_2, \Box_I \varphi_2 \in cl(\varphi)$

C4. $\Box_J\varphi_1 \in cl(\varphi)$ implies $\Box_{J-1}\varphi_1 \in cl(\varphi)$
C5. $\Diamond_J\varphi_1 \in cl(\varphi)$ implies $\Diamond_{J-1}\varphi_1 \in cl(\varphi)$.

For example, $cl(\Box_{(1,\infty)}p \wedge \Diamond_{\{1\}}q) = \{\bot, p, q, \Diamond_{\{1\}}q, \Box_{(1,\infty)}p, \Box_{(0,\infty)}p, \Box_{(1,\infty)}p \wedge \Diamond_{\{1\}}q\}$.

It is straightforward to verify that $cl(\varphi)$ has cardinality $O(|\varphi| \cdot M_\varphi)$. We note also that if $\varphi \in \mathsf{MITL}$, then $cl(\varphi) \subseteq \mathsf{MITL}$; in particular, the interval $I - 1$ is a singleton only if I is a singleton.

Given an MTL formula φ in constraint normal form, we define a *closure labelling* to be a signal $f\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ such that Rules CL1–CL10 below are satisfied for all $s \in \mathbb{R}_+$. Closure labellings are continuous-time counterparts of Hintikka sequences [19]. Here we denote by P the set of propositions mentioned in φ. We also assume that Rules CL5 and CL6 apply to $\Diamond\varphi_1$ and $\Box\varphi_1$, respectively, under the identifications $\Diamond\varphi_1 \equiv \top\ U\ \varphi_1$ and $\Box\varphi_1 \equiv \bot\ \widetilde{U}\ \varphi_1$.

CL1. $\bot \notin f(s)$;
CL2. exactly one of p and $\neg p$ lies in $f(s)$ for any $p \in P$;
CL3. $\varphi_1 \wedge \varphi_2 \in f(s)$ implies $\varphi_1 \in f(s)$ and $\varphi_2 \in f(s)$;
CL4. $\varphi_1 \vee \varphi_2 \in f(s)$ implies $\varphi_1 \in f(s)$ or $\varphi_2 \in f(s)$;
CL5. $\varphi_1\ U\ \varphi_2 \in f(s)$ implies there exists $t > s$ such that $\varphi_2 \in f(t)$ and $\varphi_1\ U\ \varphi_2$, $\varphi_1 \in f(u)$ for all $u \in (s,t)$;
CL6. $\varphi_1\ \widetilde{U}\ \varphi_2 \in f(s)$ implies for all $t > s$, if $\varphi_2 \notin f(t)$ then there exists $u \in (s,t)$ with $\varphi_1 \in f(u)$, and if $\varphi_1\ \widetilde{U}\ \varphi_2 \notin f(t)$ then there exists $u \in (s,t]$ with $\varphi_1 \in f(u)$;
CL7. $\varphi_1\ U_I\ \varphi_2 \in f(s)$ implies $\varphi_1\ U\ \varphi_2 \in f(s)$ and $\Diamond_I\varphi_2 \in f(s)$;
CL8. $\varphi_1\ \widetilde{U}_I\ \varphi_2 \in f(s)$ implies $\varphi_1\ \widetilde{U}\ \varphi_2 \in f(s)$ or $\Box_I\varphi_2 \in f(s)$;
CL9. $\Box_J\varphi_1 \in f(s)$ implies $\Box_{J-1}\varphi_1 \in f(s{+}1)$ and $\varphi_1 \in f(s{+}\delta)$ for all $\delta \in (0,1] \cap J$;
CL10. $\Diamond_J\varphi_1 \in f(s)$ implies $\Diamond_{J-1}\varphi_1 \in f(s+1)$ unless $\varphi_1 \in f(s+\delta)$ for some $\delta \in (0,1] \cap J$.

