Journée commune au CT SED et au GT AFSEC

January 30th, 2025
Paris, France

Ensuring timed-opacity in timed systems

Dylan Marinho, PhD

Sorbonne Université, CNRS, LIP6,
Dylan.Marinho@lip6.fr

Based on works with Étienne André, Sarah Dépernet, Laetitia Laversa,

Engel Lefaucheux, Didier Lime, and Sun Jun

These works are partially supported by the ANR-NRF research program ProMiS (ANR-19-CE25-0015)
and the ANR research program BisoUS (ANR-22-CE48-0012).

Motivation

- Real-time systems:
  - Not only the functional correctness but also the time to answer is important

Motivation

- Critical Real-time systems:
  - Not only the functional correctness but also the time to answer is important
  - Failures (in correctness or timing) may result in dramatic consequences

Motivation

- Critical Real-time systems:
  - Not only the functional correctness but also the time to answer is important
  - Failures (in correctness or timing) may result in dramatic consequences

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

- Example
  - Number of pizzas (and order time) ordered by the white house prior to major war announcements [1]

[1] http://home.xnet.com/~warinner/pizzacites.html

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

- Example
  - Number of pizzas (and order time) ordered by the white house prior to major war announcements [1]

[1] `http://home.xnet.com/~warinner/pizzacites.html`

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd), len(attempt)) − 1 do
if pwd[i] ≠ attempt[i] then
return false
done
return true
```

A simple example of timing attack

```
# input pwd     : Real password
# input attempt : Tentative password
for i = 0 to min(len(pwd), len(attempt)) − 1 do
if pwd[i] ≠ attempt[i] then
return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time:

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd), len(attempt)) − 1 do
 if pwd[i] ≠ attempt[i] then
 return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: ϵ

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd), len(attempt)) - 1 do
if pwd[i] ≠ attempt[i] then
return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon$

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd), len(attempt)) − 1 do
if pwd[i] ≠ attempt[i] then
return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon + \epsilon$

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd), len(attempt)) − 1 do
if pwd[i] ≠ attempt[i] then
return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon + \epsilon$

- Problem: The execution time is proportional to the number of consecutive correct characters from the beginning of `attempt`

Methodology

A program

A specification

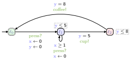
"The program must be secure"

Methodology

A program

→

A model



A specification

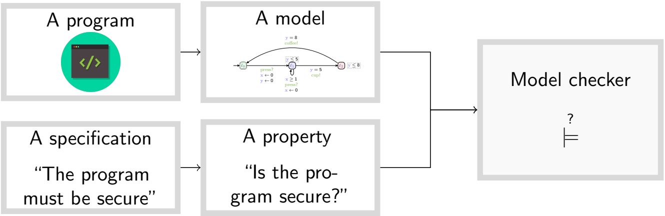
"The program must be secure"

→

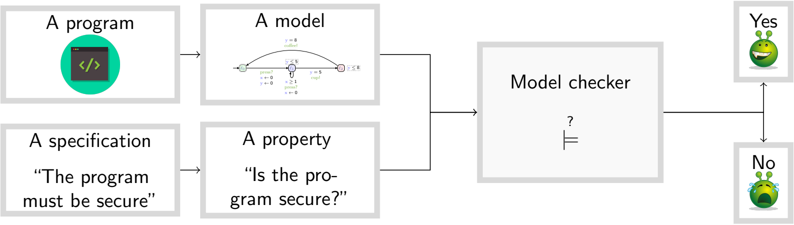
A property

"Is the program secure?"

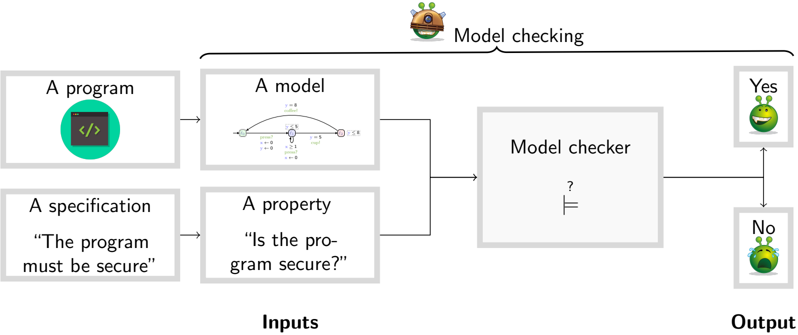
Methodology



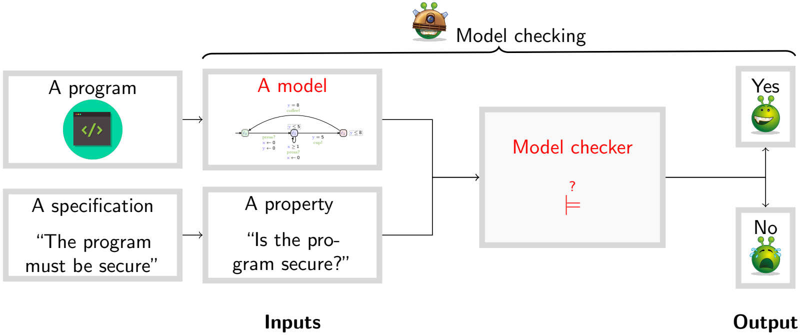
Methodology



Methodology



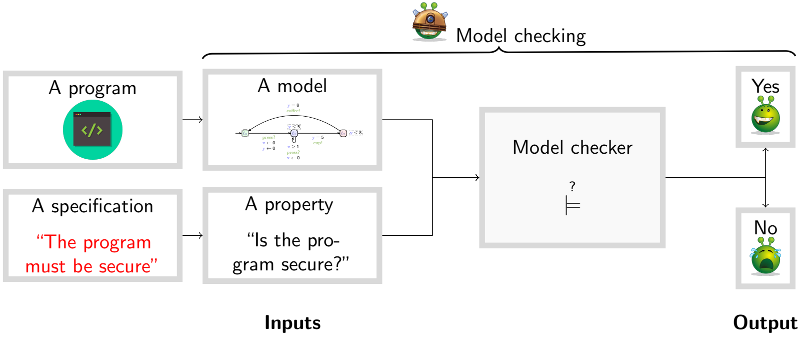
Outline



Outline

1. Preliminaries: Timed model checking

Outline



Outline

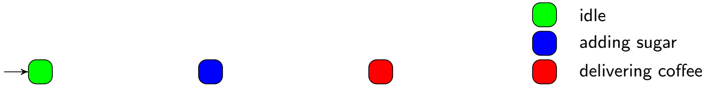
1. Preliminaries: Timed model checking
2. Timed opacity (& execution-timed opacity)

