Real-time Property Preservation
in Approximations of Timed Systems

Jinfeng Huang, Jeroen Voeten and Marc Geilen
Eindhoven University of Technology
Faculty of Electrical Engineering
Eindhoven, 5600MB, The Netherlands
E-mail: {J.Huang, J.P.M.Voeten, M.C.W.Geilen}@tue.nl

Abstract

Formal techniques have been widely applied in the design of real-time systems and have significantly helped detect design errors by checking real-time properties of the model. However, a model is only an approximation of its realization in terms of the issuing time of events. Therefore, a real-time property verified in the model can not always be directly transferred to the realization. In this paper, both the model and the realization are viewed as sets of timed state sequences. In this context, we first investigate the real-time property preservation between two neighbouring timed state sequences (execution traces of timed systems), and then extend the results to two “neighbouring” timed systems. The study of real-time property preservation gives insight in building a formal link between real-time properties satisfied in the model and those in the realization.

1 Introduction

Over the past decades, we have witnessed a significant increase in the application of formal techniques to the design of real-time systems. Various studies have been carried out in the advance of real-time property verification techniques [2, 3, 11, 10] and their applications [9, 13]. These real-time properties are formal representations of timing requirements in a realization, whose behaviour is abstracted in the corresponding model. Typically, timed systems and metric interval temporal logic (MITL)[3] have been used to formalize behaviour and timing requirements respectively and have achieved success in yielded a wide range of real-time systems.

However, a model is generally only an approximation of its realization with respect to time (or vise versa). For example, it has been argued in [12] that a real-time formal model $M$ often over-specifies its corresponding physical realizations. In model $M$, the issuing time of an event $p$ is usually specified as some time instant $t_M$. However, it cannot be guaranteed that the issuing time $t_M$ of $p$ in $M$ is identical to its counterpart $t_R$ in $R$ (a realization of $M$). There is a time difference $\delta_t$ between them ($\delta_t = t_M - t_R$).

Example 1. Consider an intelligent lighting controller, which can adjust the light intensity according to different input sequences. If there is a click action at the Light off state, the controller goes into the Wait state and a timer is activated. If there is a second click taking place within 2 seconds, the controller goes into the Bright state. Otherwise, it goes into the Normal state. The state transitions of the controller are illustrated in Figure 1.

![Figure 1. Model of the intelligent light controller](image-url)
In this paper, a real-time system is formalized as a timed system consisting of a set of timed state sequences. Each sequence represents a possible execution path of the real-time system. The related concepts are introduced as follows:

- **Proposition set** $\text{Prop}$ is a (finite) set of atomic propositions. An observable state of a system can be interpreted by a subset of $\text{Prop}$ in which all true-valued propositions are included.

- A state sequence $\sigma$ over proposition set $\text{Prop}$ is an infinite or finite sequence $\sigma_0 \sigma_1 \sigma_2 \ldots$, where $\sigma_i \in 2^{\text{Prop}}$, for $i \in \mathbb{N}$. We use $\sigma(i)$ to denote $\sigma_i$ in the sequence.

- A time interval $I$ has one of the following forms: $[a,b)$ and $(a,\infty)$, where $a < b$ and $a,b \in \mathbb{R}_{\geq 0}$. The lower(upper) bound of the interval is represented by $l(I) (r(I))$. $|I| = r(I) - l(I)$ denotes the length of time interval $I$. If the time interval $I$ is unbounded, $|I|$ is infinite.

- A time interval sequence $T = I_0 I_1 I_2 \ldots$ is an infinite or finite sequence of time intervals. The length (number of time intervals) of a sequence $T$ is represented by $N(T)$ which can be finite or countable infinite. $T$ is adjacent, i.e. $r(I_i) = l(I_{i+1})$ for every $i < N(T)$ and $i \in \mathbb{N}$. $T$ is diverging, i.e. for any $t \geq l(I_0)$, there exists some $i \in \mathbb{N}$, such that $t \in I_i$. Hence, a finite time interval sequence ends with an unbounded interval.

- A metric $d$ on time interval sequences is given as follows$^1$.

\[
d(T,T') = \begin{cases} 
\sup \{|l(I_i) - l(I'_i)| \mid i < N(T) \land i \in \mathbb{N}\} & \text{if } N(T) = N(T') \\
\infty & \text{otherwise}.
\end{cases}
\]

$^1$In $\mathbb{R}$, the supremum ($\sup$) is only exists on bounded sets. In this definition, we define $\sup S = \infty$ when $S \subseteq \mathbb{R}_{\geq 0}$ and is unbounded.
2.2 Properties of real-time systems

 Basically temporal logics can be classified into branch-time logics and linear-time logics [5]. The main difference between them is that they are interpreted over different structures of states. Branch-time logics are interpreted over tree structures of states such as in $CTL$ [6, 7] and $TCTL$ [11]. Linear-time logics, on the other hand, are interpreted over trace semantics (linear structures of states) such as in $TPTL$ [4] and $MITL$ [3].

