ETH zürich

ESORICS

Real-time Policy Enforcement with Metric First-Order Temporal Logic

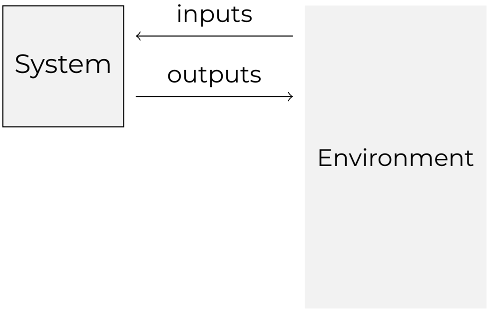
ESORICS 2022 | Copenhagen, September 26, 2022

François Hublet · David Basin · Srđan Krstić

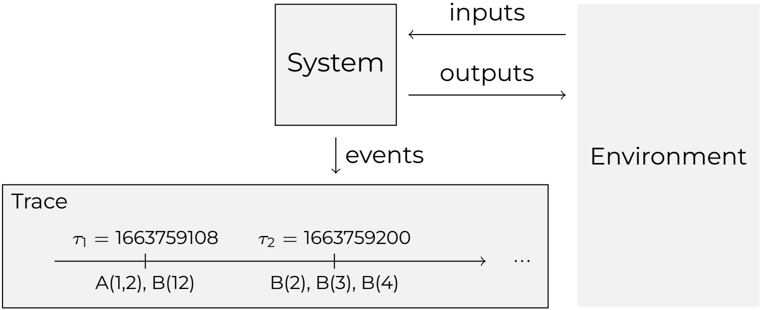
francois.hublet@inf.ethz.ch · basin@inf.ethz.ch · srdan.krstic@inf.ethz.ch

Institute of Information Security, Department of Computer Science, ETH Zürich

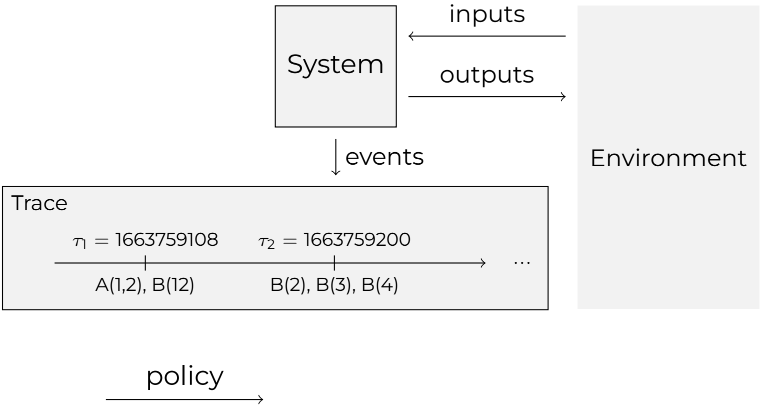
Runtime enforcement



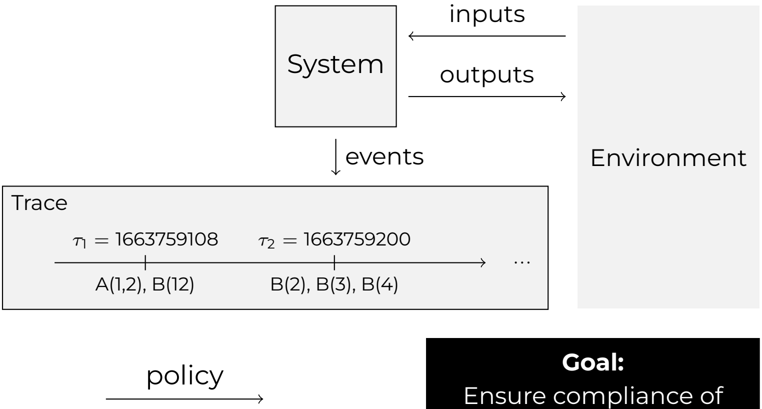
Runtime enforcement



Runtime enforcement

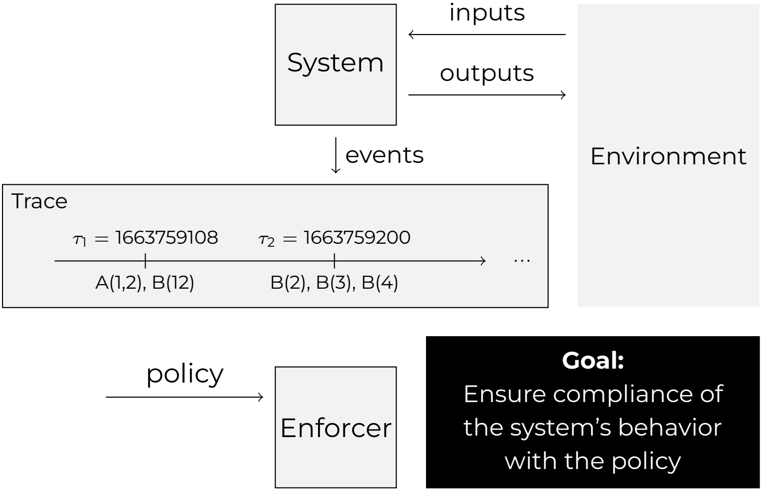


Runtime enforcement



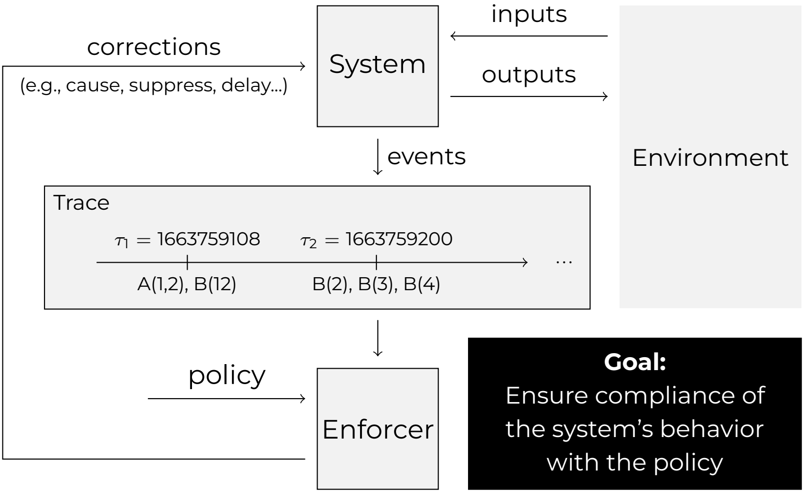
Goal:
Ensure compliance of the system's behavior with the policy