Outline

Timed automaton (TA)

[AD94]

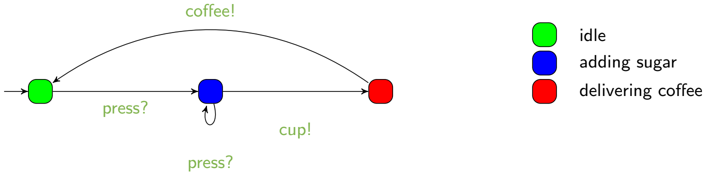
- Finite state automaton (sets of locations)



Timed automaton (TA)

[AD94]

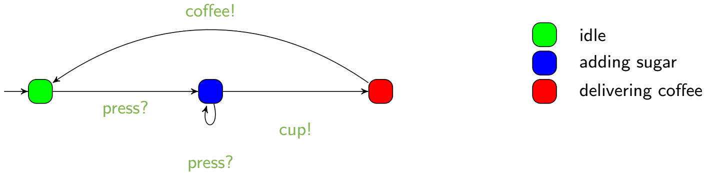
- Finite state automaton (sets of locations and actions)



Timed automaton (TA)

[AD94]

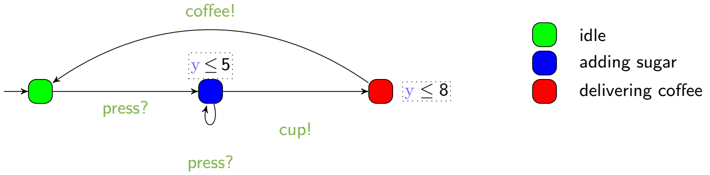
- Finite state automaton (sets of locations and actions) augmented with a set X of clocks
  - Real-valued variables evolving linearly at the same rate



Timed automaton (TA)

[AD94]

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants
- Features
  - Location invariant: property to be verified to stay at a location

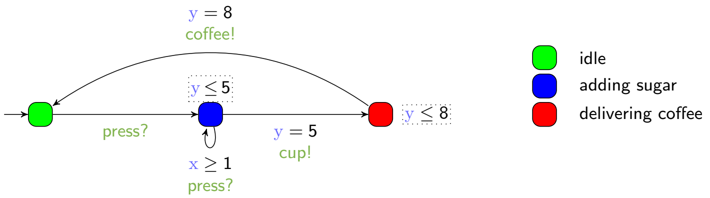


Timed automaton (TA)

[AD94]

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition

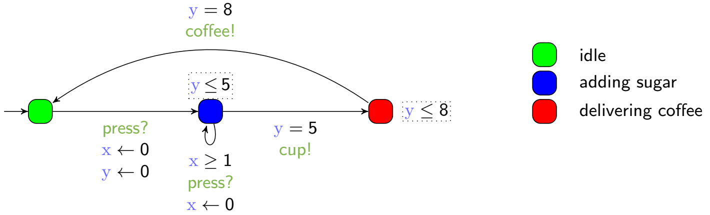


Timed automaton (TA)

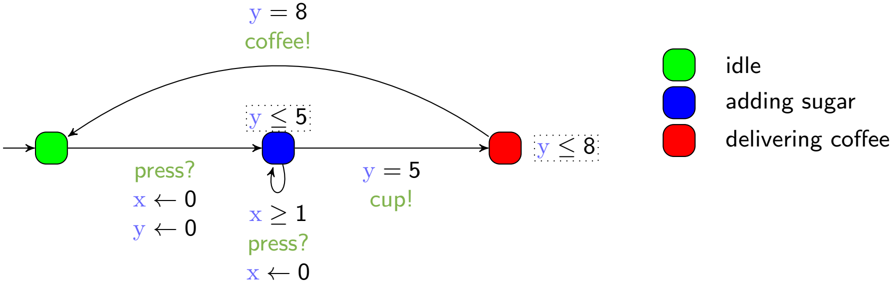
[AD94]

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

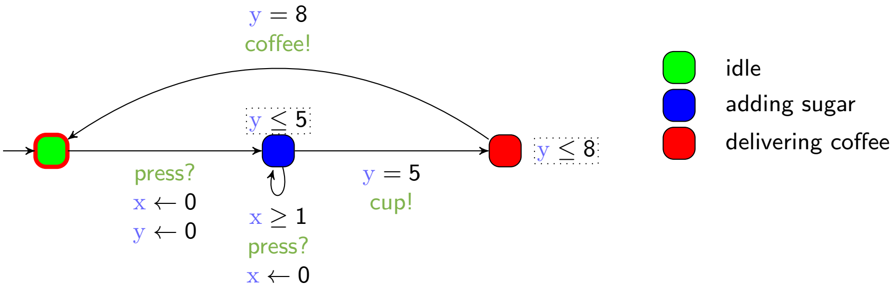
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition
  - Clock reset: some of the clocks can be set to 0 along transitions



The most critical system: The coffee machine



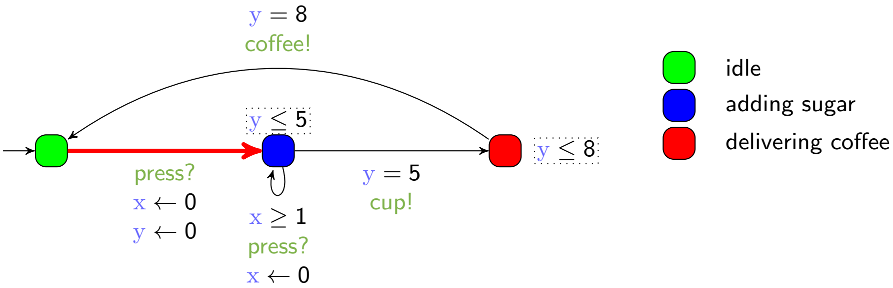
The most critical system: The coffee machine



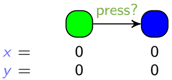
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