 In this paper, we adopt $MITL_{\mathbb{R}}$ (an extension of real-time logic $MITL$) to formalize properties of real-time systems. $MITL_{\mathbb{R}}$ formulas are formed by the following structures:

$$\varphi ::= p | \neg p | \varphi_1 \lor \varphi_2 | \varphi_1 \land \varphi_2 | \varphi_1 U_I \varphi_2 | \varphi_1 V_I \varphi_2$$

where $I$ is a time-bound interval of nonnegative reals. It takes one of the following forms: $[a, b], [a, b), (a, b), (a, b)$, $[a, \infty)$ and $(a, \infty)$, where $a \leq b$ for $a, b \in \mathbb{R}_{\geq 0}$. Now we define two scaling operators on time-bound interval. Let $\epsilon \in \mathbb{R}_{\geq 0}$, $\oplus$ and $\ominus$ are defined:

$$I \oplus \epsilon = \{ t \in \mathbb{R}_{\geq 0} \mid \exists t' \in I \land |t - t'| \leq \epsilon \}$$
$$I \ominus \epsilon = \{ t \in \mathbb{R}_{\geq 0} \mid \forall t' \in \mathbb{R}_{\geq 0} \land |t - t'| \leq \epsilon \rightarrow t' \in I \}$$

Informally speaking, $I \oplus \epsilon$ represents the time interval which elongates the end points of $I$ to $l(I) + \epsilon$ and $r(I) + \epsilon$ respectively in nonnegative reals. Similarly, $I \ominus \epsilon$ represents the time interval which shrinks the end points of $I$ to $l(I)$ and $r(I) - \epsilon$ respectively. Several examples are:

- if $I = [2\pi, 4\pi]$ then $I \oplus \pi = [\pi, 5\pi]$;
- if $I = (3, \infty)$ then $I \ominus 5 = [0, \infty)$;
- if $I = [1.1, 2]$ then $I \ominus 3 = \emptyset$.

$MITL_{\mathbb{R}}$ extends $MITL$ in the following aspects: In order to make model-checking feasible, there are two constraints on time-bound intervals in the $MITL$ formulas:

- Every time-bound interval $I$ should be nonsingular and nonempty.
- The end-points of $I$ are restricted to integer values.

Since we only use $MITL_{\mathbb{R}}$ to express real-time properties of a system, it is not necessary to employ the same constraints on the representation of time bound in $MITL_{\mathbb{R}}$ formulas.

The interpretation of $MITL_{\mathbb{R}}$ formulas over timed state sequences is standard and given in Definition 1.

Definition 1. Let $\pi$ be a timed state sequence and let $t \in \mathbb{R}_{\geq 0}$. For any $MITL_{\mathbb{R}}$ formula $\varphi$, the interpretation of $\varphi$ over $(\pi, t)$ is given as follows:

- $(\pi, t) \models p$ iff $p \in \pi(t)$;
- $(\pi, t) \models \neg p \text{ iff } p \notin \pi(t)$;
- $(\pi, t) \models \varphi_1 \lor \varphi_2$ iff $(\pi, t) \models \varphi_1$ or $(\pi, t) \models \varphi_2$;
- $(\pi, t) \models \varphi_1 \land \varphi_2$ iff $(\pi, t) \models \varphi_1$ and $(\pi, t) \models \varphi_2$;
- $(\pi, t) \models \varphi_1 U_I \varphi_2$ iff there is some $t_2 \in I$, such that $(\pi, t + t_2) \models \varphi_2$ and for all $0 \leq t_1 < t_2$, $(\pi, t + t_1) \models \varphi_1$;
- $(\pi, t) \models \varphi_1 V_I \varphi_2$ iff for all $t_2 \in I$, $(\pi, t + t_2) \models \varphi_2$ or there is some $0 \leq t_1 < t_2$, $(\pi, t + t_1) \models \varphi_1$.

In case that $I$ is empty, $(\pi, t) \models \varphi_1 U_I \varphi_2$ is always false and $(\pi, t) \models \varphi_1 V_I \varphi_2$ always holds. we use $(\pi, 0) \models \varphi$ ($\pi \models \varphi$, in short) to denote that a timed state sequence $\pi$ satisfies $MITL_{\mathbb{R}}$ formula $\varphi$. We extend the interpretation of $MITL_{\mathbb{R}}$ to sets of timed state sequences as follows.

Definition 2. Let $\varphi$ be an $MITL_{\mathbb{R}}$ formula, and let $T$ be a set of timed state sequences. $T \models \varphi$ iff for each timed state sequence $\pi \in T$, $\pi \models \varphi$.

Now we introduce two additional operators: $\Diamond_I \varphi$ (time-bounded eventually) and $\Box_I \varphi$ (time-bounded always) abbreviate $true \cup_I \varphi$ and $false \cup_I \varphi$ respectively.