Rules CL1–CL10 encode the semantics of MTL in a natural way. However it is worth noting though that constrained until U_I and dual until $\widetilde{U}_I$ are handled indirectly, via Rules CL7 and CL8. Note also that the correctness of CL7 and CL8 depends on the assumption that the interval I appearing in these rules is initial, which holds because φ is in constraint normal form.

The following straightforward proposition expresses the expected property of closure labellings.

Proposition 2. *A* Flat-MTL *formula φ is satisfiable iff there is a closure labelling $g\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ with $\varphi \in g(0)$.*

5.1 The Partition Lemma

Next we identify some structure on the closure labellings of Flat-MTL formulas. To this end, say that $E \subseteq \mathbb{R}_+$ is a *basic set* if it can be written as a finite union of compact intervals with integer end-points: $E = E_1 \cup E_2 \cup \cdots \cup E_n$. We define $length(E)$ in the obvious manner as the sum of the lengths of the E_i. Given a basic set $E \subseteq \mathbb{R}_+$,

we say that a closure labelling g is *E-rigid* if $g(t)$ contains punctual formulas[1] only when $t \in E$. The term *rigid* anticipates the development in Section 6.2.

The following result crucially relies on the flatness of φ:

Lemma 1 (Partition Lemma). *Let φ be a* Flat-MTL *formula and $g\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ a closure labelling with $\varphi \in g(0)$. Then there is a basic set E with $length(E) \leqslant M_\varphi \cdot 2^{|\varphi|}$ and an E-rigid closure labelling h with $\varphi \in h(0)$.*

Remark 1. In case $\varphi \in$ Bounded-MTL, the Partition Lemma can be strengthened by requiring that $length(E) \leqslant M_\varphi \cdot |\varphi|$. This makes our algorithm be in PSPACE for Bounded-MTL with unary-encoded integers.

Given a closure labelling $f\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$, the *non-punctual part* of f is the function $f_{np} \subseteq f$, where $f_{np}(t)$ consists of the set of formulas in $f(t)$ of the form $\Box_I\varphi_1$ or $\Diamond_I\varphi_1$ with I non-singular.

Consider a signal $f\colon \mathbb{R}_+ \to 2^P$, and assume that $t_1 < t_2 < t_3 < t_1 + 1$, and that f^{t_1} and f^{t_3} both satisfy $\Box_I\psi$ with I non-singular; then it is easily shown that f^{t_2} also satisfies $\Box_I\psi$. Thus $\Box_I\psi$ changes its truth value at most 3 times in any unit interval. By duality, it also holds that $\Diamond_I\psi$ also changes its truth value at most 3 times in any unit interval. Following this line of reasoning we can assume in Proposition 2 that g_{np} has variability at most $3 \cdot M_\varphi \cdot |\varphi|$. Moreover the construction underlying the proof of the Partition Lemma is such that the only discontinuities in h, other than those in g, are integer-valued. In summary we have:

Proposition 3. *A* Flat-MTL *formula φ is satisfiable if, and only if, there is a basic set E with $length(E) \leq M_\varphi \cdot 2^{|\varphi|}$ and an E-rigid closure labelling $h\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ such that $\varphi \in h(0)$ and h_{np} has variability at most $3 \cdot M_\varphi \cdot |\varphi|$.*

6 The Decision Procedure

In this section we describe an EXPSPACE decision procedure for the Flat-MTL satisfiability problem. As explained in Section 3, this implies that the model checking problem for coFlat-MTL is also in EXPSPACE. To achieve this we utilise a technique, inspired by [11], to give a translation of Flat-MTL into LTL+Past[2] that respects the satisfiability of formulas.

6.1 Tableaux

The rules CL9 and CL10 in Section 5 treat punctual and non-punctual connectives alike. We now introduce a modified notion of closure labelling, called a *tableau*, in which punctual and non-punctual connectives are handled differently. To motivate the definition of a tableau, consider the following 'stacking' construction on a closure labelling $g\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$. Given an integer $k \geq 1$, define $T\colon \mathbb{R}_+ \to \left(2^{cl(\varphi)}\right)^k$

[1] We recall (see Section 3) that a formula is *punctual* if its outermost connective is a temporal modality with a singular constraining interval.

