Flattening Metric Temporal Logic

Deepak D'Souza[1], Raj Mohan M[1], and Pavithra Prabhakar[2]

[1] Dept. of Computer Science & Automation
Indian Institute of Science, Bangalore 560012, India.
{deepakd,raj}@csa.iisc.ernet.in
[2] Dept. of Computer Science
University of Illinois at Urbana-Champaign, USA.
pprabha2@cs.uiuc.edu

Abstract. We show that MTL can be *flattened* in the sense that for any formula in the logic we can construct a satisfiability equivalent formula which has no nested distance subformulas. This holds for both the pointwise and continuous semantics and for past operators and modalities. We also show that the S ("since") modality can be eliminated using U and some new propositions. Finally, we show that in the continuous semantics we can eliminate S_I subformulas using the $\Diamond_I$ operator, the U modality, and some new propositions.

1 Introduction

Metric Temporal Logic (MTL) introduced by Koymans [Koy90] is a popular logic for specifying quantitative timing properties of real-time systems. It is interpreted over timed behaviours and extends the "until" modality of LTL [P77] with an interval index, allowing formulas of the form $\varphi U_I \psi$ which say that ψ is satisfied in the future at a timepoint whose distance lies in the interval I, and φ is held on at all time points in between. The modality S_I expresses the symmetric property in the past. What is considered a possible timepoint to assert a formula depends on whether one considers the "pointwise" or "continuous" interpretation of the logic. In the pointwise semantics we are allowed to make assertions only at time points corresponding to event occurrences, while in the continuous semantics we are allowed to make assertions at arbitrary timepoints. In general, the continuous semantics is strictly more expressive than the pointwise semantics [BCM05,PD06].

In this paper we show some results about equivalence of various fragments of MTL in terms of satisfiability-preserving translations. As a basic stepping stone we first show that the formulas of MTL can be "flattened" in the sense that for any formula in the logic we can construct a satisfiability-equivalent formula which has no occurrences of nested U_I formulas (though it does make use of the untimed U modality). In fact, the only subformulas involving the U_I modality are of the form pU_Iq, where p and q are propositions. We call this the "flat" or "non-recursive" version of MTL. A similar claim is true for formulas of the logic extended with the S or S_I modalities: we can give a satisfiability-equivalent formula in which the only S or S_I subformulas are of the form pSq

or pS_Iq. The idea we use is quite simple: to flatten U_I formulas for example, we introduce new propositions p_0 and p_1 for each subformula of the form $\varphi U_I \psi$, replace each occurrence of $\varphi U_I \psi$ by $p_0 U_I p_1$, and add formulas which ensure that p_0 and p_1 correctly capture the truth of φ and ψ along the model. As a simple illustrative example, the flattened form of the formula $(pUq)S_{(0,1)}(p \wedge (qUr))$ is $(p_0 S_{(0,1)} p_1) \wedge \Box(p_0 \Leftrightarrow (pUq)) \wedge \Box(p_1 \Leftrightarrow (p \wedge (qUr)))$ (here $\Box\varphi$ stands for $\neg(\top U \neg\varphi)$). This result is shown for both the pointwise and continuous versions of the logic.

To point out an immediate application of this result, we recall that the pointwise version of MTL over infinite models was shown to be undecidable [OW06] via a reduction from channel systems to the general (recursive) version of MTL. From our result above it now follows that the corresponding non-recursive fragment of the logic is also undecidable.

Next we show that the S modality can be eliminated from MTL formulas (in both the pointwise and continuous semantics) using only the U modality and introducing some new propositions. The technique is similar: we first flatten the S subformulas, replace each pSq by a new proposition r, and add formulas which force r to reflect the truth of pSq along the model. The situation is somewhat more complicated in the continuous case due to the lack of the notion of a "next" time point, but the essential idea is the same.

The implication of the above result is that adding the past modality S to a decidable variant of MTL, cannot lead to undecidability. Thus the logics MITL^c (continuous MTL in which only non-singular intervals are allowed) over both finite and infinite words, and MTL^{pw} (pointwise MTL) over finite words, which were shown to be decidable in [AFH96] and [OW06] respectively, remain decidable even when we add the S modality.

Finally, using similar techniques, we show that in continuous time we can eliminate S_I distance formulas using U and the distance operator $\Diamond_I$ (which is the same as $\top U_I-$). This implies that adding S_I formulas to a decidable variant of MTL in the continuous semantics, cannot lead to undecidability. We note that our technique of eliminating S_I does not work for $\mathrm{MITL}^c_{S_I}$ since our translation may introduce non-singular intervals. Further, such an elimination result cannot hold for the pointwise semantics, since it is known that introducing S_I in pointwise MTL over finite words leads to undecidability [DM05].

We should point out here that our translations only preserve the equivalence of models up to projection onto the original set of propositions (and hence satisfiability), and not expressiveness in general. In fact, MTL^c with S, for example, is known to be strictly more expressive than MTL^c [BCM05,PD06] over a given set of propositions.

In related work, the well-known work of [GPSS80] shows how to eliminate S from pointwise classical LTL, without expanding the set of propositions, thus preserving expressiveness in addition to satisfiability. However no similar result is known for continuous time. In another related piece of work Hirshfeld and Rabinovich [HR04] show how to eliminate the future distance operator $\Diamond_I$ (with I non-singular) using the past distance operator $\diamondminus_I$, and the U and S modalities.

2 MTL in the pointwise semantics

We use $\mathbb{N}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Q}_{\geq 0}$ to denote the set of natural numbers, non-negative reals and non-negative rationals, respectively. We will make use of the standard notation to denote intervals of reals. These intervals can be open or closed, have endpoints in $\mathbb{R}_{\geq 0}$, and we allow ∞ as an open right end. These intervals denote a subset of reals in the usual manner – for example $[2, \infty)$ denotes the set $\{t \in \mathbb{R}_{\geq 0} \mid 2 \leq t\}$. In particular, we will use rational bounded intervals (i.e. with end-points in $\mathbb{Q}_{\geq 0}$ and right-ends possibly ∞) as constraints in MTL. We denote the set of all rational bounded intervals as $\mathcal{I}_{\mathbb{Q}}$. Wherever convenient, we will use '$\langle$' to denote a left open or a left closed bracket, and similarly '$\rangle$' to denote a right open or right closed bracket.

