**SynCoP 2023**

**Paris, France**

22nd April 2023

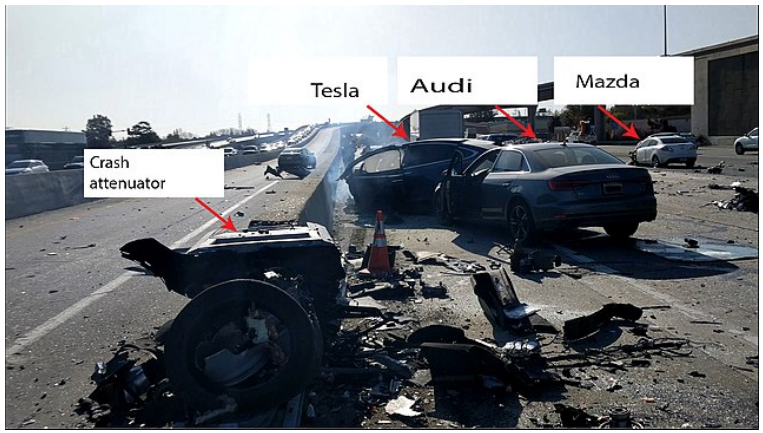
# Monitoring cyber-physical systems under uncertainty

Étienne André

Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030, F-93430 Villetaneuse, France
Joint work with Masaki Waga and Ichiro Hasuo

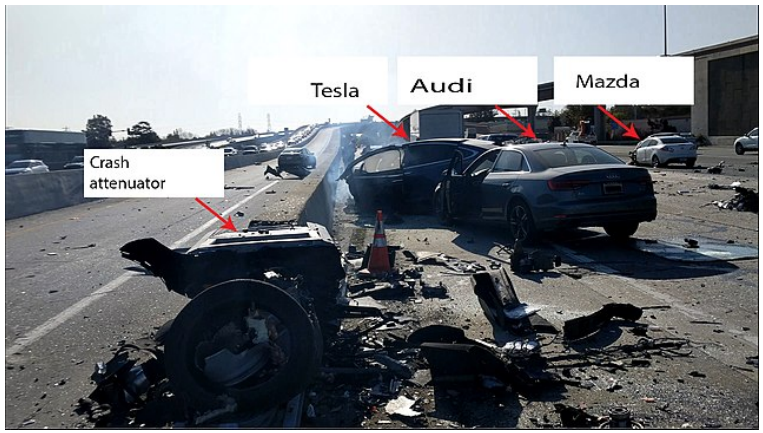
# Outline

# Context: safety-critical cyber-physical systems




Images illustrating Tesla fatal crashes: Williston, Florida, USA [May 7, 2016]; Mountain View, California, USA [March 23, 2018]

# Context: safety-critical cyber-physical systems



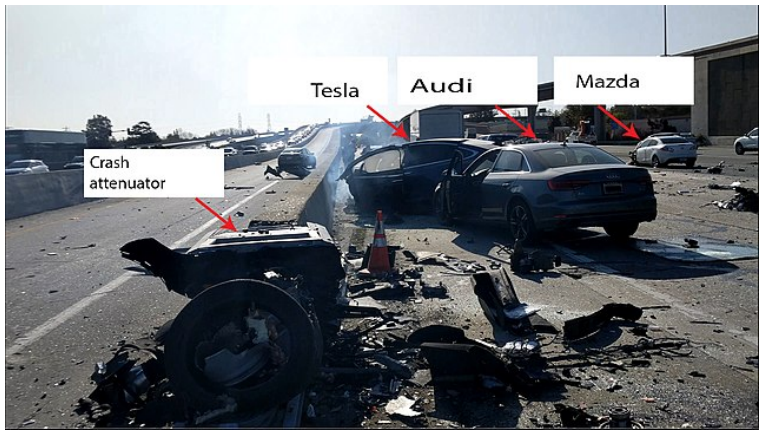

Images illustrating Tesla fatal crashes: Williston, Florida, USA [May 7, 2016]; Mountain View, California, USA [March 23, 2018]

Formal verification of complex cyber-physical systems: out of reach?

Lightweight verification

- Testing
- Monitoring, runtime verification

# Context: safety-critical cyber-physical systems




Images illustrating Tesla fatal crashes: Williston, Florida, USA [May 7, 2016]; Mountain View, California, USA [March 23, 2018]

Formal verification of complex cyber-physical systems: out of reach?

Lightweight verification

- Testing
- Monitoring, runtime verification

# Outline

# Motivation: monitoring logs from the automotive industry

- Modern cars embed several processors and produce logs

# Motivation: monitoring logs from the automotive industry

- Modern cars embed several processors and produce logs


- Log: sequences of events and timestamps

```
start    2.3
gear1    5.8
gear2    9.2
gear3   18.5
gear2   42.1
```

# Motivation: monitoring logs from the automotive industry

- Modern cars embed several processors and produce logs


  - Log: sequences of events and timestamps

```
start    2.3
gear1    5.8
gear2    9.2
gear3   18.5
gear2   42.1
```

- How to ensure on-the-fly that some properties are satisfied on a log?
  - "It never happens that gear1 and gear3 are separated by less than 5 s"

# Motivation: monitoring logs from the automotive industry

- Modern cars embed several processors and produce logs


  - Log: sequences of events and timestamps

```
start    2.3
gear1    5.8
gear2    9.2
gear3   18.5
gear2   42.1
```

- How to ensure on-the-fly that some properties are satisfied on a log?
  - "It never happens that gear1 and gear3 are separated by less than 5 s"

$\Rightarrow$ Monitoring

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

```
start walking                                    2.3
```

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

```
start walking                    2.3
walk faster                      6.3
```

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

```
start walking          2.3
walk faster            6.3
receive SMS           15.8
```

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

| | |
|---|---|
| `start walking` | 2.3 |
| `walk faster` | 6.3 |
| `receive SMS` | 15.8 |
| `read SMS` | 19.2 |

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

```
start walking                              2.3
walk faster                                6.3
receive SMS                               15.8
read SMS                                  19.2
sound of someone bumping into a lamp      22.5
```

# Larger motivation: data collection and management

- Personal mobile devices collect large amounts of data

These data can also come in the form of a timed log

```
start walking                              2.3
walk faster                                6.3
receive SMS                               15.8
read SMS                                  19.2
sound of someone bumping into a lamp      22.5
```

- Key challenge: manage these data
  - Verify properties: “has the owner bumped into a street lamp”?
    - key applications (health, …)
  - Deduce information:
    - “what are the minimum/maximum intervals without visiting this shop”?
    - “is the user visiting this place more or less periodically?” (without knowing the actual period)

# Outline

# Untimed pattern matching: example

- Naive algorithm for pattern matching

$$\texttt{c}\quad\texttt{r}\quad\texttt{e}\quad\texttt{p}\quad\texttt{e}\quad\texttt{s}\quad \in ?L\left(\{c|i|d\}^?r^*e\right)$$

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | | | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^{?}r^{*}e\right)$ |
|---|---|---|---|---|---|---|
| c | r | | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^{?}r^{*}e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^?r^*e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ?L\left(\{c\|i\|d\}^?r^*e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | $\surd$ |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | $\surd$ |
| | | | p | | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | $\surd$ |
| | | | p | | | $\times$ |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | $\surd$ |
| | | | p | | | $\times$ |
| | | | | e | | |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | $\surd$ |
| | r | e | | | | $\surd$ |
| | | e | | | | $\surd$ |
| | | | p | | | $\times$ |
| | | | | e | | $\surd$ |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | ✓ |
| | r | e | | | | ✓ |
| | | e | | | | ✓ |
| | | | p | | | × |
| | | | | e | | ✓ |
| | | | | | s | |

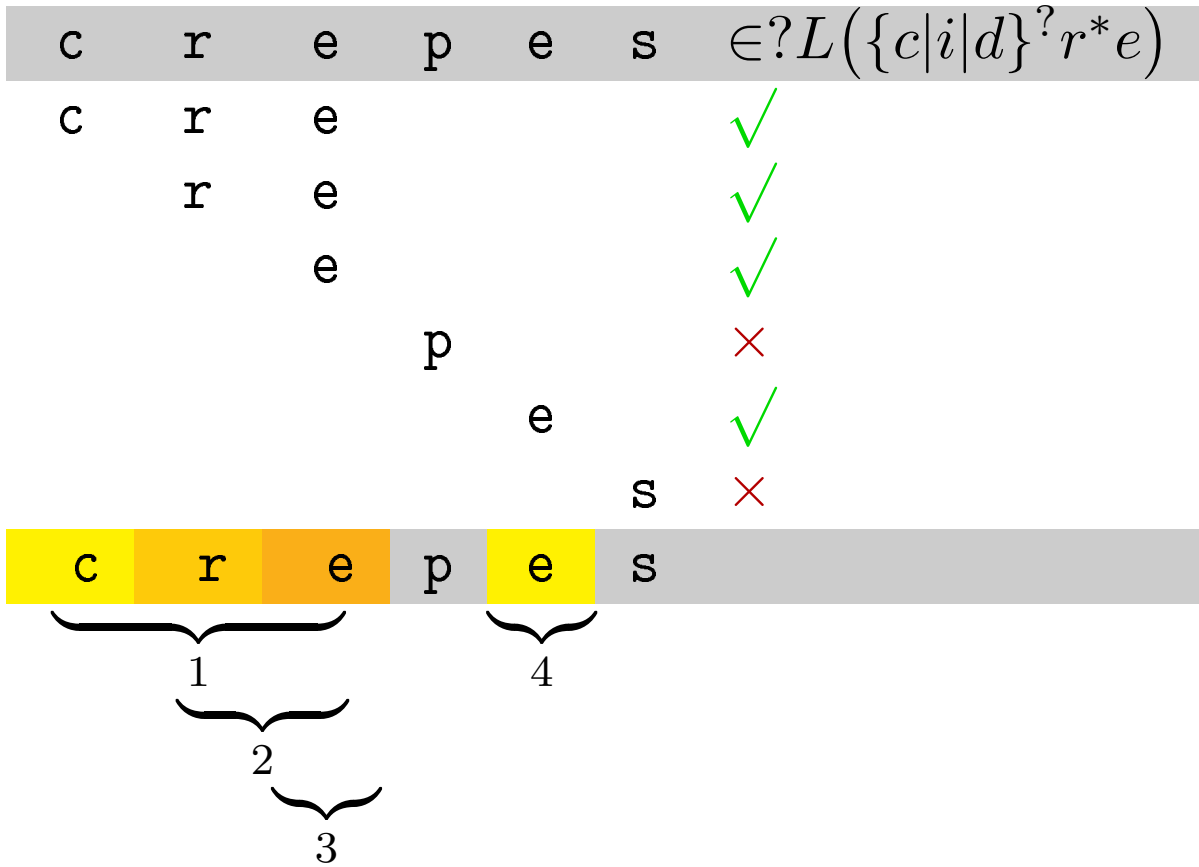
# Untimed pattern matching: example

- Naive algorithm for pattern matching

| c | r | e | p | e | s | $\in ? L\left(\{c \mid i \mid d\}^? r^* e\right)$ |
|---|---|---|---|---|---|---|
| c | r | e | | | | ✓ |
| | r | e | | | | ✓ |
| | | e | | | | ✓ |
| | | | p | | | × |
| | | | | e | | ✓ |
| | | | | | s | × |

# Untimed pattern matching: example

- Naive algorithm for pattern matching

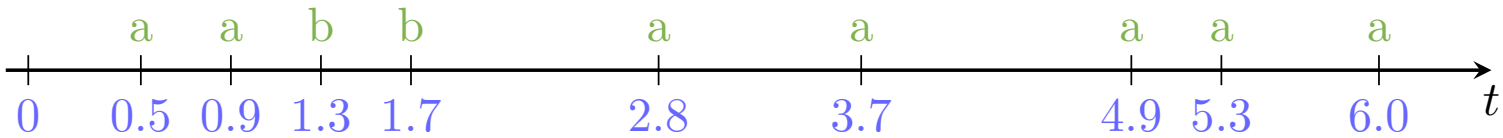


# Timed pattern matching: timed word

## Timed word

[AD94]

= sequence of actions and timestamps





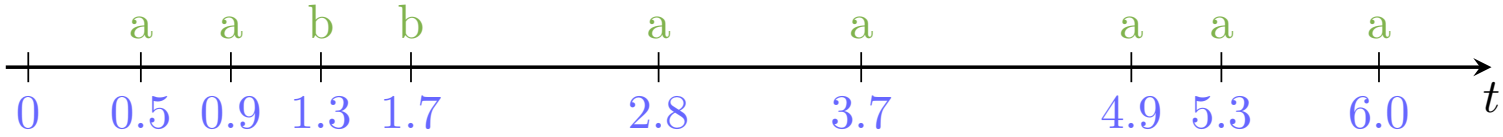
[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

[WAH16] Masaki Waga, Takumi Akazaki, and Ichiro Hasuo. "A Boyer-Moore Type Algorithm for Timed Pattern Matching". In: *FORMATS*. vol. 9884. LNCS. Springer, 2016, pp. 121–139

# Timed pattern matching: timed word

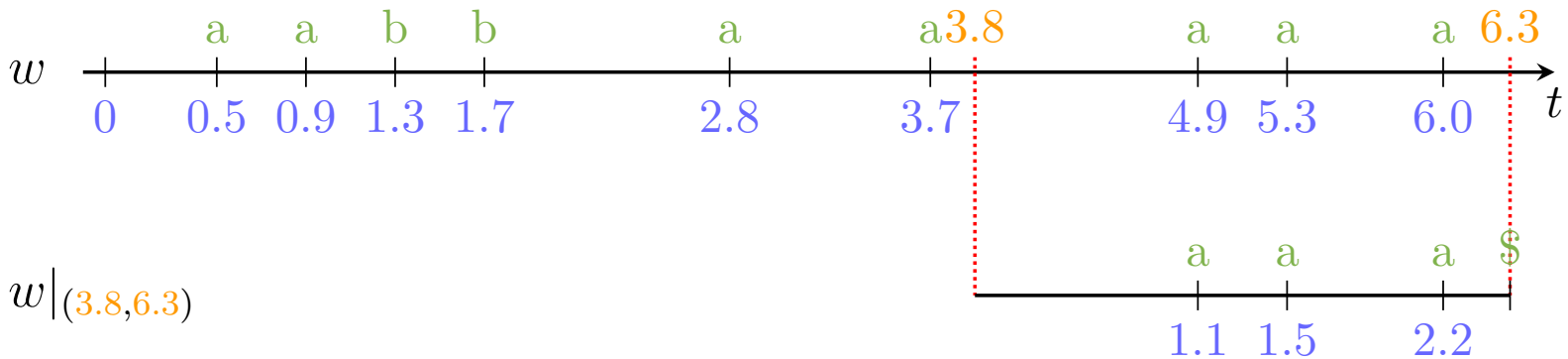
## Timed word [AD94]

= sequence of actions and timestamps




## Timed word segment [WAH16]

= projection of a segment of the timed word onto a given interval




---


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

[WAH16] Masaki Waga, Takumi Akazaki, and Ichiro Hasuo. "A Boyer-Moore Type Algorithm for Timed Pattern Matching". In: *FORMATS*. vol. 9884. LNCS. Springer, 2016, pp. 121–139

# Timed pattern matching: timed automaton

How to express a (timed) property on a log?

## Example

"At least 1 time unit after the start of the segment, $\mathrm{a}$ is observed.
Then, within strictly less than 1 time unit, another $\mathrm{a}$ is observed.
Then, within strictly less than 1 time unit, another $\mathrm{a}$ is observed."

[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

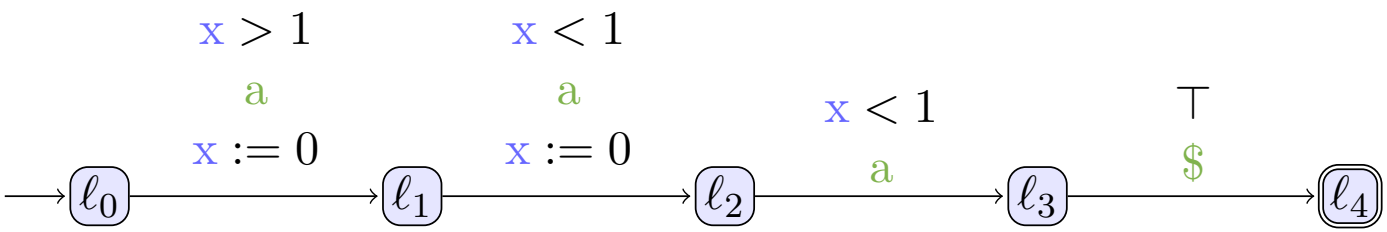
# Timed pattern matching: timed automaton

How to express a (timed) property on a log?

## Example

"At least 1 time unit after the start of the segment, a is observed.
Then, within strictly less than 1 time unit, another a is observed.
Then, within strictly less than 1 time unit, another a is observed."

A solution: timed automata [AD94]




- expressive
- well-studied
- supported by well-established model-checkers

---

[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

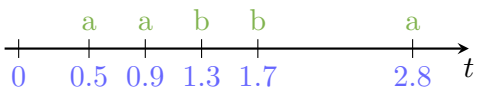
# Timed pattern matching: principle

Timed pattern matching

- Inputs

**A log**

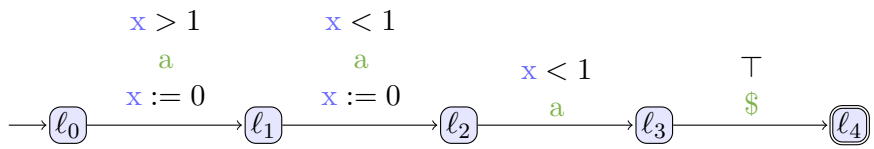
(timed word)




**A property**

usually a specification of faults

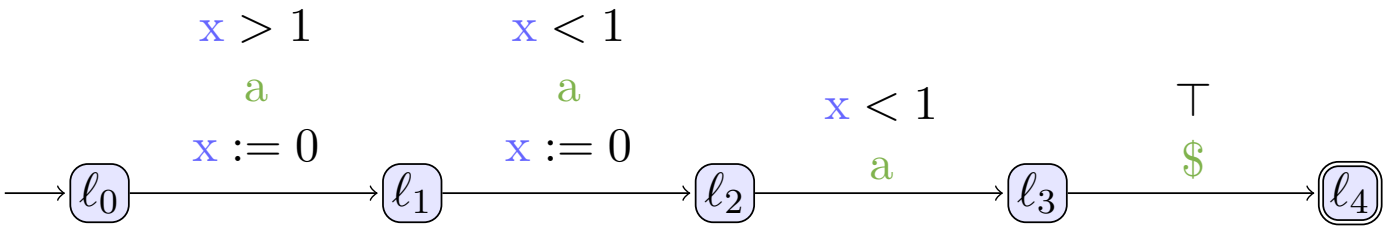
(timed automaton)



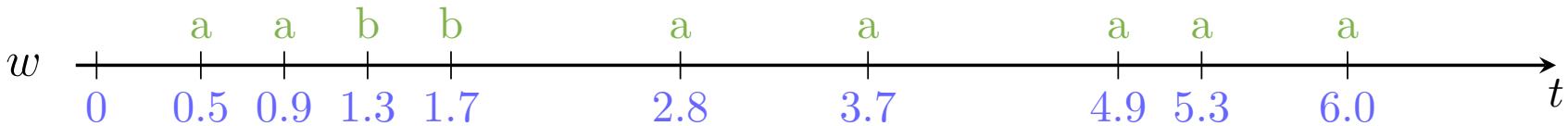

- Output
  - The set of time intervals where faults are detected
    $\Rightarrow$ Set of matching intervals $\{(t, t') \mid w|_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$

# Timed pattern matching: example

Our property:

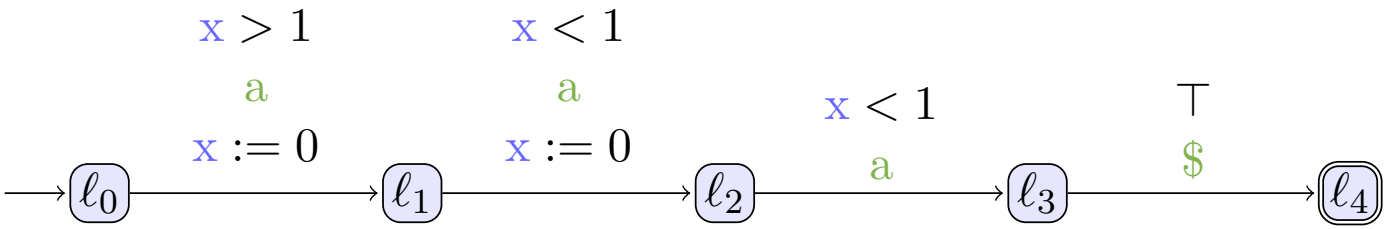



Our log:

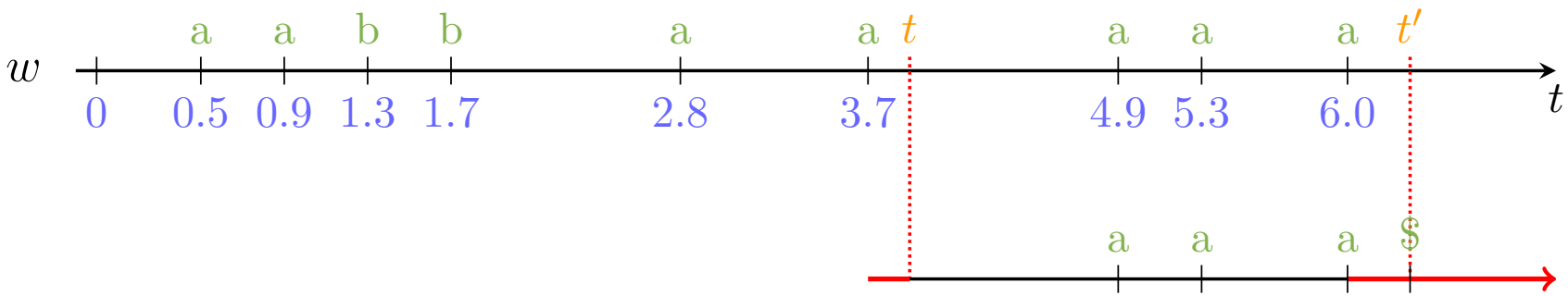


# Timed pattern matching: example

Our property:

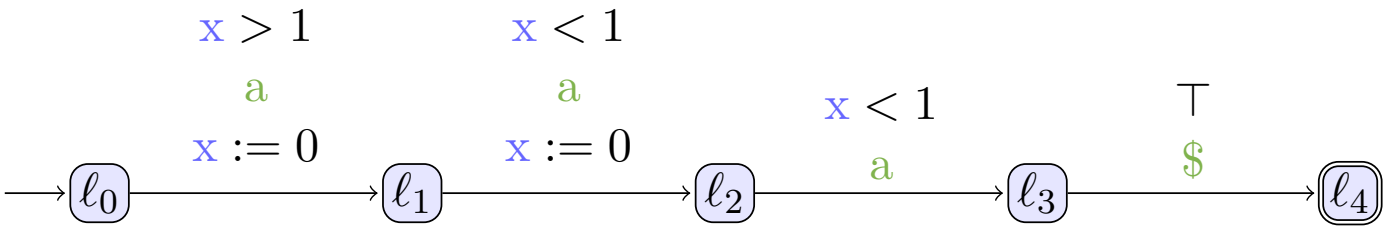



Our log:

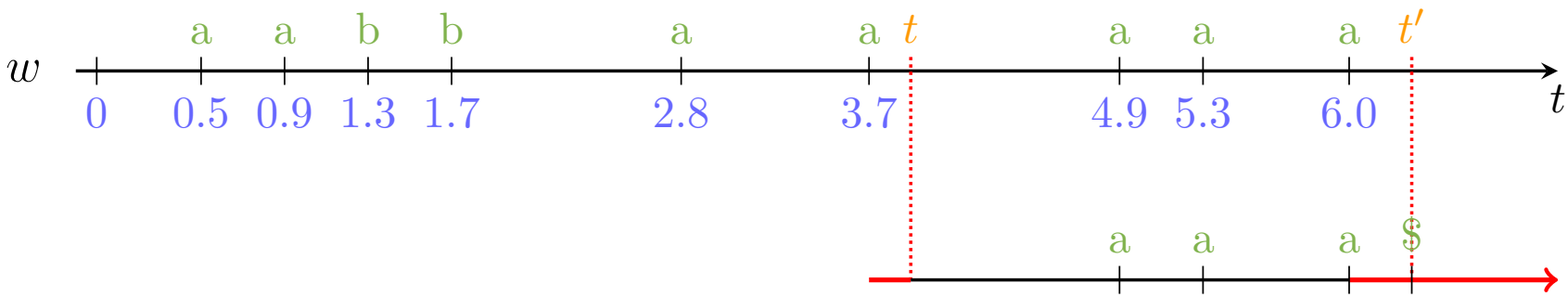


# Timed pattern matching: example

Our property:




Our log:




Set of matching intervals:

$$\{(t,t') \mid w|_{(t,t')} \in \mathcal{L}(\mathcal{A})\} = \{(t,t') \mid t \in (3.7, 3.9), t' \in [6.0, \infty)\}$$

# Previous works

- Timed pattern matching with signals [Ulu+14][Ulu+16][Ulu17]
  - logs are encoded by signals (i. e., values that vary over time)
    - *state-based* view, while our timed words are *event-based*
  - specification is encoded by timed regular expressions (TREs)

- Timed pattern matching with timed words and timed automata
  - [WAH16]: brute-force and Boyer-Moore algorithm
  - [WHS17]: online algorithm that employs skip values from the Franek–Jennings–Smyth string matching algorithm [FJS07]

---


[Ulu+14] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. "Timed Pattern Matching". In: *FORMATS*. vol. 8711. LNCS. Springer, 2014, pp. 222–236

[Ulu+16] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. "Online Timed Pattern Matching Using Derivatives". In: *TACAS*. vol. 9636. LNCS. Springer, 2016, pp. 736–751

[Ulu17] Dogan Ulus. "Montre: A Tool for Monitoring Timed Regular Expressions". In: *CAV, Part I*. vol. 10426. LNCS. Springer, 2017, pp. 329–335

[WAH16] Masaki Waga, Takumi Akazaki, and Ichiro Hasuo. "A Boyer-Moore Type Algorithm for Timed Pattern Matching". In: *FORMATS*. vol. 9884. LNCS. Springer, 2016, pp. 121–139

[WHS17] Masaki Waga, Ichiro Hasuo, and Kohei Suenaga. "Efficient Online Timed Pattern Matching by Automata-Based Skipping". In: *FORMATS*. vol. 10419. LNCS. Springer, 2017, pp. 224–243

[FJS07] Frantisek Franek, Christopher G. Jennings, and William F. Smyth. "A simple fast hybrid pattern-matching algorithm". In: *Journal of Discrete Algorithms* 5.4 (2007), pp. 682–695

# Goal: Extend timed pattern matching for uncertainty

## Challenges

- The property may not be known with full certainty:
  - Detect a periodic event but without knowing the period
    - "is the user visiting this place more or less periodically?" (without knowing the actual period)
- Optimization problems
  - Find minimal/maximal timings for which some property holds
    - "what are the minimum/maximum intervals without visiting this shop"?

---

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Goal: Extend timed pattern matching for uncertainty

Challenges

- The property may not be known with full certainty:
  - Detect a periodic event but without knowing the period
    - "is the user visiting this place more or less periodically?" (without knowing the actual period)
- Optimization problems
  - Find minimal/maximal timings for which some property holds
    - "what are the minimum/maximum intervals without visiting this shop"?

## Objective [WAH22b]

Find intervals of time and values of parameters for which a property holds

| Problem | log (target) | specification (pattern) | output |
|---|---|---|---|
| string matching | word | word $pat \in \Sigma^*$ | $\{(i,j) \in (\mathbb{Z}_{>0})^2 \mid w(i,j) = pat\}$ |
| pattern matching (PM) | word | NFA $\mathcal{A}$ | $\{(i,j) \in (\mathbb{Z}_{>0})^2 \mid w(i,j) \in \mathcal{L}(\mathcal{A})\}$ |
| timed PM | timed word | TA $\mathcal{A}$ | $\{(t,t') \in (\mathbb{R}_{>0})^2 \mid w\vert_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$ |
| parametric timed PM | timed word | PTA $\mathcal{A}$ | $\{(t,t',v) \mid w\vert_{(t,t')} \in \mathcal{L}(v(\mathcal{A}))\}$ |

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Outline

# Methodology

## Main idea

Use parametric timed model checking on parametric timed automata [AHV93]

- Toolkit: IMITATOR [And21]

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

[And21] Étienne André. "IMITATOR 3: Synthesis of timing parameters beyond decidability". In: *CAV*. vol. 12759. LNCS. Springer, 2021, pp. 1–14

# Methodology

## Main idea

Use parametric timed model checking on parametric timed automata [AHV93]

- Toolkit: IMITATOR [And21]

Methodology step by step

1. Encode the property using a PTA
2. Add two parameters $t$ and $t'$
3. Apply a (mild) transformation to the property PTA
4. Transform the timed word into a PTA
5. Perform the composition of both PTA
6. Apply reachability synthesis to the product

---


[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

[And21] Étienne André. "IMITATOR 3: Synthesis of timing parameters beyond decidability". In: *CAV*. vol. 12759. LNCS. Springer, 2021, pp. 1–14

# Methodology

## Main idea

Use parametric timed model checking on parametric timed automata [AHV93]

- Toolkit: IMITATOR [And21]

Methodology step by step

1. Encode the property using a PTA
2. Add two parameters $t$ and $t'$
3. Apply a (mild) transformation to the property PTA
4. Transform the timed word into a PTA
5. Perform the composition of both PTA
6. Apply reachability synthesis to the product

## Teaser

Our method is scalable!

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

[And21] Étienne André. "IMITATOR 3: Synthesis of timing parameters beyond decidability". In: *CAV*. vol. 12759. LNCS. Springer, 2021, pp. 1–14

# Outline

# Property: parametric timed automaton

Expressing a parametric timed property on a log

## Example

"At least $p_1$ time units after the start of the segment, $\mathrm{a}$ is observed.
Then, within strictly less than $p_2$ time units, another $\mathrm{a}$ is observed.
Then, within strictly less than $p_2$ time units, another $\mathrm{a}$ is observed."

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

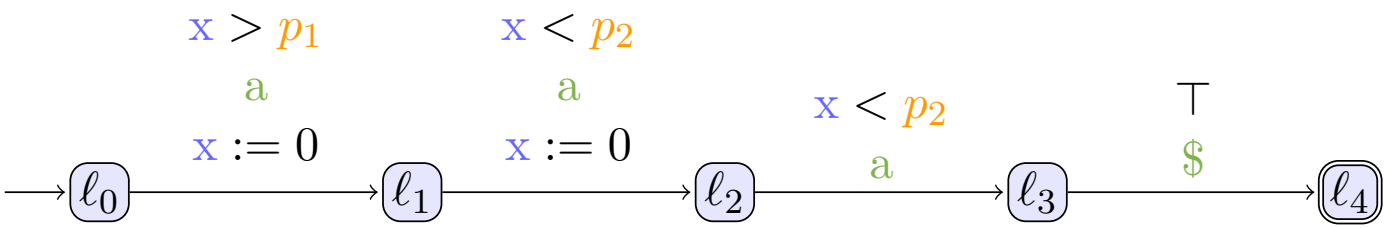
# Property: parametric timed automaton

Expressing a parametric timed property on a log

## Example

"At least $p_1$ time units after the start of the segment, $\mathrm{a}$ is observed.
Then, within strictly less than $p_2$ time units, another $\mathrm{a}$ is observed.
Then, within strictly less than $p_2$ time units, another $\mathrm{a}$ is observed."

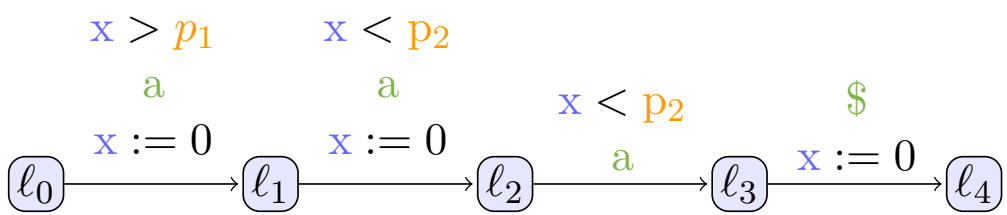
Our solution: use parametric timed automata [AHV93]




[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

# Modifying the property pattern

Add some start and end gadgets for completeness of the method [WAH22b]

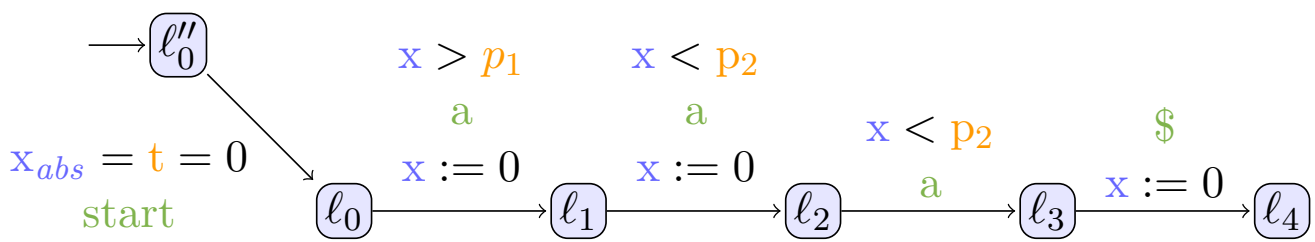



[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Modifying the property pattern

Add some start and end gadgets for completeness of the method [WAH22b]

1. Add an initial transition in $0$-time
   - Captures segments starting from $0$



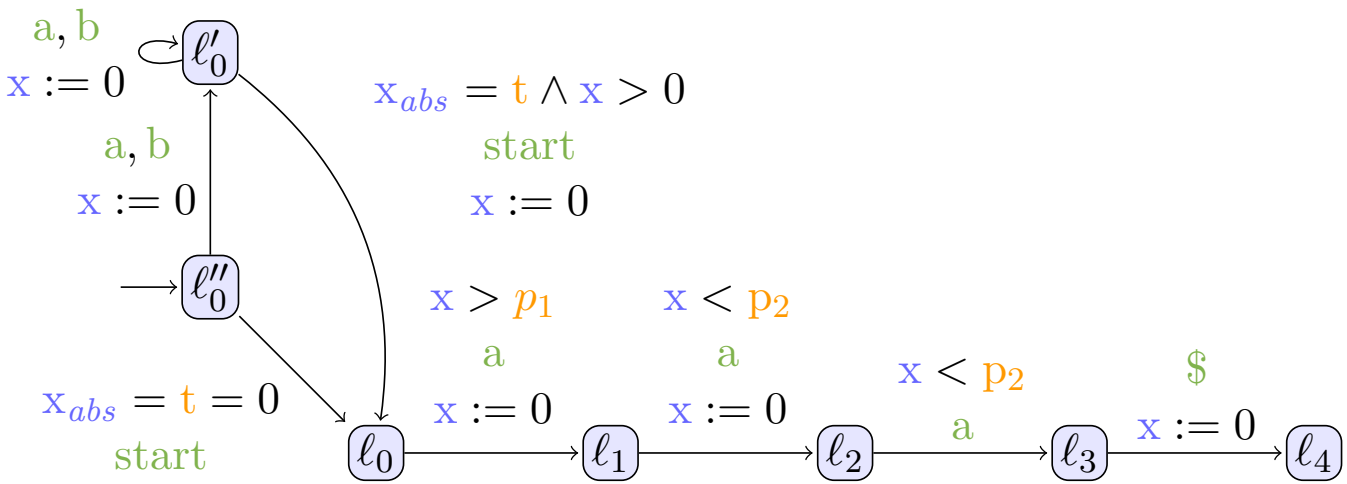


[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Modifying the property pattern

Add some start and end gadgets for completeness of the method [WAH22b]

1. Add an initial transition in $0$-time
   - Captures segments starting from $0$
2. Add a new location with a self-loop
   - Captures segments not starting from the beginning of the word



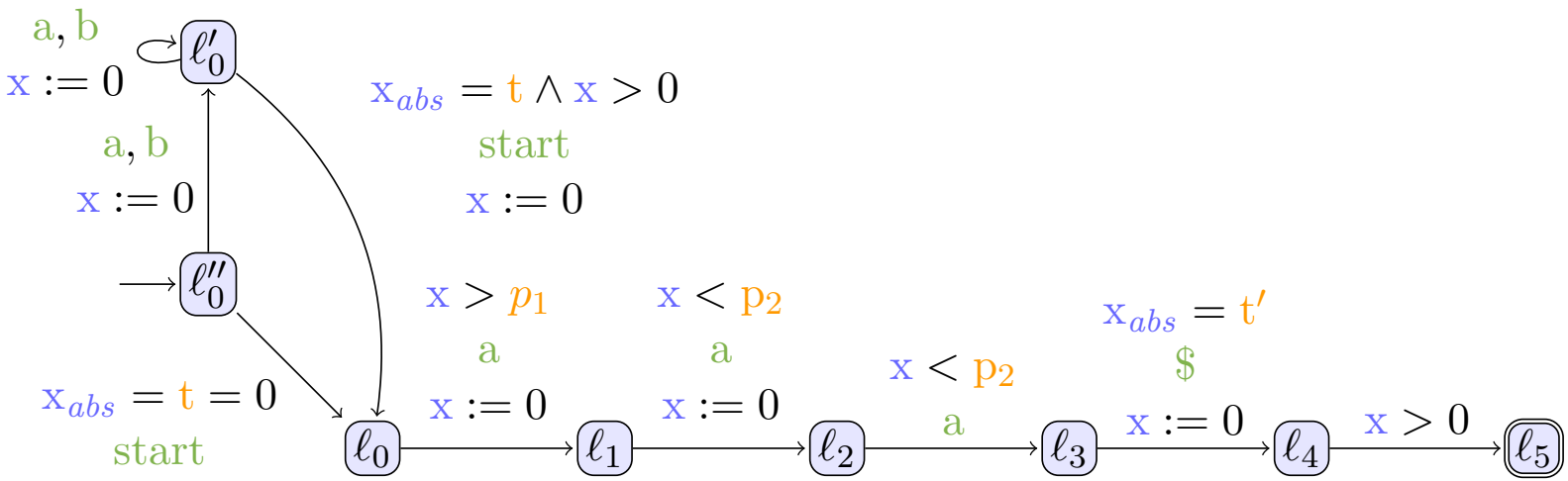


[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Modifying the property pattern

Add some start and end gadgets for completeness of the method [WAH22b]

1. Add an initial transition in $0$-time
   - Captures segments starting from $0$
2. Add a new location with a self-loop
   - Captures segments not starting from the beginning of the word
3. Add a new final transition in $> 0$ time
   - To match the usual definition that the segment must end in $> 0$ time after the last action




[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Transforming a log into a (parametric) timed automaton

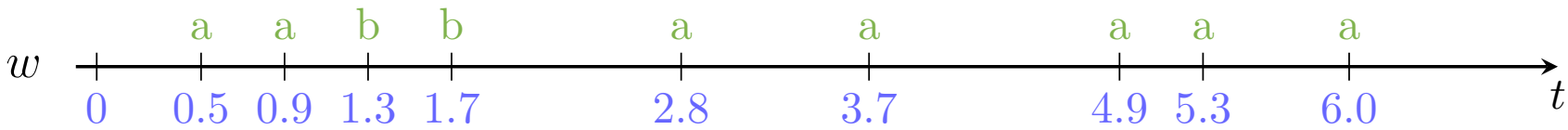
Essentially easy:

1. Add one clock never reset (absolute time)
2. Convert pairs (action, time) into transitions

# Transforming a log into a (parametric) timed automaton

Essentially easy:

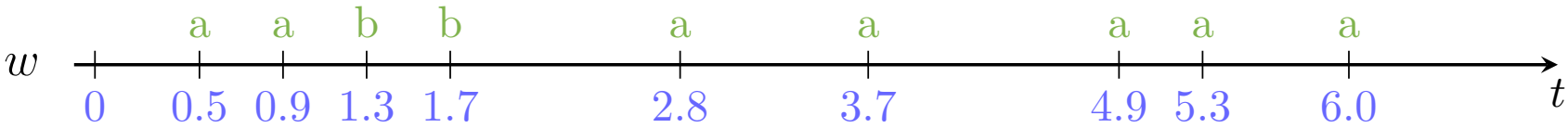
1. Add one clock never reset (absolute time)
2. Convert pairs (action, time) into transitions

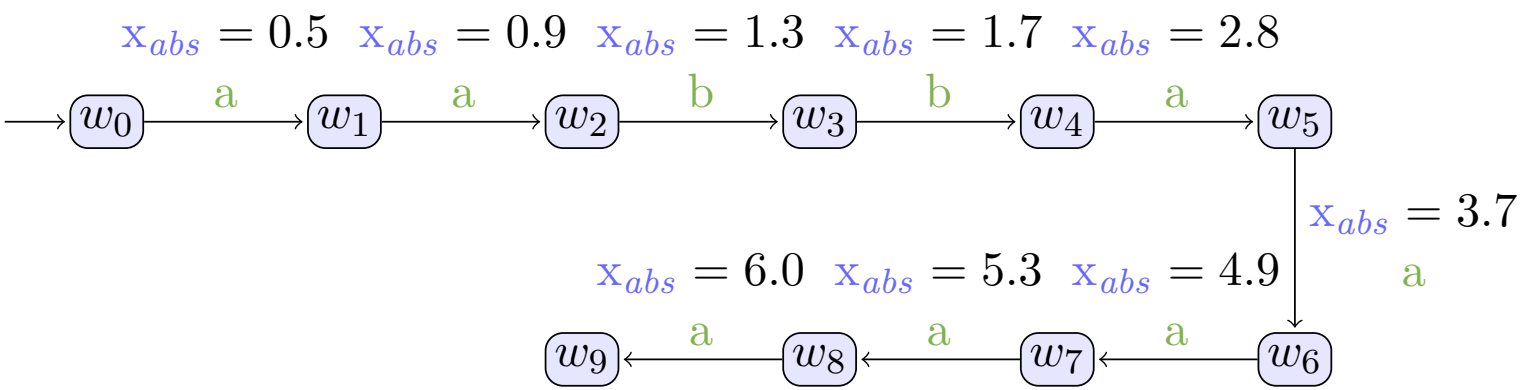


# Transforming a log into a (parametric) timed automaton

Essentially easy:

1. Add one clock never reset (absolute time)
2. Convert pairs (action, time) into transitions



# Product and reachability synthesis

## Result

The set of parameter valuations $\mathrm{t}, \mathrm{t}', p_1, p_2 \ldots$ reaching the final location of the property is exactly the answer to the parametric pattern matching problem

[And19] Étienne André. "What's decidable about parametric timed automata?" In: *International Journal on Software Tools for Technology Transfer* 21.2 (Apr. 2019), pp. 203–219

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Product and reachability synthesis

## Result

The set of parameter valuations $\mathrm{t}, \mathrm{t}', p_1, p_2 \ldots$ reaching the final location of the property is exactly the answer to the parametric pattern matching problem

## Remark

This problem is decidable… in contrast to most problems using PTAs!

[And19]

See formal result in paper [WAH22b]

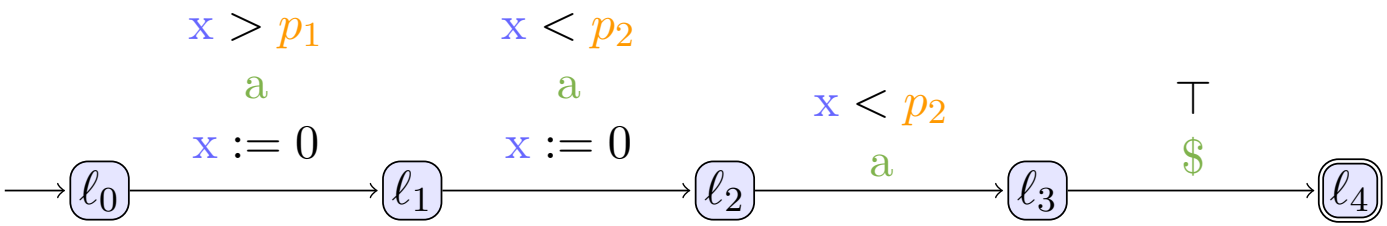
---


[And19] Étienne André. "What's decidable about parametric timed automata?" In: *International Journal on Software Tools for Technology Transfer* 21.2 (Apr. 2019), pp. 203–219

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Product and reachability synthesis: example

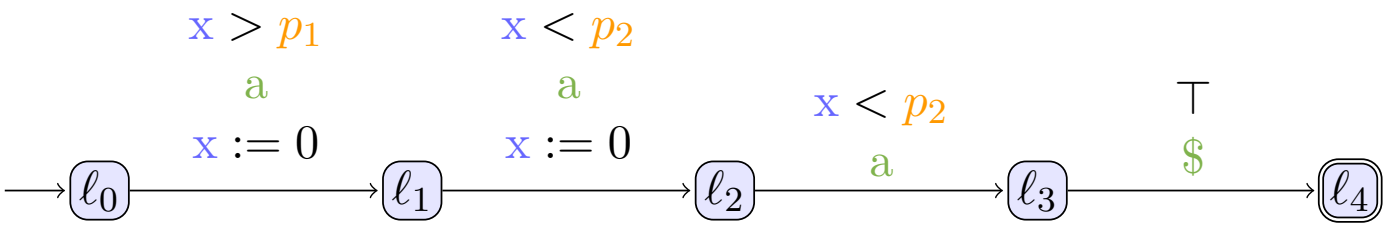
Our property:



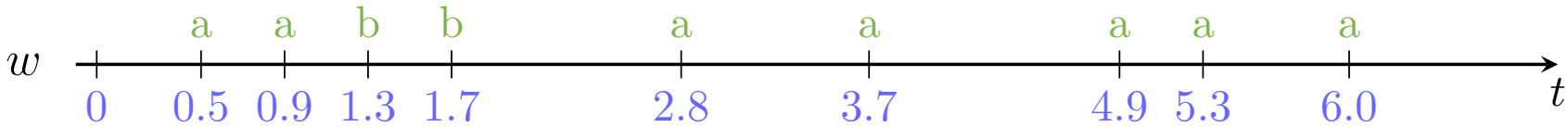

Our log:

# Product and reachability synthesis: example

Our property:




Our log:

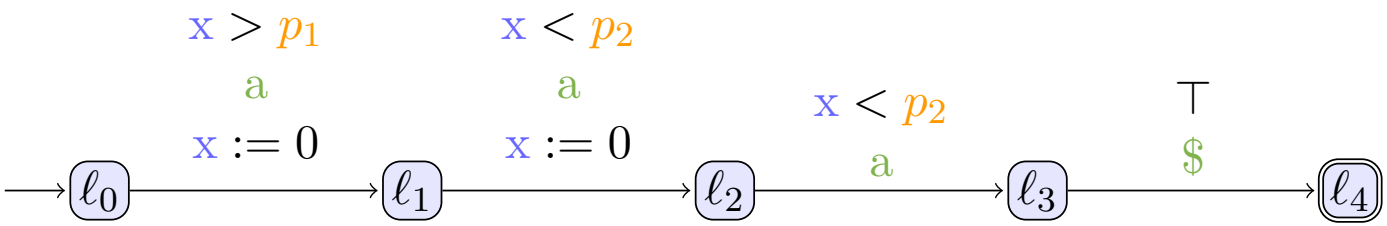



Set of matching intervals:

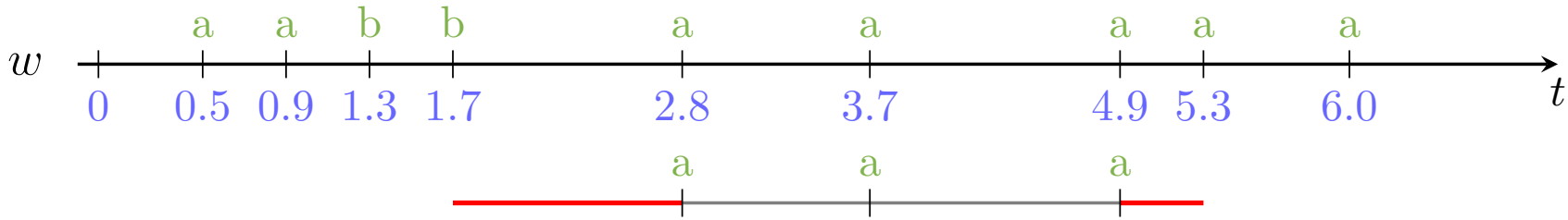
$$
\begin{aligned}
& 1.7 < \mathrm{t} < 2.8 - \mathrm{p}_1 \wedge 4.9 \leq \mathrm{t}' < 5.3 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 2.8 < \mathrm{t} < 3.7 - \mathrm{p}_1 \wedge 5.3 \leq \mathrm{t}' < 6 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 3.7 < \mathrm{t} < 4.9 - \mathrm{p}_1 \wedge \mathrm{t}' \geq 6 \wedge \mathrm{p}_2 > 0.7
\end{aligned}
$$

# Product and reachability synthesis: example

Our property:




Our log:

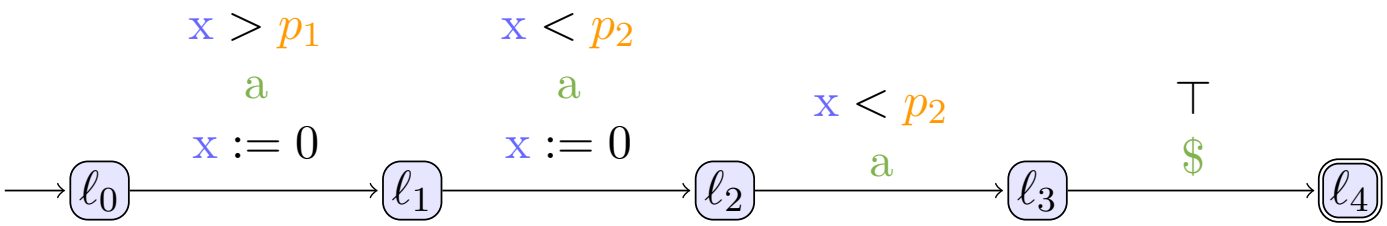



Set of matching intervals:

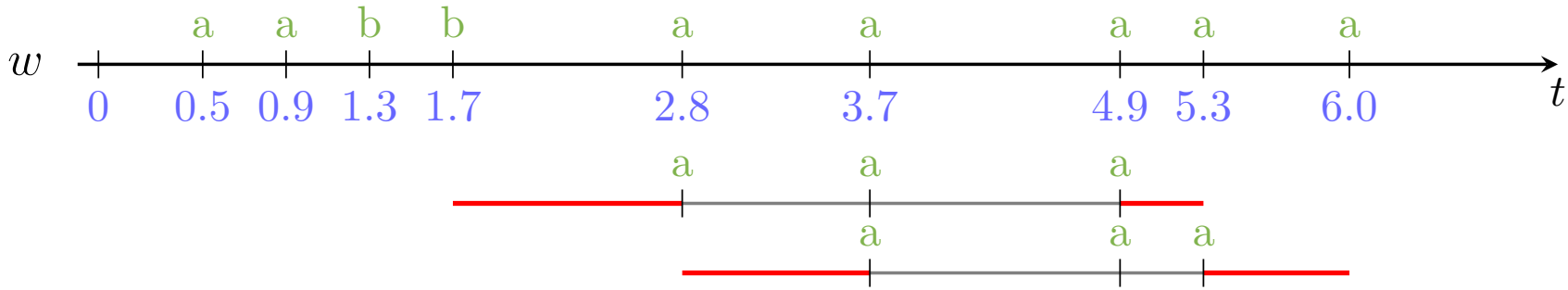
$$
\begin{aligned}
& 1.7 < \mathrm{t} < 2.8 - \mathrm{p}_1 \wedge 4.9 \leq \mathrm{t}' < 5.3 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 2.8 < \mathrm{t} < 3.7 - \mathrm{p}_1 \wedge 5.3 \leq \mathrm{t}' < 6 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 3.7 < \mathrm{t} < 4.9 - \mathrm{p}_1 \wedge \mathrm{t}' \geq 6 \wedge \mathrm{p}_2 > 0.7
\end{aligned}
$$

# Product and reachability synthesis: example

Our property:




Our log:

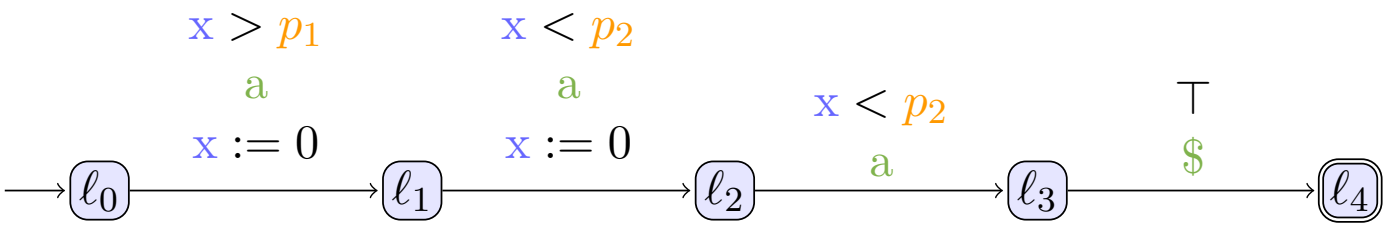



Set of matching intervals:

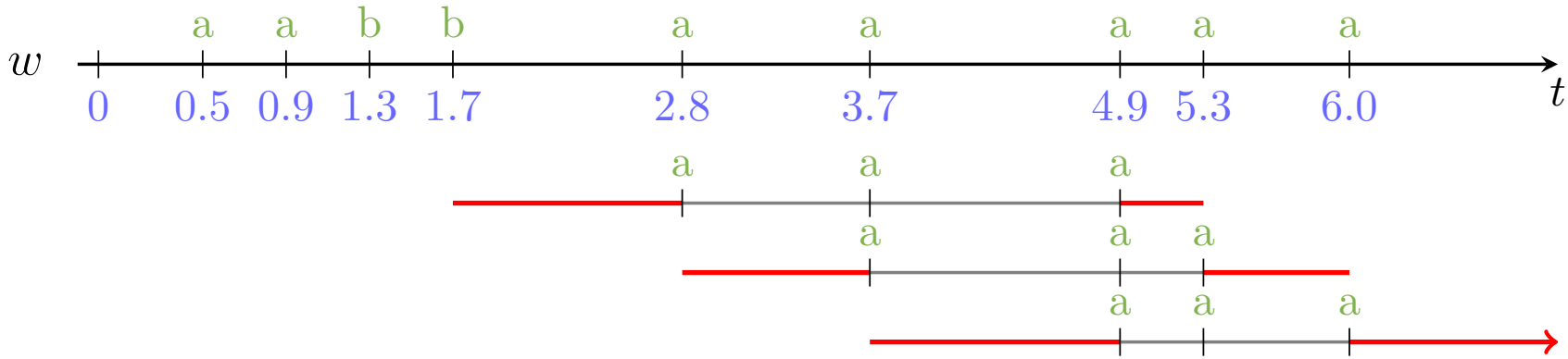
$$
\begin{aligned}
& 1.7 < \mathrm{t} < 2.8 - \mathrm{p}_1 \wedge 4.9 \leq \mathrm{t}' < 5.3 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 2.8 < \mathrm{t} < 3.7 - \mathrm{p}_1 \wedge 5.3 \leq \mathrm{t}' < 6 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 3.7 < \mathrm{t} < 4.9 - \mathrm{p}_1 \wedge \mathrm{t}' \geq 6 \wedge \mathrm{p}_2 > 0.7
\end{aligned}
$$

# Product and reachability synthesis: example

Our property:




Our log:



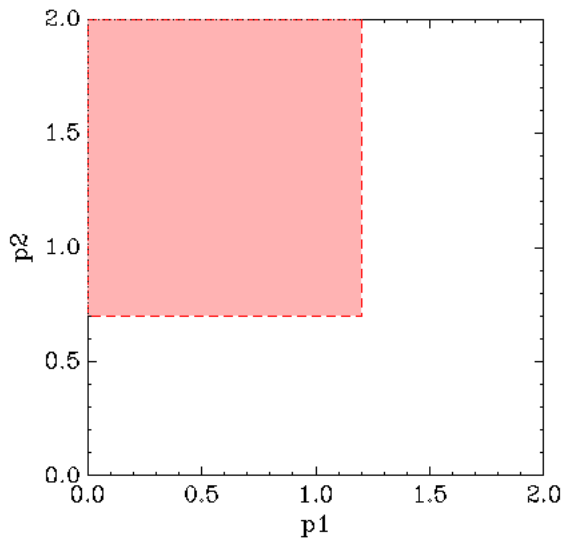

Set of matching intervals:

$$
\begin{aligned}
& 1.7 < \mathrm{t} < 2.8 - \mathrm{p}_1 \wedge 4.9 \leq \mathrm{t}' < 5.3 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 2.8 < \mathrm{t} < 3.7 - \mathrm{p}_1 \wedge 5.3 \leq \mathrm{t}' < 6 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 3.7 < \mathrm{t} < 4.9 - \mathrm{p}_1 \wedge \mathrm{t}' \geq 6 \wedge \mathrm{p}_2 > 0.7
\end{aligned}
$$

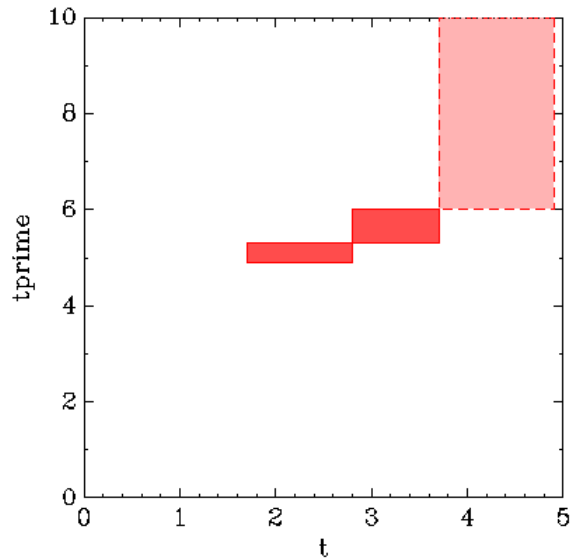
# Exemple: graphical representation

$$
\begin{aligned}
& 1.7 < \mathrm{t} < 2.8 - \mathrm{p}_1 \wedge 4.9 \leq \mathrm{t}' < 5.3 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 2.8 < \mathrm{t} < 3.7 - \mathrm{p}_1 \wedge 5.3 \leq \mathrm{t}' < 6 \wedge \mathrm{p}_2 > 1.2 \\
\vee\ & 3.7 < \mathrm{t} < 4.9 - \mathrm{p}_1 \wedge \mathrm{t}' \geq 6 \wedge \mathrm{p}_2 > 0.7
\end{aligned}
$$