[2] LTL+Past is the classical extension of LTL with past-time modalities [13].

by $T(t) = \langle g(t), g(t+1), \ldots, g(t+k)\rangle$. We can think of T as a multi-track closure labelling in which the i-th track is the function $T_i\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ defined by $T_i(t) = T(t)_i$. Notice that the $(i+1)$-th track is one time unit ahead of the i-th track. Motivated by this construction, we axiomatise the notion of a *tableau* below.

Given an integer $k \geq 1$, we say that a signal $T\colon \mathbb{R}_+ \to \left(2^{cl(\varphi)}\right)^k$ is a tableau if the following rules are satisfied for each $0 \leq i \leq k-1$ and $s \in \mathbb{R}_+$:

TH1. T_i satisfies the closure labelling axioms CL1–CL8;

TH2. T_i satisfies the versions of CL9 and CL10 in which the constraining interval J is non-singular;

TV1. If $0 \leq i < k-1$ then $\Box_J \psi \in T(s)_i$ implies $\Box_{J-1}\psi \in T(s)_{i+1}$, $\psi \in T_i(s+\delta)$ for all $\delta \in (0,1] \cap J$ such that $s+\delta \leq \lceil s \rceil$, and $\psi \in T_{i+1}(s+\delta-1)$ for all $\delta \in (0,1] \cap J$ such that $\lfloor s \rfloor \leq s+\delta-1$;

TV2. If $0 \leq i < k-1$ then $\Diamond_J \psi \in T(s)_i$ implies that either $\Diamond_{J-1}\psi \in T(s)_{i+1}$, or there exists $\delta \in (0,1] \cap J$ such that either $s+\delta \leq \lceil s \rceil$ and $\psi \in T(s+\delta)_i$, or $s+\delta-1 \geq \lfloor s \rfloor$ and $\psi \in T(s+\delta-1)_{i+1}$;

TV3. if $0 \leq i < k-1$ then for each $n \in \mathbb{N}$ such that $n > 0$ we have $T(n)_i = T(n-1)_{i+1}$.

We think of TH1 and TH2 as *horizontal* rules, since they concern individual tracks of T. They say that each track T_i would be a closure labelling but for the fact that rules CL9 and CL10 need only hold for non-punctual connectives.

Next come the *vertical* rules TV1–TV3, which relate points on different tracks of T. TV1 and TV2 are vertical counterparts of CL9 and CL10 for punctual and non-punctual formulas. Note how TV3 reflects the intuition that $T_{i+1}(s) = T_i(s+1)$.

Since TH2 does not apply to punctual connectives, the tableau axioms do not accurately capture the semantics of arbitrary MTL formulas. However, for Flat-MTL formulas the notion of rigidity, defined in Section 5, comes to the rescue. Intuitively in a tableau of an Flat-MTL formula we rely on the vertical rules to handle punctual subformulas and we rely on the horizontal rules otherwise. Since the number of tracks of a tableau is finite, the correctness of this idea depends on the existence of bounds on the parts of the tableau containing punctual subformulas. To this end we first extend the notion of rigidity to tableaux by saying that a tableau T is E-rigid iff each track T_i is E_i-rigid, where $E_i = \{t : t+i \in E\}$. Then we have the following result:

Proposition 4. *Given a basic set E and $k \geq$ length(E), there is an E-rigid closure labelling $g\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ with $\varphi \in g(0)$ if, and only if, there is an E-rigid tableau $T\colon \mathbb{R}_+ \to \left(2^{cl(\varphi)}\right)^k$ with $\varphi \in T_0(0)$.*

Proof (sketch). If $g\colon \mathbb{R}_+ \to 2^{cl(\varphi)}$ is an E-rigid closure labelling with $\varphi \in g(0)$, then $T\colon \mathbb{R}_+ \to \left(2^{cl(\varphi)}\right)^k$ defined by $T(t) = \langle g(t), g(t+1), \ldots, g(t+k)\rangle$ is an E-rigid tableau with $\varphi \in T_0(0)$. Indeed, only rules TV1–TV3 need to be checked, and TV1 (resp. TV2) is simply a consequence of CL9 (resp. CL10). The rule TV3 is satisfied by construction.