The natural interpretation of MTL in the pointwise semantics is over timed words. Let us fix a finite alphabet of actions Σ. An (infinite) *timed word* σ over the alphabet Σ is an infinite word over $\Sigma \times \mathbb{R}_{\geq 0}$ of the form $(a_0, t_0)(a_1, t_1)\cdots$, satisfying the following conditions

- (*monotonicity*) $t_i < t_{i+1}$, for each $i \in \mathbb{N}$.
- (*progressiveness*) For each $t \in \mathbb{R}_{\geq 0}$ there exists an $i \in \mathbb{N}$ such that $t < t_i$.

Let $T\Sigma^{\infty}$ denote the set of timed words over Σ.

We denote the pointwise version of MTL by MTL^{pw}. The syntax of MTL^{pw} formulas over the alphabet Σ is:

$$\varphi ::= \ a \mid O\varphi \mid \varphi U_I \varphi \mid \neg\varphi \mid \varphi \vee \varphi,$$

where $a \in \Sigma$ and I is an interval in $\mathcal{I}_{\mathbb{Q}}$.

The formulas of MTL^{pw} above are interpreted over timed words over Σ. Let $\sigma \in T\Sigma^{\infty}$ with $\sigma = (a_0, t_0)(a_1, t_1)\cdots$, and let $i \in \mathbb{N}$. Then the satisfaction relation $\sigma, i \models \varphi$ is given by

$$\begin{array}{lll}
\sigma, i \models a & \text{iff} & a_i = a \\
\sigma, i \models O\psi & \text{iff} & \sigma, i+1 \models \psi \\
\sigma, i \models \psi U_I \eta & \text{iff} & \exists k \geq i : \sigma, k \models \eta,\ t_k - t_i \in I, \text{ and } \forall j : i \leq j < k,\ \sigma, j \models \psi \\
\sigma, i \models \neg\psi & \text{iff} & \sigma, i \not\models \psi \\
\sigma, i \models \psi \vee \eta & \text{iff} & \sigma, i \models \psi \text{ or } \sigma, i \models \eta.
\end{array}$$

We say that a timed word σ satisfies the MTL^{pw} formula φ if and only if $\sigma, 0 \models \varphi$, and set $L(\varphi) = \{\sigma \in T\Sigma^{\infty} \mid \sigma \models \varphi\}$.

We can also consider a version of MTL with the past modality S_I whose semantics is given by:

$$\sigma, i \models \psi S_I \eta \ \text{ iff } \ \exists k \leq i : \sigma, k \models \eta,\ t_i - t_k \in I, \text{ and } \forall j : k < j \leq i,\ \sigma, j \models \psi.$$

We denote this logic by MTL$^{pw}_{S_I}$. We will assume the definitions of the derived constants and connectives $\top$, $\bot$, $\wedge$, etc.

It will be convenient to work with a slightly different presentation of MTL. Let us define a base logic which is similar to classical discrete time LTL, and which we denote by LTL^{pw}. The syntax of the logic (over the alphabet Σ) is given by

$$\varphi ::= \ a \mid O\varphi \mid \varphi U \varphi \mid \neg\varphi \mid \varphi \vee \varphi,$$

and is interpreted over timed words in the same way as MTL^{pw} above, with the modality U having the same semantics as $U_{[0,\infty)}$.

To this base logic we can add the past-time modalities S ("since") and Θ ("previous"), and the distance operators $\Diamond$ and $\diamondminus$ to get the logic $\mathrm{LTL}^{pw}(S, \Theta, \Diamond, \diamondminus)$, whose syntax is given by

$$\varphi ::= \ a \mid O\varphi \mid \varphi U \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \Theta\varphi \mid \varphi S \varphi \mid \Diamond_I \varphi \mid \diamondminus_I \varphi.$$

The modality S is equivalent to $S_{[0,\infty)}$, and $\sigma, i \models \Theta\varphi$ iff $i \geq 1$ and $\sigma, i-1 \models \varphi$. The formula $\Diamond_I \varphi$ is equivalent to $\top U_I \varphi$, and $\diamondminus_I \varphi$ is equivalent to $\top S_I \varphi$. We will use the derived operators $\Box$ and $\boxminus$ with the semantics $\Box_I \varphi = \neg\Diamond_I \neg\varphi$ and $\boxminus_I \varphi = \neg\diamondminus_I \neg\varphi$.

We denote the sublogics of $\mathrm{LTL}^{pw}(S, \Theta, \Diamond, \diamondminus)$ by mentioning the allowed modalities and operators: For example, $\mathrm{LTL}^{pw}(\Diamond)$ denotes the sublogic which does not allow the use of S, Θ and the distance operator $\diamondminus$. We also denote the "non-recursive" versions of these logics by nr-LTL^{pw} (with the appropriate arguments), in which we restrict the use of the distance operators to only actions in Σ, i.e. we allow only distance subformulas of the form $\Diamond_I a$ and $\diamondminus_I a$, with $a \in \Sigma$.

The logic MTL^{pw} can now be seen to be expressively equivalent to the logic $\mathrm{LTL}^{pw}(\Diamond)$. It is clear that $\mathrm{LTL}^{pw}(\Diamond)$ is a sublogic of MTL^{pw}. Conversely, the U_I modality of MTL can be expressed in terms of U and $\Diamond$ as follows [HRS98,DT04]:

$$\varphi_1 U_I \varphi_2 \equiv \begin{cases} \Diamond_I \varphi_2 \wedge \Box_{[0,l)}(\varphi_1 U(\varphi_1 \wedge O\varphi_2)) & \text{if } I = [l, r\rangle, l > 0 \\ \Diamond_I \varphi_2 \wedge \Box_{[0,l]}(\varphi_1 U(\varphi_1 \wedge O\varphi_2)) & \text{if } I = (l, r\rangle, l > 0 \\ \Diamond_I \varphi_2 \wedge (\varphi_1 U \varphi_2) & \text{if } I = [0, r\rangle \\ \Diamond_I \varphi_2 \wedge (\varphi_1 U(\varphi_1 \wedge O\varphi_2)) & \text{if } I = (0, r\rangle. \end{cases}$$

In a symmetric way, the logic $\mathrm{MTL}^{pw}_{S_I}$ can be seen to be equivalent to the logic $\mathrm{LTL}^{pw}(S, \diamondminus)$.

3 Flattening MTL^{pw}

We now show that the logic $\mathrm{LTL}^{pw}(\Diamond)$ (i.e. MTL^{pw}) can be "flattened" to the logic nr-$\mathrm{LTL}^{pw}(\Diamond)$, in the sense that for any formula φ in $\mathrm{LTL}^{pw}(\Diamond)$ over Σ, we can construct a formula φ' in nr-$\mathrm{LTL}^{pw}(\Diamond)$ over an extended alphabet Σ', such that φ' is satisfiable iff φ is. In fact the set of models of φ will be precisely the projections of the models of φ' to Σ.