Runtime enforcement



Goal:
Ensure compliance of the system's behavior with the policy

Runtime enforcement



Goal:
Ensure compliance of the system's behavior with the policy

Metric First-Order Temporal Logic (MFOTL)

- First introduced by Chomicki [1]
- Expressive formalism for specifying **trace properties**

Metric First-Order Temporal Logic (MFOTL)

- First introduced by Chomicki [1]
- Expressive formalism for specifying **trace properties**

Example

"At any time, for all x and y, if $A(x, y)$ has occurred in the last 1000 seconds, then $B(y)$ must occur now"

Metric First-Order Temporal Logic (MFOTL)

- First introduced by Chomicki [1]
- Expressive formalism for specifying **trace properties**

Example

"At any time, for all x and y, if $\mathrm{A}(x, y)$ has occurred in the last 1000 seconds, then $\mathrm{B}(y)$ must occur now"

$$\Box\, [\forall x, y.\ (\blacklozenge_{[0,1000]} \mathrm{A}(x, y)) \Rightarrow \mathrm{B}(y)]$$

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

- **Monitors** for MFOTL (MonPoly [2], Verimon [3])

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

- **Monitors** for MFOTL (MonPoly [2], Verimon [3])
- **Enforcers** for less expressive policy languages

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

- **Monitors** for MFOTL (MonPoly [2], Verimon [3])
- **Enforcers** for less expressive policy languages
  - LTL (Acacia [4]…)

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

- **Monitors** for MFOTL (MonPoly [2], Verimon [3])
- **Enforcers** for less expressive policy languages
  - LTL (Acacia [4]…)
  - MTL (TACoS [5]…)

Runtime enforcement with MFOTL

Existing enforcement tools do not support MFOTL

Related work includes:

- **Monitors** for MFOTL (MonPoly [2], Verimon [3])
- **Enforcers** for less expressive policy languages
  - LTL (Acacia [4]...)
  - MTL (TACoS [5]...)
  - Security automata (GREP [6], TiPEX [7]...)

This work

Focus: Enforcement of MFOTL policies

This work

Focus: Enforcement of MFOTL policies

Contributions:

This work

Focus: Enforcement of MFOTL policies

Contributions:

1. **Characterization** of enforceable MFOTL policies

This work

Focus: Enforcement of MFOTL policies

Contributions:

1. **Characterization** of enforceable MFOTL policies
2. Enforcement **algorithm** for an expressive fragment

This work

Focus: Enforcement of MFOTL policies

Contributions:

1. **Characterization** of enforceable MFOTL policies
2. Enforcement **algorithm** for an expressive fragment
3. First MFOTL enforcement **tool**

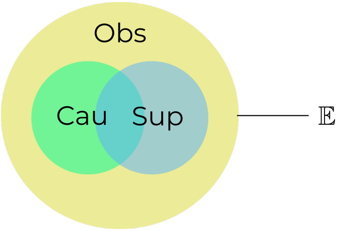
Table of contents

Runtime enforcement setup

Signature	$\Sigma = (\mathbb{D}, \mathbb{E}, a)$	domain $\mathbb{D}$, event names $\mathbb{E}$, arities $a : \mathbb{E} \to \mathbb{N}$
Events	$e(d_1, \ldots, d_{a(e)}) \in \mathrm{Ev}$	$e \in \mathbb{E}$, $d_i \in \mathbb{D}$
Databases	$D = \{e_1, ..., e_k\} \in \mathbb{DB}$	$e_i \in \mathrm{Ev}$
Traces	$\sigma = (\tau_i, D_i)_{1 \le i \le k} \in \mathbb{T}$	$\tau_1 \le \tau_2 \le \ldots \quad \ldots \le \tau_i \in \mathbb{N}$ $D_1 \; D_2 \; \ldots \qquad D_i \in \mathbb{DB}$

Runtime enforcement setup

Signature	$\Sigma = (\mathbb{D}, \mathbb{E}, a)$	domain $\mathbb{D}$, event names $\mathbb{E}$, arities $a : \mathbb{E} \to \mathbb{N}$
Events	$e(d_1, \dots, d_{a(e)}) \in \mathsf{Ev}$	$e \in \mathbb{E}$, $d_i \in \mathbb{D}$
Databases	$D = \{e_1, ..., e_k\} \in \mathbb{DB}$	$e_i \in \mathsf{Ev}$
Traces	$\sigma = (\tau_i, D_i)_{1 \leq i \leq k} \in \mathbb{T}$	$\tau_1 \leq \tau_2 \leq \dots \quad \dots \leq \tau_i \in \mathbb{N}$ $D_1 \; D_2 \; \dots \quad D_i \in \mathbb{DB}$

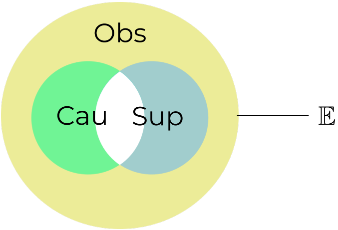


We distinguish:

- *suppressable* (Sup) vs. *causable* (Cau) events [8]
- controllable (Sup ∪ Cau) vs. *only-observable* (Obs) events [9]

Runtime enforcement setup

Signature	$\Sigma = (\mathbb{D}, \mathbb{E}, a)$	domain $\mathbb{D}$, event names $\mathbb{E}$, arities $a : \mathbb{E} \to \mathbb{N}$
Events	$e(d_1, \ldots, d_{a(e)}) \in \mathrm{Ev}$	$e \in \mathbb{E}$, $d_i \in \mathbb{D}$
Databases	$D = \{e_1, ..., e_k\} \in \mathbb{DB}$	$e_i \in \mathrm{Ev}$
Traces	$\sigma = (\tau_i, D_i)_{1 \leq i \leq k} \in \mathbb{T}$	$\tau_1 \leq \tau_2 \leq \ldots \quad \ldots \leq \tau_i \in \mathbb{N}$ $D_1 \;\; D_2 \;\; \ldots \qquad D_i \in \mathbb{DB}$



We distinguish:

- *suppressable* (Sup) vs. *causable* (Cau) events [8]
- controllable (Sup ∪ Cau) vs. *only-observable* (Obs) events [9]

Additional assumption: $\mathrm{Sup} \cap \mathrm{Cau} = \emptyset$

Metric First-Order Temporal Logic

Syntax

MFOTL is defined by the grammar

$$\varphi ::= r(t_1, \ldots, t_{a(r)}) \mid \neg\varphi \mid \varphi \lor \varphi \mid \exists x.\, \varphi$$
$$\mid \underbrace{\bullet_I}_{\text{“previous”}} \varphi \mid \underbrace{\bigcirc_I}_{\text{“next”}} \varphi \mid \varphi \underbrace{\mathsf{S}_I}_{\text{“since”}} \varphi \mid \varphi \underbrace{\mathsf{U}_I}_{\text{“until”}} \varphi$$

with $r \in \mathbb{E}$, V variables, $t_i \in V \cup \mathbb{D}$, $I \subseteq \{[a, b) \mid (a, b) \in \mathbb{N}^2, b > a\}$.

Metric First-Order Temporal Logic

Syntax

MFOTL is defined by the grammar

$$\varphi ::= r(t_1, \ldots, t_{a(r)}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\ \varphi$$
$$\mid \underbrace{\bullet_I}_{\text{"previous"}} \varphi \mid \underbrace{\bigcirc_I}_{\text{"next"}} \varphi \mid \varphi \underbrace{\mathsf{S}_I}_{\text{"since"}} \varphi \mid \varphi \underbrace{\mathsf{U}_I}_{\text{"until"}} \varphi$$

with $r \in \mathbb{E}$, V variables, $t_i \in V \cup \mathbb{D}$, $I \subseteq \{[a, b) \mid (a, b) \in \mathbb{N}^2, b > a\}$.

Some derived operators: $\underbrace{\blacklozenge_I}_{\text{"once"}} \varphi$, $\underbrace{\square}_{\text{"always"}} \varphi$...

Metric First-Order Temporal Logic

Syntax

MFOTL is defined by the grammar

$$\varphi ::= r(t_1, \ldots, t_{a(r)}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\, \varphi$$
$$\mid \underbrace{\bullet_I}_{\text{“previous”}} \varphi \mid \underbrace{\bigcirc_I}_{\text{“next”}} \varphi \mid \varphi \underbrace{\mathsf{S}_I}_{\text{“since”}} \varphi \mid \varphi \underbrace{\mathsf{U}_I}_{\text{“until”}} \varphi$$

with $r \in \mathbb{E}$, V variables, $t_i \in V \cup \mathbb{D}$, $I \subseteq \{[a, b) \mid (a, b) \in \mathbb{N}^2, b > a\}$.

Some derived operators: $\underbrace{\blacklozenge_I}_{\text{“once”}} \varphi$, $\underbrace{\square}_{\text{“always”}} \varphi$...

$v, i \vDash_\sigma \varphi \iff$ “φ is true on trace σ at timepoint i under valuation v”

Metric First-Order Temporal Logic

Syntax

MFOTL is defined by the grammar

$$\varphi ::= r(t_1, \ldots, t_{a(r)}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\, \varphi$$
$$\mid \underbrace{\bullet_I}_{\text{“previous”}} \varphi \mid \underbrace{\bigcirc_I}_{\text{“next”}} \varphi \mid \varphi \underbrace{\mathsf{S}_I}_{\text{“since”}} \varphi \mid \varphi \underbrace{\mathsf{U}_I}_{\text{“until”}} \varphi$$

with $r \in \mathbb{E}$, V variables, $t_i \in V \cup \mathbb{D}$, $I \subseteq \{[a, b) \mid (a, b) \in \mathbb{N}^2, b > a\}$.

Some derived operators: $\underbrace{\blacklozenge_I}_{\text{“once”}} \varphi$, $\underbrace{\Box}_{\text{“always”}} \varphi$...