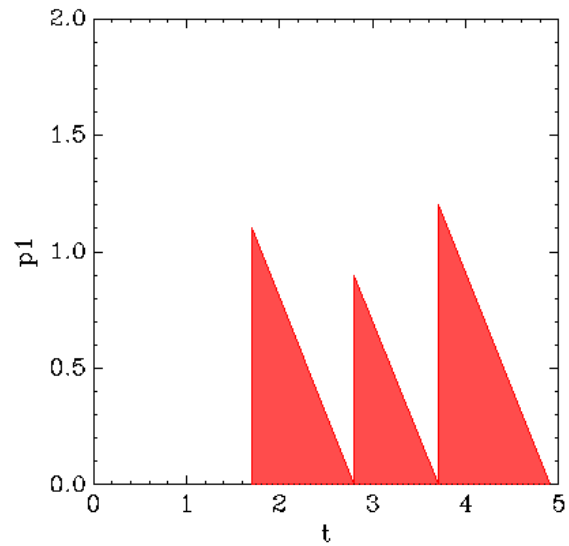
Projections in 2 dimensions:




On $p_1$ and $p_2$




On $\mathrm{t}$ and $\mathrm{t}'$

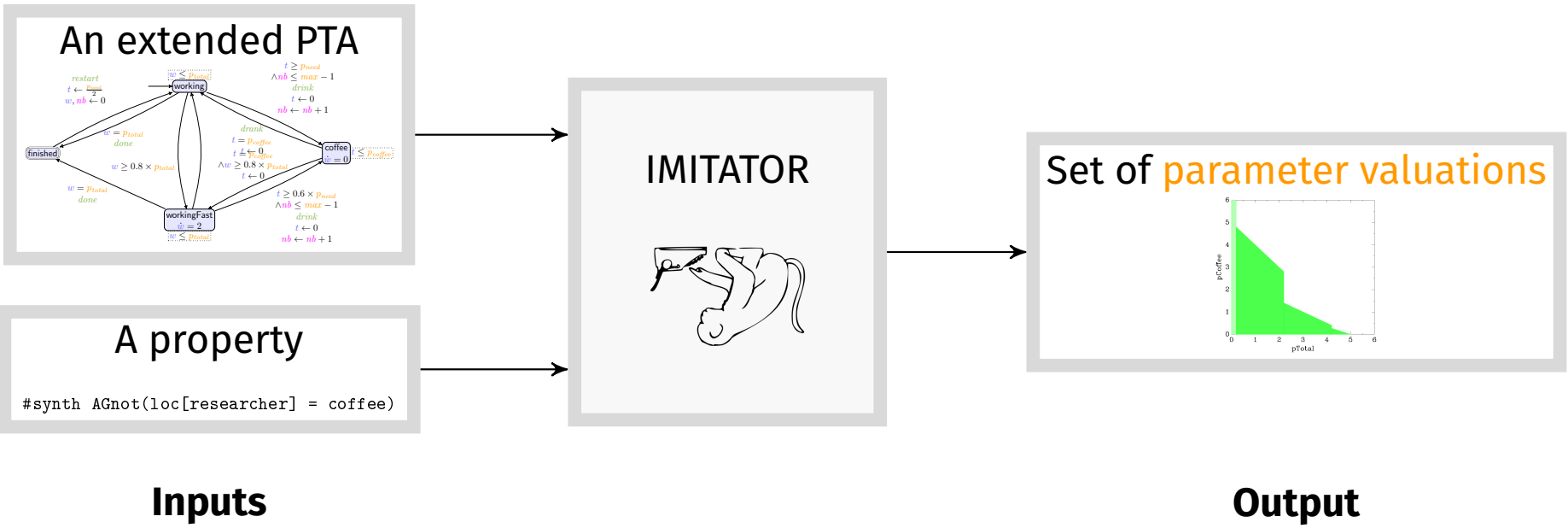



On $\mathrm{t}$ and $p_1$

# Outline

# Parameter synthesis using IMITATOR

IMITATOR: a parametric timed model checker

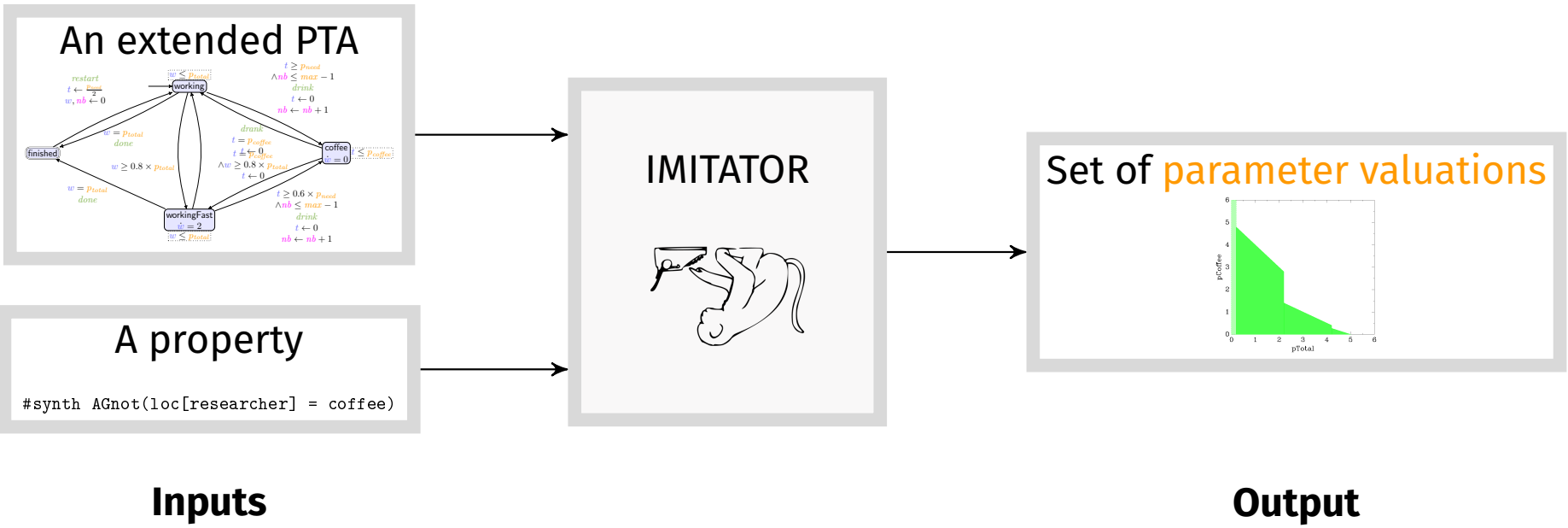



The set of parameter valuations is symbolic

- Symbolic: finite set of linear constraints (polyhedra)

# Parameter synthesis using IMITATOR

IMITATOR: a parametric timed model checker




The set of parameter valuations is symbolic

- Symbolic: finite set of linear constraints (polyhedra)
- Two categories of properties
  - Synthesis: “(try to) synthesize all valuations for which the property holds”
  - Exhibition: “(try to) synthesize at least one valuation for which the property holds”

# Distribution

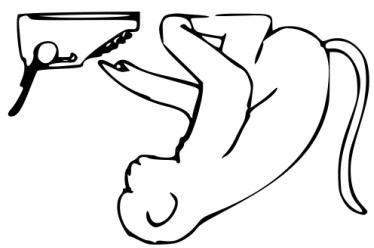
Free and open source software: Available under the GNU-GPL license

Distribution:

- Binaries available for Linux platforms (no dependency, no install)
- Docker version
- Integrated as a virtual machine
- Comes with a user manual and an extensive benchmarks library [AMP21]

doi.org/10.5281/zenodo.4723415

Try it!



www.imitator.fr

[AMP21] Étienne André, Dylan Marinho, and Jaco van de Pol. "A Benchmarks Library for Extended Timed Automata". In: *TAP*. vol. 12740. LNCS. Springer, 2021, pp. 39–50

# Some success stories

- Verification of an asynchronous memory circuit by ST-Microelectronics
- Parametric schedulability analyses for flight control systems for ASTRIUM Space Transportation / ArianeGroup [Fri+12]
- Verification of software product lines [Lut+17]
- Formal timing analysis of music scores [FJ13]
- Solution to a challenge related to a distributed video processing system by Thales
- Parametric timed pattern matching and online monitoring [WAH22b]

---


[Fri+12] Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. "Robustness Analysis for Scheduling Problems using the Inverse Method". In: *TIME*. IEEE Computer Society Press, Sept. 2012, pp. 73–80

[Lut+17] Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. "Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints". In: *SPLC, Volume A*. ACM, 2017, pp. 104–113

[FJ13] Léa Fanchon and Florent Jacquemard. "Formal Timing Analysis Of Mixed Music Scores". In: *ICMC*. Michigan Publishing, Aug. 2013

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Outline

# Experimental environment

Toolkit

- Simple Python script to transform timed words into IMITATOR PTAs
- Slightly modified version of IMITATOR
  - To handle PTAs with dozens of thousands of locations
  - To manage $n$-parameter constraints with dozens of thousands of disjuncts

Two algorithms:

- PTPM: parametric timed pattern matching
- $\text{PTPM}_{\text{opt}}$: parametric timed pattern matching with parameter optimization
  - e. g., "where in the log is the smallest value of the parameter $p$ s.t. the property is satisfied/violated?"

Sources, binaries, models, logs can be found at `imitator.fr/static/ICECCS18`

# Case study 1: GEAR (description)

Monitoring the gear change of an automatic transmission system

- Obtained by simulation of the Simulink model of an automatic transmission system [HAF14]
- S-TaLiRo [Ann+11] used to generate an input to this model (generates a gear change signal that is fed to the model)
- Gear chosen from $\{g_1, g_2, g_3, g_4\}$
- Generated gear change recorded in a timed word

## Property

"If the gear is changed to 1, it should not be changed to 2 within $p$ seconds."

This condition is related to the requirement $\phi_5^{AT}$ proposed in [HAF14] (the nominal value for $p$ in [HAF14] is 2).

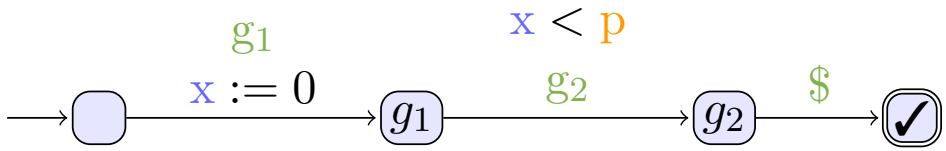
---


[HAF14] Bardh Hoxha, Houssam Abbas, and Georgios E. Fainekos. "Benchmarks for Temporal Logic Requirements for Automotive Systems". In: *ARCH@CPSWeek*. Vol. 34. EPiC Series in Computing. EasyChair, 2014, pp. 25–30

[Ann+11] Yashwanth Annpureddy, Che Liu, Georgios E. Fainekos, and Sriram Sankaranarayanan. "S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems". In: *TACAS*. vol. 6605. LNCS. Springer, 2011, pp. 254–257

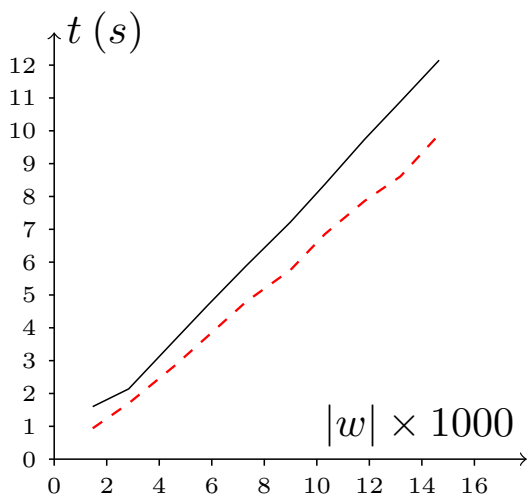
# Case study 1: GEAR (experiments)

Property: "If the gear is changed to 1, it should not be changed to 2 within $p$ seconds."




Experiments data:

| Model | | PTPM | | | | $\text{PTPM}_{\text{opt}}$ | |
|---|---|---|---|---|---|---|---|
| Length | Time frame | States | Matches | Parsing (s) | Comp. (s) | States | Comp. (s) |
| 1,467 | 1,000 | 4,453 | 379 | 0.02 | 1.60 | 3,322 | 0.94 |
| 2,837 | 2,000 | 8,633 | 739 | 0.33 | 2.14 | 6,422 | 1.70 |
| 4,595 | 3,000 | 14,181 | 1,247 | 0.77 | 3.63 | 10,448 | 2.85 |
| 5,839 | 4,000 | 17,865 | 1,546 | 1.23 | 4.68 | 13,233 | 3.74 |
| 7,301 | 5,000 | 22,501 | 1,974 | 1.94 | 5.88 | 16,585 | 4.79 |
| 8,995 | 6,000 | 27,609 | 2,404 | 2.96 | 7.28 | 20,413 | 5.76 |
| 10,316 | 7,000 | 31,753 | 2,780 | 4.00 | 8.38 | 23,419 | 6.86 |
| 11,831 | 8,000 | 36,301 | 3,159 | 5.39 | 9.75 | 26,832 | 7.87 |
| 13,183 | 9,000 | 40,025 | 3,414 | 6.86 | 10.89 | 29,791 | 8.61 |
| 14,657 | 10,000 | 44,581 | 3,816 | 8.70 | 12.15 | 33,141 | 9.89 |




$\text{PTPM}_{\text{opt}}$: alternative procedure to find the minimum/maximum value of a parameter along the log

# Case study 2: ACCEL (description)

Monitoring the acceleration of an automated transmission system

- Also obtained by simulation from the Simulink model of [HAF14]
- (discretized) value of three state variables recorded in the log:
  - engine RPM (discretized to "high" and "low" with a certain threshold)
  - velocity (discretized to "high" and "low" with a certain threshold)
  - 4 gear positions

## Property

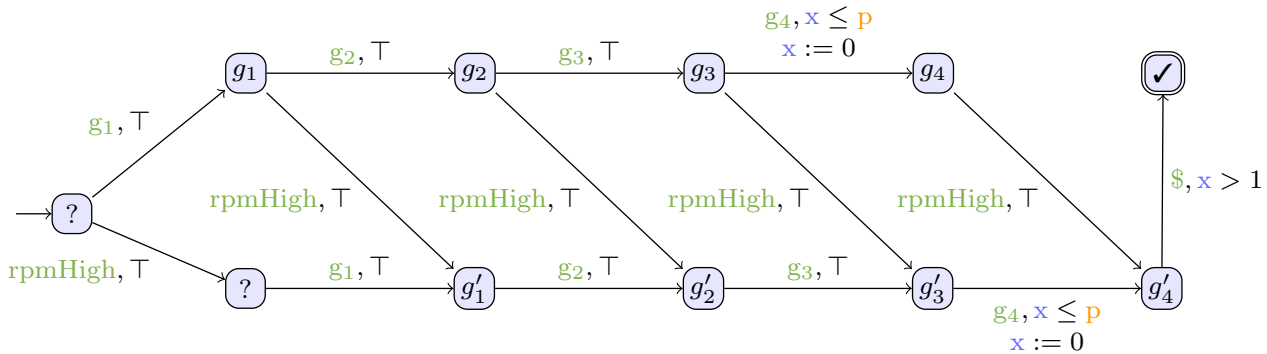
"If a gear changes from 1 to 2, 3, and 4 in this order in $p$ seconds and engine RPM becomes large during this gear change, then the velocity of the car must be sufficiently large in one second."

This condition models the requirement $\phi_8^{AT}$ proposed in [HAF14] (the nominal value for $p$ in [HAF14] is 10).

[HAF14] Bardh Hoxha, Houssam Abbas, and Georgios E. Fainekos. "Benchmarks for Temporal Logic Requirements for Automotive Systems". In: *ARCH@CPSWeek*. Vol. 34. EPiC Series in Computing. EasyChair, 2014, pp. 25–30

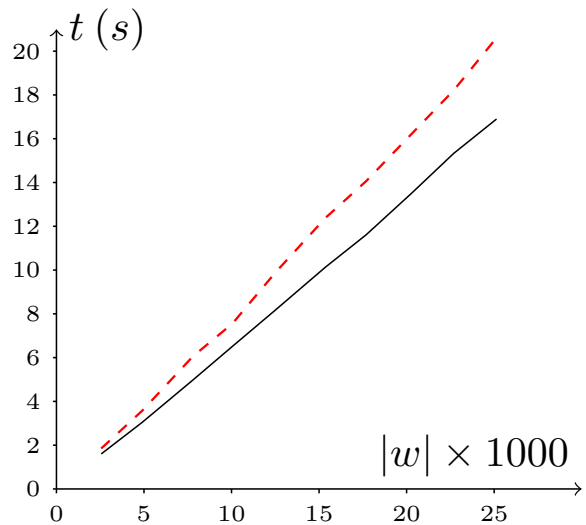
# Case study 2: ACCEL (experiments)

Property: "If a gear changes from 1 to 2, 3, and 4 in this order in $p$ seconds and engine RPM becomes large during this gear change, then the velocity of the car must be sufficiently large in one second."



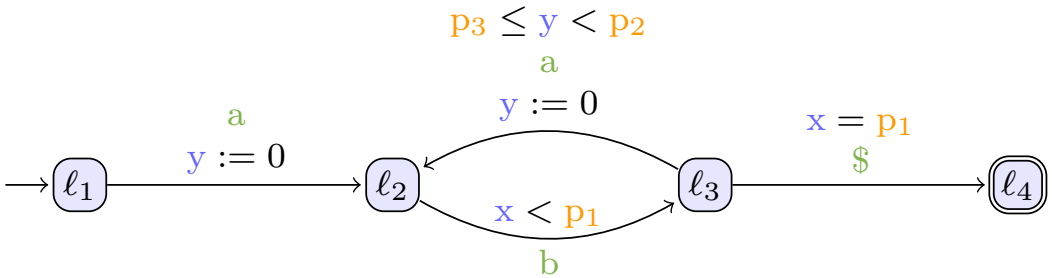

Experiments data:

| Model | | PTPM | | | | $\text{PTPM}_{\text{opt}}$ | |
|---|---|---|---|---|---|---|---|
| Length | Time frame | States | Matches | Parsing (s) | Comp. (s) | States | Comp. (s) |
| 2,559 | 1,000 | 6,504 | 2 | 0.27 | 1.60 | 6,502 | 1.85 |
| 4,894 | 2,000 | 12,429 | 2 | 0.86 | 3.04 | 12,426 | 3.57 |
| 7,799 | 3,000 | 19,922 | 7 | 2.21 | 4.98 | 19,908 | 6.06 |
| 10,045 | 4,000 | 25,520 | 3 | 3.74 | 6.51 | 25,514 | 7.55 |
| 12,531 | 5,000 | 31,951 | 9 | 6.01 | 8.19 | 31,926 | 9.91 |
| 15,375 | 6,000 | 39,152 | 7 | 9.68 | 10.14 | 39,129 | 12.39 |
| 17,688 | 7,000 | 45,065 | 9 | 13.40 | 11.61 | 45,039 | 14.06 |
| 20,299 | 8,000 | 51,660 | 10 | 18.45 | 13.52 | 51,629 | 16.23 |
| 22,691 | 9,000 | 57,534 | 11 | 24.33 | 15.33 | 57,506 | 18.21 |
| 25,137 | 10,000 | 63,773 | 13 | 31.35 | 16.90 | 63,739 | 20.61 |