Let us first define a few necessary notions. We assume the standard notion of subformulas: thus the subformulas of the formula $\varphi = aU(\Diamond_{[1,2]}(b \vee \Diamond_{(0,\infty)} c))$ are φ, a, $\Diamond_{[1,2]}(b \vee \Diamond_{(0,\infty)} c)$, $b \vee \Diamond_{(0,\infty)} c$, b, $\Diamond_{(0,\infty)} c$ and c. The *distance subformulas* of a formula are all its subformulas of the form $\Diamond_I \psi$.

We define the *level* of a $\mathrm{LTL}^{pw}(\Diamond)$ formula φ as a measure of the nesting depth of distance subformulas in φ. Inductively, the level of a formula without any distance subformulas is 0; the level of a formula φ is $i+1$ if it has a distance subformula of the form $\Diamond_I\psi$ with ψ a level i formula, and no distance subformula of the form $\Diamond_J\eta$ with the level of η more than i. A *top-level* distance subformula of φ is a distance subformula which has at least one occurrence outside the scope of any other distance subformula. More formally, the set of distance subformulas of φ, denoted *top-dsf*(φ) is defined inductively as:

$$\begin{array}{ll} \textit{top-dsf}(a) & = \{\} \\ \textit{top-dsf}(O\varphi) & = \textit{top-dsf}(\varphi) \\ \textit{top-dsf}(\psi U\eta) & = \textit{top-dsf}(\psi) \cup \textit{top-dsf}(\eta) \\ \textit{top-dsf}(\psi \vee \eta) & = \textit{top-dsf}(\psi) \cup \textit{top-dsf}(\eta) \\ \textit{top-dsf}(\Diamond_I\psi) & = \{\Diamond_I\psi\}. \end{array}$$

Let us call a set of formulas X *closed* if for every distance subformula $\Diamond_I\psi$ of a formula in X, the formula ψ also belongs to X.

Let $X = \{\psi_0, \ldots, \psi_n\}$ be a closed set of formulas, with $\psi_0, \ldots, \psi_n$ being in increasing order of level. Let $\sigma = (a_0, t_0)(a_1, t_1)\cdots$ be a timed word over Σ. Then we define the *canonical extension* of σ wrt X, denoted by $\textit{can-ext}_X(\sigma)$, to be the timed word over the extended alphabet $\Sigma' = \Sigma \times \{0,1\}^{n+1}$, defined as follows. Let us denote an element of Σ' by (a, v) where $a \in \Sigma$ and $v \in \{0,1\}^{n+1}$. Then $\textit{can-ext}_X(\sigma) = ((a_0, v_0), t_0)((a_1, v_1), t_1)\cdots$, where for each i in $\mathbb{N}$ and $j \in \{0, \ldots, n\}$, $v_i[j] = 1$ iff $\sigma, i \models \psi_j$.

For a timed word σ' over Σ' let us denote the timed word over Σ obtained by erasing the extension vectors (i.e projecting the alphabet to Σ and keeping the timestamps), by $\sigma'\lceil\Sigma$. Note that σ' can only be a canonical extension (if at all) of its projection to Σ. We extend the notion of projection to a set of timed words over Σ' in the natural way.

As an example, consider the formula $\varphi = \Diamond_{(0,1)}\psi_2$ where $\psi_2 = \Diamond_{[0,1]}\psi_1$ and $\psi_1 = b \wedge \Diamond_{[1,1]}c$. Let X be the set $\{c, \psi_1, \psi 2\}$. Then the canonical extension of the timed word $(a, 0.2)(a, 1.4)(b, 1.8)(c, 2.8)(a, 4)(a, 5)\cdots$ with respect to X is:

T	0.2	1.4	1.8	2.8	4.0	5.0	$\cdots$
Σ	a	a	b	c	a	a	$\cdots$
ψ_2	0	1	1	0	0	0	$\cdots$
ψ_1	0	0	1	0	0	0	$\cdots$
c	0	0	0	1	0	0	$\cdots$

Let us now define the "canonical translation" of a formula φ. Let X be any closed finite set of formulas containing the set of formulas ψ such that φ has a distance subformula of the form $\Diamond_I\psi$. Then the *canonical translation* of φ wrt X, denoted $\textit{can-tr}_X(\varphi)$, is obtained from φ by replacing every top-level distance subformula of the form $\Diamond_I\psi_k$ by

$$\bigvee_{a\in\Sigma,\, v[k]=1} \Diamond_I(a, v)$$

and replacing each remaining atomic subformula a by $\bigvee_v (a, v)$. Note that for any φ, $\mathit{can}\text{-}\mathit{tr}_X(\varphi)$ is always a formula in nr-LTL$^{pw}(\Diamond)$ over Σ'.

We now argue that the truth of φ in σ is preserved by $\mathit{can}\text{-}\mathit{tr}_X(\varphi)$ in $\mathit{can}\text{-}\mathit{ext}_X(\sigma)$.

Lemma 1. *Let φ and X be as above, and let σ be a timed word over Σ. Then $\sigma, i \models \varphi$ iff $\mathit{can}\text{-}\mathit{ext}_X(\sigma), i \models \mathit{can}\text{-}\mathit{tr}_X(\varphi)$.*

Proof. The proof follows easily by induction on the structure of φ. For the base case, clearly $\sigma, i \models a$ iff $\mathit{can}\text{-}\mathit{ext}_X(\sigma), i \models \bigvee_v (a, v)$. For the inductive step, the only interesting case is $\Diamond_I \psi_j$. But $\sigma, i \models \Diamond_I \psi_j$ iff $\mathit{can}\text{-}\mathit{ext}_X(\sigma), i \models \bigvee_{a \in \Sigma,\, v[j]=1} \Diamond_I (a, v)$, since ψ_j is true at a position k in σ precisely when the vector v at position k in $\mathit{can}\text{-}\mathit{ext}_X(\sigma)$ is such that $v[j] = 1$. ☐

