Runtime Enforcement of Timed Properties

Srinivas Pinisetty[1], Yliès Falcone[2], Thierry Jéron[1], Hervé Marchand[1], Antoine Rollet[3] and Omer Nguena Timo[3]

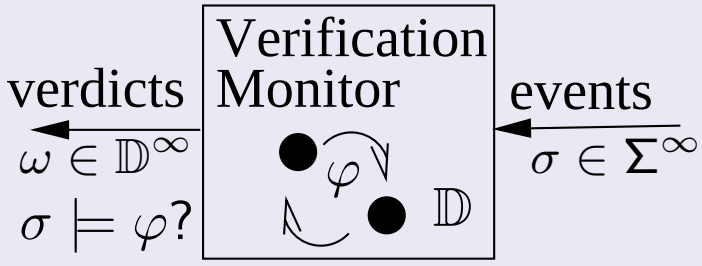
INRIA Rennes - Bretagne Atlantique, France
LIG, Université Grenoble I, France
LaBRI, Université de Bordeaux - CNRS, France

MOVEP 2012, December 05, Marseille

Outline

Verification and enforcement monitors

Runtime verification



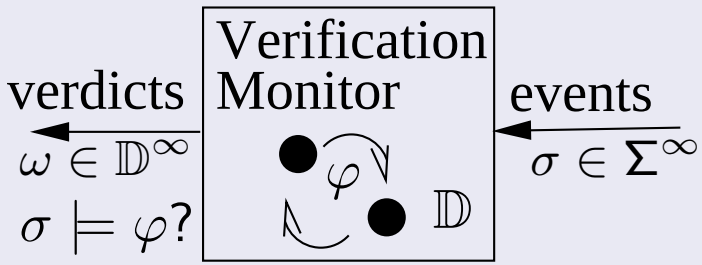
- Does the run satisfy the property?
- Monitoring an executing system.
- No system model.
- Input: stream of events.
- Output: stream of verdicts.

Runtime enforcement

- The run should satisfy the property.
- Monitoring an executing system.
- No system model.
- Input: stream of events (may violate the property).
- Output: stream of events (should satisfy the property).

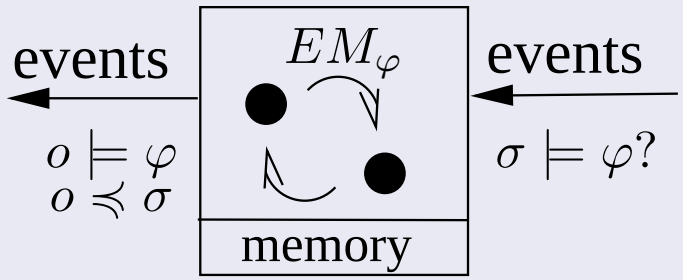
Verification and enforcement monitors

Runtime verification



- Does the run satisfy the property?
- Monitoring an executing system.
- No system model.
- Input: stream of events.
- Output: stream of verdicts.

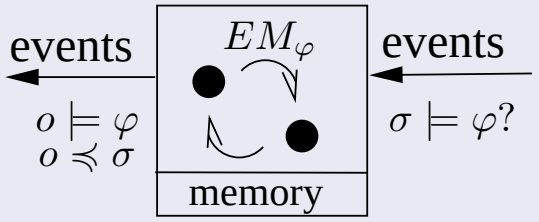
Runtime enforcement



- The run should satisfy the property.
- Monitoring an executing system.
- No system model.
- Input: stream of events (may violate the property).
- Output: stream of events (should satisfy the property).

Enforcement monitor (untimed case)

- Dedicated to a property φ.
- Possibly augmented with a memorization mechanism.



Enforcement mechanism

An EM modifies the current execution sequence (sometimes like a "filter").

- reads an input sequence $\sigma \in \Sigma^*$.
- outputs a new sequence $o \in \Sigma^*$.
- endowed with a set of enforcement primitives.
  - operates on the memorization mechanism.
  - delete or insert events using the memory content and the current input.

An EM behaves as a function $E : \Sigma^* \to \Sigma^*$.

Motivation for timed enforcement

Specifying the timing behavior

Allow specifying desired behavior of a system more precisely (time constraints between events).

- After an action "a", action "b" should occur with a delay of at least 5 time units between them.

Many application domains

- Domains: Real-time embedded systems, monitor hardware failures, communication protocols, web services and many more.
- Examples
  - Monitor a firewall to prevent DOS attack ensuring minimal delay between input events.
  - Monitor a web application to check if pre-conditions are met to provide a service.