---

2The $MITL$ logic employs the nonnegative real-valued time domain, but it constrains the endpoints of $I$ as integers.
2.3 Normalization of timed state sequences

A timed state sequence \( \tau = (\sigma, T) \) can be viewed as a function from a time domain \( \mathbb{R} \geq 0 \) to a state space \( 2^\text{Prop} \) [3]. Through this function, every time instant \( t \) along the time line is assigned a state \( \tau(t) \). From this point of view, some timed state sequences with different state sequences and time interval sequences may represent the same function and can not be discriminated by \( \text{MITL}_R \).

Example 2. Define two timed state sequences \( \tau \) and \( \tau' \) as follows. \( \tau = (\sigma, T) \) where for all \( i \in \mathbb{N}, \sigma(i) = \delta_i \) and \( T(i) = [2i, 2i + 2] \). \( \tau' = (\sigma', T') \) where for all \( i \in \mathbb{N}, \sigma'(2i) = \delta_i \) and \( T'(i) = [i, i+1] \). It is not hard to see that for any \( t \in \mathbb{R} \geq 0, \tau(t) = \tau'(t) \). Therefore, the timed state sequences \( \tau \) and \( \tau' \) represent the same function from time domain \( \mathbb{R} \geq 0 \) to state set \( \{\delta_i | i \in \mathbb{N}\} \).

Definition 3. Two timed state sequences \( \tau = (\sigma, T) \) and \( \tau' = (\sigma', T') \) are equivalent (\( \equiv \)) if for all \( t \in \mathbb{R} \geq 0, \tau(t) = \tau'(t) \).

Any timed state sequence \( \tau \) can be normalized to a timed state sequence \( \tau^* \), which is the normal form of \( \tau \), by performing the following operations. Along the state sequence of \( \tau \), successive identical states are replaced by one single state, and their corresponding time intervals are also merged into one single time interval. It can be shown that \( \tau \equiv \tau' \) iff they have the same normal form. In Example 2, if no two sequential states \( \delta_i \) and \( \delta_{i+1} \) are identical, we can conclude that \( \tau \equiv \tau' \).

Recall the interpretation of \( \text{MITL}_R \) formulas over uniform timed state sequences in Definition 1. Notice that the satisfaction relation \( \vdash \varphi \) depends on the mapping from the time domain to its state set only, so it is independent of various choices of time interval sequences.

Proposition 1. Let \( \varphi \) be an \( \text{MITL}_R \) formula. If two timed state sequences \( \tau \) and \( \tau' \) are equivalent, then \( \tau \vdash \varphi \) iff \( \tau' \vdash \varphi \).

Since \( \text{MITL}_R \) formulas cannot distinguish equivalent timed state sequences, we assume in the sequel that timed state sequences are in normal form if not explicitly stated otherwise.

2.4 Measuring timed state sequences

Let set \( S_{\text{Prop}} \) consist of all timed state sequences in normal form over a proposition set \( \text{Prop} \). We define another equivalence relation (\( \equiv_s \)) that divides \( S_{\text{Prop}} \) into groups sharing the same state sequence \( \tau \).

Definition 4. An equivalence relation (\( \equiv_s \)) on set \( S_{\text{Prop}} \) is defined as follows. For any two timed state sequences \( \tau, \tau' \in S_{\text{Prop}}, \) if \( \tau = (\sigma, T) \) and \( \tau' = (\sigma', T') \), then \( \tau \equiv_s \tau' \) iff \( \sigma = \sigma' \).

In order to evaluate the distance between two timed state sequences in \( S_{\text{Prop}} \), we adopt a metric \( d_{\text{sup}} \). For two timed sequences in the same partition of \( S_{\text{Prop}} \), \( d_{\text{sup}} \) is defined as the least upper bound of the absolute difference between time-stamps of corresponding state transitions in two state sequences. For example, Figure 2 shows two timed state sequences which have the same state sequence. There are four state transitions in each timed state sequence. \( d_{\text{sup}}(\tau_1, \tau_2) \) is computed as \( \sup\{0 - 0, |1.1 - 1.2|, |2.3 - 2.2|, |3.3 - 3.4|, |4.4 - 4.2| \} = 0.2 \).

\[
\tau_1; \begin{array}{ccccccc}
0 & 1.1 & 2.3 & 3.3 & 4.4 & \\
\end{array};
\tau_2; \begin{array}{ccccccc}
0 & 1.2 & 2.2 & 3.4 & 4.2 & \\
\end{array};
\]

Figure 2. Two finite timed state sequences

Definition 5. The metric \( d_{\text{sup}} \) on \( S_{\text{Prop}} \) is formally defined as follows. For \( \tau, \tau' \in S_{\text{Prop}} \),

\[
d_{\text{sup}}(\tau, \tau') = \begin{cases} d(T, T') & \text{if } \tau \equiv_s \tau' \\ \infty & \text{otherwise} \end{cases}
\]

where \( d(T, T') \), which is defined in Section 2.1, denotes the distance between two time interval sequences.