We now show that the property of an extended timed word being a canonical extension of some timed word wrt a closed set of formulas X is definable in nr-LTL$^{pw}(\Diamond)$. Let $X = \{\psi_0, \ldots, \psi_n\}$ be a closed set of formulas, with $\psi_0, \ldots, \psi_n$ being in increasing order of level. The formula ν_{can}^X basically says that each 0/1 entry in each row of an extended word is correct. We define

$$\nu_{can}^X = \bigwedge_{j=0}^{n} \Box\Big(\Big(\bigvee_{a \in \Sigma, v[j]=1} (a, v)\Big) \Leftrightarrow \mathit{can}\text{-}\mathit{tr}_X(\psi_j)\Big).$$

Lemma 2. *Let X be as above. Let $\Sigma' = \Sigma \times \{0,1\}^{n+1}$ and let σ' be a timed word over Σ'. Then σ' is a canonical extension of $\sigma' \restriction \Sigma$ wrt X iff $\sigma' \models \nu_{can}^X$.*

Proof. Let σ' be a canonical extension of $\sigma' \restriction \Sigma$. Then we can argue by induction on j going from 0 to n that $\sigma' \restriction \Sigma, i \models \psi_j$ iff $\sigma', i \models \mathit{can}\text{-}\mathit{tr}_X(\psi_j)$. This gives us that $\sigma' \models \nu_{can}^X$.

Conversely, suppose $\sigma' \models \nu_{can}^X$. Then once again we argue by induction on j going from 0 to n that $\sigma' \restriction \Sigma, i \models \psi_j$ iff $\sigma', i \models \mathit{can}\text{-}\mathit{tr}_X(\psi_j)$. By the definition of ν_{can}^X we then have that $v_i[j] = 1$ iff $\sigma' \restriction \Sigma, i \models \psi_j$, which means that σ' is a canonical extension wrt X. ☐

We can now construct the required flattening of a formula φ in LTL$^{pw}(\Diamond)$. Let X be the set of all ψ such that there is a distance subformula of φ of the form $\Diamond_I \psi$. Note that X is closed. We now define φ' to be $\mathit{can}\text{-}\mathit{tr}_X(\varphi) \wedge \nu_{can}^X$. By Lemmas 1 and 2, it follows that $L(\varphi) = L(\varphi') \restriction \Sigma$.

To summarise the developments of this section:

Theorem 1. *Let $\varphi \in$ LTL$^{pw}(\Diamond)$ over the alphabet Σ. Then we can construct a formula $\varphi' \in \mathit{nr}$-LTL$^{pw}(\Diamond)$ over an alphabet Σ' which is an extension of Σ, such that $L(\varphi) = L(\varphi') \restriction \Sigma$. In particular φ is satisfiable iff φ' is.* ☐

The flattening construction carried out in this section for $\Diamond_I$ subformulas, clearly works also for the $\diamondminus_I$ subformulas. Further, it can be extended to the modality S, in the sense that for any formula φ we can give a satisfiability-equivalent formula over an extended alphabet, in which there are no nested S modalities – in fact the only S formulas left will be of the form $a' S b'$ for some letters a' and b' of the extended alphabet. The idea is similar: consider canonical extensions wrt the set of formulas ψ and η for each subformula $\psi S \eta$ of φ.

4 MTL in the continuous semantics

In the continuous semantics MTL is typically interpreted over "timed state sequences". Before we define these, let first introduce the notion of a finitely varying function. Let B be a finite non-empty set, and let $l, r \in \mathbb{R}_{\geq 0}$ with $l \leq r$. Let $f : [l, r] \to B$. Then $t \in [l, r]$ is a point of discontinuity of f if there does not exist $\epsilon > 0$ such that f is constant in the interval $(t - \epsilon, t + \epsilon)$. The function f is called *finitely varying* if it has only a finite number of discontinuities.

Let P be a finite set of propositions. A *timed state sequence* τ over P is a map $\tau : \mathbb{R}_{\geq 0} \to 2^P$ satisfying the condition that it is finitely varying in any bounded interval of its domain (or, equivalently, that it has a countable number of discontinuities). Yet another equivalent definition (as given in [AFH96]) is that there exists a sequence of subsets of propositions $s_0, s_1, \ldots$ and a sequence of intervals $I_0, I_1, \ldots$ satisfying:

1. I_0 is of the form $[0, r\rangle$ for some $r \in \mathbb{R}_{\geq 0}$.
2. Every pair of intervals I_j and I_{j+1} are *adjacent* in the sense that I_j and I_{j+1} are disjoint, and $I_j \cup I_{j+1}$ forms an interval.
3. The sequence of intervals is "progressive" in that for every $t \in \mathbb{R}_{\geq 0}$, there exists $j \in \mathbb{N}$ such that $t \in I_j$.
4. The function τ is constant and equal to s_j in each I_j.

We call the sequence $(s_0, I_0)(s_1, I_1)\cdots$ above an *interval representation* of the function τ. It is easy to see that a timed state sequence τ has a "canonical" interval representation of the form $(s_0, I_0)(s_1, I_1)\cdots$ where the I_i's are an alternating sequence of singular and open intervals (i.e. for each i, I_{2i} is of the form $[l, l]$ and I_{2i+1} is of the form (l, r)), where the singular intervals are precisely the points of discontinuity of τ. We denote the set of timed state sequences over P by $TSS(P)$.

The continuous version of MTL will be denoted by MTL^c. The syntax of MTL^c formulas over a set of propositions P is given by:

$$\varphi ::= \; p \mid \varphi U_I \varphi \mid \neg\varphi \mid \varphi \vee \varphi,$$

where $p \in P$ and I is an interval in $\mathcal{I}_{\mathbb{Q}}$. The "next" operator O is absent as it will have no meaning in our continuous semantics, and the "until" operator is "strict" in its first argument, as defined below.

The formulas of MTL^c above are interpreted over timed state sequences over P. Let τ be a timed state sequence over P, and let $t \in \mathbb{R}_{\geq 0}$. Then the satisfaction relation $\tau, t \models \varphi$ is given by:

$$\begin{array}{lll}
\tau, t \models p & \text{iff} & p \in \tau(t) \\
\tau, t \models \psi U_I \eta & \text{iff} & \exists t' \geq t : \tau, t' \models \eta,\ t' - t \in I, \text{ and } \forall t'' : t < t'' < t',\ \tau, t'' \models \psi \\
\tau, i \models \neg\psi & \text{iff} & \tau, i \not\models \psi \\
\tau, i \models \psi \vee \eta & \text{iff} & \tau, i \models \psi \text{ or } \tau, i \models \eta.
\end{array}$$

We say that a timed word τ satisfies the MTL^c formula φ if and only if $\tau, 0 \models \varphi$, and set $L(\varphi) = \{\tau \in TSS(P) \mid \tau \models \varphi\}$.