Motivation for timed enforcement

Specifying the timing behavior

Allow specifying desired behavior of a system more precisely (time constraints between events).

- After an action "a", action "b" should occur with a delay of at least 5 time units between them.

Many application domains

- Domains: Real-time embedded systems, monitor hardware failures, communication protocols, web services and many more.
- Examples
  - Monitor a firewall to prevent DOS attack ensuring minimal delay between input events.
  - Monitor a web application to check if pre-conditions are met to provide a service.

Motivation for timed enforcement

Specifying the timing behavior

Allow specifying desired behavior of a system more precisely (time constraints between events).

- After an action "a", action "b" should occur with a delay of at least 5 time units between them.

Many application domains

- Domains: Real-time embedded systems, monitor hardware failures, communication protocols, web services and many more.
- Examples
  - Monitor a firewall to prevent DOS attack ensuring minimal delay between input events.
  - Monitor a web application to check if pre-conditions are met to provide a service.

Motivation for timed enforcement

Specifying the timing behavior

Allow specifying desired behavior of a system more precisely (time constraints between events).

- After an action "a", action "b" should occur with a delay of at least 5 time units between them.

Many application domains

- Domains: Real-time embedded systems, monitor hardware failures, communication protocols, web services and many more.
- Examples
  - Monitor a firewall to prevent DOS attack ensuring minimal delay between input events.
  - Monitor a web application to check if pre-conditions are met to provide a service.

Motivation for timed enforcement

Specifying the timing behavior

Allow specifying desired behavior of a system more precisely (time constraints between events).

- After an action "a", action "b" should occur with a delay of at least 5 time units between them.

Many application domains

- Domains: Real-time embedded systems, monitor hardware failures, communication protocols, web services and many more.
- Examples
  - Monitor a firewall to prevent DOS attack ensuring minimal delay between input events.
  - Monitor a web application to check if pre-conditions are met to provide a service.

Outline

Enforcement of timed properties

From untimed to timed properties enforcement

New elements have to be taken into account

- Input/output sequences are timed words:
  $\sigma = (\delta_1, a_1) \cdot (\delta_2, a_2) \cdots (\delta_n, a_n), \delta_i \in \mathbb{R}_{\geq}, a_i \in \Sigma$.
- Property φ described by a timed automaton or a timed logic.

Synthesis of the corresponding enforcer?

- Class of enforceable properties?
  $\rightarrow$ Focus on safety and co-safety properties modeled by TA.
- Model of the enforcer?
  $\rightarrow$ Memory + similar operations (Store, Dump).
  $\rightarrow$ No finite structure.
  $\rightarrow$ Requirements (What should the enforcer do?).

Enforcement of timed properties

From untimed to timed properties enforcement

New elements have to be taken into account

- Input/output sequences are timed words:
  $\sigma = (\delta_1, a_1) \cdot (\delta_2, a_2) \cdots (\delta_n, a_n), \delta_i \in \mathbb{R}_{\geq}, a_i \in \Sigma$.
- Property φ described by a timed automaton or a timed logic.

Synthesis of the corresponding enforcer?

- Class of enforceable properties?
  $\rightarrow$ Focus on safety and co-safety properties modeled by TA.
- Model of the enforcer?
  $\rightarrow$ Memory + similar operations (Store, Dump).
  $\rightarrow$ No finite structure.
  $\rightarrow$ Requirements (What should the enforcer do?).

Property

Defined by a timed language $\varphi \subseteq (\mathbb{R}_{\geq 0} \times \Sigma)^*$.
A timed word σ satisfies φ (noted $\sigma \models \varphi$) if $\sigma \in \varphi$.
Focus on properties specified by a TA $\mathcal{A}_\varphi$.