$x =$	0
$y =$	0

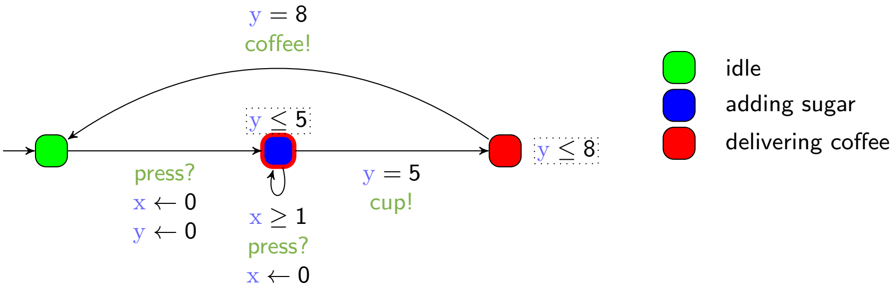
The most critical system: The coffee machine



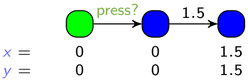
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



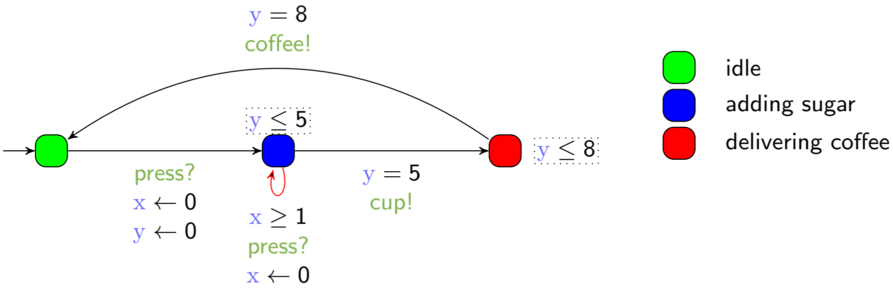
The most critical system: The coffee machine



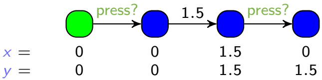
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



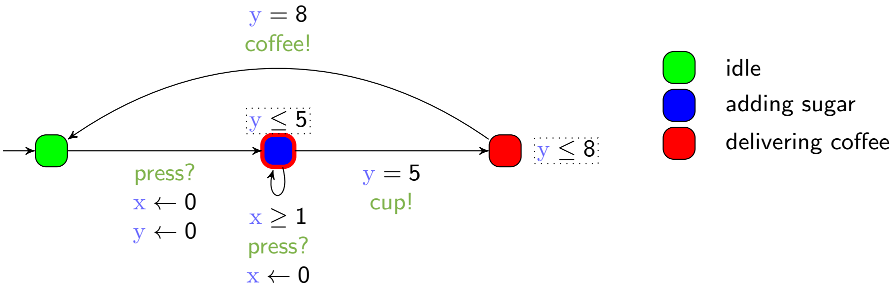
The most critical system: The coffee machine



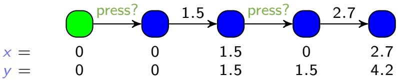
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



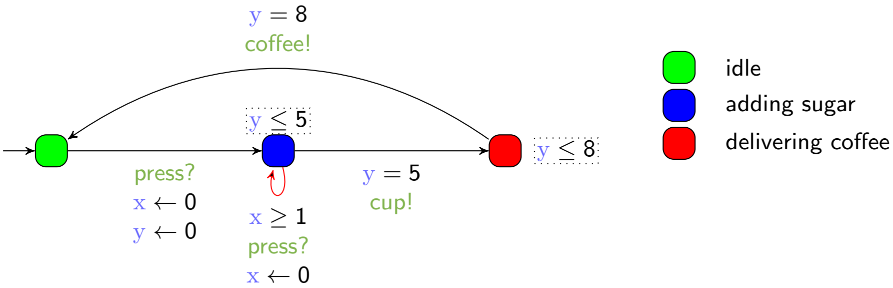
The most critical system: The coffee machine



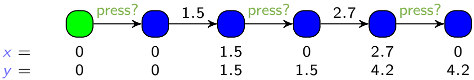
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



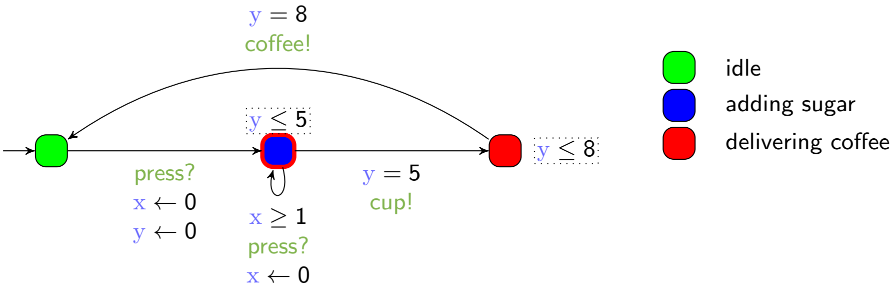
The most critical system: The coffee machine



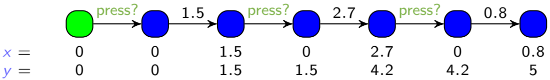
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



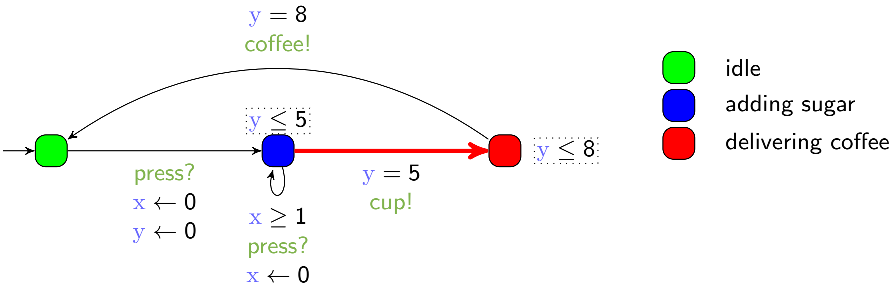
The most critical system: The coffee machine



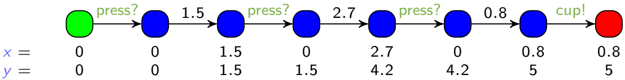
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