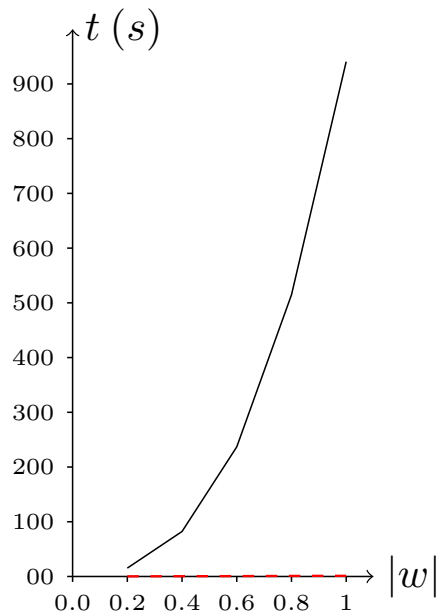
# Case study 3: Blowup

Property made on purpose to test our scalability




Experiments data:

| Model | | PTPM | | | | $PTPM_{opt}$ | |
|---|---|---|---|---|---|---|---|
| Length | Time frame | States | Matches | Parsing (s) | Comp. (s) | States | Comp. (s) |
| 200 | 101 | 20,602 | 5,050 | 0.01 | 15.31 | 515 | 0.24 |
| 400 | 202 | 81,202 | 20,100 | 0.02 | 82.19 | 1,015 | 0.49 |
| 600 | 301 | 181,802 | 45,150 | 0.03 | 236.80 | 1,515 | 0.71 |
| 800 | 405 | 322,402 | 80,200 | 0.05 | 514.57 | 2,015 | 1.05 |
| 1,000 | 503 | 503,002 | 125,250 | 0.06 | 940.74 | 2,515 | 1.24 |



# Outline

# Summary of part I

- New method to monitor logs of real-time systems with parametric specifications
- Methodology: parametric timed model checking
- Applications: automotive industry
  - Linear in the size of the log
  - Able to handle logs of dozens of thousands of events
    $\Rightarrow$ scalable
- Both online and offline

---

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Summary of part I

- New method to monitor logs of real-time systems with parametric specifications
- Methodology: parametric timed model checking
- Applications: automotive industry
  - Linear in the size of the log
  - Able to handle logs of dozens of thousands of events
    $\Rightarrow$ scalable
- Both online and offline
- Improvements: ad-hoc method enhanced with skipping [WAH22b]

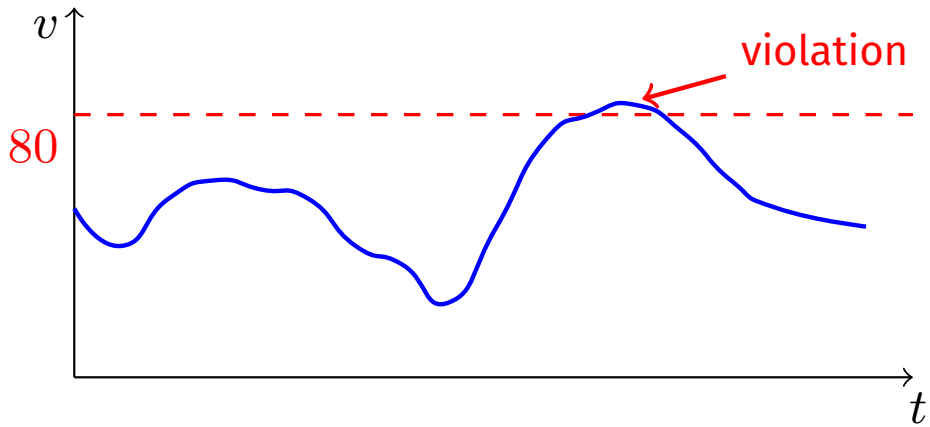
---

[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022)

# Outline

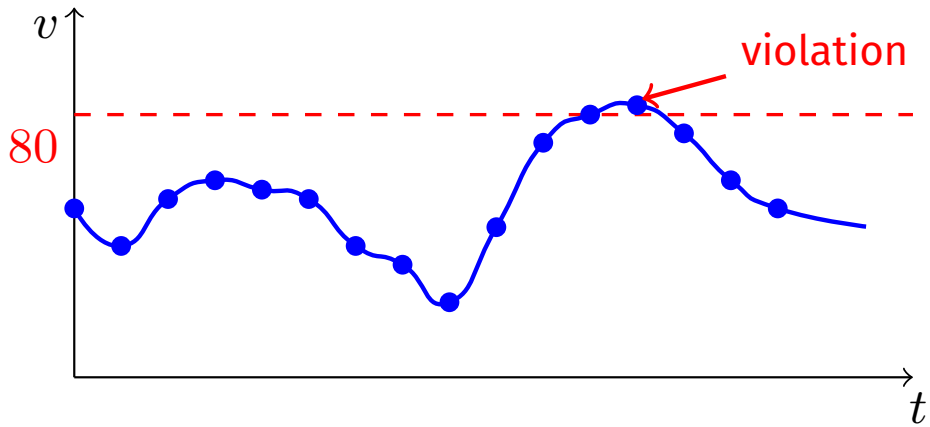
# Monitoring cyber-physical systems

Specification: $\neg(v > 80)$



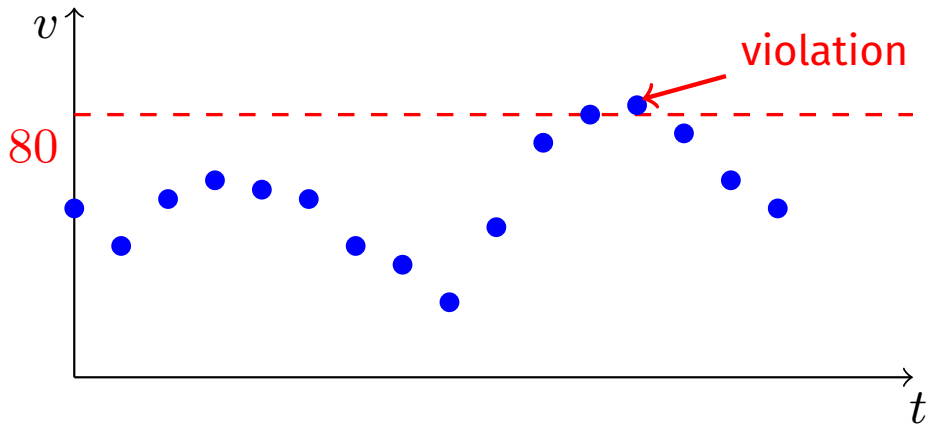
# Monitoring cyber-physical systems with sampling

Specification: $\neg(v > 80)$



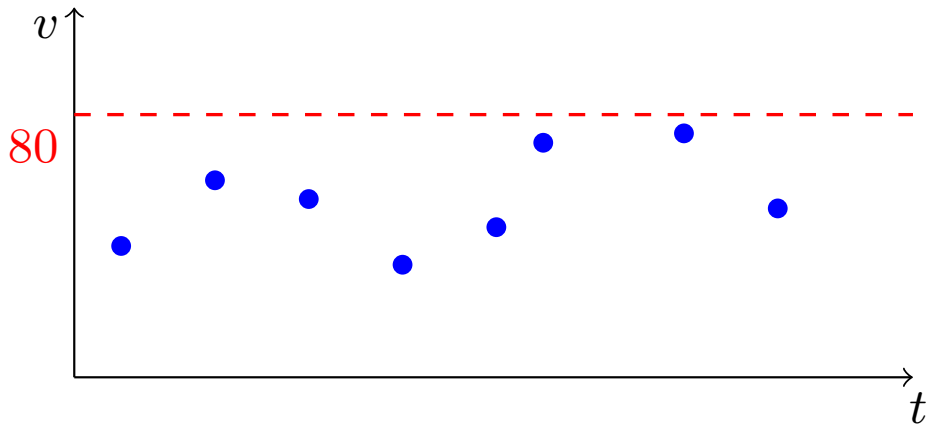
# Monitoring cyber-physical systems with sampling

Specification: $\neg(v > 80)$



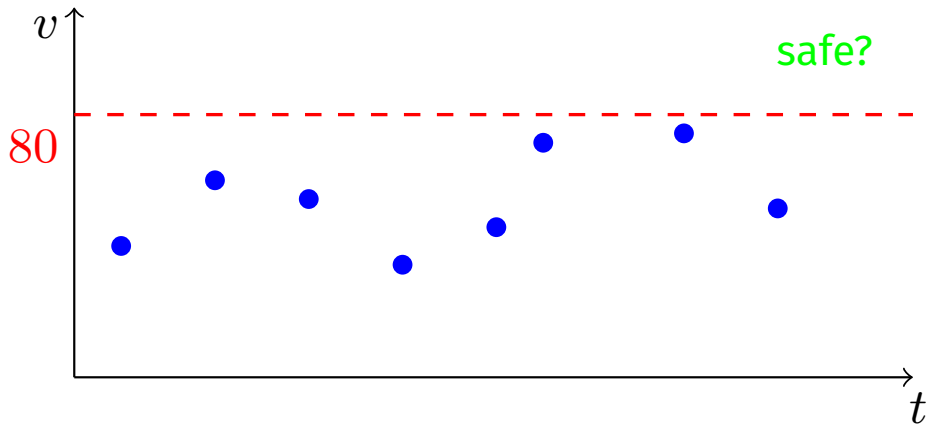
# Monitoring cyber-physical systems with sampling

Specification: $\neg(v > 80)$



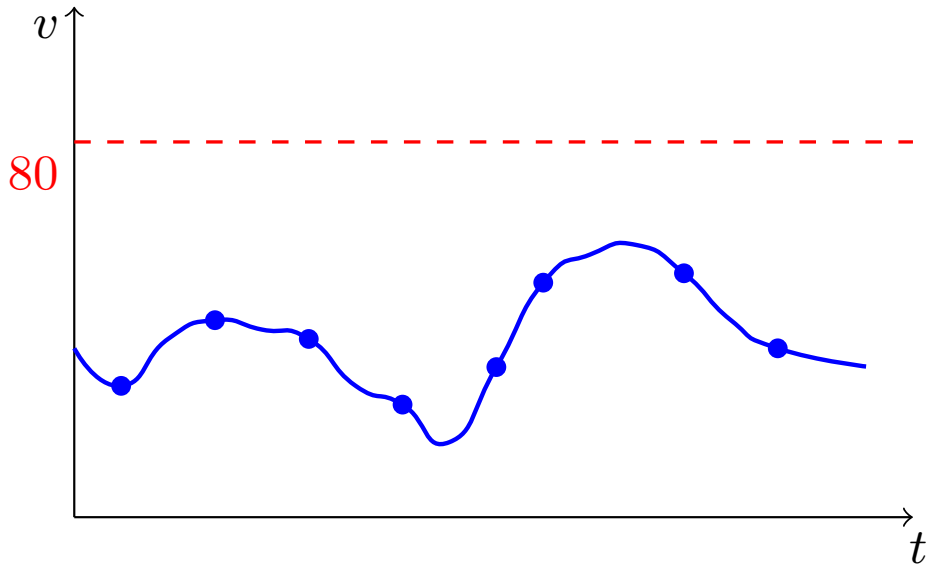
# Monitoring cyber-physical systems with sampling

Specification: $\neg(v > 80)$



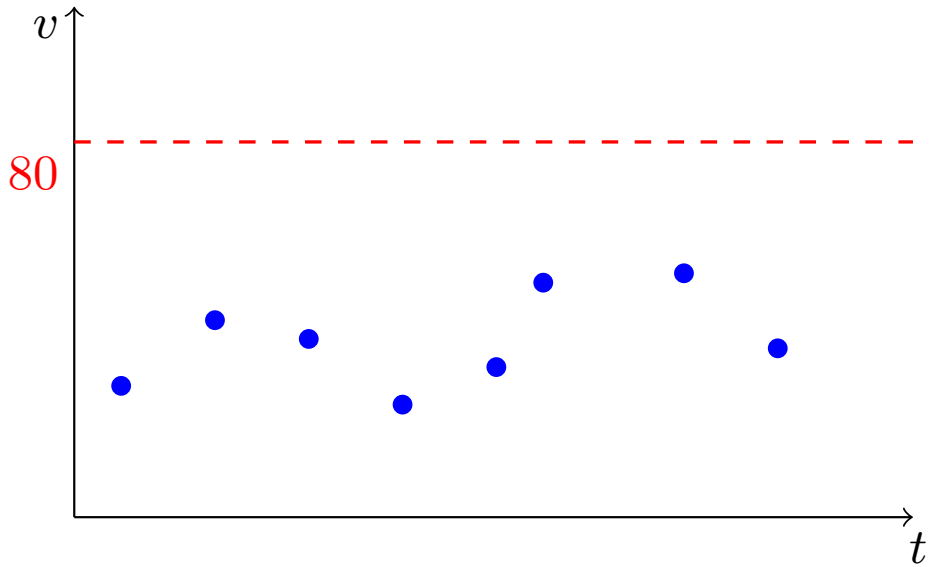
# Signal interpolation

Specification: $\neg(v > 80)$



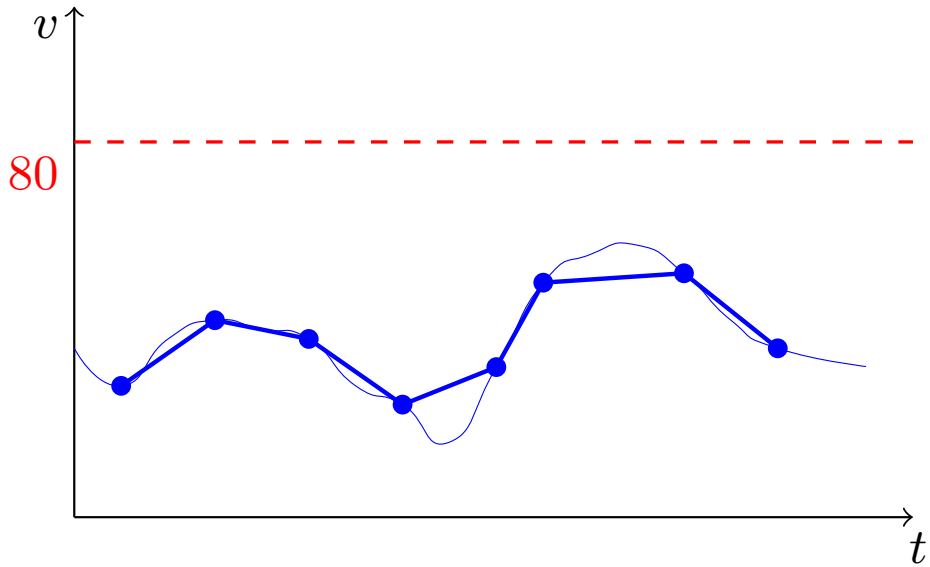
# Signal interpolation

Specification: $\neg(v > 80)$



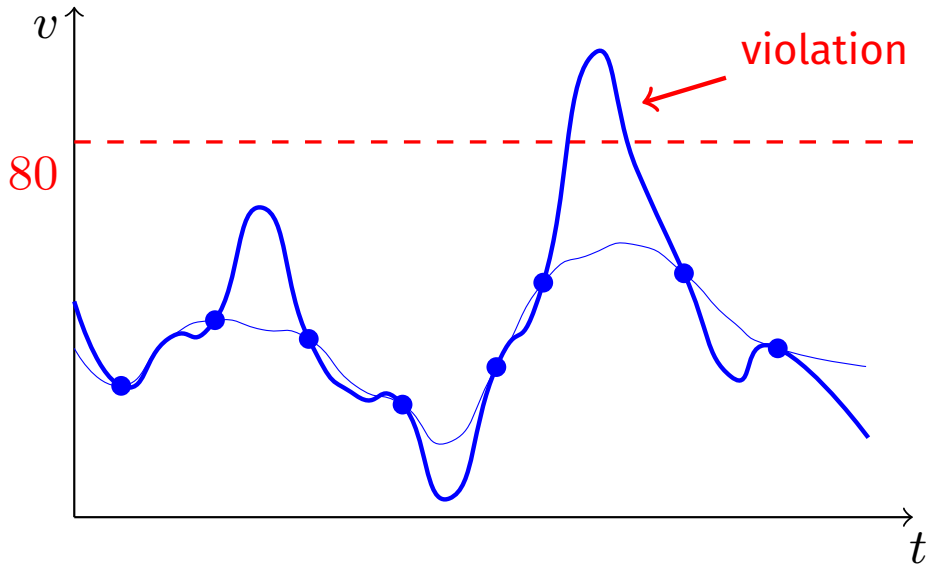
# Signal interpolation

Specification: $\neg(v > 80)$



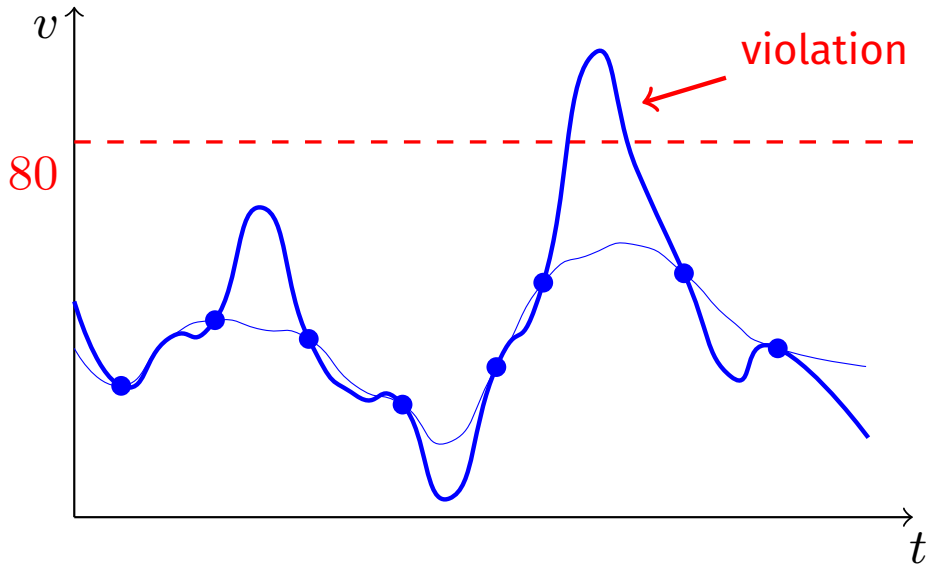
# Signal interpolation

Specification: $\neg(v > 80)$



# Signal interpolation with prior knowledge

Specification: $\neg(v > 80)$




Impossible violation because we know that $\frac{dv}{dt} < K$ (for some known $K$)

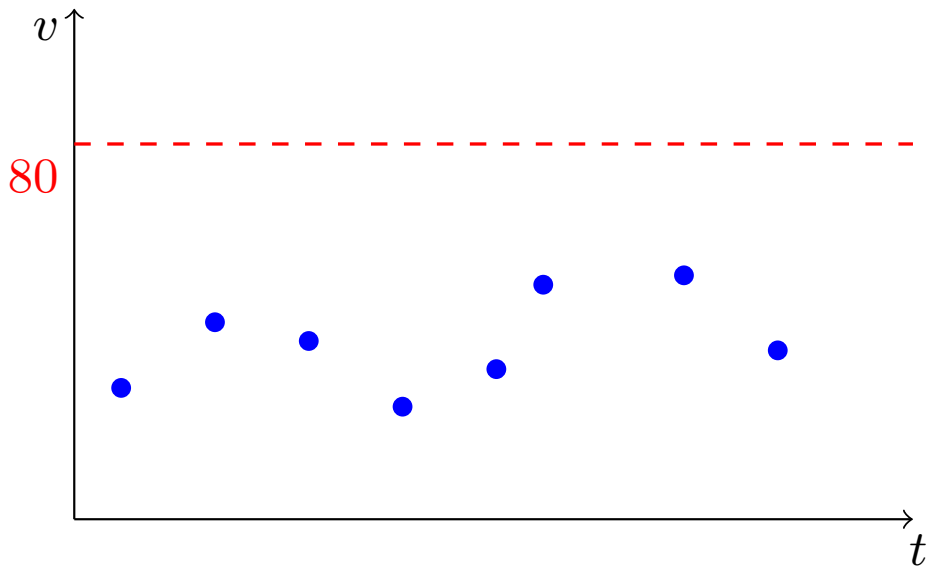
Example: a car cannot accelerate arbitrarily fast