Safety and co-safety properties specified by TA

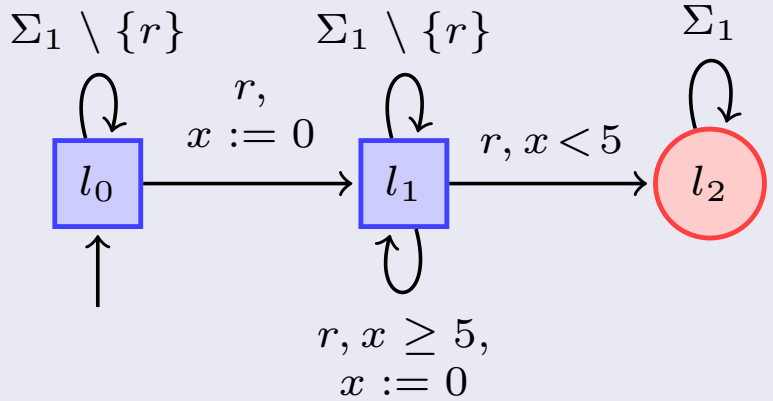
- Safety: nothing bad should ever happen (prefix closed).
- Co-safety: something good will eventually happen within a finite amount of time (extension closed).

Property

Defined by a timed language $\varphi \subseteq (\mathbb{R}_{\geq 0} \times \Sigma)^*$.
A timed word σ satisfies φ (noted $\sigma \models \varphi$) if $\sigma \in \varphi$.
Focus on properties specified by a TA $\mathcal{A}_\varphi$.

Safety and co-safety properties specified by TA

- Safety: nothing bad should ever happen (prefix closed).
- Co-safety: something good will eventually happen within a finite amount of time (extension closed).

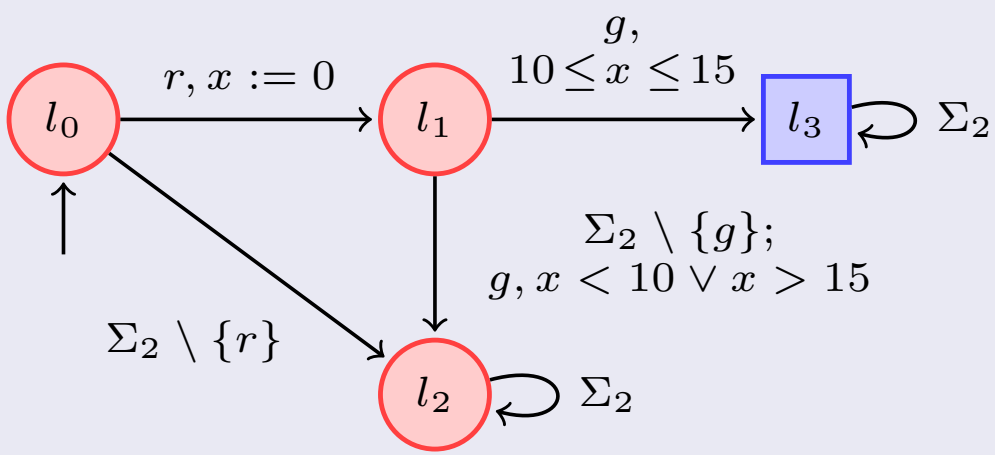


Property

Defined by a timed language $\varphi \subseteq (\mathbb{R}_{\geq 0} \times \Sigma)^*$.
A timed word σ satisfies φ (noted $\sigma \models \varphi$) if $\sigma \in \varphi$.
Focus on properties specified by a TA $\mathcal{A}_\varphi$.

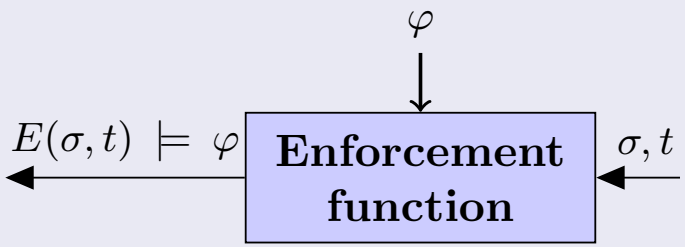
Safety and co-safety properties specified by TA

- Safety: nothing bad should ever happen (prefix closed).
- Co-safety: something good will eventually happen within a finite amount of time (extension closed).



Enforcement monitoring in a timed context

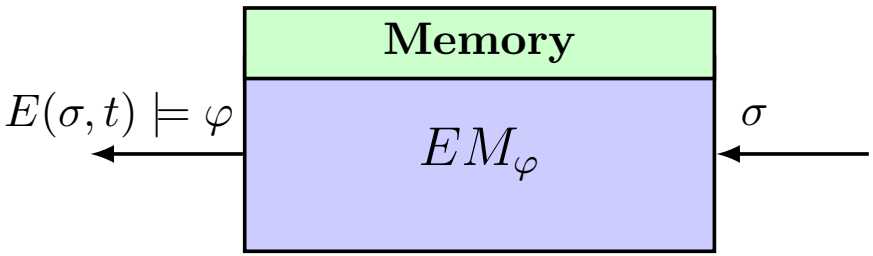
Enforcement function - E



$E : (\mathbb{R}_{\geq 0} \times \Sigma)^* \times \mathbb{R}_{\geq 0} \rightarrow (\mathbb{R}_{\geq 0} \times \Sigma)^*$.