$v, i \vDash_\sigma \varphi \iff$ “φ is true on trace σ at timepoint i under valuation v”

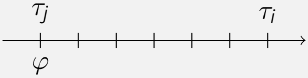
The trace property specified by φ is $\mathcal{L}(\varphi) := \{\sigma \mid \emptyset, 1 \vDash_\sigma \varphi\}$

Metric First-Order Temporal Logic

Semantics (selected operators)

Semantics (selected operators)

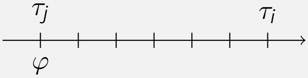
$v, i \models_\sigma \blacklozenge_I \varphi \iff \exists j.\ \sigma =$



$\wedge\ \tau_i - \tau_j \in I$

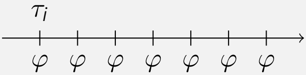
Semantics (selected operators)

$v, i \models_\sigma \blacklozenge_I \varphi \iff \exists j.\ \sigma =$



$\land\ \tau_i - \tau_j \in I$

$v, i \models_\sigma \Box \varphi \iff \sigma =$



$\cdots$

MFOTL for enforcement

Standard monitoring setup

MFOTL for enforcement

Standard monitoring setup

Consider only (safety) policies of the form $\Box\, \varphi$

MFOTL for enforcement

Standard monitoring setup

Consider only (safety) policies of the form $\Box\,\varphi$

+ for practical enforcement

φ *future-free* ($\approx$ independent of the future)

$\text{MFOTL}^{\mathcal{F}}_{\Box} := \{\Box\,\varphi \mid \varphi \text{ future free}\}$

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*,

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*, i.e., a function $\mu : \mathbb{T} \to \underbrace{\mathbb{DB}}_{\text{ev. to suppress}} \times \underbrace{\mathbb{DB}}_{\text{ev. to cause}}$ that always restores compliance with φ.

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*, i.e., a function $\mu : \mathbb{T} \to \underbrace{\mathbb{DB}}_{\text{ev. to suppress}} \times \underbrace{\mathbb{DB}}_{\text{ev. to cause}}$ that always restores compliance with φ.

Formally, for all σ, τ, D:

$$\sigma \in \mathcal{L}(\varphi) \land \tau \geq \text{last\_timestamp}(\sigma) \land \sigma' = \sigma \cdot (\tau, D)$$

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*, i.e., a function $\mu : \mathbb{T} \to \underbrace{\mathbb{DB}}_{\text{ev. to suppress}} \times \underbrace{\mathbb{DB}}_{\text{ev. to cause}}$ that always restores compliance with φ.

Formally, for all σ, τ, D:

$$\sigma \in \mathcal{L}(\varphi) \land \tau \geq \text{last\_timestamp}(\sigma) \land \sigma' = \sigma \cdot (\tau, D)$$
$$\land\, \mu(\sigma') = (D_-, D_+)$$

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*, i.e., a function $\mu : \mathbb{T} \to \underbrace{\mathbb{DB}}_{\text{ev. to suppress}} \times \underbrace{\mathbb{DB}}_{\text{ev. to cause}}$ that always restores compliance with φ.

Formally, for all σ, τ, D:

$$\begin{aligned} & \sigma \in \mathcal{L}(\varphi) \land \tau \geq \text{last\_timestamp}(\sigma) \land \sigma' = \sigma \cdot (\tau, D) \\ & \land \mu(\sigma') = (D_-, D_+) \\ \Longrightarrow\ & \text{all events in } D_- \text{ are suppressable} \\ & \land \text{all events in } D_+ \text{ are causable} \\ & \land \sigma \cdot (\tau, (D \setminus D_-) \cup D_+) \in \mathcal{L}(\varphi). \end{aligned}$$

Enforceability

Definition (Enforceability)

A policy φ is enforceable iff it has a *correct enforcer*, i.e., a function $\mu : \mathbb{T} \to \underbrace{\mathbb{DB}}_{\text{ev. to suppress}} \times \underbrace{\mathbb{DB}}_{\text{ev. to cause}}$ that always restores compliance with φ.

Formally, for all σ, τ, D:

$$\begin{aligned} & \sigma \in \mathcal{L}(\varphi) \wedge \tau \geq \text{last\_timestamp}(\sigma) \wedge \sigma' = \sigma \cdot (\tau, D) \\ & \wedge\, \mu(\sigma') = (D_-, D_+) \\ \Longrightarrow\; & \text{all events in } D_- \text{ are suppressable} \\ & \wedge \text{ all events in } D_+ \text{ are causable} \\ & \wedge\, \sigma \cdot (\tau, (D \setminus D_-) \cup D_+) \in \mathcal{L}(\varphi). \end{aligned}$$

Example

Assume $\mathsf{B} \in \mathsf{Cau}$.

The policy $\Box[\forall x, y.\ \blacklozenge_{[0,1000]} \mathsf{A}(x, y) \Rightarrow \mathsf{B}(y)]$ is enforceable.

MFOTL: negative result

Question

Is there an algorithm to decide if an $\text{MFOTL}^{\mathcal{F}}_{\square}$ policy is enforceable?

MFOTL: negative result

Question

Is there an algorithm to decide if an $\text{MFOTL}^{\mathcal{F}}_{\square}$ policy is enforceable?

Theorem 1

Assume that Sup contains an event of arity > 1 and Obs $\neq \emptyset$. The set $\{\varphi \in \text{MFOTL}^{\mathcal{F}}_{\square} \mid \varphi \text{ is enforceable}\}$ is not computable.

MFOTL: negative result

Question

Is there an algorithm to decide if an $\text{MFOTL}^{\mathcal{F}}_{\Box}$ policy is enforceable?

Theorem 1

Assume that Sup contains an event of arity > 1 and Obs $\neq \emptyset$.
The set $\{\varphi \in \text{MFOTL}^{\mathcal{F}}_{\Box} \mid \varphi \text{ is enforceable}\}$ is not computable.

Proof: Reduction to the Entscheidungsproblem of FOL.

MFOTL: positive result

Define *guarded MFOTL* (GMFOTL) as

$$\psi ::= \bot \mid s(t_1, \ldots, t_n) \mid \neg c(t_1, \ldots, t_n) \mid \psi \wedge \varphi \mid \psi \vee \psi \mid \exists x.\, \psi$$

where $s \in \mathsf{Sup}, c \in \mathsf{Cau}$, and $\varphi \in \mathsf{MFOTL}$.

Lemma

For all $\psi \in \mathsf{GMFOTL}$, $\Box \neg \psi$ is enforceable.