Conversely, given an E-rigid tableau T we construct a closure labelling g by splicing together unit-length segments from different tracks of T. The idea is that if a given segment contains a punctual formula then it is concatenated with the segment immediately

below on the next track; otherwise it is concatenated with its right neighbour on the same track. More precisely, define $\sigma\colon \mathbb{N} \to \{0, \ldots, k-1\}$ by $\sigma(0) = 0$ and

$$\sigma(n+1) = \begin{cases} \sigma(n)+1 & \text{if } [n, n+1] \subseteq E \\ \sigma(n) & \text{otherwise.} \end{cases}$$

Then $g(t) = T(t - \sigma(\lfloor t \rfloor))_{\sigma(\lfloor t \rfloor)}$ is an E-rigid closure labelling ($k \geq \mathit{length}(E)$). □

Combining Lemma 1 and Proposition 4, we obtain the following result.

Corollary 1. *A* Flat-MTL *formula φ is satisfiable if, and only if, there is a basic set E with $\mathit{length}(E) \leq M_\varphi \cdot 2^{|\varphi|}$ and an E-rigid tableau $T\colon \mathbb{R}_+ \to \left(2^{cl(\varphi)}\right)^k$ with $\varphi \in T_0(0)$ and $k = \mathit{length}(E)$. Moreover we can assume that T_{np} has variability $3 \cdot M_\varphi^2 \cdot |\varphi| \cdot 2^{|\varphi|}$.*

6.2 The Stretching Lemma

Say that two signals $f, g\colon \mathbb{R}_+ \to X$ are *stretching equivalent*, denoted $f \sim g$, if there is a monotone bijection $h\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that $g = f \circ h$. In this case it is easy to see that f and g satisfy the same LTL+Past properties. Our translation from Flat-MTL to LTL+Past relies on an observation of [11] that simple metric properties can be specified in LTL+Past up to stretching equivalence.

Given an integer N, and set of atomic propositions $\Delta_N = \{d_j, d'_j : 1 \leq j \leq N\} \cup \{p_\checkmark\}$, let θ_N be an LTL formula enforcing the following properties: *(i)* the propositions d_j, d'_j and $p_\checkmark$ all hold punctually; *(ii)* the propositions d_j are mutually exclusive and the d'_j are also mutually exclusive; *(iii)* $p_\checkmark$ holds at time 0 and thereafter holds infinitely often; *(iv)* in between each occurrence of $p_\checkmark$ each d_j holds exactly once, and the d_j hold in the order $d_1, d_2, \ldots, d_N$ (and similarly for the d'_j). We omit the formal definition of θ_N, which is straightforward. Lemma 2, below, states that a signal f that satisfies *(i)–(iv)* can be stretched into one in which $p_\checkmark$ holds precisely at integer time-points and every time d_j holds then d'_j holds one time unit later. The proof uses a construction from [11, Lemma 10].

Lemma 2 (Stretching Lemma). *If $f \models \theta_N$ then there exists a signal $g \sim f$ such that $g^t \models p_\checkmark$ iff $t \in \mathbb{N}$, and $g^{t+1} \models d'_j$ whenever $g^t \models d_j$.*