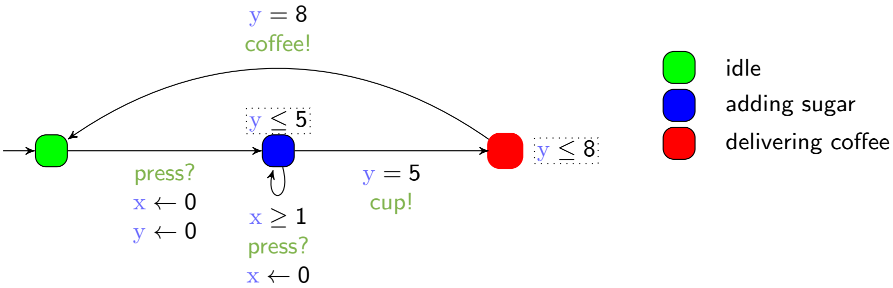
The most critical system: The coffee machine



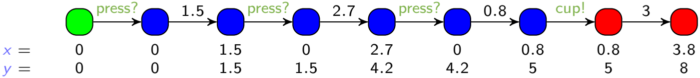
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



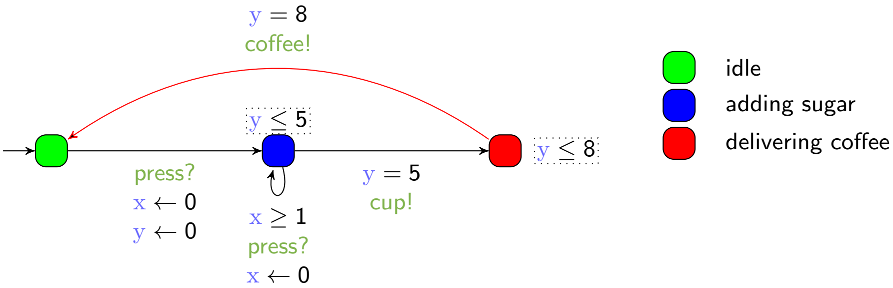
The most critical system: The coffee machine



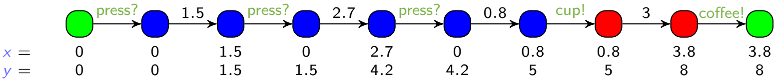
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



The most critical system: The coffee machine



- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



Outline

A first attacker model

Attacker capabilities

- Has access to the model (white box)
- Can observe an execution

A first attacker model

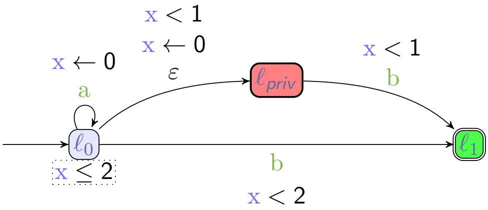
Attacker capabilities

- Has access to the model (white box)
- Can observe an execution

Attacker goal

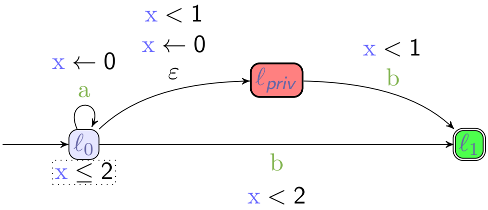
- Wants to deduce some private information based on these observations
  $\rightarrow$ visit of a private location

Attacker Setting



- Observed trace: $(a, 0.7)(b, 1.3)$

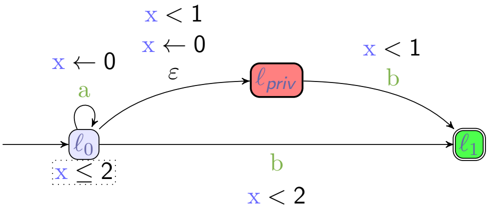
Attacker Setting



- Observed trace: $(a, 0.7)(b, 1.3)$

Question: Can they infer if ℓ_{priv} has been visited ?

Attacker Setting



- Observed trace: $(a, 0.7)(b, 1.3)$

Question: Can they infer if ℓ_{priv} has been visited ?

No: there is

- a run visiting ℓ_{priv}
- a run not visiting ℓ_{priv} of trace $(a, 0.7)(b, 1.3)$ too.

Opacity in Timed Automata

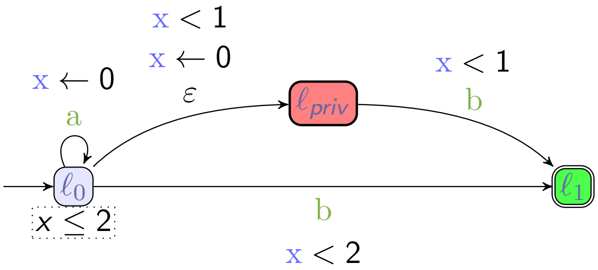
The TA is opaque iff all traces can be obtained **both**

- by runs visiting ℓ_{priv}
- and by runs not visiting it.

Opacity in Timed Automata

The TA is opaque iff all traces can be obtained **both**

- by runs visiting ℓ_{priv}
- and by runs not visiting it.

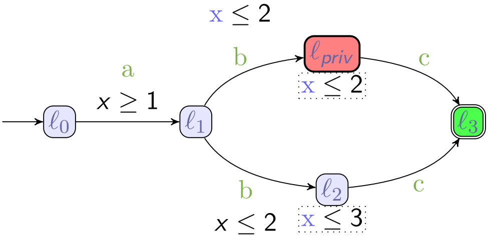


OPAQUE

Opacity in Timed Automata

The TA is opaque iff all traces can be obtained **both**

- by runs visiting ℓ_{priv}
- and by runs not visiting it.



NON OPAQUE

non-opaque trace: $(a, 1)(b, 2)(c, 3)$

Decision problem

Opacity Decision Problem

Is the given timed automaton opaque?

[Cas09] Franck Cassez. "The Dark Side of Timed Opacity". In: ISA (2009). LNCS. Springer, 2009

Decision problem

Opacity Decision Problem

Is the given timed automaton opaque?

Franck Cassez, *The Dark Side of Timed Opacity* (2009)
$\longrightarrow$ Opacity is undecidable for timed automata!

So... is it the end?

[Cas09] Franck Cassez. "The Dark Side of Timed Opacity". In: ISA (2009). LNCS. Springer, 2009

Decision problem

Opacity Decision Problem

Is the given timed automaton opaque?