# Outline

# Model-bounded monitoring [WAH22a]

Specification: $\neg(v > 80)$

System log

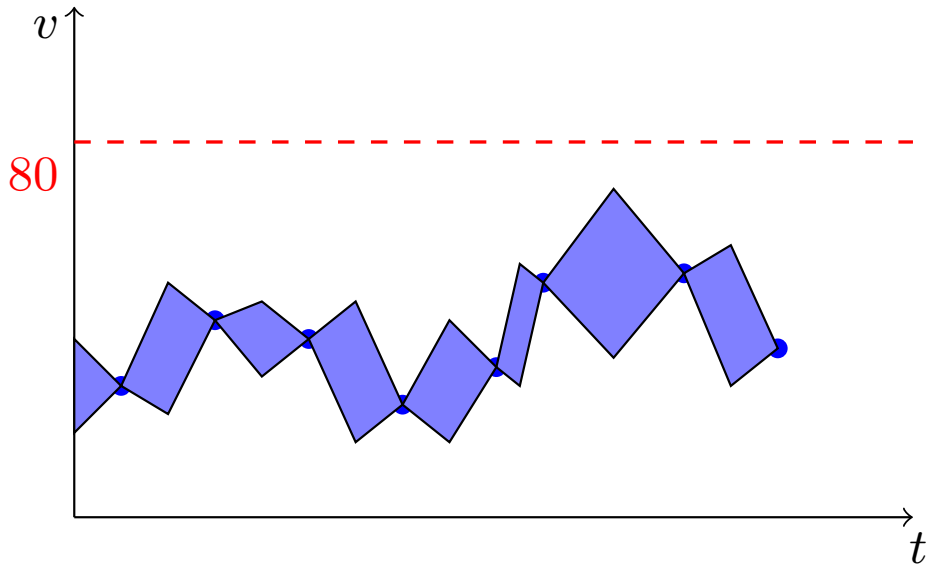



[WAH22a] Masaki Waga, Étienne André, and Ichiro Hasuo. "Model-Bounded Monitoring of Hybrid Systems". In: *ACM Transactions on Cyber-Physical Systems* 6.4 (Nov. 2022), 30:1–30:26

# Model-bounded monitoring [WAH22a]

Specification: $\neg(v > 80)$

System log + knowledge (bounded model) ($\frac{dv}{dt} < K$)

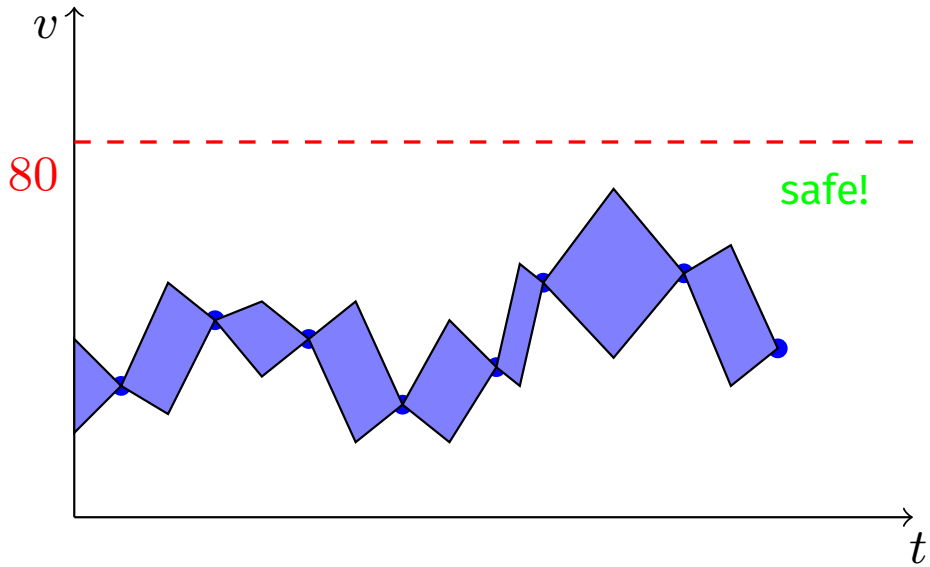



[WAH22a] Masaki Waga, Étienne André, and Ichiro Hasuo. "Model-Bounded Monitoring of Hybrid Systems". In: *ACM Transactions on Cyber-Physical Systems* 6.4 (Nov. 2022), 30:1–30:26

# Model-bounded monitoring [WAH22a]

Specification: $\neg(v > 80)$

System log + knowledge (bounded model) ($\frac{dv}{dt} < K$)




[WAH22a] Masaki Waga, Étienne André, and Ichiro Hasuo. "Model-Bounded Monitoring of Hybrid Systems". In: *ACM Transactions on Cyber-Physical Systems* 6.4 (Nov. 2022), 30:1–30:26

# The bounding model: a linear hybrid automaton

A bounding model should:

- ☺ be expressive
- ☺ yet allow for efficient computation

[Hen96] Thomas A. Henzinger. "The Theory of Hybrid Automata". In: *LiCS*. IEEE Computer Society, 1996, pp. 278–292

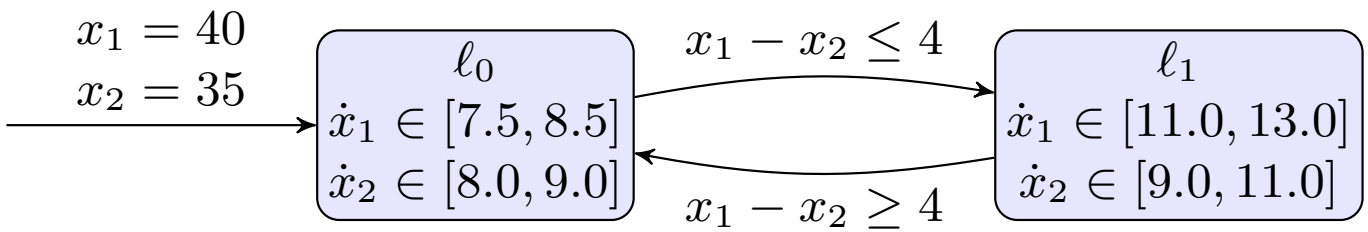
# The bounding model: a linear hybrid automaton

A bounding model should:

- ☺ be expressive
- ☺ yet allow for efficient computation

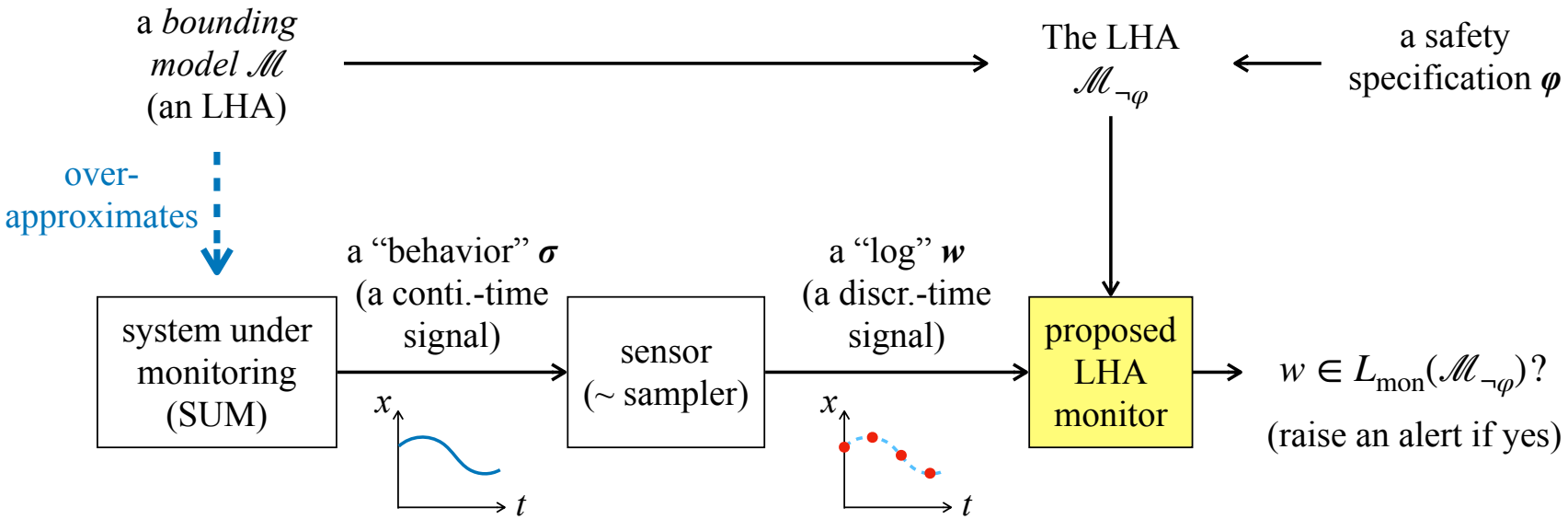
Our formalism: linear hybrid automata [Hen96]

- discrete modes
- invariants, guards, derivatives expressed by polyhedra

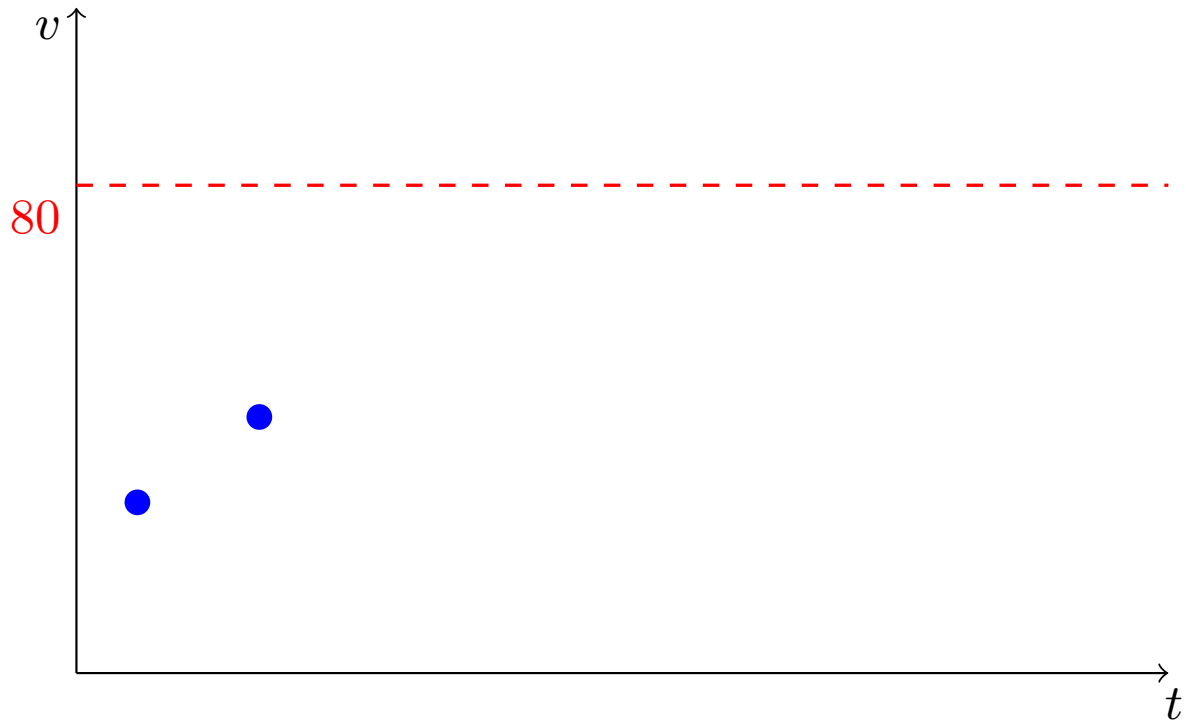



[Hen96] Thomas A. Henzinger. "The Theory of Hybrid Automata". In: *LiCS*. IEEE Computer Society, 1996, pp. 278–292