MFOTL: positive result

Define *guarded MFOTL* (GMFOTL) as

$$\psi ::= \bot \mid s(t_1, \ldots, t_n) \mid \neg c(t_1, \ldots, t_n) \mid \psi \land \varphi \mid \psi \lor \psi \mid \exists x.\, \psi$$

where $s \in \mathsf{Sup}, c \in \mathsf{Cau}$, and $\varphi \in$ MFOTL.

Lemma

For all $\psi \in$ GMFOTL, $\Box \neg\psi$ is enforceable.

Theorem 2

$\varphi \in \text{MFOTL}^{\mathcal{F}}_{\Box}$ is enforceable $\Longleftrightarrow \exists \psi \in$ GMFOTL. $\varphi \equiv \Box \neg\psi$.

MFOTL: positive result

Define *guarded MFOTL* (GMFOTL) as

$$\psi ::= \bot \mid s(t_1, \ldots, t_n) \mid \neg c(t_1, \ldots, t_n) \mid \psi \wedge \varphi \mid \psi \vee \psi \mid \exists x.\, \psi$$

where $s \in \text{Sup}, c \in \text{Cau}$, and $\varphi \in$ MFOTL.

Lemma

For all $\psi \in$ GMFOTL, $\Box \neg\psi$ is enforceable.

Theorem 2

$\varphi \in \text{MFOTL}^{\mathcal{F}}_{\Box}$ is enforceable $\iff \exists \psi \in$ GMFOTL. $\varphi \equiv \Box \neg\psi$.

Example

$$\Box[\forall x, y.\ \blacklozenge_{[0,1000]} \mathrm{A}(x,y) \Rightarrow \mathrm{B}(y)] \equiv \Box[\neg(\underbrace{\exists x, y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x,y)}_{\in \text{GMFOTL}})]$$

Table of contents

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL
- Compute corrections following the structure top-down

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box\neg\varphi$ where $\varphi \in$ GMFOTL
- Compute corrections following the structure top-down
- Use state-of-the-art monitoring algorithm by Basin et al. [10] as a subroutine

Enforcing MFOTL policies

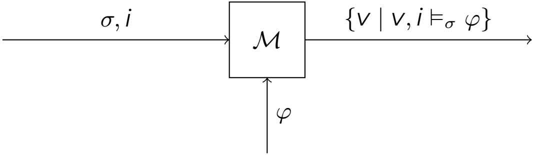
The idea, in a nutshell:

- Focus on $\Box\neg\varphi$ where $\varphi \in$ GMFOTL
- Compute corrections following the structure top-down
- Use state-of-the-art monitoring algorithm by Basin et al. [10] as a subroutine
  - For monitorable class of policies of the form $\Box\varphi$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL
- Compute corrections following the structure top-down
- Use state-of-the-art monitoring algorithm by Basin et al. [10] as a subroutine
  - For monitorable class of policies of the form $\Box \varphi$
  - Interface:



Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforcing MFOTL policies

Example

Consider $\varphi = \square \neg\psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathsf{A}(1, 2), \mathsf{B}(12)\})$.

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg \psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathsf{A}(1, 2), \mathsf{B}(12)\})$.

falsify $\exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathsf{A}(1, 2), \mathsf{B}(12)\})$.

falsify $\exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ

$\rightsquigarrow$ call $\mathcal{M}(\exists y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y); \sigma, i)$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)$.

Enforce φ on $\sigma = (0, \{A(1, 2), B(12)\})$.

$$\text{falsify } \exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)) \text{ on } \sigma$$
$$\rightsquigarrow \text{call } \mathcal{M}(\exists y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1)\}$$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)$.

Enforce φ on $\sigma = (0, \{A(1,2), B(12)\})$.

falsify $\exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y))$ on σ

$\rightsquigarrow$ call $\mathcal{M}(\exists y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y); \sigma, i)$ $\rightsquigarrow$ output $= \{(1)\}$

$\rightsquigarrow$ falsify $\exists y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y))$ on σ for $x = 1$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathsf{A}(1, 2), \mathsf{B}(12)\})$.

falsify $\exists x, y.\ \neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ

$\rightsquigarrow$ call $\mathcal{M}(\exists y.\ \neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y); \sigma, i)$ $\rightsquigarrow$ output $= \{(1)\}$

$\rightsquigarrow$ falsify $\exists y.\ \neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ for $x = 1$

$\rightsquigarrow$ call $\mathcal{M}(\neg \mathsf{B}(y) \land \blacklozenge_{[0,1000]} \mathsf{A}(x, y); \sigma, i)$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathrm{A}(1, 2), \mathrm{B}(12)\})$.

$$
\begin{aligned}
&\text{falsify } \exists x, y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)) \text{ on } \sigma \\
\rightsquigarrow\ &\text{call } \mathcal{M}(\exists y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1)\} \\
\rightsquigarrow\ &\text{falsify } \exists y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)) \text{ on } \sigma \text{ for } x = 1 \\
\rightsquigarrow\ &\text{call } \mathcal{M}(\neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1, 2)\}
\end{aligned}
$$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)$.