Gupta et al. proposed several metrics over finite timed state sequences in [12]. We only adopt \( d_{\text{sup}} \) (similar to the metric \( d_{\text{max}} \) in [12]) as a measure over \( S_{\text{Prop}} \) including both infinite and finite timed state sequences.

Definition 6. For \( \tau = (\sigma, T) \) and \( \epsilon \in \mathbb{R} \geq 0 \) we define an \( \epsilon \)-close set \( T^\tau_\epsilon \) \( \subseteq \{\tau \in S_{\text{Prop}} \mid d_{\text{sup}}(\tau, \tau') \leq \epsilon \} \).

Set \( T^\tau_\epsilon \) includes all timed state sequences whose distance from \( \tau \) is less than or equal to \( \epsilon \) in set \( S_{\text{Prop}} \).

3 Real-time property preservation

Real-time property preservation between timed state sequences is based on their distance in \( S_{\text{Prop}} \). Given a real-time property of one timed state sequence and (an upper bound of) its distance to another timed state sequence, a related property of the latter one is guaranteed without additional computation.

In this section, the results of real-time property preservation are presented in the following order.

1. The introduction of the \( \epsilon \)-neighbouring function (\( \epsilon \in \mathbb{R} \geq 0 \)) and the proof of its existence for two timed interval sequences whose distance is less than or equal
to $\epsilon$ based on metric $d_{sup}$ (see Section 3.1). In fact, $\epsilon$-neighbouring function between two timed interval sequences provide another representation of the distance between them.

2. The definition of a weakening function $R^\mu$ (parameterized with $\mu \in \mathbb{R}_{\geq 0}$) over the set of MILP$_2$ formulas in which $R^\mu(\varphi)$ is called the $\mu$-weakened formula of $\varphi$ (see Section 3.2).

3. The results of real-time property preservation are explained and proven (see Sections 3.3 and 3.4).

3.1 $\epsilon$-neighbouring function

Recall that metric $d_{sup}$ defines the distance between two timed state sequences in set $S_{Prop}$ by calculating the least upper bound of the absolute time difference of their corresponding state transitions. In other words, the distance between two timed state sequences which share the same state sequence is equal to the distance between their timed interval sequences.

Given two time interval sequences $\mathcal{T}$ and $\mathcal{T}'$, an $\epsilon$-neighbouring function establishes a one-to-one mapping from a time instant $t$ in $\mathcal{T}$ to a time instant $t'$ in $\mathcal{T}'$. The definition of the $\epsilon$-neighbouring function is given in Definition 7.

**Definition 7.** Let $\epsilon \in \mathbb{R}_{\geq 0}$. A function $F: \mathbb{R}_{\geq t(\ell_0)} \rightarrow \mathbb{R}_{\geq t'(\ell_0)}$ is called an $\epsilon$-neighbouring function from time interval sequence $\mathcal{T}$ to $\mathcal{T}'$, if and only if it has all of the following properties:

- **Interval consistency:** For every $t \in \mathbb{R}_{\geq t(\ell_0)}$, $t \in I_k$ implies $F(t) \in I'_k$ where $k < N(T)$ and $k \in \mathbb{N}$.

- **One to one mapping:** For all $t_1, t_2 \in \mathbb{R}_{\geq t(\ell_0)}$, $t_1 \neq t_2$ implies $F(t_1) \neq F(t_2)$. Furthermore, for every $t' \in \mathbb{R}_{\geq t'(\ell_0)}$, there exists a $t \in \mathbb{R}_{\geq t(\ell_0)}$, such that $t' = F(t)$.

- **Monotone increasing mapping:** For any $t_1, t_2 \in \mathbb{R}_{\geq t(\ell_0)}$, $t_1 > t_2$ implies $F(t_1) > F(t_2)$.

- **$\epsilon$-bound:** For every $t \in \mathbb{R}_{\geq t(\ell_0)}$, $|F(t) - t| \leq \epsilon$.

If there exists an $\epsilon$-neighbouring function $F$ from $\mathcal{T}$ to $\mathcal{T}'$, timed interval sequences $\mathcal{T}$ and $\mathcal{T}'$ are called $\epsilon$-neighbouring.

**Lemma 1.** If $F$ is an $\epsilon$-neighbouring function from $\mathcal{T}$ to $\mathcal{T}'$, then $F^{-1}$ is an $\epsilon$-neighbouring function from $\mathcal{T}'$ to $\mathcal{T}$.

---

**Figure 3.** Two time interval sequences

![Figure 3](image)

**Proof.** The lemma follows straightforwardly from the definition of $\epsilon$-neighbouring function.

**Example 3.** Given two time interval sequences $\mathcal{T} = I_1I_2I_3\ldots$ (where $I_k = [k, k+1)$) and $\mathcal{T}' = I'_1I'_2I'_3\ldots$ (where $I'_k = [k + 0.2, k + 1.2]$) as in Figure 3. $F(t) = t + 0.2$ ($t \in \mathbb{R}_{\geq 0}$) is a mapping from $\mathcal{T}$ to $\mathcal{T}'$. It is not hard to check that $F$ is a 0.2-neighbouring function from $\mathcal{T}$ to $\mathcal{T}'$. $F^{-1}(t) = t - 0.2$ ($t \in \mathbb{R}_{\geq -0.2}$) is a 0.2-neighbouring function from $\mathcal{T}'$ to $\mathcal{T}$.

