Bounded-Memory Runtime Enforcement of Timed Properties

This repository contains additional material for the work titled "Bounded-Memory Runtime Enforcement of Timed Properties". This work defines a bounded-memory RE framework for a regular timed property specified as a timed automaton. Specifically, this appendix contains proofs of the propositions included in the paper.

Definition 4 is recalled below:

An enforcer for a timed property $\varphi \subseteq tw(\Sigma)$ is a function $E^\varphi : tw(\Sigma) \to tw(\Sigma)$, satisfying the following constraints:

Soundness	**(Snd)**	$\forall \sigma \in tw(\Sigma) : E^\varphi(\sigma) \models \varphi \vee E^\varphi(\sigma) = \epsilon$
Monotonicity	**(Mo)**	$\forall \sigma, \sigma' \in tw(\Sigma) : \sigma \preccurlyeq \sigma' \implies E^\varphi(\sigma) \preccurlyeq E^\varphi(\sigma')$
Transparency	**(Tr1)**	$Tr_1 : \forall \sigma \in tw(\Sigma), \text{delayable1}_\varphi(\sigma) = \emptyset \implies E^\varphi(\sigma) \triangleleft_d \sigma$
	(Tr2)	$Tr_2 : \forall \sigma \in tw(\Sigma), \text{delayable1}_\varphi(\sigma) \neq \emptyset \implies E^\varphi(\sigma) \preccurlyeq_d \sigma$

Definition 5 is recalled below:

The enforcer for a property $\varphi \subseteq tw(\Sigma)$ is the function $E^\varphi : tw(\Sigma) \to tw(\Sigma)$ defined as: $\forall \sigma \in tw(\Sigma), \forall t \in \mathbb{R}_{\geq 0}, \forall a \in \Sigma$,

$$E^\varphi(\sigma) = \Pi_1(\text{store}^\varphi(\sigma)), \text{ where}$$

$\text{store}^\varphi : tw(\Sigma) \to tw(\Sigma) \times tw(\Sigma)$ is defined as:

- $\text{store}^\varphi(\epsilon) = (\epsilon, \epsilon)$

$$\text{store}^\varphi(\sigma \cdot (t,a)) = \begin{cases} (\sigma_s \cdot min_{\preceq_{lex},end}(k^\varphi(\sigma_s, \sigma_{ca})), \epsilon) & \text{if } k^\varphi(\sigma_s, \sigma_{ca}) \neq \emptyset, \\ (\sigma_s, \sigma_c) & \text{if } k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) = \emptyset, \\ (\sigma_s, \sigma_{ca}) & \text{otherwise} \end{cases}$$

- with $(\sigma_s, \sigma_c) = \text{store}^\varphi(\sigma)$; $\quad \sigma_{ca} = \sigma_c \cdot (t,a)$

Definition 11 is recalled below:

A bounded enforcer for a timed property $\varphi \subseteq tw(\Sigma)$, equipped with a buffer of size k, is a function $E^{\varphi,k}$ satisfying the following constraints: $E^{\varphi,k} : tw(\Sigma) \to tw(\Sigma) \times \{\bot, \top, stop\}$

Soundness	**(SndB)**	$\forall \sigma \in tw(\Sigma) : E^{\varphi,k}_{out}(\sigma) \models \varphi \vee E^{\varphi,k}_{out}(\sigma) = \epsilon$
Monotonicity	**(Mo1B)**	$Mo_1 : \forall \sigma, \sigma' \in tw(\Sigma) : \sigma \preccurlyeq \sigma' \implies E^{\varphi,k}_{out}(\sigma) \preccurlyeq E^{\varphi,k}_{out}(\sigma')$
	(Mo2B)	$Mo_2 : \forall \sigma, \sigma' \in tw(\Sigma) : \sigma \preccurlyeq \sigma', (E^{\varphi,k}_{\text{mode}}(\sigma) = \bot \implies E^{\varphi,k}_{\text{mode}}(\sigma') = \bot)$
Transparency	**(Tr1B)**	$Tr_1 : \forall \sigma \in tw(\Sigma), \text{delayable1}_\varphi(\sigma) = \emptyset \vee E^{\varphi,k}_{\text{mode}}(\sigma) = \bot \implies E^{\varphi,k}_{out}(\sigma) \triangleleft_d \sigma$
	(Tr2B)	$Tr_2 : \forall \sigma \in tw(\Sigma), \text{delayable1}_\varphi(\sigma) \neq \emptyset \wedge E^{\varphi,k}_{\text{mode}}(\sigma) = \top \implies E^{\varphi,k}_{out}(\sigma) \preccurlyeq_d \sigma$