Franck Cassez, *The Dark Side of Timed Opacity* (2009)
$\longrightarrow$ Opacity is undecidable for timed automata!

So... is it the end? Not yet!

[Cas09] Franck Cassez. "The Dark Side of Timed Opacity". In: ISA (2009). LNCS. Springer, 2009

Outline

Our Contributions

- *change the system:*
  subclasses of TA for which opacity can be decided
  - restriction on the number of actions
  - restriction on the number of clocks
  - discrete time

- *change the problem* $\rightarrow$ weaker attackers
  - bounded number of observations
  - limited observation

Outline

Changing the System

Subclass	Opacity
One-action TAs	$\times$
One-clock TAs without silent actions	$\surd$ non-primitive rec.-c.
One-clock TAs with silent actions	$\times$
($>$1)-clock TAs	$\times$
Discrete-time TAs	$\surd$EXPSPACE-c.[2]
Observable ERAs	$\surd$ PSPACE-c.

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024

[2]Fun fact: decidability result also proved this year in *Verifying opacity of discrete-timed automata*, Klein and al., FormaliSE'24 and in *The opacity of timed automata*, An and al., FM 2024

Outline

Weakening the Attacker

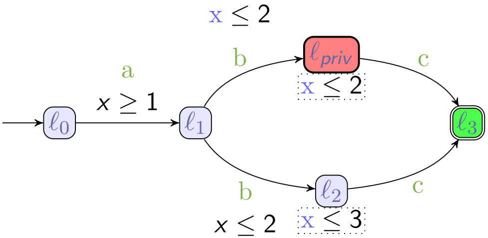
What if the attacker has a limited observation budget?

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024

Weakening the Attacker

What if the attacker has a limited observation budget?

The attacker can only see the first N observations of the run.



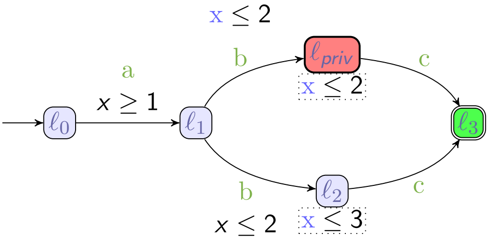
Possible traces with $N = 2$: $(a, \tau_1)(b, \tau_2)$ with $1 \leq \tau_1 \leq \tau_2 \leq 2$

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024

Weakening the Attacker

What if the attacker has a limited observation budget?

The attacker can only see the first N observations of the run.



Possible traces with $N = 2$: $(a, \tau_1)(b, \tau_2)$ with $1 \leq \tau_1 \leq \tau_2 \leq 2$

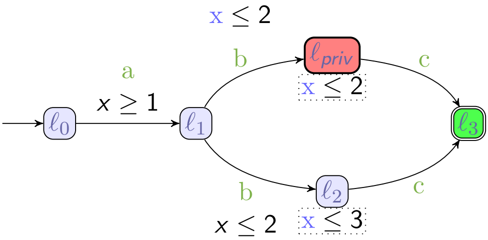
- OPAQUE with $N = 2$
- NON OPAQUE with $N = 3$: $(a, 1)(b, 2)(c, 3)$

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024

Weakening the Attacker

What if the attacker has a limited observation budget?

The attacker can only see the first N observations of the run.



Result

The problem of opacity with a bounded number of observations is decidable, and moreover we have a **2EXPSPACE** algorithm.

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024

Outline

Preliminaries: (Parametric) Timed model checking

Timed opacity

Solutions

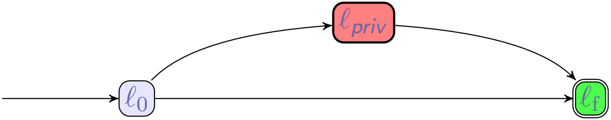
- Low dimension
- Bounded number of observations
- Execution-time opacity

Conclusion & Perspectives

Formalization

Hypotheses: [AS19][TOSEM22]

- A start location ℓ_0 and an end location ℓ_f
- A special private location ℓ_{priv}

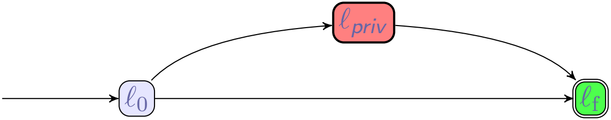


[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM (2022)

Formalization

Hypotheses: [AS19][TOSEM22]

- A start location ℓ_0 and an end location ℓ_f
- A special private location ℓ_{priv}



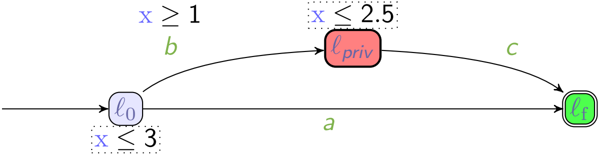
Definition (execution-time opacity)

The system is ET-opaque for a duration d if there exist two runs to ℓ_f of duration d

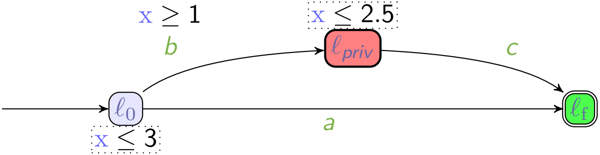
1. one visiting ℓ_{priv}
2. one not visiting ℓ_{priv}

[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM (2022)

Example

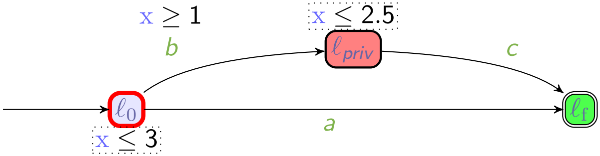


Example



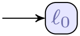
- There exist (at least) two runs of duration $d = 2$:

Example

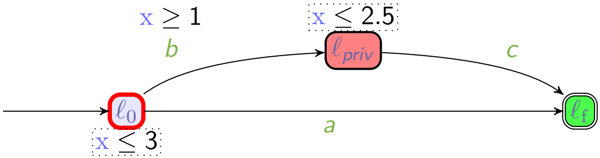


- There exist (at least) two runs of duration $d = 2$:

visiting ℓ_{priv}

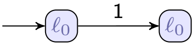


Example

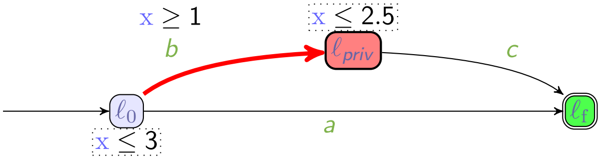


- There exist (at least) two runs of duration $d = 2$:

visiting ℓ_{priv}

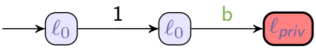


Example

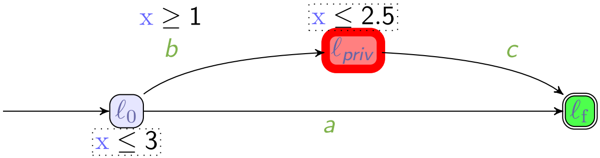


- There exist (at least) two runs of duration $d = 2$:

visiting ℓ_{priv}

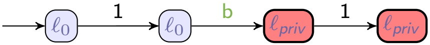


Example

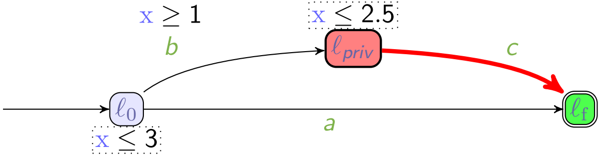


- There exist (at least) two runs of duration $d = 2$:

visiting ℓ_{priv}

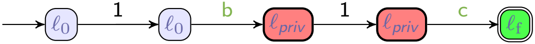


Example

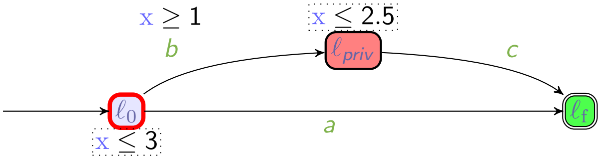


- There exist (at least) two runs of duration $d = 2$:

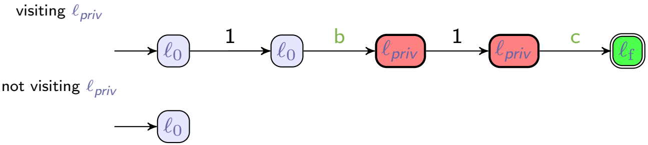
visiting ℓ_{priv}



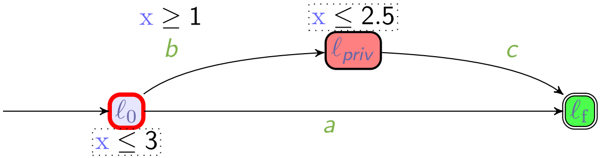
Example



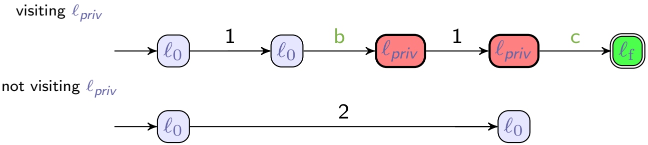
- There exist (at least) two runs of duration $d = 2$:



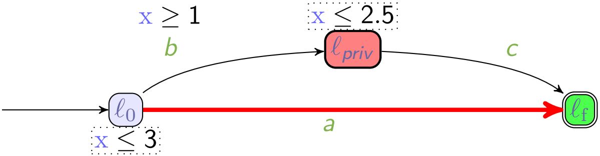
Example



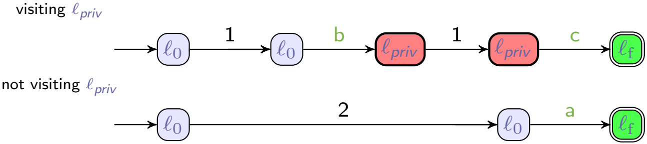
- There exist (at least) two runs of duration $d = 2$:



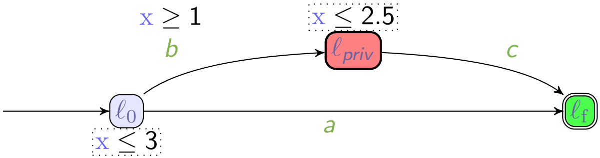
Example



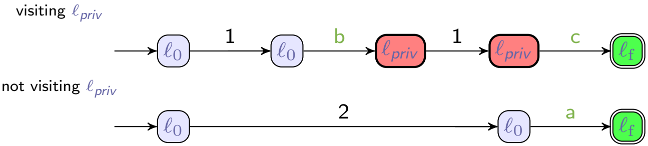
- There exist (at least) two runs of duration $d = 2$:



Example



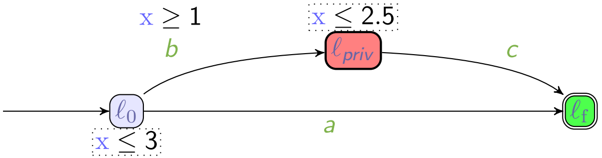
- There exist (at least) two runs of duration $\mathrm{d} = 2$:



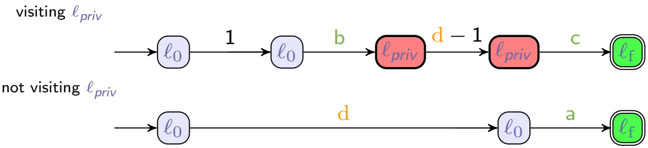
The system is ET-opaque for a duration $\mathrm{d} = 2$

The system is $\exists$-ET-opaque

Example



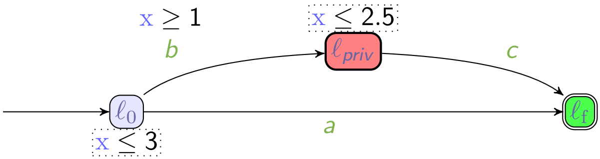
- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$:



The system is ET-opaque for all durations in $[1, 2.5]$

The system is ∃-ET-opaque

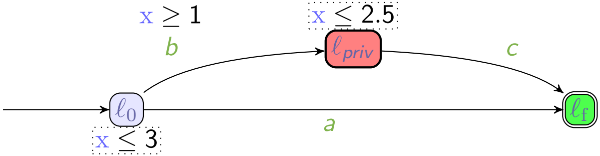
Example



- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

The system is ∃-ET-opaque

Example

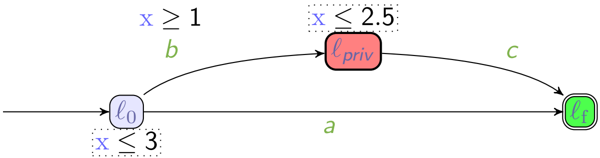


- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

The system is ∃-ET-opaque

- private durations are $[1, 2.5]$
  public durations are $[0, 3]$

Example

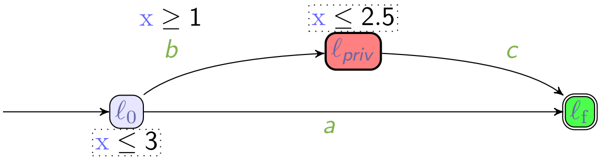


- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

The system is ∃-ET-opaque

- private durations are $[1, 2.5]$
  public durations are $[0, 3]$
- private durations $\subseteq$ public durations

Example



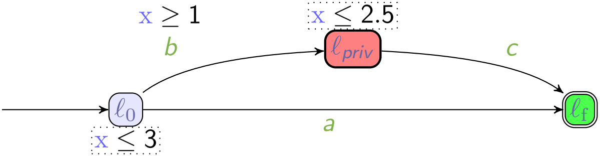
- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

The system is ∃-ET-opaque

- private durations are $[1, 2.5]$
  public durations are $[0, 3]$
- private durations $\subseteq$ public durations

The system is weakly ET-opaque

Example



- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

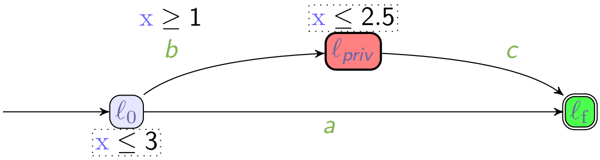
The system is ∃-ET-opaque

- private durations are $[1, 2.5]$
  public durations are $[0, 3]$
- private durations $\subseteq$ public durations

The system is weakly ET-opaque

- private durations $\neq$ public durations

Example



- There exist (at least) two runs of duration d for all durations $d \in [1, 2.5]$

The system is $\exists$-ET-opaque

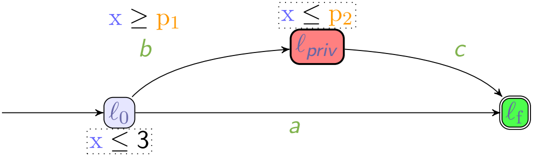
- private durations are $[1, 2.5]$
  public durations are $[0, 3]$
- private durations $\subseteq$ public durations

The system is weakly ET-opaque

- private durations $\neq$ public durations

The system is not fully ET-opaque

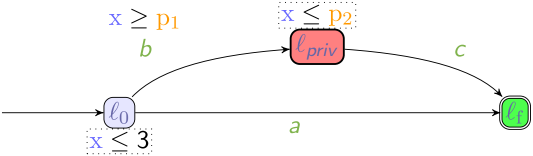
A parametric extension



Private	$[p_1, p_2]$
Public	$[0, 3]$

ET-opacity notion	∃	Weak	Full
p-Emptiness			
p-Synthesis			

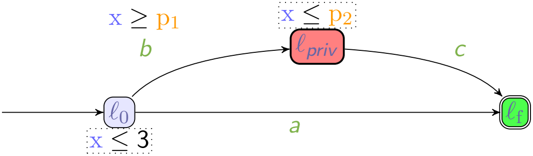
A parametric extension



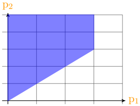
Private	$[p_1, p_2]$
Public	$[0, 3]$

ET-opacity notion	$\exists$	Weak	Full
p-Emptiness	$\times(\exists v)$	$\times(\exists v)$	$\times(\exists v)$
p-Synthesis			

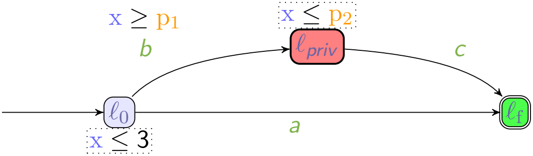
A parametric extension



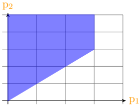
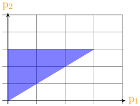
Private	$[p_1, p_2]$
Public	$[0, 3]$

ET-opacity notion	$\exists$	Weak	Full
p-Emptiness	$\times(\exists v)$	$\times(\exists v)$	$\times(\exists v)$
p-Synthesis	$0 \leq p_1 \leq 3$ $\wedge\ p_1 \leq p_2$ 		

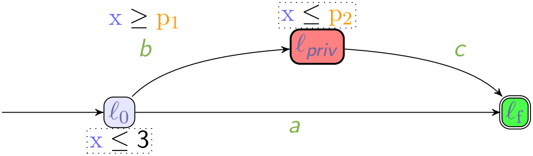
A parametric extension



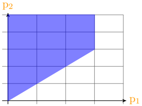
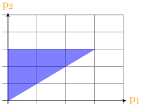
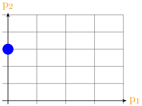
Private	$[p_1, p_2]$
Public	$[0, 3]$

ET-opacity notion	∃	Weak	Full
p-Emptiness	$\times(\exists v)$	$\times(\exists v)$	$\times(\exists v)$
p-Synthesis	$0 \leq p_1 \leq 3$ $\wedge\ p_1 \leq p_2$ 	$0 \leq p_1 \wedge p_2 \leq 3$ $\wedge\ p_1 \leq p_2$ 	

A parametric extension



Private	$[p_1, p_2]$
Public	$[0, 3]$

ET-opacity notion	∃	Weak	Full
p-Emptiness	$\times(\exists v)$	$\times(\exists v)$	$\times(\exists v)$
p-Synthesis	$0 \leq p_1 \leq 3 \wedge p_1 \leq p_2$ 	$0 \leq p_1 \wedge p_2 \leq 3 \wedge p_1 \leq p_2$ 	$p_1 = 0 \wedge p_2 = 3$ 

Decidability results for ET-opacity

		∃-ET-opaque	weakly ET-opaque	fully ET-opaque
Decision	TA	√	√	√
p-emptiness	L/U-PTA	√	×	×
	PTA	×	×	×
p-synthesis	L/U-PTA	×	×	×
	PTA	×	×	×

- L/U-PTA (Lower/Upper-PTA): subclass of PTA where the parameters are partitioned into two sets (either compared to clocks as upperbound, or as lower bound) [Hun+02]
- *Proofs are based on the region automaton (for TAs) and by reduction from EF-emptiness (for PTAs). (see formal proofs in [TOSEM22])*

[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM (2022)

Expiring ET-opacity

- How to deal with outdated secrets?
  e. g., cache values, status of the memory, . . .