We can also consider a version of MTL with the past modality S_I whose semantics is given by:

$$\tau, t \models \psi S_I \eta \text{ iff } \exists t' \leq t : \tau, t' \models \eta,\ t - t' \in I, \text{ and } \forall t'' : t' < t'' < t\ \tau, t'' \models \psi.$$

We denote this logic by $\mathrm{MTL}^c_{S_I}$.

As in the pointwise case, it will be convenient to work with a slightly different presentation of MTL. Let us define a base logic which is similar to classical continuous time LTL [K68], and which we denote by LTL^c. The syntax of the logic (over the set of propositions P) is given by $\varphi ::= \ p \mid \varphi U \varphi \mid \neg\varphi \mid \varphi \vee \varphi$, and is interpreted over timed words in the same way as MTL^c above, with the modality U interpreted as $U_{[0,\infty)}$. To this base logic we can add the past-time modalities S ("since") and the distance operators $\Diamond$ and $\overleftarrow{\Diamond}$ to get the logic $\mathrm{LTL}^c(S, \Diamond, \overleftarrow{\Diamond})$, whose syntax is given by

$$\varphi ::= \ p \mid \varphi U \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi S \varphi \mid \Diamond_I \varphi \mid \overleftarrow{\Diamond}_I \varphi.$$

The modality S is interpreted as $S_{[0,\infty)}$, while the formula $\Diamond_I \varphi$ is equivalent to $\top U_I \varphi$, and $\overleftarrow{\Diamond}_I \varphi$ is equivalent to $\top S_I \varphi$. As in the pointwise case, we assume the notation for the sublogics and the "non-recursive" versions of this logic.

Once again the logic MTL^c can be seen to be expressively equivalent to the logic $\mathrm{LTL}^c(\Diamond)$ as the U_I modality of MTL^c can be expressed in terms of U and $\Diamond$. For example, if $I = [l, l]$, then $\psi U_I \eta = (\Box_{(0,l)}\psi) \wedge (\Diamond_{[l,l]}\eta)$; and if $I = (l, r)$ then $\psi U_I \eta = (\Box_{(0,l]}\psi) \wedge (\Diamond_{[l,l]}(\psi U \eta)) \wedge \Diamond_{(l,r)}\eta$. In a symmetric way, the logic $\mathrm{MTL}^c_{S_I}$ can be seen to be equivalent to the logic $\mathrm{LTL}^c(S, \Diamond, \overleftarrow{\Diamond})$.

5 Flattening MTL^c

We now show that each of the sublogics of $\mathrm{LTL}^c(S, \Diamond, \overleftarrow{\Diamond})$ can be flattened to its non-recursive version. As in the pointwise case we illustrate this for the logic $\mathrm{LTL}^c(\Diamond)$. We show that every $\mathrm{LTL}^c(\Diamond)$ formula (equivalently a MTL^c formula) over a set of propositions P, can be flattened to an nr-$\mathrm{LTL}^c(\Diamond)$ formula over a set of propositions P' which is an extension of P. We carry over the notions of distance subformulas, level etc. defined for LTL^{pw}, for LTL^c as well. We fix a set of propositions P for the rest of this section.

Let $X = \{\psi_0, \ldots, \psi_n\}$ be a closed set of formulas, in increasing order of level. Let $\tau = (s_0, I_0)(s_1, I_1)\cdots$ be a timed state sequence over P. Let P' be the extended set of propositions $P \cup \{p_0, \ldots, p_n\}$, where each proposition p_j is meant to capture the truth of ψ_j. Then we define the *canonical extension* of τ wrt X, denoted by $\textit{can-ext}_X(\tau)$, to be the timed state sequence over P', given by $\textit{can-ext}_X(\tau)(t) = \tau(t) \cup \{p_j \mid \tau, t \models \psi_j\}$.

We need to argue that $\textit{can-ext}_X(\tau)$ is indeed a timed state sequence, in that it is has only a countable number of discontinuities. For this it is enough to argue that the truth of each ψ changes at most countably many times along τ. But this follows from the argument in [AFH96] which says that for every timed state sequence ν and every $\mathrm{LTL}^c(\Diamond)$ formula φ, there is an equivalent interval representation of τ (i.e. denoting the same function as τ) which is "φ-fine" – i.e. φ is constant throughout each interval.

For a timed state sequence τ' over P', let us denote the timed state sequence over P obtained by projecting each $\tau'(t)$ to P by $\tau' \lceil P$. Thus $(\tau' \lceil P)(t) = \tau(t) \cap P$. We extend the notion of projection to a set of timed state sequences over P' in the natural way.

As an example, consider the formula $\varphi = \Diamond_{(0,1)}\psi_2$ where $\psi_2 = \Diamond_{[0,1]}\psi_1$ and $\psi_1 = p \wedge \Diamond_{[1,1]}q$. Let us take X to be the set $\{q, \psi_1, \psi_2\}$. Then the canonical extension of the timed state sequence $\tau = (\{p\}, [0, 1.5))(\{q\}, [1.5, 2])(\{p, q\}, (2, \infty)$ with respect to X is:

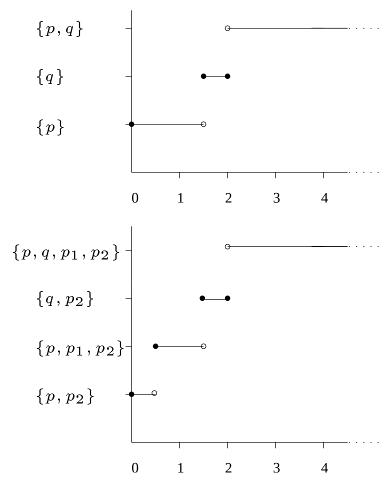

Fig. 1. TSS τ and its canonical extension wrt X.

Let us now define the "canonical translation" of a formula φ. Let X be any closed finite set of formulas containing the set of formulas ψ such that φ has a distance subformula of the form $\Diamond_I\psi$. Then the *canonical translation* of φ wrt X, denoted $\textit{can-tr}_X(\varphi)$, is obtained from φ by replacing every top-level distance subformula of the form $\Diamond_I\psi_k$ by $\Diamond_I p_k$. Once again, we note that for any φ, $\textit{can-tr}_X(\varphi)$ is always a formula in nr-LTL$^c(\Diamond)$ over P'.