Definition 10 is recalled below:

A bounded enforcer for a property $\varphi \subseteq tw(\Sigma)$ is the function $E^{\varphi,k} : tw(\Sigma) \rightarrow tw(\Sigma) \times \{\top, \bot, stop\}$, and is defined as:
$\forall \sigma \in tw(\Sigma), \forall t \in \mathbb{R}_{\geq 0}, \forall a \in \Sigma,$

$$E^{\varphi,k}(\sigma) = (\Pi_1(\text{store}^{\varphi,k}(\sigma)), \Pi_3(\text{store}^{\varphi,k}(\sigma))), \text{ where:}$$

$\text{store}^{\varphi,k} : tw(\Sigma) \rightarrow tw(\Sigma) \times tw(\Sigma) \times \{\top, \bot, stop\}$ is defined as:

- $\text{store}^{\varphi,k}(\epsilon) = (\epsilon, \epsilon, \top)$
- $\text{store}^{\varphi,k}(\sigma \cdot (t, a)) =$

$$\begin{cases} (\sigma_s \cdot min_{\preceq_{lex},end}(k^{\varphi}(\sigma_s, \sigma_{ca})), \epsilon, mode \setminus \{stop\}) & \text{if } k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset, \\ (\sigma_s, \sigma_c, \bot) & \text{if } k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) = \emptyset, \\ (\sigma_s, \sigma_{ca}, mode \setminus \{stop\}) & \text{if } k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset \ \wedge |\sigma_{ca}| \leq k \\ (\sigma_s, \sigma_c, stop) & \text{if } k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset \ \wedge \ |\sigma_{ca}| > k \\ & \quad \wedge \text{ Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) = \emptyset \\ (\sigma_s, \text{Clean}^{\varphi,k}(\sigma_s, \sigma_{ca}), \bot) & \text{if } k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset \ \wedge |\sigma_{ca}| > k \\ & \quad \wedge \text{ Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) \neq \emptyset \end{cases}$$

- $(\sigma_s, \sigma_c, \{\top, \bot\}) = \text{store}^{\varphi,k}(\sigma)$, and $\sigma_{ca} = \sigma_c \cdot (t, a)$
- $E^{\varphi,k}_{\text{out}}(\sigma) = \Pi_1(E^{\varphi,k}(\sigma))$, $E^{\varphi,k}_{\text{mode}}(\sigma) = \Pi_3(E^{\varphi,k}(\sigma))$, and $\text{buff}(E^{\varphi,k}(\sigma)) = \Pi_2(E^{\varphi,k}(\sigma))$
- $\text{Clean}^{\varphi,k} : tw(\Sigma) \times tw(\Sigma) \rightarrow tw(\Sigma)$

$$\text{Clean}^{\varphi,k}(\sigma_s, \sigma_{ca}) = \sigma' \in \text{Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) : \forall \sigma'' \in \text{Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}),$$
$$\sigma' \neq \sigma'' \ \wedge \ |\sigma'| > |\sigma''| \ \wedge \ (\text{index}(\sigma', \sigma_{ca}) \leq \text{index}(\sigma'', \sigma_{ca}))$$

- $\text{index}(\sigma', \sigma_{ca}) = (i \in \mathbb{N} \mid i \in [1, |\sigma_{ca}|] : \sigma_{ca[i]} \neq \sigma'_{[i]})$
- $\text{Get\_SW}^{\varphi,k} : tw(\Sigma) \times tw(\Sigma) \rightarrow 2^{tw(\Sigma)}$

$$\text{Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) = \{\sigma'' \in tw(\Sigma) \mid \exists \sigma' \in \text{delayable2}^{\varphi}(\sigma_s, \sigma_{ca}) \wedge$$
$$\exists i, j, k \in \mathbb{N} \wedge 1 \leq i \leq j < k :$$
$$\sigma'' = \sigma'_{[1\ldots i-1]} \cdot \sigma'_{[j+1\ldots k]} \wedge \sigma'_{[1\ldots i-1]} \cdot \sigma'_{[i\ldots j]} \cdot \sigma'_{[j+1\ldots k]} \sim_{\varphi} \sigma''\}$$

C.1 Sketch of the proof (of Proposition 6)

Proposition 6: Given some property φ, its enforcement function E^{φ} as per Definition 5 satisfies the Soundness, Monotonicity, and Transparency constraints as per Definition 4.