It should be noticed that for any two time interval sequences and any $\epsilon$ ($\epsilon \in \mathbb{R}_{\geq 0}$), the $\epsilon$-neighbouring function does not always exist. For example, there is no 0.1-neighbouring function from $\mathcal{T}$ to $\mathcal{T}'$ defined in Example 3. The following theorem establishes the necessary and sufficient condition for the existence of an $\epsilon$-neighbouring function over two time interval sequences.

**Lemma 2.** If $F$ is an $\epsilon$-neighbouring function from $\mathcal{T}$ to $\mathcal{T}'$, then $F(l(I_i)) = l(I'_i)$, for all $i < N(T)$ and $i \in \mathbb{N}$.

**Proof.** Suppose $F(l(I_i)) = t$ and $t \neq l(I'_i)$. By the interval consistency property of $F$, it is easy to see that $t > l(I'_i)$. For any $l(I_i') < t' < l$ there exists $t''$ and $F(t'') = t'$. By the monotone increasing mapping property of $F$, it is easy to see that $t'' < l(I_i)$, which contradicts the interval consistency property of $F$.

**Theorem 1.** Let $\mathcal{T}$ and $\mathcal{T}'$ be two time interval sequences. $\mathcal{T}$ and $\mathcal{T}'$ are $\epsilon$-neighbouring, iff $d(\mathcal{T}, \mathcal{T}') \leq \epsilon$.

**Proof.** ($\Rightarrow$) It is not hard to prove by Lemma 2 and the $\epsilon$-bound property of the $\epsilon$-function.

($\Leftarrow$) Construct a function $G: \mathbb{R}_{\geq t(\ell_0)} \rightarrow \mathbb{R}_{\geq t'(\ell_0)}$ as follows.

$$G(t) = (t - l(I_k)) \frac{|I'_k|}{|I_k|} + l(I'_k), t \in I_k, k < N(T) and k \in \mathbb{N}$$

In case that both $|I_k|$ and $|I'_k|$ are finite, we define $|I'_k|/|I_k| = 1$.

As shown in Figure 4(b), the function $G$ is a monotone increasing linear function on every time interval. It is not hard to show that $G$ satisfies the first three properties of the $\epsilon$-neighbouring function.

Next we show that $G$ also satisfies the $\epsilon$-bound property.
For any $t \in \mathbb{R}^{|I(t_o)|}$, there exists a time interval $I_k$, $k \in \mathbb{N}$ such that $t \in I_k$. Construct a function $\text{diff}(t) = G(t) - t$. Since $\text{diff}(t)$ is a linear function within interval $I_k$, $|\text{diff}(t)| \leq \max\{|\text{diff}(l(I_k))|, \lim_{t \to r(I_k)} |\text{diff}(t)|\}$. By the continuity of $\text{diff}(t)$ and the adjacent property of time interval sequence $T$, it is not hard to see that $|\text{diff}(t)| \leq \max\{|\text{diff}(l(I_k))|, |\text{diff}(l(I_{k+1}))|\}$. By the definition of $G$, it is easy to prove that $\text{diff}(l(I_k)) = l(I'_k) - l(I_k)$. Since $d_{\text{sup}}(T, T') < \epsilon$, by the definition of $d_{\text{sup}}$, we can see $|\text{diff}(t)| < \epsilon$. Hence, $G$ satisfies the $\epsilon$-bound property.

We have shown that $G$ is an $\epsilon$-neighbouring function from $T$ to $T'$ and hence, $T$ and $T'$ are $\epsilon$-neighbouring.

From Theorem 1, we can see that the existence of $\epsilon$-neighbouring functions between two time interval sequences give another characterization of the distance between them.

### 3.2 Weakening functions $R^\mu$ over formulas

$MITL_\mathbb{R}$ incorporates quantitative timing constraints in its operators which enables $MITL_\mathbb{R}$ formulas to express quantitative timing properties. At the same time, these quantitative timing constraints in the formulas can also be used to show weakening or strengthening relation between formulas. For example, in $MITL_\mathbb{R}$, $p \cup_{[0, \mu]} q$ specifies a real-time property that $q$ happens within $\mu$ time after $p$. For different values of $\mu$, formulas have different quantitative timing constraints. Formula $p \cup_{[0, \mu]} q$ has stronger requirement on the issuing time of $q$ than formula $p \cup_{[0, 2]} q$, because the satisfaction of the former formula always implies the satisfaction of the latter one. In Definition 8, function $R^\mu$ defines such a weakening function on formulas. It offers a quantitative way to weaken timing requirements of $MITL_\mathbb{R}$ formulas.