As in Lemma 1, we can argue that the truth of φ in τ is preserved by $\textit{can-tr}_X(\varphi)$ in $\textit{can-ext}_X(\tau)$:

Lemma 3. *Let φ and X be as above, and let τ be a timed state sequence over P. Then $\tau, t \models \varphi$ iff $\textit{can-ext}_X(\tau), t \models \textit{can-tr}_X(\varphi)$.* □

Once again, we show that the property of an extended timed state sequence being a canonical extension of a timed state sequence wrt a closed set of formulas X is definable in nr-LTL$^c(\Diamond)$. Let $X = \{\psi_0, \ldots, \psi_n\}$ be a closed set of formulas, with $\psi_0, \ldots, \psi_n$ being in increasing order of level. The formula ν^X_{can} basically says that the value of each p_i at a point in the extended word is correct. We define

$$\nu^X_{can} = \Box(\bigwedge_{j=0}^{n}(p_j \Leftrightarrow \mathit{can\text{-}tr}_X(\psi_j))).$$

Lemma 4. *Let X be as above. Let $P' = P \cup \{p_0, \ldots, p_n\}$ and let τ' be a timed state sequence over P'. Then τ' is a canonical extension of $\tau' \restriction P$ wrt X iff $\tau' \models \nu^X_{can}$.*

Proof. Let τ' be a canonical extension of $\tau' \restriction P$. Then we can argue by induction on j going from 0 to n that $\tau' \restriction P, t \models \psi_j$ iff $\tau', t \models \mathit{can\text{-}tr}_X(\psi_j)$. This gives us that $\tau' \models \nu^X_{can}$.

Conversely, suppose $\tau' \models \nu^X_{can}$. Then once again we argue by induction on j going from 0 to n that $\tau' \restriction P, t \models \psi_j$ iff $\tau', t \models \mathit{can\text{-}tr}_X(\psi_j)$. By the definition of ν^X_{can} we then have that $v_i[j] = 1$ iff $\sigma' \restriction \Sigma, i \models \psi_j$, which means that σ' is a canonical extension wrt X. $\Box$

We can now construct the required flattening of a formula φ in $\mathrm{LTL}^c(\Diamond)$. Let X be the set of all ψ such that there is a distance subformula of φ of the form $\Diamond_I \psi$. Note that X is closed. We now define φ' to be $\mathit{can\text{-}tr}_X(\varphi) \wedge \nu^X_{can}$. By Lemmas 3 and 4, it follows that $L(\varphi) = L(\varphi') \restriction \Sigma$. To summarise:

Theorem 2. *Let $\varphi \in \mathrm{LTL}^c(\Diamond)$ over the set of propositions P. Then we can construct a formula $\varphi' \in nr\text{-}\mathrm{LTL}^c(\Diamond)$ over a an extended set of propositions P', such that $L(\varphi) = L(\varphi') \restriction P$. In particular φ is satisfiable iff φ' is.* $\Box$

Once again, we observe that the flattening carried out here can be extended to the past operator $\diamondminus$ and the past modality S.

6 Eliminating S in LTL^{pw} and LTL^c

In this section we show how to go from a formula in any of our logics containing the S modality to a satisfiability-equivalent formula over an extended alphabet that has no S subformulas.

We begin with the pointwise case. Let φ be a formula in $\mathrm{LTL}^{pw}(S)$ over the alphabet Σ. Without loss of generality we can assume that the S subformulas in φ have been flattened (cf. Section 3). Thus the only S subformulas in φ are of the form aSb with $a, b \in \Sigma$. Let $\{b_0 S c_0, \ldots, b_n S c_n\}$ be all the since formulas in φ. We introduce a new proposition r_i for each $b_i S c_i$, which is meant to capture its truth, and is encoded in the extended alphabet $\Sigma' = \Sigma \times \{0,1\}^{n+1}$. We define the canonical extension of a timed word $\sigma = (a_0, t_0)(a_1, t_1)\cdots$ to be the timed word $\sigma' = ((a_0, v_0), t_0)((a_1, v_1), t_1)\cdots$ where for each $i \in \mathbb{N}$, and each $j \in \{0, \ldots, n\}$, $v_i[j] = 1$ iff $\sigma, i \models b_j S c_j$. The canonical translation of φ is obtained by replacing each a by $\bigvee_v (a, v)$, and each $b_j S c_j$ by r_j, where $r_j = \bigvee_{a \in \Sigma, v[j]=1} (a, v)$. It is clear that the canonical translation preserves the truth of φ in canonical extensions.

Further, we can express the fact that σ' in $T\Sigma'^{\infty}$ is a canonical extension, by a formula ν_{can} in LTL^{pw} over Σ' defined as follows.

$$\bigwedge_{j=0}^{n}((c_j \Leftrightarrow r_j)\wedge\Box(((r_j\wedge Ob_j) \Rightarrow Or_j)\wedge(c_j \Rightarrow r_j)\wedge(((\neg r_j)\wedge(\neg Oc_j)) \Rightarrow \neg Or_j))).$$

It is easy to see that a canonical extension will satisfy ν_{can}, and conversely, by induction on i, that any σ' which satisfies ν_{can} will have correct extension entries at each point i, and hence is canonical.

The required formula in LTL^{pw} over Σ' which is satisfiability-equivalent to φ is thus $\varphi' \wedge \nu_{can}$.

For the continuous case we need to take a slightly different approach. Given a formula in LTL$^{c}(S)$, we want to give a satisfiability equivalent formula in LTL^{c}. Let φ be an LTL$^{c}(S)$ formula over the set of propositions P. Once more we can assume that φ has been flattened, and thus the only occurrences of S formulas are of the form pSq with $p, q \in P$. Let $\{p_0Sq_0, \ldots, p_nSq_n\}$ be all the S subformulas in φ. We introduce a proposition r_j for each p_jSq_j which is meant to capture the truth of p_jSq_j, and call this extended set of propositions P'. The canonical extension of τ in $TSS(P)$ is τ' in $TSS(P')$, given by $\tau'(t) = \tau(t) \cup \{r_j \mid \tau, t \models p_jSq_j\}$. The translation of φ obtained by replacing every occurrence of p_iSq_i by r_i, clearly preserves the truth of φ in the canonical extension.

It is now sufficient if we can define a formula ν_{can} in LTL^{c} over P' which describes precisely the timed state sequences over P' which are canonical extensions.