- φ: Property which should be enforced (specified by a TA).
- σ: Input timed word.
- Output at time t: $E(\sigma, t)$ should satisfy some additional constraints [*Soundness*, *Transparency*, *Optimality*].
- E realized as a Enforcement Monitor (*EM*).

Enforcement Monitor - EM



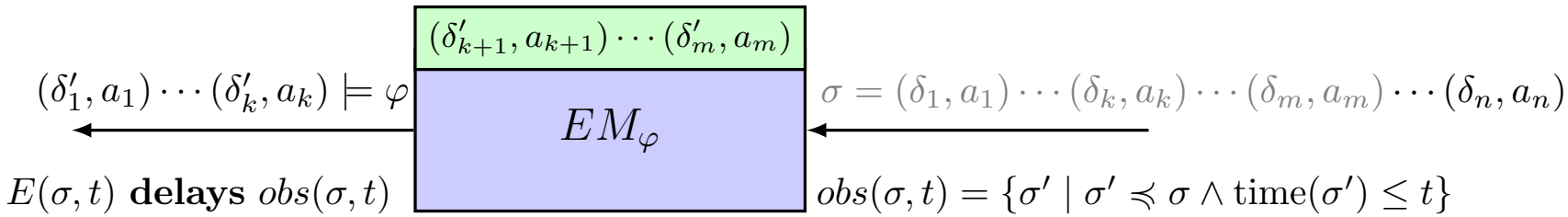
Memory

Timed word

Operations

- **Store**: stores the received event and a delay in the memory.
- **Dump**: removes the event from memory and releases it as output.

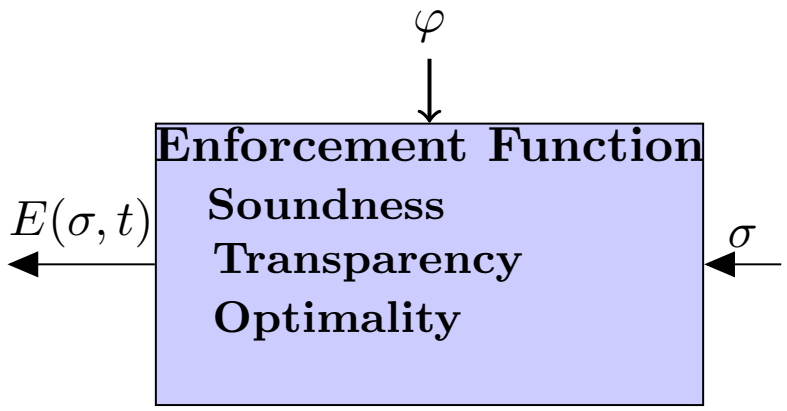
Decisions



What can the enforcer do?

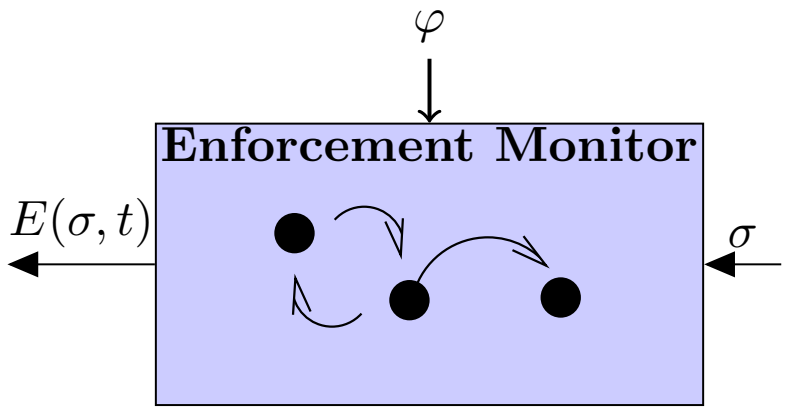
- No insertion, deletion of events.
- Order of events cannot be changed.
- Allow to increase the delay between actions.