# Model-bounded monitoring in a nutshell

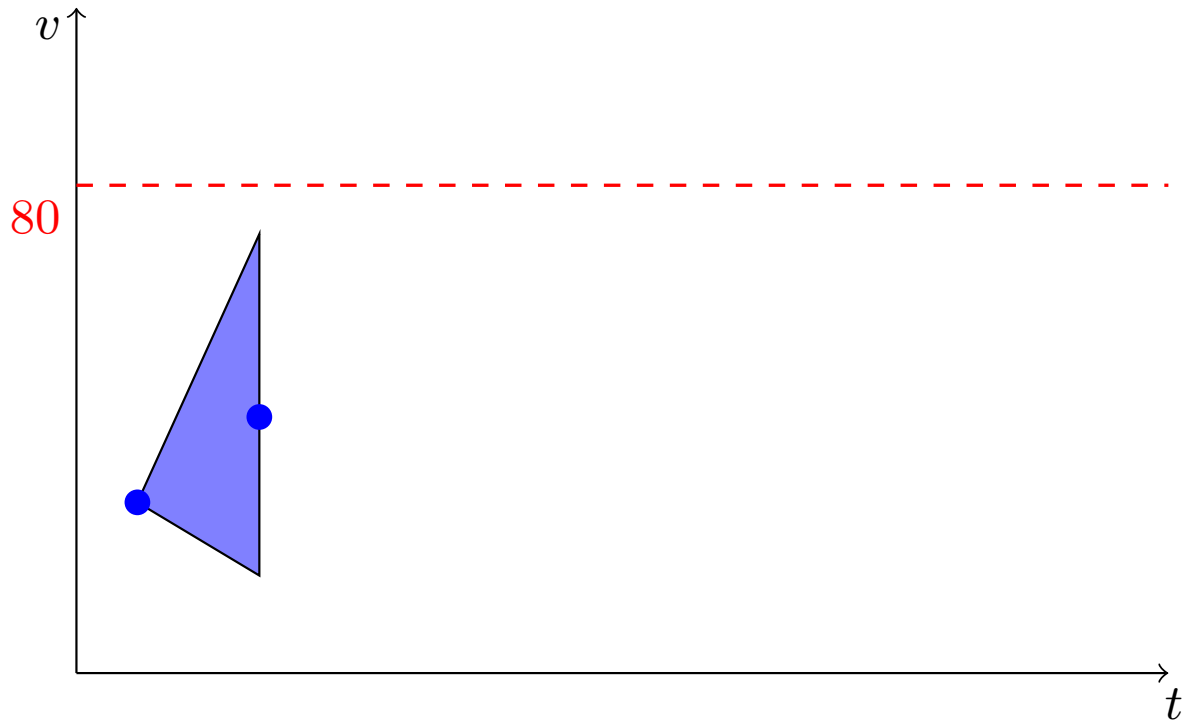


# Algorithm: bounded-time reachability



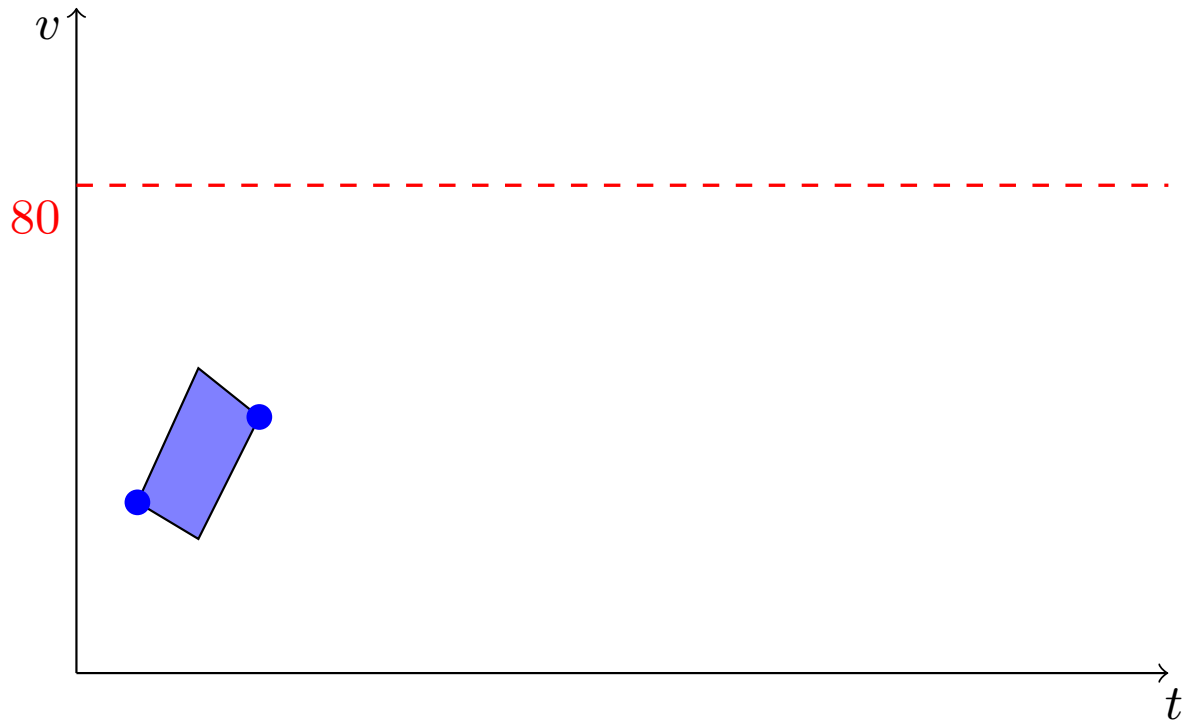

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



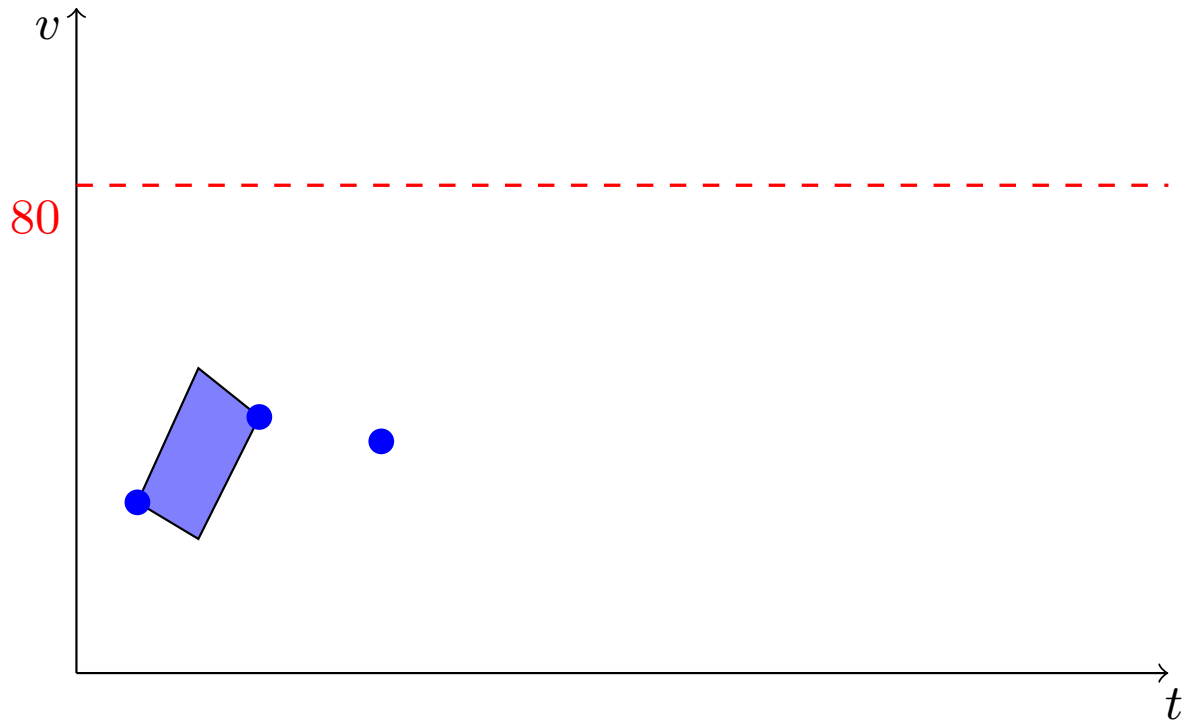

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



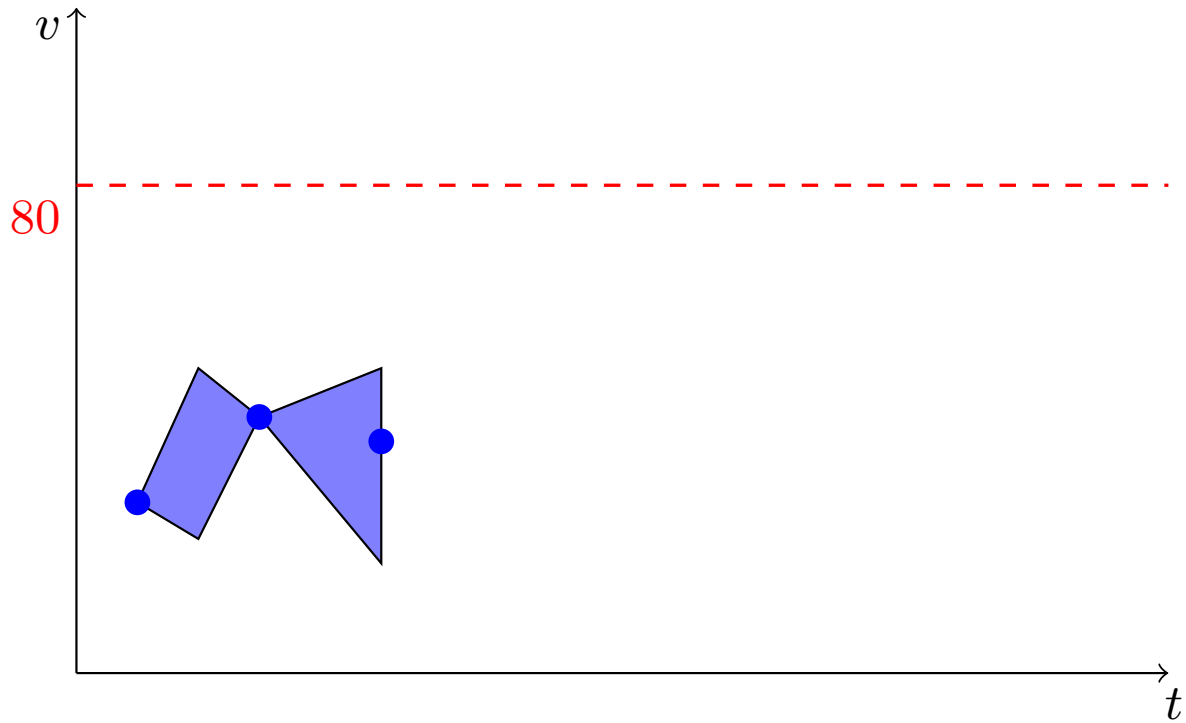

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



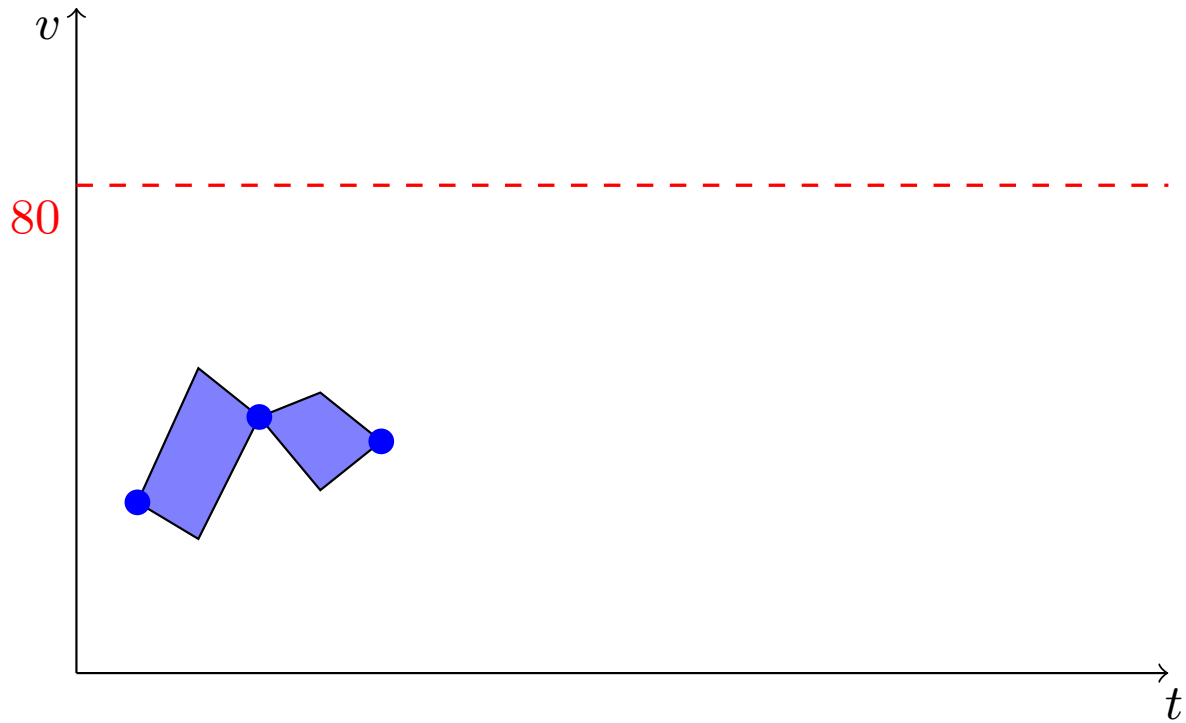

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



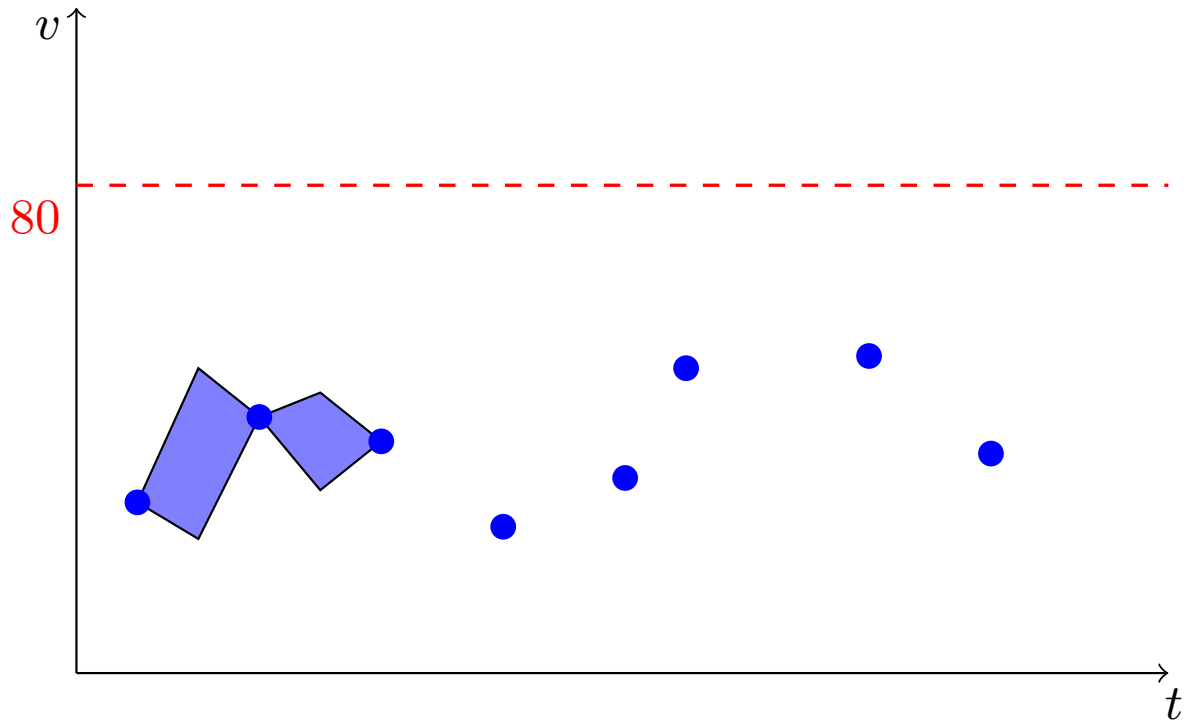

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



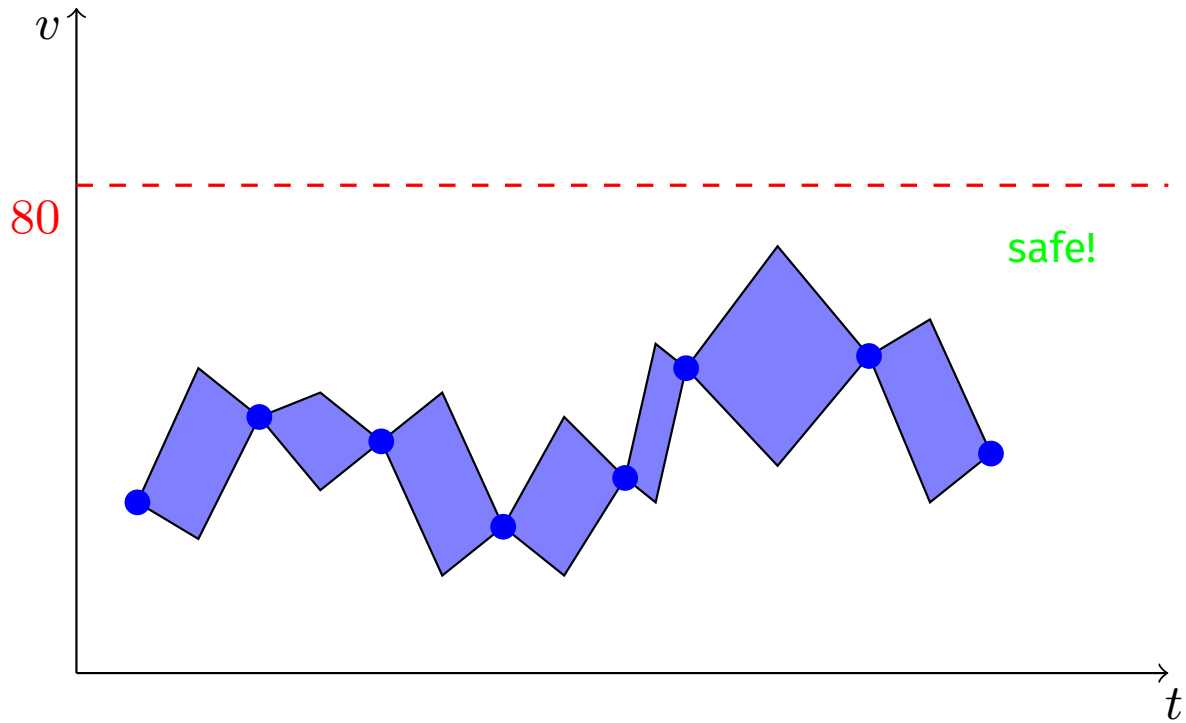

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability



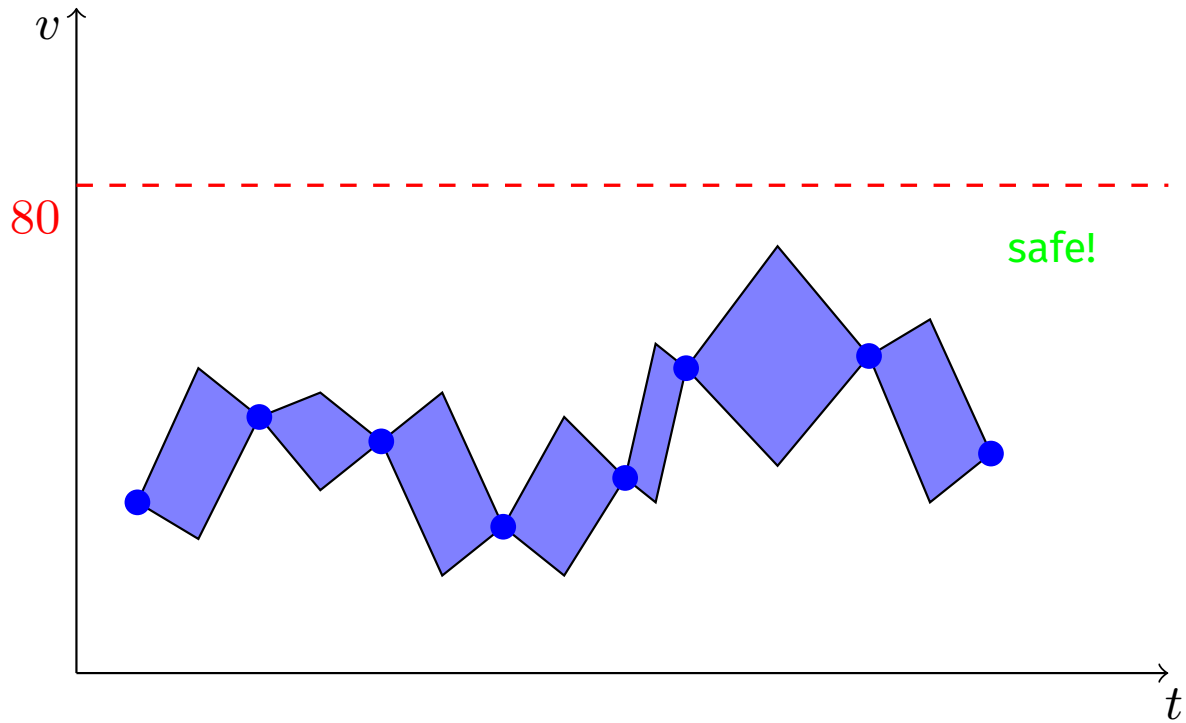

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability




[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability

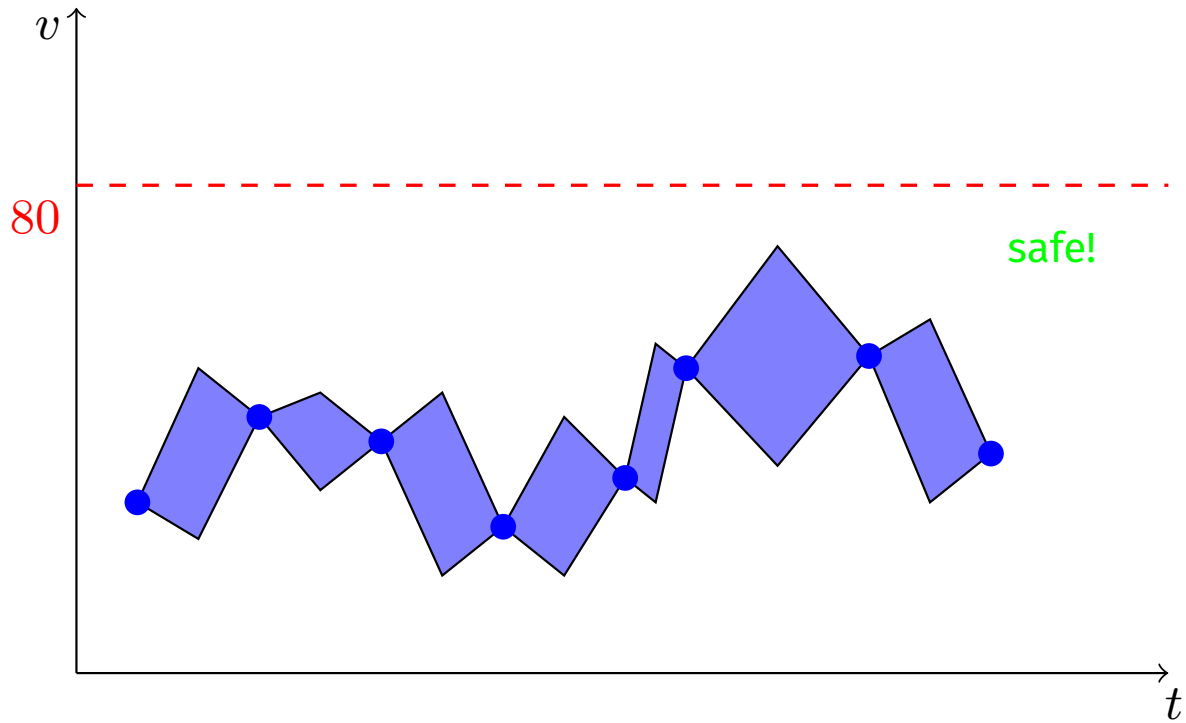



Small detail: undecidable [Bri+11]

---

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Algorithm: bounded-time reachability




Small detail: undecidable [Bri+11] (but do we care?)

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II*. vol. 6756. LNCS. Springer, 2011, pp. 416–427

# Are linear hybrid automata efficient?

☹ Reachability analysis in (linear) hybrid automata is very hard

- Long line of research (e. g., [Bu+19][Bog+20])

---

[Bu+19] Lei Bu, Jiawan Wang, Yuming Wu, and Xuandong Li. "From Bounded Reachability Analysis of Linear Hybrid Automata to Verification of Industrial CPS and IoT". In: *SETSS*. vol. 12154. LNCS. Springer, 2019, pp. 10–43

[Bog+20] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Kostiantyn Potomkin, and Christian Schilling. "Reachability Analysis of Linear Hybrid Systems via Block Decomposition". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 4018–4029

# Are linear hybrid automata efficient?

- ☹ Reachability analysis in (linear) hybrid automata is very hard
  - Long line of research (e. g., [Bu+19][Bog+20])

- ☺ But in our scheme we "reset" the uncertainty at each new sample
  - No error accumulation, no divergence
  - Possible to define sufficient condition for guaranteed termination

---

[Bu+19] Lei Bu, Jiawan Wang, Yuming Wu, and Xuandong Li. "From Bounded Reachability Analysis of Linear Hybrid Automata to Verification of Industrial CPS and IoT". In: *SETSS*. vol. 12154. LNCS. Springer, 2019, pp. 10–43

[Bog+20] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Kostiantyn Potomkin, and Christian Schilling. "Reachability Analysis of Linear Hybrid Systems via Block Decomposition". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 4018–4029

# Outline

# Two implementations

Approach 1: existing model-checker PHAVerLite [BZ19]

- Light fork of PHAVerLite [Fre08]
- ☺ Highly optimized reachability analysis

Approach 2: *ad hoc* dedicated monitor HAMoNi (by Masaki Waga)

- ☺ Best performance in theory

---

[BZ19] Anna Becchi and Enea Zaffanella. "Revisiting Polyhedral Analysis for Hybrid Systems". In: *SAS*. vol. 11822. LNCS. Springer, 2019, pp. 183–202

[Fre08] Goran Frehse. "PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech". In: *International Journal on Software Tools for Technology Transfer* 10.3 (May 2008), pp. 263–279. ISSN: 1433-2779

# Benchmarks

Three main benchmarks:

1. Adaptive cruise controller [BRS19]
2. Robot navigation benchmark [FI04]
3. Shared Gas-Burner [DHR05]

---

[BRS19] Lei Bu, Rajarshi Ray, and Stefan Schupp. "ARCH-COMP19 Category Report: Bounded Model Checking of Hybrid Systems with Piecewise Constant Dynamics". In: *ARCH@CPSIoTWeek*. Vol. 61. EPiC Series in Computing. EasyChair, 2019, pp. 120–128