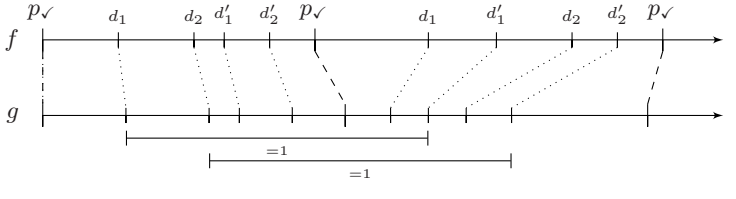

Fig. 1. The stretching lemma

6.3 Translation to LTL+Past

Given a Flat-MTL formula φ we define an LTL+Past formula φ° such that φ is satisfiable iff φ° is satisfiable. The idea is that φ° encodes the tableau rules for φ and the E-rigidity condition. To this end, φ° uses the set of propositions $Q = \{p_{\psi,i} : \psi \in cl(\varphi), 0 \leq i \leq k-1\}$, where k, which represents the height of the tableau, will be chosen later. Then given a signal $f\colon \mathbb{R}_+ \to 2^Q$, the stretching equivalent signal g given by Lemma 2 naturally encodes a function $T\colon \mathbb{R}_+ \to 2^{(cl(\varphi))^k}$ by $T_i(t) = \{\psi : p_{\psi,i} \in g(t)\}$. The definition of φ° is such that $f \models \varphi^\circ$ iff $g \models \varphi^\circ$ iff T is a tableau for φ.

Most of the tableau rules can be straightforwardly encoded in φ°. For example, TH1 is captured by formulas such as $\Box(p_{\varphi_1 \wedge \varphi_2,i} \rightarrow (p_{\varphi_1,i} \wedge p_{\varphi_2,i}))$ and $\Box(p_{\varphi_1 U \varphi_2,i} \rightarrow p_{\varphi_1,i}\ U\ p_{\varphi_2,i})$; corresponding to Rule TV1 we have formulas such as $\Box(p_{\Box_I \psi,i} \rightarrow p_{\Box_{I-1}\psi,i+1})$.

The most interesting part of the translation concerns the rule TH2: the horizontal rule dealing with the constrained and non-punctual connectives $\Box_J$ and $\Diamond_J$. To help encode this, we choose a suitable constant N (more on this later) and include formula θ_N, from Section 6.2, as a conjunct of φ°. We furthermore specify in φ° that propositions of the form $p_{\Diamond_I \psi,i}$ and $p_{\Box_I \psi,i}$ with I non-singular only change truth value when one of the d_j holds (think of the d_j as marking discontinuities in T_{np}). Now consider some signal that satisfies φ°; let s_j^n denote the time-point of the n-th occurrence of d_j and let t_j^n denote the time-point of the n-th occurrence of d'_j. By Lemma 2 we can assume without loss of generality that $t_j^n = s_j^n + 1$. But now it is easy to encode TH2. For instance, by referring to the propositions d_j and d'_j we can specify in φ° that if $p_{\Box_I \psi,i}$ holds in an interval (s_j^n, s_{j+1}^n) then $p_{\Box_{I-1}\psi,i}$ holds in the interval $(t_j^{n+1}, t_{j+1}^{n+1})$.

It only remains to choose the constants k and N. The choice should be such that if φ is satisfiable then there should exist a tableau T with k tracks such that T_{np} has variability at most N. But then Corollary 1 shows that we can take $k = M_\varphi \cdot 2^{|\varphi|}$ and $N = 3 \cdot M_\varphi^2 \cdot |\varphi| \cdot 2^{|\varphi|}$. Note that since k and N are both exponential in the size of the description of φ, formula φ° may be exponentially bigger than φ. The correctness of the construction is stated below.

Theorem 2. *Let φ be a* Flat-MTL *formula, and φ° be the* LTL+Past *formula defined above. Then, φ is satisfiable iff φ° is satisfiable.*

In summary, we have a satisfiability-respecting exponential translation from Flat-MTL to LTL+Past. Now it is known that the satisfiability problem for LTL+Past over $\mathbb{R}_+$ is PSPACE-complete [18,14] and we conclude that the satisfiability problem for Flat-MTL and the model checking problem for coFlat-MTL are both in EXPSPACE. As a final remark we observe that, due to the factor $2^{|\varphi|}$ in the expressions for N and k, the exponential blow-up in the translation from Flat-MTL to LTL+Past arises even if the timing constraints in formulas are encoded in unary (as mentioned at the end of Section 4, this exponential blow-up is unavoidable).