**Definition 8.** Let $\mu \in \mathbb{R}^\geq 0$. The weakening function $R^\mu : MITL_\mathbb{R} \rightarrow MITL_\mathbb{R}$ is defined as follows:

- $R^\mu(p) = p$;
- $R^\mu(\neg p) = \neg p$;
- $R^\mu(p \lor q) = R^\mu(p) \lor R^\mu(q)$;
- $R^\mu(p \land q) = R^\mu(p) \land R^\mu(q)$;
- $R^\mu(p \land \neg q) = R^\mu(p) \land \neg R^\mu(q)$;
- $R^\mu(p \lor \neg q) = R^\mu(p) \lor \neg R^\mu(q)$.

For every $\mu \in \mathbb{R}^\geq 0$, $R^\mu$ defines a function over $MITL_\mathbb{R}$. $R^\mu(\phi)$ relaxes the quantitative timing constraints in formula $\phi$ and is called the $\mu$-weakened formula of $\phi$. Next, we will prove that formula $R^\mu(\phi)$ is indeed weaker than formula $\phi$.

**Lemma 3.** For any $\mu \in \mathbb{R}^\geq 0$, $t \in \mathbb{R}^\geq 0$, $\phi \in MITL_\mathbb{R}$ and $\tau \in S_{\text{Prop}}$ if $(\tau, t) \models \phi$, then $(\tau, t) \models R^\mu(\phi)$.

**Proof.** Suppose $(\tau, t) \models \phi$. We will show that $(\tau, t) \models R^\mu(\phi)$ by induction on the structure of formula $\phi$.

**Case 1:** $\phi = p$.
By definition of function $R^\mu$, $R^\mu(\phi) = p$. Hence, $(\tau, t) \models R^\mu(\phi)$.

**Case 2:** $\phi = \neg p$.
The proof is similar to the pervious case.

**Case 3:** $\phi = \phi_1 \lor \phi_2$.
By the interpretation of $MITL_\mathbb{R}$ over timed state sequences, $(\tau, t) \models \phi_1$ or $(\tau, t) \models \phi_2$. By induction we thus have $(\tau, t) \models R^\mu(\phi_1)$ or $(\tau, t) \models R^\mu(\phi_2)$. Hence, $(\tau, t) \models R^\mu(\phi_1) \lor R^\mu(\phi_2) = R^\mu(\phi)$.

**Case 4:** $\phi = \phi_1 \land \phi_2$.
The proof of this case is similar to the previous case.

**Case 5:** $\phi = \phi_1 \lor \neg \phi_2$.
There is some $t_2 \in I$, such that $(\tau, t + t_2) \models \phi_2$ and for all $0 \leq t_1 < t_2$, $(\tau, t + t_1) \models \neg \phi_1$. It is obvious that $t_2 \in I \cup \mu$. By induction we have $(\tau, t + t_2) \models R^\mu(\phi_2)$ and for all $0 \leq t_1 < t_2$, $(\tau, t + t_1) \models R^\mu(\phi_1)$. Hence, $(\tau, t) \models R^\mu(\phi_1) \lor R^\mu(\phi_2) = R^\mu(\phi)$.

**Case 6:** $\phi = \phi_1 \land \neg \phi_2$.
For all $t_2 \in I$, $(\tau, t + t_2) \models \phi_2$ or there is some $0 \leq t_1 < t_2$, $(\tau, t + t_1) \models \phi_1$. By induction we thus have for all $t_2 \in I \cup \mu$, $(\tau, t + t_2) \models R^\mu(\phi_2)$ or there is some
0 ≤ t₁ < t₂, (τ, t + t₁) ∨ R^⁰(ϕ₁). Hence, (τ, t) ∨ R^⁰(ϕ₁) = R^⁰(ϕ₁) = R^⁰(ϕ) .
This completes our inductive proof of Lemma 3. □

Theorem 2. (Relaxation property of R^⁰) For any μ ∈ R^≥₀, ϕ ∈ MITL_R and τ ∈ S^prop, if τ ∨ ϕ, then τ ∨ R^⁰(ϕ).

Proof. Follows directly from Lemma 3. □

In general, the larger the value of μ is, the weaker is formula R^⁰(ϕ).

3.3 Real-time property preservation between timed state sequences

In the previous sections, the ε-neighbouring function is used to characterize the distance between two neighbouring timed state sequences in set S^prop. Then we introduced the weakening functions R^⁰ to quantitatively weaken MITL_R formulas. In this section, we investigate the properties that are preserved between neighbouring sequences in S^prop by employing the above two functions.

Lemma 4. Let ε ∈ R^≥₀, t ∈ R^≥₀ and ϕ ∈ MITL_R.
Further let τ and τ’ be two ε-neighbouring timed state sequences and let G be an ε-neighbouring function from the time interval sequence of τ to that of τ’.
Then (τ, t) ∨ ϕ implies (τ’, G(t)) ∨ R^ε(ϕ).

Proof. We show that (τ’, G(t)) ∨ R^ε(ϕ) by induction on the structure of formula ϕ.