Idea

The secret can expire: beyond a certain duration, knowing the secret is useless to the attacker (e. g., a cache value) [Amm+21]

Expiring ET-opacity

Assumption

Knowing an expired secret is equivalent to not knowing a secret

	Secret runs	**Non-secret runs**
ET-opacity	Runs visiting the private location (= private runs)	Runs not visiting the private location (= public runs)
expiring-ET-opacity	Private runs with ℓ_{priv} visit $\leq \Delta$ before the system completion	(i) Public runs and (ii) Private runs with ℓ_{priv} visit $> \Delta$ before the system completion

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023

Decidability results for expiring-ET-opacity

		weakly expiring-ET-opaque	fully expiring-ET-opaque
Δ-emptiness	TA	√	√
Δ-synthesis	TA	√	?
$(p + \Delta)$-emptiness	L/U-PTA	×	×
	PTA	×	×
$(p + \Delta)$-synthesis	L/U-PTA	×	×
	PTA	×	×

- ∃-expiring ET-opacity was left as a future work.
- L/U-PTA (Lower/Upper-PTA): subclass of PTA where the parameters are partitioned into two sets (either compared to clocks as upperbound, or as lower bound) [Hun+02]

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023

Decidability results for expiring-ET-opacity

		weakly expiring-ET-opaque	fully expiring-ET-opaque
Δ-emptiness	TA	√	√
Δ-synthesis	TA	√	?
$(p + \Delta)$-emptiness	L/U-PTA	×	×
	PTA	×	×
$(p + \Delta)$-synthesis	L/U-PTA	×	×
	PTA	×	×

- ∃-expiring ET-opacity was left as a future work.
- L/U-PTA (Lower/Upper-PTA): subclass of PTA where the parameters are partitioned into two sets (either compared to clocks as upperbound, or as lower bound) [Hun+02]
- *Proofs are based on the region automaton (for TAs) and by reduction from EF-emptiness (for PTAs). (see formal proofs in [ICECCS23])*

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023

Outline

Conclusion

Context: vulnerability by timing-attacks

- Goal: avoid leaking information on whether some discrete state has been visited
- Variations of the notion of timed opacity
  - Model: weaker models considered
  - Attacker: limited number of observations & observability of the global execution time

Several problems studied for timed automata

- ☹ Mostly undecidable with observations
- ☺ Mostly decidable for weaker attackers

Conclusion

Extension of ET-opacity to parametric timed automata

- ☹ Quickly undecidable
- ☺ One procedure for one synthesis problem

Other contributions

- Untimed and timed control
- $\exists$ and weak timed opacity with observations

Perspectives

Theoretical perspectives

- Existential version of expiring ET-opacity
- Δ-synthesis for full expiring ET-opacity

Algorithmic perspectives

- Synthesis for weak and full ET-opacity
- Synthesis for expiring problems

Automatic translation of programs to PTAs

- Our translation required non-trivial creativity
  $\rightarrow$ Translation with Petri nets including cache system

Perspectives

Theoretical perspectives

- Existential version of expiring ET-opacity
- Δ-synthesis for full expiring ET-opacity

Algorithmic perspectives

- Synthesis for weak and full ET-opacity
- Synthesis for expiring problems

Automatic translation of programs to PTAs

- Our translation required non-trivial creativity
  $\rightarrow$ Translation with Petri nets including cache system **see you in SAC'25!**

References I

[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: TCS 126 (Apr. 1994).

[Amm+21] Ikhlass Ammar, Yamen El Touati, Moez Yeddes, and John Mullins. "Bounded opacity for timed systems". In: Journal of Information Security and Applications 61 (Sept. 2021).

[AS19] Étienne André and Jun Sun. "Parametric Timed Model Checking for Guaranteeing Timed Opacity". In: ATVA (2019). LNCS. Springer, 2019.

[Cas09] Franck Cassez. "The Dark Side of Timed Opacity". In: ISA (2009). LNCS. Springer, 2009.

[ÉL24] Sarah Dépernet Étienne André and Engel Lefaucheux. "The Bright Side of Timed Opacity". In: ICFEM. 2024.

References II

[Hun+02] Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. "Linear parametric model checking of timed automata". In: Journal of Logic and Algebraic Programming 52-53 (2002).

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023.

[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM 31 (2022).

Licensing

Source of the graphics used I

Title: Smiley green alien big eyes (aaah)
Author: LadyofHats
Source: `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License: public domain

Title: Smiley green alien big eyes (cry)
Author: LadyofHats
Source: `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License: public domain

Title: Smiley green alien exterminate
Author: LadyofHats
Source: `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_exterminate.svg`
License: public domain

Title: Piratey, vector version
Author: Gustavb
Source: `https://commons.wikimedia.org/wiki/File:Piratey,_vector_version.svg`
License: CC by-sa

Title: Expired
Author: RRZEicons
Source: `https://commons.wikimedia.org/wiki/File:Expired.svg`
License: CC by-sa

License of this document

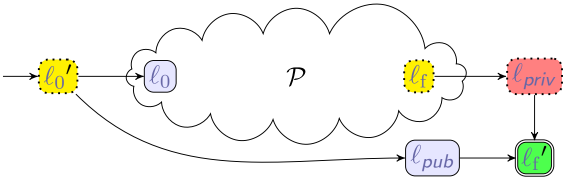
Authors: **Étienne André**, **Sarah Dépernet**, and **Dylan Marinho**

ET-opacity synthesis is (very) difficult

Theorem (Undecidability of ∃-ET-opacity p-emptiness)

Given $\mathcal{P}$, the mere existence of a parameter valuation v s. t. $v(\mathcal{P})$ ∃-ET-opacity is undecidable.

Proof idea: reduction from reachability-emptiness for PTAs



Remark: L/U-PTA is a decidable subclass