Summary of the approach



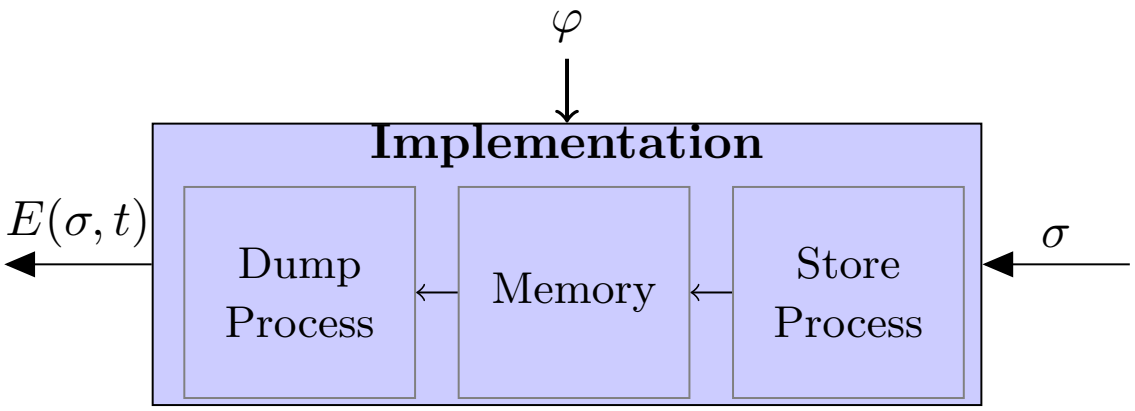
- Requirements: Soundness, Transparency and Optimality conditions.
- Enforcement Monitor: Defined as a transition system (which should satisfy the requirements).
- Implementation: Translation of the EM semantic rules into algorithms.

Summary of the approach



- Requirements: Soundness, Transparency and Optimality conditions.
- Enforcement Monitor: Defined as a transition system (which should satisfy the requirements).
- Implementation: Translation of the EM semantic rules into algorithms.

Summary of the approach



- Requirements: Soundness, Transparency and Optimality conditions.
- Enforcement Monitor: Defined as a transition system (which should satisfy the requirements).
- Implementation: Translation of the EM semantic rules into algorithms.

Outline

Enforcement of a safety property

Soundness

$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \models \varphi.$

Transparency

At any time instant t, the output $E(\sigma, t)$ delays the input $\mathrm{obs}(\sigma, t)$:
$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \preccurlyeq_d \mathrm{obs}(\sigma, t) \wedge \mathrm{time}(E(\sigma, t)) \leq t.$

Optimality

If E is sound and transparent, it is *optimal* for any σ, t if

- (Op1) $E(\sigma, t)$ is among the longest correct timed words delaying $\mathrm{obs}(\sigma, t)$.
- (Op2) Every prefix of $E(\sigma, t)$ has the shortest possible last delay.

Enforcement of a safety property

Soundness

$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \models \varphi.$

Transparency

At any time instant t, the output $E(\sigma, t)$ delays the input $\mathrm{obs}(\sigma, t)$:
$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \preccurlyeq_d \mathrm{obs}(\sigma, t) \wedge \mathrm{time}(E(\sigma, t)) \leq t.$

Optimality

If E is sound and transparent, it is *optimal* for any σ, t if

- (Op1) $E(\sigma, t)$ is among the longest correct timed words delaying $\mathrm{obs}(\sigma, t)$.
- (Op2) Every prefix of $E(\sigma, t)$ has the shortest possible last delay.

Enforcement of a safety property

Soundness

$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \models \varphi.$

Transparency

At any time instant t, the output $E(\sigma, t)$ delays the input $\mathrm{obs}(\sigma, t)$:
$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \preccurlyeq_d \mathrm{obs}(\sigma, t) \wedge \mathrm{time}(E(\sigma, t)) \leq t.$

Optimality

If E is sound and transparent, it is *optimal* for any σ, t if

- (Op1) $E(\sigma, t)$ is among the longest correct timed words delaying $\mathrm{obs}(\sigma, t)$.
- (Op2) Every prefix of $E(\sigma, t)$ has the shortest possible last delay.