Case 1: ϕ = p.
By definition of function R^ε, R^ε(ϕ) = p. By the interpretation of MITL_R formulas over timed state sequences, p ∈ τ(t).
Since G is an ε-neighbouring function from the time interval sequence of τ to that of τ’ and since τ and τ’ share the same state sequence, we know that (τ(t) = τ’(G(t))) by the interval consistency property of G. Hence, p ∈ τ’(G(t)). But then, (τ’, G(t)) ∨ R^ε(ϕ).

Case 2: ϕ = ¬p.
The proof is similar to the previous case.

Case 3: ϕ = ϕ₁ ∨ ϕ₂.
(τ, t) ∨ ϕ₁ or (τ, t) ∨ ϕ₂. By induction we have (τ’, G(t)) ∨ R^ε(ϕ₁) or (τ’, G(t)) ∨ R^ε(ϕ₂). But then (τ’, G(t)) ∨ R^ε(ϕ₁) ∨ R^ε(ϕ₂) = R^ε(ϕ).

Case 4: ϕ = ϕ₁ ∧ ϕ₂.
The proof is similar to the previous case.

Case 5: ϕ = ϕ₁ Uϕ₂.
There is some t₂ ∈ I, such that (τ, t + t₂) ∨ ϕ₂ and for all 0 ≤ t₁ < t₂, (τ, t + t₁) ∨ ϕ₁. By induction we have (τ’, G(t + t₂)) ∨ R^ε(ϕ₂). For any 0 ≤ t’₁ < G(t + t₂) - G(t), by the one to one mapping and monotone increasing mapping properties of G, there is a 0 ≤ t’₁ < t₂, such that G(t + t’₁) = G(t) + t’₁. By induction we have (τ’, G(t + t’₁)) ∨ R^ε(ϕ₁). Hence (τ’, G(t + t’₁)) ∨ R^ε(ϕ₁).

Now, we will show that G(t + t₂) - G(t) ∈ I ⊕ 2ε. By the Triangle Inequality and the ε-bound property of G,

G(t + t₂) - G(t) = G(t + t₂) - (t + t₂) - (G(t) - t) + t₂ ≤ |G(t + t₂) - (t + t₂)| + |G(t) - t| + t₂ ≤ ε + ε + t₂ = t₂ + 2ε

Similarly, it is easy to prove that G(t + t₂) - G(t) ≥ t₂ - 2ε. By the monotone increasing mapping property of G, we know G(t + t₂) - G(t) ≥ 0. Therefore, max(t₂ - 2ε, 0) ≤ G(t + t₂) - G(t) ≤ t₂ + 2ε. Since t₂ ∈ I, it is not hard to check that G(t + t₂) - G(t) ∈ I ⊕ 2ε.

Hence, (τ’, G(t)) ∨ R^ε(ϕ₁)U₁⊗₂εR^ε(ϕ₂) = R^ε(ϕ).

Case 6: ϕ = ϕ₁ V₁ϕ₂.
The proof is similar to the previous case. A brief proof is given as follows.

For all t’₂ ∈ I ⊕ 2ε, it is not hard to prove that there exist t₂ ∈ I such that G(t₂ + t) = t₂’ + G(t). There are two possibilities:

• (τ, t + t₂) ∨ ϕ₁. By induction we have (τ’, G(t + t₂)) ∨ R^ε(ϕ₂), that is, (τ’, G(t) + t’) ∨ R^ε(ϕ₂).

• There is some 0 ≤ t₁ < t₂, such that (τ, t + t₁) ∨ ϕ₁. Let t’₁ = G(t₁+t₁) - G(t) and it is not hard to prove that 0 ≤ t’₁ < t’₂. By induction we have (τ’, G(t) + t’₁) ∨ R^ε(ϕ₁).

Hence, (τ’, G(t)) ∨ R^ε(ϕ₁)U₁⊗₂εR^ε(ϕ₂) = R^ε(ϕ).

This completes our inductive proof of Lemma 4. □

Theorem 3. Let ε ∈ R^≥₀, t ∈ R^≥₀, and ϕ ∈ MITL_R. Further let τ and τ’ be two ε-neighbouring timed state sequences then τ ∨ ϕ implies τ’ ∨ R^ε(ϕ).

Proof. It is not hard to prove by Lemma 4 and Lemma 2. □

Recall Definition 2 and Definition 6, for any timed state sequence τ ∈ S^prop, T^σ is set which contains all timed state sequences whose distance from (τ, t) is less than or equal to ε. We can extend the real-time preservation between two timed state sequences to sets of timed state sequences. This extension is given in Corollary 1.