Sketch of the proof of proposition 6:

The proof of **Mo** is straightforward by noticing that function store^{φ} is monotonic on its first output $(\forall \sigma, \sigma' \in tw(\Sigma) : \sigma \preccurlyeq \sigma' \implies \Pi_1(\text{store}^{\varphi}(\sigma)) \preccurlyeq \Pi_1(\text{store}^{\varphi}(\sigma')))$.
Let us prove **Snd**, **Tr1**, and **Tr2** using induction on the input sequence σ.
Induction basis. If $\sigma = \epsilon$, $\text{store}^{\varphi}(\epsilon) = (\epsilon, \epsilon)$. The proposition holds trivially.
Induction step. We assume for $\sigma \in tw(\Sigma)$, if $\text{store}^{\varphi}(\sigma) = (\sigma_s, \sigma_c)$ and $\sigma_c \neq \epsilon$, then this proposition holds.
Let us prove, it holds for all $(t, a) \in (\mathbb{R}_{\geq 0} \times \Sigma)$. Following definition of store^{φ}, We distinguish two cases:

- Case 1: if $k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$
  In this case, we have $E^{\varphi}(\sigma \cdot (t, a)) = \sigma_s \cdot min_{\preceq_{lex},end}(k^{\varphi}(\sigma_s, \sigma_{ca}))$ and from definition of k^{φ}, we have $k^{\varphi}(\sigma_s, \sigma_{ca}) \in \sigma_s^{-1} \cdot \varphi$. Thus, $E^{\varphi}(\sigma \cdot (t, a)) \models \varphi$ and **Snd** holds.
  Since, $k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$, $\text{delayable1}_{\varphi}(\sigma_{ca}) \neq \emptyset$, Thus, **Tr1** holds trivially.
  Since, $\text{delayable}_{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$ and $E^{\varphi}(\sigma \cdot (t, a)) = \sigma_s \cdot min_{\preceq_{lex},end}(k^{\varphi}(\sigma_s, \sigma_{ca})) \preccurlyeq_d \sigma$ (from Definition of k^{φ}), **Tr2** holds.

- Case 1: if $k^{\varphi}(\sigma_s, \sigma_{ca}) = \emptyset$
  In this case, we have $E^{\varphi}(\sigma \cdot (t,a)) = \sigma_s$. Since, from induction hypothesis, we assume that the proposition holds for σ, meaning when $E^{\varphi}(\sigma) = \sigma_s$, **Snd**, **Tr1**, and **Tr2** holds; Thus, when $E^{\varphi}(\sigma \cdot (t,a)) = \sigma_s$ **Snd**, **Tr1**, and **Tr2** will hold too.

C.2 Sketch of the proof (of Proposition 7)

Proposition 7 [Optimal Suppression]: Given some property φ, its enforcement function E^{φ} as per Definition 5 satisfies the following constraint:

$$\begin{aligned}
&\forall \sigma \in tw(\Sigma),\\
&\quad \exists \sigma_s, \sigma_c \in tw(\Sigma) : \text{store}^{\varphi}(\sigma) = (\sigma_s, \sigma_c) \wedge\\
&\quad \forall (t,a) \in (\mathbb{R}_{\geq 0} \times \Sigma),\ t \geq end(\sigma_c) : \\
&\quad (\text{delayable2}_{\varphi}(\sigma_s, \sigma_c \cdot (t,a)) = \emptyset \implies \forall \sigma_{\text{con}} \in tw(\Sigma) : start(\sigma_{\text{con}}) \geq t,\\
&\qquad\qquad E^{\varphi}(\sigma \cdot (t,a) \cdot \sigma_{\text{con}}) = E^{\varphi}(\sigma \cdot \sigma_{\text{con}}))
\end{aligned} \tag{\textbf{Opts}}$$

Sketch of the proof of Proposition 7:

Let us prove this using induction on the input sequence σ.
Induction basis. If $\sigma = \epsilon$, $\text{store}^{\varphi}(\epsilon) = (\epsilon, \epsilon)$, the proposition holds trivially.
Induction step. We assume for $\sigma \in tw(\Sigma)$, $\text{store}^{\varphi}(\sigma) = (\sigma_s, \sigma_c)$.
Let us proove, it holds for all $(t,a) \in (\mathbb{R}_{\geq 0} \times \Sigma)$. Following definition of store^{φ}, we have three cases:

- Case 1: if $k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$
  In this case, we will have a set of delayed words w of $\sigma_c \cdot (t,a)$ s.t. $\sigma_s \cdot w \in \varphi$. Thus, $\text{delayable2}_{\varphi}(\sigma_s, \sigma_c \cdot (t,a))$ will also be $\neq \emptyset$. Thus, proposition holds trivially.
- Case 2: if $k^{pref}(\sigma_s, \sigma_{ca}) = \emptyset$
  Since $k^{pref}(\sigma_s, \sigma_{ca}) = \emptyset$, $\text{delayable}_{\varphi}$ will also be $= \emptyset$, and according to **Opts** the event (t,a) should be suppressed. and, case 2 of store^{φ} is also suppressing the received event (t,a), thus, proposition holds.
- Case 3: if $k^{\varphi}(\sigma_s, \sigma_{ca}) = \emptyset$ and $k^{pref}(\sigma_s, \sigma_{ca}) \neq \emptyset$
  Since, $k^{pref}(\sigma_s, \sigma_{ca}) \neq \emptyset$, thus, $\text{delayable2}_{\varphi}(\sigma_s, \sigma_c \cdot (t,a))$ will also be $\neq \emptyset$. Thus, proposition holds trivially.

C.3 Sketch of the proof (of Proposition 12)

Proposition 12: Given some property φ and the maximum buffer size k, let $n \in \mathbb{N}$ be the number of locations in $\mathcal{A}_{\varphi}$. If $k \geq n$, then the enforcement function $E^{\varphi,k}$ as per Definition 11 satisfies the Soundness, Monotonicity, and Transparency constraints as per Definition 10.

Sketch of the proof of proposition 12:

The proof of **Mo1B** is straightforward by noticing that function $\text{store}^{\varphi,k}$ is monotonic on its first output $(\forall \sigma, \sigma' \in tw(\Sigma) : \sigma \preccurlyeq \sigma' \implies \Pi_1(\text{store}^{\varphi,k}(\sigma)) \preccurlyeq \Pi_1(\text{store}^{\varphi,k}(\sigma')))$.
The proof of **Mo2B** is straightforward by noticing that function $\text{store}^{\varphi,k}$ does not bring the nominal mode back in any of the cases once the mode has changed to $\perp$.
Let us now prove **SndB**, **Tr1B**, and **Tr2B** by an induction on the length of the input timed word σ.

Induction basis. If $\sigma = \epsilon$, $\text{store}^{\varphi,k}(\epsilon) = (\epsilon, \epsilon, \top)$. Since, $E^{\varphi,k}_{\text{out}}(\sigma) = \Pi_1(E^{\varphi,k}(\sigma)) = \Pi_1(\text{store}^{\varphi,k}(\sigma) = \epsilon$, thus **SndB**, **Tr1B**, and **Tr2B** holds trivially.
Induction step. We assume for $\sigma \in tw(\Sigma)$, if $\text{store}^{\varphi,k}(\sigma) = (\sigma_s, \sigma_c)$ and $\sigma_c \neq \epsilon$, then this proposition holds.
Let us prove, it holds for all $(t, a) \in (\mathbb{R}_{\geq 0} \times \Sigma)$. Following definition of $\text{store}^{\varphi,k}$, we have following cases to examine:

- Case 1: $k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$
  $E^{\varphi,k}_{\text{out}}(\sigma \cdot (t, a)) = \sigma_s \cdot min_{\preceq_{lex},end}(k^{\varphi}(\sigma_s, \sigma_{ca})$.
  Since, from definition of k^{φ}, $k^{\varphi}(\sigma_s, \sigma_{ca}) \in \sigma_s^{-1} \cdot \varphi$, thus, **SndB** holds.
  Since, $k^{\varphi}(\sigma_s, \sigma_{ca}) \neq \emptyset$, $\text{delayable1}_{\varphi}(\sigma_{ca})$ will also be $\neq \emptyset$ and $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = mode \setminus \{stop\} = \top$, Thus, **Tr1B** holds trivially.
  Since, $\text{delayable1}_{\varphi}(\sigma_{ca}) \neq \emptyset$ and $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = mode \setminus \{stop\} = \top$ and $E^{\varphi,k}(\sigma \cdot (t, a)) = \sigma_s \cdot min_{\preceq_{lex},end}(k^{\varphi}(\sigma_s, \sigma_{ca})) \preccurlyeq_d \sigma \cdot (t, a)$ (from Definition of k^{φ}), **Tr2B** holds.