Remark 2. In Remark 1 we noticed that the length of the basic set for Bounded-MTL formulas can be bounded by $M_\varphi \cdot |\varphi|$ when constants are encoded in unary. In that case, we can take $k = M_\varphi \cdot |\varphi|$ and $N = 3 \cdot M_\varphi^2 \cdot |\varphi|^2$, and the size of the LTL+Past formula is now polynomial. Hence, under the hypothesis that constants are encoded in unary, the satisfiability and model checking problems for Bounded-MTL are in PSPACE.

References

1. Alur, R., Dill, D.: A theory of timed automata. TCS 126(2), 183–235 (1994)
2. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. of the ACM 43(1), 116–146 (1996)
3. Alur, R., Henzinger, T.A.: Logics and models of real time: A survey. In: Real-Time: Theory in Practice, Proc. REX Workshop 1991. LNCS, vol. 600, pp. 74–106. Springer, Heidelberg (1992)
4. Alur, R., Henzinger, T.A.: Real-time logics: Complexity and expressiveness. Inf. & Comp. 104(1), 35–77 (1993)
5. Alur, R., Madhusudan, P.: Decision problems for timed automata: A survey. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 1–24. Springer, Heidelberg (2004)
6. Bouyer, P., Markey, N., Ouaknine, J., Worrell, J.: The cost of punctuality. In: Proc. 22nd Ann. IEEE Symp. Logic in Computer Science (LICS 2007), pp. 109–118. IEEE, Los Alamitos (2007)
7. Comon, H., Cortier, V.: Flatness is not a weakness. In: Clote, P.G., Schwichtenberg, H. (eds.) CSL 2000. LNCS, vol. 1862, pp. 262–276. Springer, Heidelberg (2000)
8. Demri, S., Lazić, R., Nowak, D.: On the freeze quantifier in constraint LTL: Decidability and complexity. Inf. & Comp. 205(1), 2–24 (2007)
9. Henzinger, T.A.: It's about time: Real-time logics reviewed. In: Sangiorgi, D., de Simone, R. (eds.) CONCUR 1998. LNCS, vol. 1466, pp. 439–454. Springer, Heidelberg (1998)
10. Henzinger, T.A., Raskin, J.-F., Schobbens, P.-Y.: The regular real-time languages. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 580–591. Springer, Heidelberg (1998)
11. Hirshfeld, Y., Rabinovich, A.: Timer formulas and decidable metric temporal logic. Inf. & Comp. 198(2), 148–178 (2005)
12. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems 2(4), 255–299 (1990)
13. Lichtenstein, O., Pnueli, A., Zuck, L.D.: The glory of the past. In: Proc. Conference on Logics of Programs. LNCS, vol. 193, pp. 413–424. Springer, Heidelberg (1985)
14. Lutz, C., Walther, D., Wolter, F.: Quantitative temporal logics over the reals: PSPACE and below. Inf. & Comp. 205(1), 99–123 (2007)
15. Maler, O., Nickovic, D., Pnueli, A.: From MITL to timed automata. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 274–289. Springer, Heidelberg (2006)
16. Ouaknine, J., Worrell, J.: On metric temporal logic and faulty Turing machines. In: Aceto, L., Ingólfsdóttir, A. (eds.) FOSSACS 2006 and ETAPS 2006. LNCS, vol. 3921, pp. 217–230. Springer, Heidelberg (2006)
17. Raskin, J.-F.: Logics, Automata and Classical Theories for Deciding Real-Time. PhD thesis, Université de Namur, Belgium (1999)
18. Reynolds, M.: The complexity of the temporal logic over the reals (submitted 2004)
19. Wolper, P.: Constructing automata from temporal logic formulas: A tutorial. In: European Educational Forum: School on Formal Methods and Performance Analysis. LNCS, vol. 2090, pp. 261–277. Springer, Heidelberg (2000)