Enforce φ on $\sigma = (0, \{A(1, 2), B(12)\})$.

$$\begin{aligned}
&\text{falsify } \exists x, y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)) \text{ on } \sigma \\
\rightsquigarrow\ &\text{call } \mathcal{M}(\exists y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1)\} \\
\rightsquigarrow\ &\text{falsify } \exists y.\ \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y)) \text{ on } \sigma \text{ for } x = 1 \\
\rightsquigarrow\ &\text{call } \mathcal{M}(\neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1, 2)\} \\
\rightsquigarrow\ &\text{falsify } \neg B(y) \wedge \blacklozenge_{[0,1000]} A(x, y) \text{ on } \sigma \text{ for } x = 1, y = 2
\end{aligned}$$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathrm{A}(1,2), \mathrm{B}(12)\})$.

$$
\begin{aligned}
& \text{falsify } \exists x, y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)) \text{ on } \sigma \\
\rightsquigarrow\ & \text{call } \mathcal{M}(\exists y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1)\} \\
\rightsquigarrow\ & \text{falsify } \exists y.\ \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y)) \text{ on } \sigma \text{ for } x = 1 \\
\rightsquigarrow\ & \text{call } \mathcal{M}(\neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y); \sigma, i) \rightsquigarrow \text{output} = \{(1, 2)\} \\
\rightsquigarrow\ & \text{falsify } \neg \mathrm{B}(y) \wedge \blacklozenge_{[0,1000]} \mathrm{A}(x, y) \text{ on } \sigma \text{ for } x = 1, y = 2 \\
\rightsquigarrow\ & \text{make } \neg \mathrm{B}(y) \text{ false on } \sigma \text{ for } x = 1, y = 2
\end{aligned}
$$

Enforcing MFOTL policies

Example

Consider $\varphi = \Box \neg\psi$ where $\psi = \exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$.

Enforce φ on $\sigma = (0, \{\mathsf{A}(1, 2), \mathsf{B}(12)\})$.

falsify $\exists x, y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ

$\rightsquigarrow$ call $\mathcal{M}(\exists y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y); \sigma, i) \rightsquigarrow$ output $= \{(1)\}$

$\rightsquigarrow$ falsify $\exists y.\ \neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y))$ on σ for $x = 1$

$\rightsquigarrow$ call $\mathcal{M}(\neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y); \sigma, i) \rightsquigarrow$ output $= \{(1, 2)\}$

$\rightsquigarrow$ falsify $\neg \mathsf{B}(y) \wedge \blacklozenge_{[0,1000]} \mathsf{A}(x, y)$ on σ for $x = 1, y = 2$

$\rightsquigarrow$ make $\neg \mathsf{B}(y)$ false on σ for $x = 1, y = 2$

$\rightsquigarrow$ cause $\mathsf{B}(y) = \mathsf{B}(2)$

Enforcement algorithm

- In general, one top-down iteration is not sufficient (e.g., for $\psi = \neg\mathsf{B}(1) \vee (\neg\mathsf{B}(2) \wedge \mathsf{B}(1))$)

Enforcement algorithm

- In general, one top-down iteration is not sufficient (e.g., for $\psi = \neg$B(1) $\vee$ ($\neg$B(2) $\wedge$ B(1)))
- Perform a fixpoint computation

Enforcement algorithm

- In general, one top-down iteration is not sufficient (e.g., for $\psi = \neg \mathrm{B}(1) \vee (\neg \mathrm{B}(2) \wedge \mathrm{B}(1)))$
- Perform a fixpoint computation
- We prove termination for monitorable and enforceable policies

About the condition $\mathsf{Sup} \cap \mathsf{Cau} = \emptyset$

- Needed for our enforceability condition to hold

About the condition $\mathrm{Sup} \cap \mathrm{Cau} = \emptyset$

- Needed for our enforceability condition to hold

Example

If $\mathrm{B} \in \mathrm{Sup} \cap \mathrm{Cau}$, we have $\mathrm{B}(1) \vee \neg \mathrm{B}(1) \in \mathrm{GMFOTL}$, but $\Box[\neg(\mathrm{B}(1) \vee \neg \mathrm{B}(1))] = \Box \bot$ is not enforceable.

About the condition $\mathsf{Sup} \cap \mathsf{Cau} = \emptyset$

- Needed for our enforceability condition to hold

Example

If $B \in \mathsf{Sup} \cap \mathsf{Cau}$, we have $B(1) \vee \neg B(1) \in \mathsf{GMFOTL}$, but $\square[\neg(B(1) \vee \neg B(1))] = \square \perp$ is not enforceable.

- Needed for our algorithm to terminate

About the condition $\mathsf{Sup} \cap \mathsf{Cau} = \emptyset$

- Needed for our enforceability condition to hold

Example

If $B \in \mathsf{Sup} \cap \mathsf{Cau}$, we have $B(1) \vee \neg B(1) \in$ GMFOTL, but $\square[\neg(B(1) \vee \neg B(1))] = \square \bot$ is not enforceable.

- Needed for our algorithm to terminate
- Slightly relaxed in the implementation

Table of contents

Implementation

- Extension of Basin et al.'s MonPoly [2]
- Ca. 500 loc OCaml code
- Runtime performance equivalent or better than GREP's [6] on LTL fragment
- Overhead < 50% with respect to MonPoly

https://gitlab.ethz.ch/fhublet/mfotl-enforcement

Conclusion

In this talk, we have seen:

- A **characterization** of enforceable MFOTL policies
- An **algorithm** for enforcing MFOTL policies
- The first enforcement **tool** for MFOTL

Conclusion

In this talk, we have seen:

- A **characterization** of enforceable MFOTL policies
- An **algorithm** for enforcing MFOTL policies
- The first enforcement **tool** for MFOTL

Open questions:

- Can we obtain a characterization for the case $\mathsf{Sup} \cap \mathsf{Cau}$?
- If such a characterization does not exist, how can we still extend the supported fragment?

Bibliography

[1] Jan Chomicki and Damian Niwinski. On the feasibility of checking temporal integrity constraints. *Journal of Computer and System Sciences*, 51(3):523–535, 1995.

[2] David Basin, Felix Klaedtke, and Eugen Zalinescu. The MonPoly monitoring tool. In Giles Reger and Klaus Havelund, editors, *International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools (RV-CuBES)*, volume 3 of *Kalpa*, pages 19–28, 2017.

[3] Joshua Schneider, David Basin, Srđan Krstić, and Dmitriy Traytel. A formally verified monitor for metric first-order temporal logic. In Bernd Finkbeiner and Leonardo Mariani, editors, *International Conference on Runtime Verification (RV)*, volume 11757 of *LNCS*, pages 310–328. Springer, 2019.