Enforcement of a safety property

Soundness

$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \models \varphi.$

Transparency

At any time instant t, the output $E(\sigma, t)$ delays the input $\mathrm{obs}(\sigma, t)$:
$\forall \sigma \in (\mathbb{R}_{\geq 0} \times \Sigma)^*, \forall t \in \mathbb{R}_{\geq 0},\ E(\sigma, t) \preccurlyeq_d \mathrm{obs}(\sigma, t) \wedge \mathrm{time}(E(\sigma, t)) \leq t.$

Optimality

If E is sound and transparent, it is *optimal* for any σ, t if

- (Op1) $E(\sigma, t)$ is among the longest correct timed words delaying $\mathrm{obs}(\sigma, t)$.
- (Op2) Every prefix of $E(\sigma, t)$ has the shortest possible last delay.

EM for a safety property

$EM = \langle C, C_0, \Gamma_{EM}, \hookrightarrow \rangle$

- $C = (\mathbb{R}_{\geq 0} \times \Sigma)^* \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{B} \times Q$ is the set of configurations
- Initial configuration is $C_0 = \langle \epsilon, 0, 0, \mathtt{tt}, q_0 \rangle \in C$;
- $\Gamma_{EM} = \big((\mathbb{R}_{\geq 0} \times \Sigma) \cup \{\epsilon\}\big) \times Op \times \big((\mathbb{R}_{\geq 0} \times \Sigma) \cup \{\epsilon\}\big)$ is the input-operation-output alphabet, where $Op = \{\mathrm{Store}(\cdot), \mathrm{Dump}(\cdot), \mathrm{Delay}(\cdot)\}$;
- $\hookrightarrow \subseteq C \times \Gamma_{EM} \times C$

EM should fulfill the soundness, transparency and optimality conditions.

Operations

1. Store

$$\langle \sigma_s, \delta, d, \mathtt{tt}, q \rangle \overset{(\delta,a)/\mathrm{Store}(\delta',a)/\epsilon}{\hookrightarrow} \langle \sigma_s \cdot (\delta', a), 0, d, (\delta' \neq \infty), q' \rangle \text{ with:}$$

- $\delta' =$ shortest delay $\delta' \geq \delta$ s.t. $(q \overset{(\delta',a)}{\rightarrow} q', q' \in G)$
- q' is defined as $q \overset{(\delta',a)}{\rightarrow} q'$ if $\delta' < \infty$ and $q' = q$ otherwise

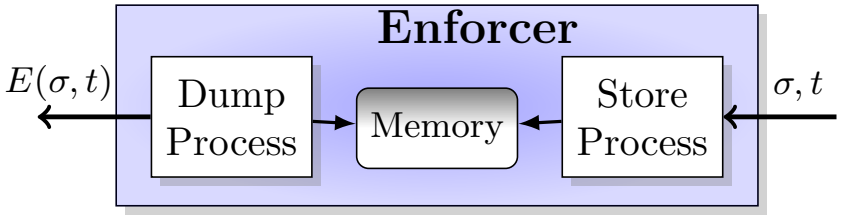
2. Dump

$$\langle (\delta, a) \cdot \sigma_s, s, \delta, b, q \rangle \overset{\epsilon/\mathrm{Dump}(\delta,a)/(\delta,a)}{\hookrightarrow} \langle \sigma_s, s, 0, b, q \rangle \text{ if } \delta \neq \infty$$

3. Delay

$$\langle \sigma_s, s, d, b, q \rangle \overset{\epsilon/\mathrm{del}(\delta)/\epsilon}{\hookrightarrow} \langle \sigma_s, s + \delta, d + \delta, b, q \rangle$$

Implementation



Algorithm: StoreProcess

$(l, X) \leftarrow (l_{\text{init}}, [X \leftarrow 0])$
while `tt` **do**
  $(\delta, a) \leftarrow$ await *event*
  if $(\text{post}(l, X, a, \delta) \notin G)$ **then**
    $\delta' \leftarrow \text{update}(l, X, a, \delta)$
    if $\delta' = \infty$ **then**
      terminate StoreProcess
    end if
  else
    $\delta' \leftarrow \delta$
  end if
  $(l, X) \leftarrow \text{post}(l, X, a, \delta')$
  enqueue (δ', a)
end while

Algorithm: DumpProcess

$d \leftarrow 0$
while `tt` **do**
  await $(|\sigma_s| \geq 1)$
  $(\delta, a) \leftarrow$ dequeue (σ_s)
  wait $(\delta - d)$
  dump (a)
  $d \leftarrow 0$
end while

Outline

Conclusion

Formal approach for enforcing timed properties

- Enforcer adds additional delay between input actions in order to satisfy the property.
- Additional constraints to ensure choosing "best" delay between actions.
- Focused on safety/co-safety properties.
- Algorithms to implement the enforcers.
- Prototypes developed using Python and UPPAAL.