We define ν_{can} to be the formula

$$\bigwedge_{j=0}^{n}((r_j \Leftrightarrow q_j) \wedge \Box(\varphi_{1j} \wedge \varphi_{2j} \wedge \varphi_{3j} \wedge \varphi_{4j} \wedge \varphi_{5j} \wedge \varphi_{6j} \wedge \varphi_{7j} \wedge \varphi_{8j})),$$

where $\mathit{immed}(\psi) = \psi \wedge (\neg\psi U \top)$, and

1. $\varphi_{1j} = ((p_j \wedge q_j)U\neg(p_j \wedge q_j)) \Rightarrow ((r_j \wedge p_j \wedge q_j)U(r_j \wedge \neg(p_j \wedge q_j)))$ $\wedge\Box(p_j \wedge q_j) \Rightarrow \Box r_j$.
2. $\varphi_{2j} = ((p_j\wedge\neg q_j)U\neg(p_j\wedge\neg q_j)) \Rightarrow (((q_j\vee r_j) \Rightarrow (r_j\wedge p_j\wedge\neg q_j)U(r_j\wedge\neg(p_j\wedge \neg q_j)))\wedge(\neg(q_j\vee r_j) \Rightarrow (\neg r_j\wedge p_j\wedge\neg q_j)U(\neg r_j\wedge\neg(p_j\wedge\neg q_j)))) \wedge\Box(p_j\wedge\neg q_j) \Rightarrow ((q_j \vee r_j) \Rightarrow \Box r_j \wedge \neg(q_j \vee r_j) \Rightarrow \Box\neg r_j)$.
3. $\varphi_{3j} = ((\neg p_j \wedge q_j)U\neg(\neg p_j \wedge q_j)) \Rightarrow ((\neg r_j \wedge \neg p_j \wedge q_j)U(\neg r_j \wedge \neg(\neg p_j \wedge q_j)))$ $\wedge\Box(\neg p_j \wedge q_j) \Rightarrow \Box\neg r_j$.
4. $\varphi_{4j} = ((\neg p_j\wedge\neg q_j)U\neg(\neg p_j\wedge\neg q_j)) \Rightarrow ((\neg r_j\wedge\neg p_j\wedge\neg q_j)U(\neg r_j\wedge\neg(\neg p_j\wedge\neg q_j)))$ $\wedge\Box(\neg p_j \wedge \neg q_j) \Rightarrow \Box\neg r_j$.
5. $\varphi_{5j} = ((p_j \wedge q_j)U\,\mathit{immed}(p_j \wedge q_j)) \Rightarrow ((r_j \wedge p_j \wedge q_j)U(r_j \wedge \mathit{immed}(p_j \wedge q_j)))$ $\wedge\Box(p_j \wedge q_j) \Rightarrow \Box r_j$.
6. $\varphi_{6j} = ((p_j \wedge \neg q_j)U\,\mathit{immed}(p_j \wedge \neg q_j)) \Rightarrow (((q_j \vee r_j) \Rightarrow (r_j \wedge p_j \wedge \neg q_j)U(r_j \wedge \mathit{immed}(p_j\wedge\neg q_j)))\wedge(\neg(q_j\vee r_j) \Rightarrow (\neg r_j\wedge p_j\wedge\neg q_j)U(\neg r_j\wedge \mathit{immed}(p_j\wedge\neg q_j))))$ $\wedge\Box(p_j \wedge \neg q_j) \Rightarrow ((q_j \vee r_j) \Rightarrow \Box r_j \wedge \neg(q_j \vee r_j) \Rightarrow \Box\neg r_j)$.
7. $\varphi_{7j} = ((\neg p_j\wedge q_j)U\,\mathit{immed}(\neg p_j\wedge q_j)) \Rightarrow ((\neg r_j\wedge\neg p_j\wedge q_j)U(\neg r_j\wedge \mathit{immed}(\neg p_j\wedge q_j))) \wedge\Box(\neg p_j \wedge q_j) \Rightarrow \Box\neg r_j$.

8. $\varphi_{8j} = ((\neg p_j \wedge \neg q_j) U\, immed(\neg p_j \wedge \neg q_j)) \Rightarrow ((\neg r_j \wedge \neg p_j \wedge \neg q_j) U (\neg r_j \wedge immed(\neg p_j \wedge \neg q_j))) \wedge \Box(\neg p_j \wedge \neg q_j) \Rightarrow \Box \neg r_j$.

The claim is now that $\tau' \in TSS(P')$ satisfies ν_{can} iff τ' is a canonical extension. If τ' is a canonical extension, then it is easy to check that each clause of ν_{can} above is satisfied in τ'. For example, the formula φ_{1j} is true of τ' since at some point t if p and q are true until some future point t', where at least one of p and q is not true, then pSq is true everywhere in $(t, t']$. Hence $r \in \tau'(t'')$ for all $t'' \in (t, t']$. Therefore at t, $(r \wedge p \wedge q) U (r \wedge \neg(p \wedge q))$ is true.

Conversely, suppose τ' satisfies ν_{can}. Let the canonical interval representation of τ' be $(s_0, I_0)(s_1, I_1)\cdots$. Then we argue by induction on i, that the value of each r_j is correct in each interval I_{2i-1} and I_{2i}. For $i = 0$, $I_0 = [0, 0]$ and by the semantics of S, at time 0, each $p_j S q_j$ is true iff q_j is true. The definition of ν_{can} forces this, and hence this is correct.

For the inductive step, let us assume the value of each r_j is correct in the singular interval I_{2i}. Consider the intervals I_{2i+1} and I_{2i+2}. There are eight different cases depending on the values p_i and p_j take in these intervals (note that by definition, the values of the propositions don't change in any of these intervals). We argue that each corresponding clause of ν_{can} ensures that r_j has the correct value in these intervals. For instance, φ_{2j} deals with the case when p is true in I_{2i+1} and q is false, and either p is false or q is true in I_{2i+2}. Then the value of pSq in I_{2i+1} and I_{2i+2} depends also on the values of q and pSq in I_{2i}. It either q or pSq is true in I_{2i} then pSq is true in I_{2i+1} and I_{2i+2}, otherwise it is false. By induction hypothesis, we know that the value of r is correct in the interval I_{2i}, hence we know correctly the value of pSq in I_{2i}. Hence the value of r is correct in I_{2i+1} and I_{2i+2}.