[4] Aaron Bohy, Véronique Bruyère, Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. Acacia+, a tool for LTL synthesis. In P. Madhusudan and Sanjit A. Seshia, editors, *International Conference Computer Aided Verification (CAV)*, volume 7358 of *LNCS*, pages 652–657. Springer, 2012.

[5] Till Hofmann and Stefan Schupp. TACoS: A tool for MTL controller synthesis. In Radu Calinescu and Corina S. Pasareanu, editors, *International Conference on Software Engineering and Formal Methods (SEFM)*, volume 13085 of *LNCS*, pages 372–379. Springer, 2021.

[6] Matthieu Renard, Antoine Rollet, and Yliès Falcone. GREP: games for the runtime enforcement of properties. In Nina Yevtushenko, Ana Rosa Cavalli, and Hüsnü Yenigün, editors, *International Conference on Testing Software and Systems (ICTSS)*, volume 10533 of *LNCS*, pages 259–275. Springer, 2017.

[7] Srinivas Pinisetty, Yliès Falcone, Thierry Jéron, and Hervé Marchand. TiPEX: A tool chain for timed property enforcement during execution. In *International Conference on Runtime Verification (RV)*, pages 306–320. Springer, 2015.

[8] Lujo Bauer, Jarred Ligatti, and David Walker. More enforceable security policies. In *Workshop on Foundations of Computer Security (FCS)*. Citeseer, 2002.

[9] David Basin, Vincent Jugé, Felix Klaedtke, and Eugen Zălinescu. Enforceable security policies revisited. *ACM Trans. Inf. Syst. Secur.*, 16(1):1–26, 2013.

[10] David Basin, Felix Klaedtke, Samuel Müller, and Eugen Zălinescu. Monitoring metric first-order temporal properties. *Journal of the ACM*, 62(2):1–45, 2015.

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \wedge \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$		

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \wedge \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$		
$\psi_1 = \exists d.\ \psi_2$		
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$		
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$		
$B(d)$		
$\mathsf{A}(c,d)$		

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \exists d.\ \underbrace{\underbrace{\neg \mathsf{B}(d) \wedge \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \models_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$		
$\psi_1 = \exists d.\ \psi_2$		
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$		
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$		
$B(d)$	$v_{41} = [(12)]$	
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box \varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \wedge \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \models_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$		
$\psi_1 = \exists d.\ \psi_2$		
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$		
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \land \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$		
$\psi_1 = \exists d.\ \psi_2$		
$\psi_2 = \neg \mathsf{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathrm{B}(d) \land \underbrace{\top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$		
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg \mathrm{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$\mathrm{A}(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg B(d) \wedge \underbrace{\top\ S_{[0,\infty)}\ A(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$		
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg B(d) \wedge \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ S_{[0,\infty)}\ A(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$A(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \exists d.\ \underbrace{\underbrace{\neg \mathrm{B}(d) \land \underbrace{\top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg \mathrm{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$\mathrm{A}(c,d)$	$v_{51} = [(1,2)]$	

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \land \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \models_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg \mathsf{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathsf{B}(d) \wedge \underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = v_{31} \triangleright_{(2,1)} v_{41} = [(1,2)]$	
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	$v_{32} = v_{52} \cup v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\square\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \exists d.\ \neg \mathsf{B}(d) \land \underbrace{\underbrace{\underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \models_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	
$\psi_2 = \neg \mathsf{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	$v_{22} = v_{32} \rhd_{(2,1)} v_{42} = []$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	$v_{32} = v_{52} \cup v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box \varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \exists d.\ \neg B(d) \wedge \underbrace{\underbrace{\underbrace{\top\ S_{[0,\infty)}\ A(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	$v_{12} = \pi_{(1)} v_{22} = []$
$\psi_2 = \neg B(d) \wedge \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	$v_{22} = v_{32} \rhd_{(2,1)} v_{42} = []$
$\psi_3 = \top\ S_{[0,\infty)}\ A(c,d)$	$v_{31} = v_{51} = [(1,2)]$	$v_{32} = v_{52} \cup v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$A(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \exists d.\ \neg \mathsf{B}(d) \wedge \underbrace{\underbrace{\underbrace{\top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \vDash_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	$v_2 = \pi_{(1)} v_{12} = []$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	$v_{12} = \pi_{(1)} v_{22} = []$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	$v_{22} = v_{32} \rhd_{(2,1)} v_{42} = []$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	$v_{32} = v_{52} \cup v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

Runtime verification of MFOTL policies

State-of-the-art algorithm by Basin et al. [10]

- For monitorable class of policies of the form $\Box\,\varphi$
- Based on (finite) table manipulation

Example

$$\neg\varphi = \exists c.\ \underbrace{\exists d.\ \underbrace{\neg \mathrm{B}(d) \land \underbrace{\top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)}_{\psi_3}}_{\psi_2}}_{\psi_1}$$

i	1	2
τ_i	1663759108	1663759200
D_i	A(1,2), B(12)	B(2), B(3), B(4)
$\emptyset, i \models_\sigma \varphi$	✗	✓
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = \pi_{(1)} v_{11} = [()]$	$v_2 = \pi_{(1)} v_{12} = []$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = \pi_{(1)} v_{21} = [(1)]$	$v_{12} = \pi_{(1)} v_{22} = []$
$\psi_2 = \neg \mathrm{B}(d) \land \psi_3$	$v_{21} = v_{31} \rhd_{(2,1)} v_{41} = [(1,2)]$	$v_{22} = v_{32} \rhd_{(2,1)} v_{42} = []$
$\psi_3 = \top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)$	$v_{31} = v_{51} = [(1,2)]$	$v_{32} = v_{52} \cup v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$	$v_{42} = [(2),(3),(4)]$
$\mathrm{A}(c,d)$	$v_{51} = [(1,2)]$	$v_{52} = []$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg \varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \models_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\square \neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathrm{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathrm{S}_{[0,\infty)}\ \mathrm{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathrm{A}(c,d)$	$v_{51} = [(1,2)]$