Ongoing/ future work

- Enforcing more expressive properties.
- New transparency conditions.
- Improve implementation.
- Test on case studies (analysis, different architectures).

Conclusion

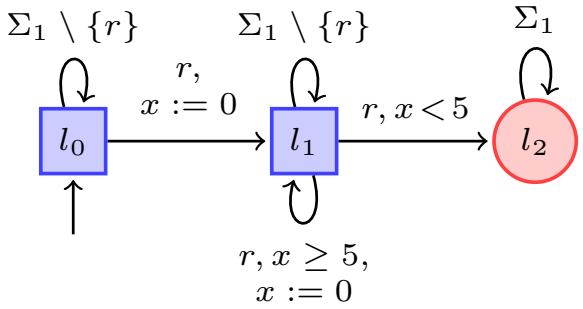
Formal approach for enforcing timed properties

- Enforcer adds additional delay between input actions in order to satisfy the property.
- Additional constraints to ensure choosing "best" delay between actions.
- Focused on safety/co-safety properties.
- Algorithms to implement the enforcers.
- Prototypes developed using Python and UPPAAL.

Ongoing/ future work

- Enforcing more expressive properties.
- New transparency conditions.
- Improve implementation.
- Test on case studies (analysis, different architectures).

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\quad \epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\quad \epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 1$ $\quad \epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Dump}$

$t = 1$ $\quad (1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $\quad (1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 4$ $\quad (1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

$\longrightarrow (1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $\quad t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $\quad t = 5$

$\downarrow \mathrm{Store}$

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $\quad t = 5$

$\downarrow \mathrm{del}(4)$

$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

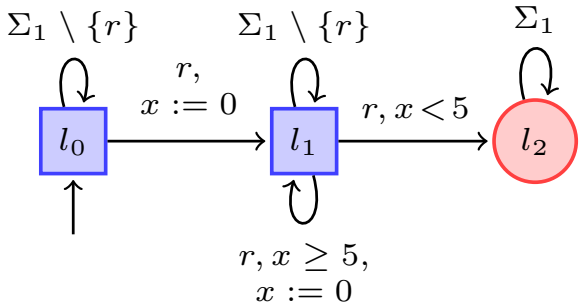
$\downarrow \mathrm{Dump}$

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

Dump

Sound, Transparent and Optimal

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\quad \epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\quad \epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow$ Store

$t = 1$ $\quad \epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow$ Dump

$t = 1$ $\quad (1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $\quad (1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow$ Store

$t = 4$ $\quad (1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

$\rightarrow (1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $\quad t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $\quad t = 5$

$\downarrow$ Store

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $\quad t = 5$

$\downarrow \mathrm{del}(4)$

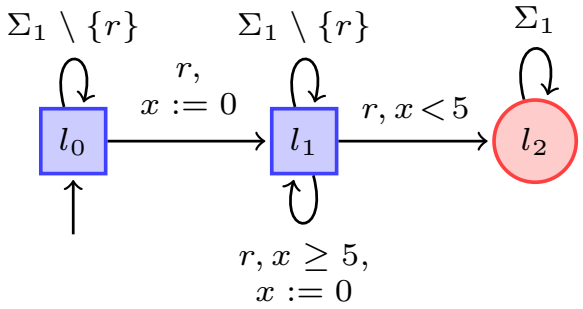
$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

$\downarrow$ Dump

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

Sound, Transparent and Optimal

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\quad \epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\quad \epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathsf{Store}$

$t = 1$ $\quad \epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathsf{Dump}$

$t = 1$ $\quad (1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $\quad (1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow \mathsf{Store}$

$t = 4$ $\quad (1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

Dump

$(1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $\quad t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $\quad t = 5$

$\downarrow \mathsf{Store}$

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $\quad t = 5$

$\downarrow \mathrm{del}(4)$

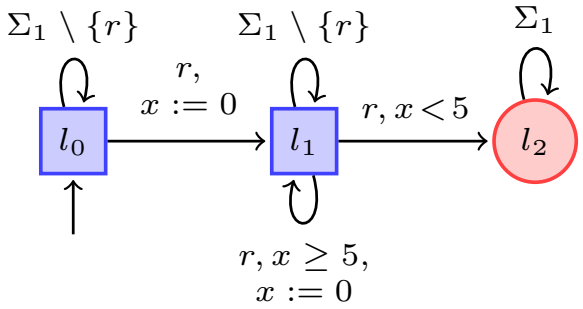
$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