Corollary 1. For any ε ∈ R^≥₀, ϕ ∈ MITL_R, τ ∈ S^prop, if τ ∨ ϕ then T^σ ∨ R^ε(ϕ).
3.4 Real-time property preservation between timed systems

In the previous part of this paper, we concentrated on real-time property preservation between timed state sequences. However, in practice we often need to analyse real-time properties of a timed system instead of those of a single timed state sequence. Since a timed system consists of a set of timed state sequences, the problem of examining the satisfaction of a formula \( \varphi \) in a timed system \( S \) is equivalent to examining its satisfaction in all of the timed state sequences in \( S \). Real-time property preservation between timed state sequences can also be extended to analyse real-time properties between timed systems. The following theorems give these realizations.

**Theorem 4.** Let \( S \) be a set of timed state sequences and let \( \varphi \) be an MITL\(_R\) formula. For any \( \epsilon \in \mathbb{R}^{\geq 0} \), if \( S \models \varphi \), then \( S \models R^\epsilon(\varphi) \).

**Proof.** Follows from Definition 2 and Theorem 2.

**Theorem 5.** Let \( S_1 \), \( S_2 \) be two sets of timed state sequences and let \( \varphi \) be an MITL\(_R\) formula. Assume there is some \( \epsilon \in \mathbb{R}^{\geq 0} \), such that for any timed state sequence \( \tau \) in \( S_2 \) there exists a sequence \( \tau' \) in \( S_1 \) such that \( \tau \) and \( \tau' \) are \( \epsilon \)-neighbouring. Then \( S_1 \models \varphi \) implies \( S_2 \models R^\epsilon(\varphi) \).

**Proof.** The theorem follows from Definition 2 and Theorem 3.

4 Problem revisited

As described in Section 1, the over-specification problem often leads to the failure of transferring verification results from a model to its realization. The issuing time of event \( p \) in the realization may have a \( \delta \) time shift from the time specified in the model \( M \). Time shifts can be different for each event in the realization. Although it is often impossible to know the exact value of time shift \( \delta \) for each pair of corresponding events in the model \( M \) and its physical realization \( R \), the upper bound of all \( \delta \) can be more easily estimated. Then for any execution trace \( \tau \) in \( R \), we can find an \( \epsilon \)-neighbouring execution trace \( \tau_m \) in \( M \). In that case, we can establish properties of \( S \) based on the verification results of \( M \).

Reconsider the example given in Section 1. Assume \( p, q \) and \( r \) are three atomic propositions as follows:

- \( p \) : The controller is in the \textit{Wait} state.
- \( q \) : The controller is in the \textit{Bright} state.
- \( r \) : The controller is in the \textit{Normal} state.

A real-time property \( \varphi = \Box(p \rightarrow (\Diamond_{[0,2]} q \lor \Diamond_{[2,\infty]} r)) \) can be described as follows: when a click happens at the \textit{Light\_off} state, it either goes into \textit{Bright} state within 2 seconds or goes into \textit{Normal} state after 2 seconds. It is not hard to check that this property holds in the model.

Suppose that an upper bound of the time shift in the realization is 0.01 second. It is not hard to check that for any execution trace \( \tau \) of such a realization, we can always find an execution trace \( \tau' \) whose distance from \( \tau \) is less or equal to 0.01. Therefore, real-time property \( R^{0.02}(\varphi) = \Box(p \rightarrow (\Diamond_{[0,2]} q \lor \Diamond_{[1,\infty]} r)) \) is satisfied in this realization. Notice that the result also can be applied the other way around.

If our objective is to check the satisfaction of the real-time property \( R^{0.02}(\varphi) = \Box(p \rightarrow (\Diamond_{[0,2]} q \lor \Diamond_{[1,\infty]} r)) \) in the realization, we can check a stronger property \( \Box(p \rightarrow (\Diamond_{[0,2]} q \lor \Diamond_{[1,\infty]} r)) \) in the model, where \( \epsilon \) is used to measure the time difference between the model and the realization.

5 Discussion and Conclusion

In this paper, we have investigated the preservation of real-time properties between timed systems. For some special cases, a better property preservation result might be achieved. This can be illustrated in the following example. Let \( \tau_1 \) and \( \tau_2 \) be two \( \epsilon \)-neighbouring timed state sequences. The issuing time of any state transition in \( \tau_1 \) is always no later (or no earlier) than the corresponding issuing time in \( \tau_2 \). Then \( \tau_1 \models \varphi \) implies that \( \tau_2 \models R^\epsilon(\varphi) \), in which \( R^\epsilon(\varphi) \) is a stronger formula than \( R^\epsilon(\varphi) \). The proof of this is similar to the proof of Theorem 3.

Since interleaving semantics is not supported in this paper, which sequentialize concurrent actions as a sequence of actions occurring at the same time instant, we are investigating the real-time property preservation with interleaving semantics for concurrent systems.

Another application of real-time property preservation is real-time software synthesis as suggested in [8]. In the model, actions/events are formalized as instantaneous events instead of events that have some time duration in the realization. If the synthesis method can guarantee that the absolute difference of the issuing time of the corresponding events in both the realization and the model is always less than or equal to a constant, we can use the real-time properties of the realization based on those of the model. We are extending and exploring this application and aim at a systematic approach to develop reliable real-time systems.
References