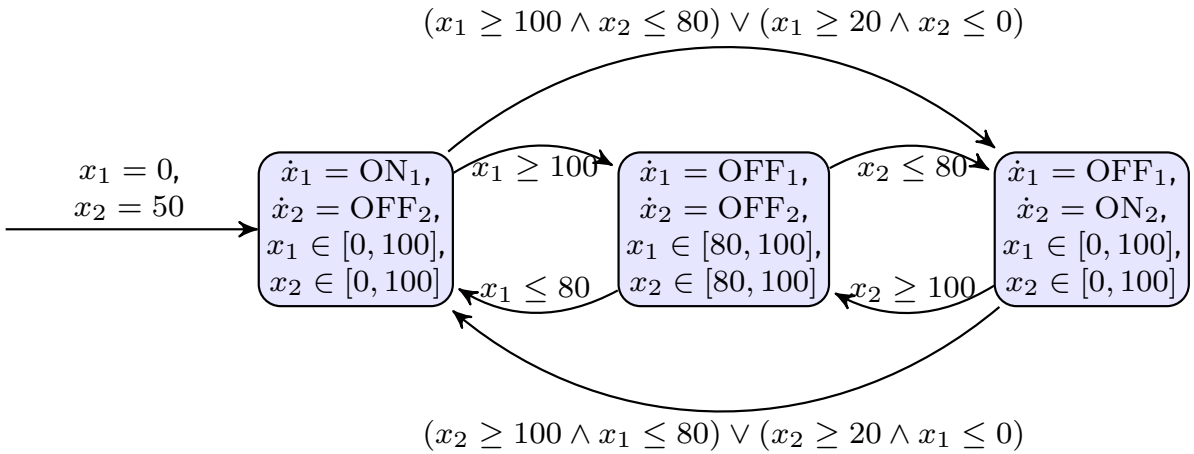
[FI04] Ansgar Fehnker and Franjo Ivancic. "Benchmarks for Hybrid Systems Verification". In: *HSCC*. vol. 2993. LNCS. Springer, 2004, pp. 326–341

[DHR05] Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. "Automatic Rectangular Refinement of Affine Hybrid Systems". In: *FORMATS*. vol. 3829. LNCS. Springer, 2005, pp. 144–161

# Benchmarks: bounding models

Bounding models: mostly taken from the literature

- Example: GasBurner:

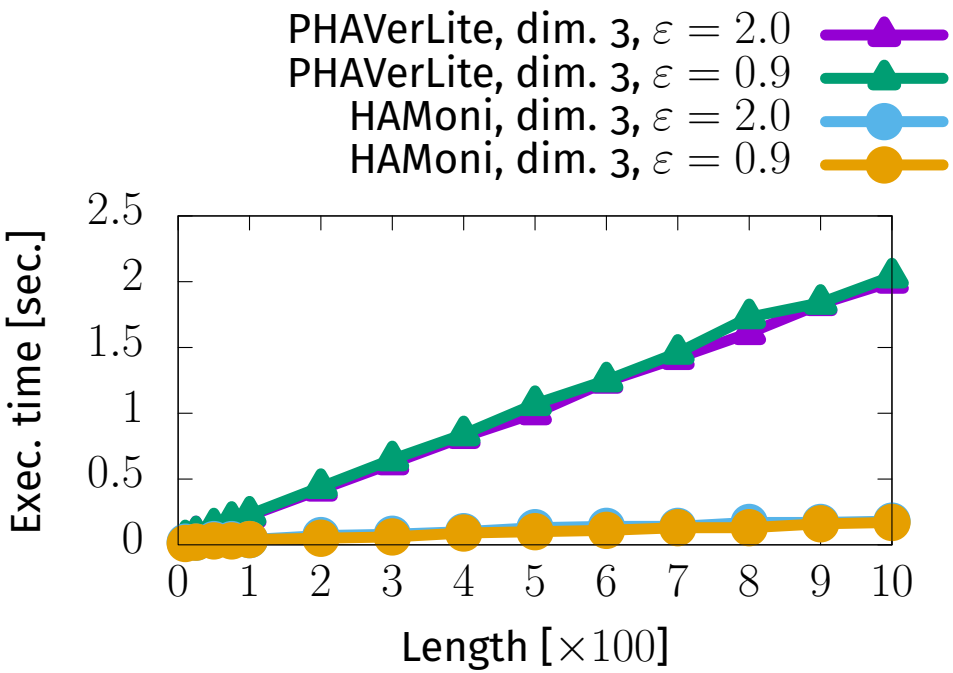



The affine hybrid automaton for the original model in GasBurner, where $h = 2, a = 0.01, b = 0.005, \mathrm{ON}_1 = h - ax_1 + bx_2$, $\mathrm{ON}_2 = h - ax_2 + bx_1, \mathrm{OFF}_1 = -ax_1 + bx_2$, and $\mathrm{OFF}_2 = -ax_2 + bx_1$.

Logs: randomly generated from the bounding models (this coincidence is not mandatory)

- Length of the logs: 150 to 100,000

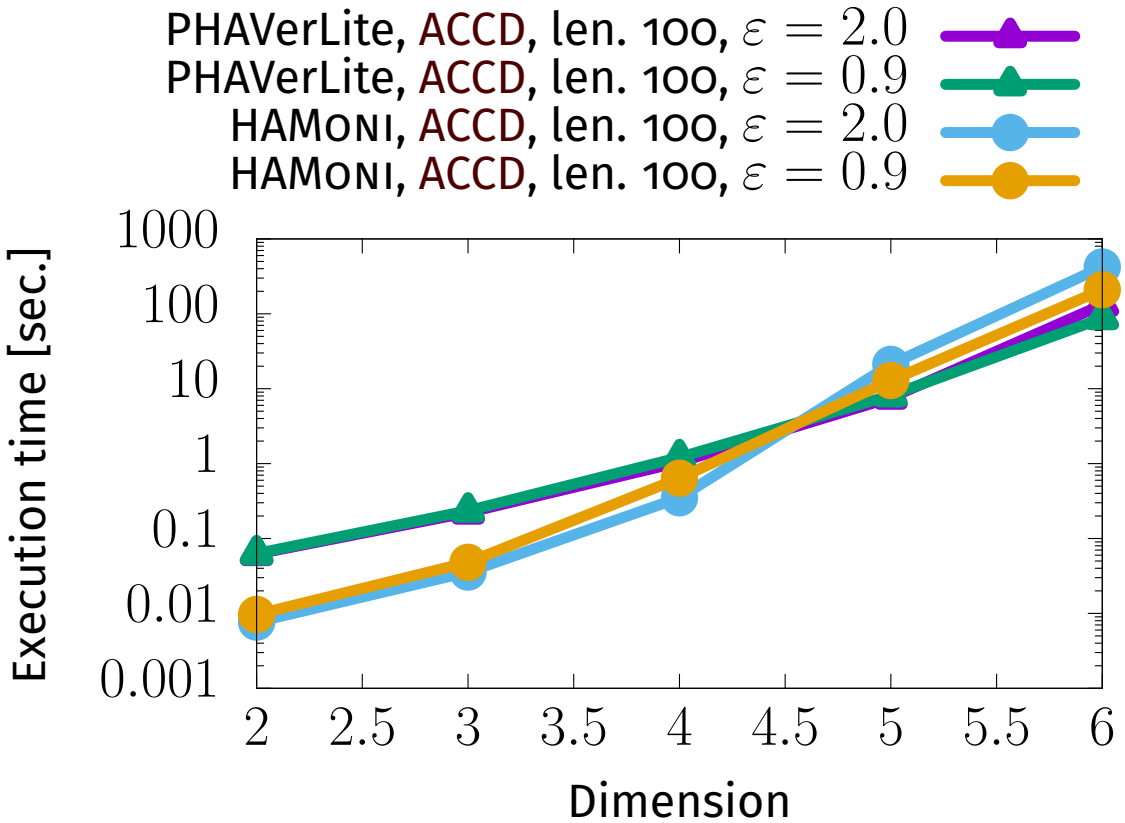
# Changing observation length




Conclusions:

- linear
- very scalable: $> 5,000$ samples / second
- Our tool HAMoni is about 10 times faster than the existing PHAVerLite

# Changing model dimension




Conclusions:

- Existing model checker PHAVerLite faster for dimensions $> 6$
- Future work: further optimization of our tool HAMoNi

# Outline

# Summary of part II

## Model-bounded monitoring

- Bounding model (prior knowledge): linear hybrid automaton

## Algorithms and implementations

- Idea: bounded-time reachability
- Crux: no error accumulation due to new sampling
- Experiments: effectively monitorable

# Outline

# General perspectives

- Monitoring beyond safety
  - Monitoring against temporal properties
- Uncertainties in the observation
  - partial logs
  - values known with uncertainties
- Uncertainty in the specification
  - Timing parameters [WAH19]
- Extension to uncertain non-linear systems [GA22]
- Quantitative monitoring
  - "By how much is the specification violated?" [Sel+22]

---


[WAH19] Masaki Waga, Étienne André, and Ichiro Hasuo. "Symbolic Monitoring against Specifications Parametric in Time and Data". In: *CAV, Part I*. vol. 11561. LNCS. Springer, 2019, pp. 520–539

[GA22] Bineet Ghosh and Étienne André. "Monitoring of scattered uncertain logs using uncertain linear dynamical systems". In: *FORTE*. vol. 13273. LNCS. Springer, 2022, pp. 67–87

[Sel+22] Daniel Selvaratnam, Michael Cantoni, J. M. Davoren, and Iman Shames. "MITL Verification Under Timing Uncertainty". In: *FORMATS*. vol. 13465. LNCS. Springer, 2022, pp. 136–152

# Bibliography

# References I


[AD94] Rajeev Alur and David L. Dill. “A theory of timed automata”. In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. DOI: 10.1016/0304-3975(94)90010-8.

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. “Parametric real-time reasoning”. In: *STOC* (May 16–18, 1993). Ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. San Diego, California, United States: ACM, 1993, pp. 592–601. DOI: 10.1145/167088.167242.

[AMP21] Étienne André, Dylan Marinho, and Jaco van de Pol. “A Benchmarks Library for Extended Timed Automata”. In: *TAP* (June 21–25, 2021). Ed. by Frédéric Loulergue and Franz Wotawa. Vol. 12740. LNCS. virtual: Springer, 2021, pp. 39–50. DOI: 10.1007/978-3-030-79379-1_3.

[And19] Étienne André. “What’s decidable about parametric timed automata?” In: *International Journal on Software Tools for Technology Transfer* 21.2 (Apr. 2019), pp. 203–219. DOI: 10.1007/s10009-017-0467-0.

[And21] Étienne André. “IMITATOR 3: Synthesis of timing parameters beyond decidability”. In: *CAV* (July 18–23, 2021). Ed. by Rustan Leino and Alexandra Silva. Vol. 12759. LNCS. virtual: Springer, 2021, pp. 1–14. DOI: 10.1007/978-3-030-81685-8_26.

[Ann+11] Yashwanth Annpureddy, Che Liu, Georgios E. Fainekos, and Sriram Sankaranarayanan. “S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems”. In: *TACAS* (Mar. 26–Apr. 3, 2011). Ed. by Parosh Aziz Abdulla and K. Rustan M. Leino. Vol. 6605. LNCS. Saarbrücken, Germany: Springer, 2011, pp. 254–257. DOI: 10.1007/978-3-642-19835-9_21.

# References II


[Bog+20] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Kostiantyn Potomkin, and Christian Schilling. "Reachability Analysis of Linear Hybrid Systems via Block Decomposition". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 4018–4029. DOI: 10.1109/TCAD.2020.3012859.

[Bri+11] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. "On Reachability for Hybrid Automata over Bounded Time". In: *ICALP Part II* (July 4–8, 2011). Ed. by Luca Aceto, Monika Henzinger, and Jirí Sgall. Vol. 6756. LNCS. Zurich, Switzerland: Springer, 2011, pp. 416–427. DOI: 10.1007/978-3-642-22012-8_33.

[BRS19] Lei Bu, Rajarshi Ray, and Stefan Schupp. "ARCH-COMP19 Category Report: Bounded Model Checking of Hybrid Systems with Piecewise Constant Dynamics". In: *ARCH@CPSIoTWeek* (Apr. 15, 2019). Vol. 61. EPiC Series in Computing. Montréal, QC, Canada: EasyChair, 2019, pp. 120–128.

[Bu+19] Lei Bu, Jiawan Wang, Yuming Wu, and Xuandong Li. "From Bounded Reachability Analysis of Linear Hybrid Automata to Verification of Industrial CPS and IoT". In: *SETSS* (Apr. 21–27, 2019). Vol. 12154. LNCS. Chongqing, China: Springer, 2019, pp. 10–43. DOI: 10.1007/978-3-030-55089-9_2.

[BZ19] Anna Becchi and Enea Zaffanella. "Revisiting Polyhedral Analysis for Hybrid Systems". In: *SAS* (Oct. 8–11, 2019). Ed. by Bor-Yuh Evan Chang. Vol. 11822. LNCS. Porto, Portugal: Springer, 2019, pp. 183–202. DOI: 10.1007/978-3-030-32304-2_10.

# References III


[DHR05] Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. “Automatic Rectangular Refinement of Affine Hybrid Systems”. In: *FORMATS* (Sept. 26–28, 2005). Ed. by Paul Pettersson and Wang Yi. Vol. 3829. LNCS. Uppsala, Sweden: Springer, 2005, pp. 144–161. DOI: 10.1007/11603009_13.

[FI04] Ansgar Fehnker and Franjo Ivancic. “Benchmarks for Hybrid Systems Verification”. In: *HSCC* (Mar. 25–27, 2004). Ed. by Rajeev Alur and George J. Pappas. Vol. 2993. LNCS. Philadelphia, PA, USA: Springer, 2004, pp. 326–341. DOI: 10.1007/978-3-540-24743-2_22.

[FJ13] Léa Fanchon and Florent Jacquemard. “Formal Timing Analysis Of Mixed Music Scores”. In: *ICMC* (Aug. 12–16, 2013). Perth, Australia: Michigan Publishing, Aug. 2013.

[FJS07] Frantisek Franek, Christopher G. Jennings, and William F. Smyth. “A simple fast hybrid pattern-matching algorithm”. In: *Journal of Discrete Algorithms* 5.4 (2007), pp. 682–695. DOI: 10.1016/j.jda.2006.11.004.

[Fre08] Goran Frehse. “PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech”. In: *International Journal on Software Tools for Technology Transfer* 10.3 (May 2008), pp. 263–279. ISSN: 1433-2779. DOI: 10.1007/s10009-007-0062-x.

[Fri+12] Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. “Robustness Analysis for Scheduling Problems using the Inverse Method”. In: *TIME* (Sept. 12–14, 2012). Ed. by Mark Reynolds, Paolo Terenziani, and Ben Moszkowski. Leicester, UK: IEEE Computer Society Press, Sept. 2012, pp. 73–80. DOI: 10.1109/TIME.2012.10.

# References IV


[GA22] Bineet Ghosh and Étienne André. “Monitoring of scattered uncertain logs using uncertain linear dynamical systems”. In: *FORTE* (June 13–17, 2022). Ed. by Mohammad Mousavi and Anna Philippou. Vol. 13273. LNCS. Lucca, Italy: Springer, 2022, pp. 67–87. DOI: 10.1007/978-3-031-08679-3_5.

[HAF14] Bardh Hoxha, Houssam Abbas, and Georgios E. Fainekos. “Benchmarks for Temporal Logic Requirements for Automotive Systems”. In: *ARCH@CPSWeek* (Apr. 14, 2014–Apr. 13, 2015). Ed. by Goran Frehse and Matthias Althoff. Vol. 34. EPiC Series in Computing. Berlin, Germany and Seattle, WA, USA: EasyChair, 2014, pp. 25–30.

[Hen96] Thomas A. Henzinger. “The Theory of Hybrid Automata”. In: *LiCS* (July 27–30, 1996). Ed. by Moshe Y. Vardi and Edmund M. Clarke. New Brunswick, New Jersey, USA: IEEE Computer Society, 1996, pp. 278–292. DOI: 10.1109/LICS.1996.561342.

[Lut+17] Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. “Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints”. In: *SPLC, Volume A* (Sept. 25–29, 2017). Ed. by Myra B. Cohen, Mathieu Acher, Lidia Fuentes, Daniel Schall, Jan Bosch, Rafael Capilla, Ebrahim Bagheri, Yingfei Xiong, Javier Troya, Antonio Ruiz Cortés, and David Benavides. Sevilla, Spain: ACM, 2017, pp. 104–113. DOI: 10.1145/3106195.3106204.

[Sel+22] Daniel Selvaratnam, Michael Cantoni, J. M. Davoren, and Iman Shames. “MITL Verification Under Timing Uncertainty”. In: *FORMATS* (Sept. 13–15, 2022). Ed. by Sergiy Bogomolov and David Parker. Vol. 13465. LNCS. Warsaw, Poland: Springer, 2022, pp. 136–152. DOI: 10.1007/978-3-031-15839-1_8.

# References V


[Ulu+14] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. "Timed Pattern Matching". In: *FORMATS* (Sept. 8–10, 2014). Ed. by Axel Legay and Marius Bozga. Vol. 8711. LNCS. Florence, Italy: Springer, 2014, pp. 222–236. DOI: 10.1007/978-3-319-10512-3_16.

[Ulu+16] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. "Online Timed Pattern Matching Using Derivatives". In: *TACAS* (Apr. 2–8, 2016). Ed. by Marsha Chechik and Jean-François Raskin. Vol. 9636. LNCS. Eindhoven, The Netherlands: Springer, 2016, pp. 736–751. DOI: 10.1007/978-3-662-49674-9_47.

[Ulu17] Dogan Ulus. "Montre: A Tool for Monitoring Timed Regular Expressions". In: *CAV, Part I* (July 24–28, 2017). Ed. by Rupak Majumdar and Viktor Kuncak. Vol. 10426. LNCS. Heidelberg, Germany: Springer, 2017, pp. 329–335. DOI: 10.1007/978-3-319-63387-9_16.

[WAH16] Masaki Waga, Takumi Akazaki, and Ichiro Hasuo. "A Boyer-Moore Type Algorithm for Timed Pattern Matching". In: *FORMATS* (Aug. 24–26, 2016). Ed. by Martin Fränzle and Nicolas Markey. Vol. 9884. LNCS. Québec, QC, Canada: Springer, 2016, pp. 121–139. DOI: 10.1007/978-3-319-44878-7_8.

[WAH19] Masaki Waga, Étienne André, and Ichiro Hasuo. "Symbolic Monitoring against Specifications Parametric in Time and Data". In: *CAV, Part I* (July 15–18, 2019). Ed. by Işil Dillig and Serdar Tasiran. Vol. 11561. LNCS. New York City, USA: Springer, 2019, pp. 520–539. DOI: 10.1007/978-3-030-25540-4_30.

[WAH22a] Masaki Waga, Étienne André, and Ichiro Hasuo. "Model-Bounded Monitoring of Hybrid Systems". In: *ACM Transactions on Cyber-Physical Systems* 6.4 (Nov. 2022), 30:1–30:26. DOI: 10.1145/3529095.

# References VI


[WAH22b] Masaki Waga, Étienne André, and Ichiro Hasuo. "Parametric Timed Pattern Matching". In: *ACM Transactions on Software Engineering and Methodology* (2022).

[WHS17] Masaki Waga, Ichiro Hasuo, and Kohei Suenaga. "Efficient Online Timed Pattern Matching by Automata-Based Skipping". In: *FORMATS* (Sept. 5–7, 2019). Ed. by Alessandro Abate and Gilles Geeraerts. Vol. 10419. LNCS. Berlin, Germany: Springer, 2017, pp. 224–243. DOI: 10.1007/978-3-319-65765-3_13.

# Licensing

# Source of the graphics used I

Title: Tesla Model S
Author: National Transportation Safety Board
Source: https://commons.wikimedia.org/wiki/File:Tesla_Model_S_(35366284636).jpg
License: public domain

Title: Mtn view tesla scene graphic
Author: National Transportation Safety Board
Source: https://commons.wikimedia.org/wiki/File:Mtn_view_tesla_scene_graphic_(28773524958).jpg
License: public domain

Title: 1960 Citroen DS19
Author: Joc281
Source: https://en.wikipedia.org/wiki/File:800px_1973_377_Citroen_DS19_automatically_guided_motor
License: CC by-sa 3.0

Title: A Cartoon Businessman Reading A Text Message
Author: Vector Toons
Source: https://en.wikipedia.org/wiki/File:800px_1973_377_Citroen_DS19_automatically_guided_motor
License: CC by-sa 4.0

# License of this document



Authors: **Étienne André** and **Masaki Waga**