$\downarrow \mathsf{Dump}$

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

Sound, Transparent and Optimal

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\quad \epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\quad \epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathsf{Store}$

$t = 1$ $\quad \epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathsf{Dump}$

$t = 1$ $\quad (1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $\quad (1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow \mathsf{Store}$

$t = 4$ $\quad (1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

$\rightarrow (1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $\quad t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $\quad t = 5$

$\downarrow \mathsf{Store}$

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $\quad t = 5$

$\downarrow \mathrm{del}(4)$

$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

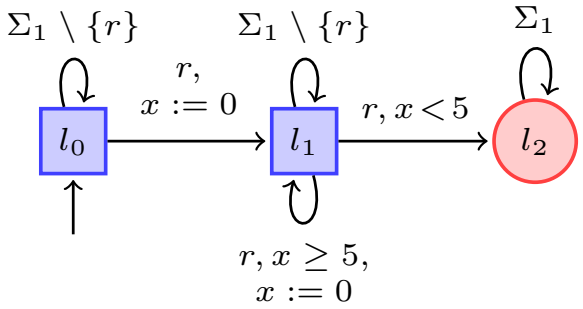
$\downarrow \mathsf{Dump}$

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

Dump

Sound, Transparent and Optimal

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\quad \epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\quad \epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 1$ $\quad \epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Dump}$

$t = 1$ $\quad (1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $\quad (1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 4$ $\quad (1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

$\rightarrow (1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $\quad t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $\quad t = 5$

$\downarrow \mathbf{Store}$

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $\quad t = 5$

$\downarrow \mathrm{del}(4)$

$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

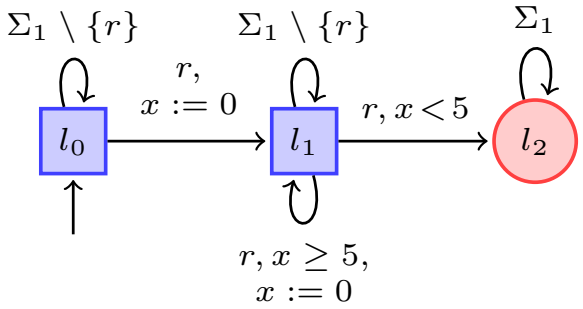
$\downarrow \mathrm{Dump}$

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $\quad t = 9$

Dump

Sound, Transparent and Optimal

Example



Input $\sigma = (1, a) \cdot (3, r) \cdot (1, r)$

$t \doteq 0$ $\epsilon/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 0 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(1)$

$t = 1$ $\epsilon/(\epsilon, 1, 1, \mathtt{tt}, < l_0, 1 >)/(1, a) \cdot (3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 1$ $\epsilon/((1, a), 0, 1, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Dump}$

$t = 1$ $(1, a)/(\epsilon, 0, 0, \mathtt{tt}, < l_0, 1 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{del}(3)$

$t = 4$ $(1, a)/(\epsilon, 3, 3, \mathtt{tt}, < l_0, 4 >)/(3, r) \cdot (1, r)$

$\downarrow \mathrm{Store}$

$t = 4$ $(1, a)/((3, r), 0, 3, \mathtt{tt}, < l_1, 0 >)/(1, r)$

Dump

$(1, a) \cdot (3, r)/(\epsilon, 0, 0, \mathtt{tt}, < l_1, 0 >)/(1, r)$ $t \doteq 4$

$\downarrow \mathrm{del}(1)$

$(1, a) \cdot (3, r)/(\epsilon, 1, 1, \mathtt{tt}, < l_1, 1 >)/(1, r)$ $t = 5$

$\downarrow \mathbf{Store}$

$(1, a) \cdot (3, r)/((5, r), 0, 1, \mathtt{tt}, < l_1, 0 >)/\epsilon$ $t = 5$

$\downarrow \mathrm{del}(4)$

$(1, a) \cdot (3, r)/((5, r), 4, 5, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $t = 9$

$\downarrow \mathrm{Dump}$

$(1, a) \cdot (3, r) \cdot (5, r)/(\epsilon, 4, 0, \mathtt{tt}, < l_1, 4 >)/\epsilon$ $t = 9$

Sound, Transparent and Optimal