- Case 2: $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) = \emptyset$
  Since, from induction hypothesis, for σ, $E^{\varphi,k}(\sigma) = \sigma_s$ we assumed that **SndB** holds; and in this case, we have $E^{\varphi,k}(\sigma \cdot (t, a)) = \sigma_s$, thus **SndB** holds here too.
  Since, $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) = \emptyset$ thus $\text{delayable1}_{\varphi}(\sigma_{ca}) = \emptyset$, thus, **Tr2B** holds trivially.
  Since, $\text{delayable1}_{\varphi}(\sigma_{ca}) = \emptyset$, $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = \perp$, and (t, a) have been suppressed, thus $E^{\varphi,k}_{\text{out}}(\sigma \cdot (t, a)) = \sigma_s \triangleleft_d \sigma \cdot (t, a)$, thus **Tr1B** holds.

- Case 3: $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset\ \wedge |\sigma_{ca}| \leq k$
  Same as case 2, **SndB** holds here too.
  Since, $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset$, thus $\text{delayable1}_{\varphi}(\sigma_{ca}) \neq \emptyset$, and $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = mode \setminus \{stop\} = \top$. Thus, **Tr1B** holds trivially.
  Also, $\text{delayable1}_{\varphi}(\sigma_{ca}) \neq \emptyset$ and $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = \top$. $E^{\varphi,k}(\sigma \cdot (t, a)) = \sigma_s \preccurlyeq_d \sigma \cdot (t, a)$ thus, **Tr2B** holds.

- Case 4: $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset\ \wedge\ |\sigma_{ca}| > k \wedge\ \text{Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) = \emptyset$
  Same as case 2, **SndB** holds here too.
  Since, $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset$, thus $\text{delayable1}_{\varphi}(\sigma_{ca}) \neq \emptyset$, $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = stop \neq \perp$, thus **Tr1B** holds trivially.
  Since, $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = stop \neq \top$, **Tr2B** holds trivially.

- Case 5: $k^{pref(\varphi)}(\sigma_s, \sigma_{ca}) \neq \emptyset\ \wedge |\sigma_{ca}| > k \wedge\ \text{Get\_SW}^{\varphi,k}(\sigma_s, \sigma_{ca}) \neq \emptyset$
  Same as case 2, **SndB** holds here too.
  $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = \perp$ and some events are removed from the buffer during cleaning, thus, $E^{\varphi,k}_{\text{out}}(\sigma \cdot (t, a)) = \sigma_s \triangleleft_d \sigma \cdot (t, a)$, **Tr1B** holds.
  Since, $E^{\varphi,k}_{\text{mode}}(\sigma \cdot (t, a)) = \perp$, **Tr2B** holds trivially.

C.4 Informal proof (of Proposition 16)

Proposition 16: Given some property $\varphi \subseteq tw(\Sigma)$, its enforcer E^{φ} as per Definition 5 satisfies **Opt** property.

Proof of above proposition 16 can be read from [10].

C.5 Informal proof (of Proposition 18)

Proposition 18: Given some property φ and the maximum buffer size k, its enforcement function $E^{\varphi,k}$ as per Definition 11 satisfies **Opt1B** and **Opt2B** properties.

Proof for **Opt2B** can be read from [10].

Informal proof of proposition 18 for **Opt2B**:

If the buffer has not reached its maximum capacity, $F^{\varphi,k}$ has to insert/add other events, in order to produce a longer output than $E^{\varphi,k}$, which Def. 10 prevents. Otherwise when the buffer is exhausted, $F^{\varphi,k}$ has to produce maximal output by removing the least number of events such that ∞-compatible($F^{\varphi,k}$) holds, which is already ensured by $\mathrm{clean}^{\varphi,\mathrm{k}}$ of Def. 11. Thus, the proposition holds.