The table below summarizes the different cases.

p, q in I_{2i+1}	p, q in I_{2i+2}	r in I_{2i+1}	r in I_{2i+2}
$(1,1)$	not $(1,1)$	1	1
$(1,0)$	not $(1,0)$	1 if q or r in I_{2i}, 0 otherwise	1 if q or r in I_{2i}, else 0
$(0,1)$	not $(0,1)$	0	0
$(0,0)$	not $(0,0)$	0	0
$(1,1)$	$(1,1)$	1	1
$(1,0)$	$(1,0)$	1 if q or r in I_{2i}, 0 otherwise	1 if q or r in I_{2i}, else 0
$(0,1)$	$(0,1)$	0	0
$(0,0)$	$(0,0)$	0	0

The rows of the table correspond to the formulas φ_{1j} through φ_{8j}.

7 Eliminating $\diamondminus$ in LTL^c

We now show how we can remove the past distance modality $\diamondminus$ from the continuous version of MTL, while preserving satisfiability of the original formula.

Let φ be a formula in any of the versions of our continuous logic LTL^c with the $\diamondminus$ operator. By the results of Section 5 we can assume the $\diamondminus$ subformulas in φ are of the form $\diamondminus_I p$ with p a proposition in P. We essentially show how to express formulas of the form $\diamondminus_I p$ in terms of new propositions, the U modality, and $\Diamond$ operator.

Let $\diamondminus_{I_0} p_0, \ldots, \diamondminus_{I_n} p_n$ be all the $\diamondminus_I$ subformulas in φ. Let τ be a timed state sequence over P. Let P' be the extended set of propositions $P \cup \{q_0, \ldots, q_n\} \cup \{r_0, \ldots, r_n\}$, where q_j and r_j are new propositions associated with each $\diamondminus_{I_j} p_j$. The proposition q_j is meant to capture the truth of $\diamondminus_{I_j} p_j$, and r_j is meant to capture the fact that we have seen a p_j sometime strictly in the past. We define the canonical extension of τ (wrt $\{\diamondminus_{I_0} p_0, \ldots, \diamondminus_{I_n} p_n\}$) to be the timed state sequence τ' over P', given by

$$\tau'(t) = \tau(t) \cup \{q_j \mid \tau, t \models \diamondminus_{I_j} p_j\} \cup \{r_j \mid \exists t' < t : \ \tau, t' \models p_j\}.$$

The canonical translation of φ is obtained by simply replacing each $\diamondminus_{I_j} p_j$ by q_j. It is clear that $\tau, t \models \varphi$ iff the canonical translation of φ is satisfied at t in the canonical extension of τ.

We now define the formula ν_{can} which characterises canonical extensions. The formula ν_{can} is the formula below

$$\bigwedge_{j=0}^{n} ((\neg r_j) \wedge \Box(p_j \Rightarrow \Box_{(0,\infty)} r_j) \wedge \neg((\neg p_j) \wedge ((\neg p_j) U r_j)))$$

in conjunction with a formula ψ_j for each j as below:

- If I_j is of the form $[l, l]$ take ψ_j to be $\Box(p_j \Leftrightarrow \Diamond_{[l,l]} q_j)$;
- If I_j is of the form (l, e) take ψ_j to be $\Box((\Diamond_{[e,e]} q_j) \Leftrightarrow \Diamond_{(0,e-l)} p_j)$;
- If I_j is of the form (l, ∞) take ψ_j to be $\Box(r_j \Leftrightarrow \Diamond_{[l,l]} q_j)$;
- And if I_j is of the form $[l, \infty)$ take ψ_j to be $\Box((r_j \vee p_j) \Leftrightarrow \Diamond_{[l,l]} q_j)$.

We note that we have introduced the proposition r_j to avoid using a formula of the form $\top S p_j$ in the translation. The construction here is simpler than if we had used S and then eliminated it using the results of Section 6.

8 Discussion

We point out here that our flattening construction can be carried out for the general class of timed temporal logics based on "input-determined" operators [DT04,CDP06]. These logics can be viewed as being of the form LTL^{pw} or LTL^{c} along with an input-determined distance operator Δ. Example of such operators include the operators $\lhd$ and $\rhd$ of [HRS98], apart from the $\Diamond$ and $\diamondminus$ of MTL.

We are also working on eliminating $\diamondminus$ from $\mathrm{MITL}^{c}_{S_I}$ without introducing singular intervals. This would tell us that $\mathrm{MITL}^{c}_{S_I}$ is also decidable.

References

[AD94] R. Alur and D. Dill: A theory of timed automata, *Theoretical Computer Science* **126**, Elsevier, 183–235 (1994).

[AFH96] R. Alur, T. Feder and T. A. Henzinger: The benefits of relaxing punctuality, *J. ACM* **43**, 116–146 (1996).

[BCM05] P. Bouyer, F. Chevalier and N. Markey: On the Expressiveness of TPTL and MTL, *FSTTCS 2005*, LNCS **3821** (2005).

[CDP06] F. Chevalier, D. D'Souza, and P. Prabhakar: On continuous timed automata with input-determined guards, *To appear in Proc. FSTTCS 2006* (2006).

[DM05] D. D'Souza and M. Raj Mohan: Eventual timed automata, *Proc. FSTTCS 2005*, LNCS **3821** (2005).

[DT04] D. D'Souza and N. Tabareau: On timed automata with input-determined guards, *Proc. FORMATS-FTRTFT*, LNCS **3253** (2004).

[GPSS80] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi: The Temporal Analysis of Fairness, *Seventh ACM POPL*, 163–173 (1980).

[HR04] Y. Hirshfeld and A. Rabinovich: Logics for Real Time: Decidability and Complexity, *Fundamenta Informaticae*, **62**, 1–28 (2004).

[HRS98] T. Henzinger, J. Raskin and P. Schobbens: The regular real-time languages, *Proc. 25th ICALP*, LNCS **1443**, 580–591 (1998).

[K68] J. A. W. Kamp: Tense Logic and the Theory of Linear Order, PhD thesis, University of California, Los Angeles (1968).

[Koy90] R. Koymans: Specifying real-time properties with metric temporal logic, *Real Time Systems*, Vol. **2**, No.4, 255–299 (1990).

[OW05] J. Ouaknine and J. Worrell: On the decidability of metric temporal logic, *Proc. LICS 2005* (2005).

[OW06] J. Ouaknine and J. Worrell: On Metric Temporal logic and Faulty Turing Machine, *Proc. FOSSACS 2006*, LNCS **3921** (2006).

[P77] A. Pnueli: The Temporal Logic of programs, *Proceedings of 18th IEEE Symposium on Foundations of Computer Science,* 46–57 (1977).

[PD06] P. Prabhakar and D. D'Souza: On the expressiveness of MTL with past operators, *To appear in Proc. FORMATS 2006*, (2006).