$\mathrm{enf}(\neg\varphi, \sigma, ()) =$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg \varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1, 2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c, d)$	$v_{31} = [(1, 2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c, d)$	$v_{51} = [(1, 2)]$

$$\mathsf{enf}(\neg\varphi, \sigma, ()) = \mathsf{enf}(\psi_1, \sigma, (1))$$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$

$\mathsf{enf}(\neg\varphi, \sigma, ()) = \mathsf{enf}(\psi_1, \sigma, (1))$

$\mathsf{enf}(\psi_1, \sigma, (1)) =$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box\neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg\mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$

$\mathsf{enf}(\neg\varphi, \sigma, ()) = \mathsf{enf}(\psi_1, \sigma, (1))$

$\mathsf{enf}(\psi_1, \sigma, (1)) = \mathsf{enf}(\psi_2, \sigma, (1, 2))$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box \neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$

$\mathsf{enf}(\neg\varphi, \sigma, ()) = \mathsf{enf}(\psi_1, \sigma, (1))$

$\mathsf{enf}(\psi_1, \sigma, (1)) = \mathsf{enf}(\psi_2, \sigma, (1,2))$

$\mathsf{enf}(\psi_2, \sigma, (1,2)) =$

Enforcing MFOTL policies

The idea, in a nutshell:

- Focus on $\Box\neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1	
τ_i	1663759108	
D_i	A(1,2), B(12)	
$\emptyset, i \models_\sigma \varphi$	✗	
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$	$\text{enf}(\neg\varphi, \sigma, ()) = \text{enf}(\psi_1, \sigma, (1))$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$	$\text{enf}(\psi_1, \sigma, (1)) = \text{enf}(\psi_2, \sigma, (1,2))$
$\psi_2 = \neg\text{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$	$\text{enf}(\psi_2, \sigma, (1,2)) = \text{enf}(\neg\text{B}(d), \sigma, (2))$
$\psi_3 = \top\ \text{S}_{[0,\infty)}\ \text{A}(c,d)$	$v_{31} = [(1,2)]$	
$B(d)$	$v_{41} = [(12)]$	
$\text{A}(c,d)$	$v_{51} = [(1,2)]$	

Enforcing MFOTL policies

The idea, in a nutshell:

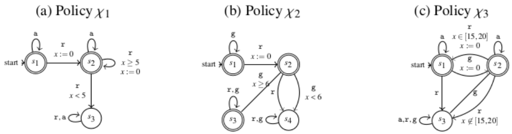
- Focus on $\square \neg\varphi$ where $\varphi \in$ GMFOTL
- Use $\mathcal{M}$ as a subroutine
- Compute corrections following the structure top-down

Example

i	1
τ_i	1663759108
D_i	A(1,2), B(12)
$\emptyset, i \vDash_\sigma \varphi$	✗
$\neg\varphi = \exists c.\ \psi_1$	$v_1 = [()]$
$\psi_1 = \exists d.\ \psi_2$	$v_{11} = [(1)]$
$\psi_2 = \neg \mathsf{B}(d) \wedge \psi_3$	$v_{21} = [(1,2)]$
$\psi_3 = \top\ \mathsf{S}_{[0,\infty)}\ \mathsf{A}(c,d)$	$v_{31} = [(1,2)]$
$B(d)$	$v_{41} = [(12)]$
$\mathsf{A}(c,d)$	$v_{51} = [(1,2)]$

$$\begin{aligned}
\mathsf{enf}(\neg\varphi, \sigma, ()) &= \mathsf{enf}(\psi_1, \sigma, (1)) \\
\mathsf{enf}(\psi_1, \sigma, (1)) &= \mathsf{enf}(\psi_2, \sigma, (1,2)) \\
\mathsf{enf}(\psi_2, \sigma, (1,2)) &= \mathsf{enf}(\neg \mathsf{B}(d), \sigma, (2)) \\
&= (\emptyset, \{B(2)\})
\end{aligned}$$

Performance



$\chi_1 = \Box\neg(\mathtt{r} \wedge ((\blacklozenge_{[1,5)}\mathtt{r}) \vee (\bullet_{[0,0]}\blacklozenge_{[0,0]}\mathtt{r})))$ Sup $= \{\mathtt{r}\}$, Obs $= \{\mathtt{a}\}$

$\chi_2 = \Box\neg((\mathtt{g} \wedge \blacklozenge_{[0,6)}(\mathtt{r} \wedge (\neg\bullet\blacklozenge\mathtt{r}))) \vee (\mathtt{r} \wedge (\bullet(\neg\mathtt{g}\,\mathsf{S}\,(\mathtt{r} \wedge (\neg\bullet\blacklozenge\mathtt{r}))))))$ Sup $= \{\mathtt{r},\mathtt{g}\}$, Obs $= \emptyset$

$\chi_3 = \Box\neg(\mathtt{r} \wedge ((\neg\mathtt{r}\,\mathsf{S}_{(20,\infty)}\,\mathtt{g}) \vee (\neg\mathtt{r}\,\mathsf{S}_{[0,15)}\,\mathtt{g}) \vee (\neg\mathtt{g}\,\mathsf{S}\,\mathtt{r})) \vee \mathtt{g} \wedge \neg\mathtt{r}\,\mathsf{S}\,\mathtt{g})$ Sup $= \{\mathtt{r},\mathtt{g}\}$, Obs $= \{\mathtt{a}\}$

Fig. 3: Policies used to compare EnfPoly to GREP