VERIMAG

28 janvier 2021

Symbolic Monitoring against Specifications Parametric in Time and Data

Étienne André[1], Ichiro Hasuo[2,3] and Masaki Waga[2,3]

[1] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
[2] National Institute of Informatics, Japan
[3] SOKENDAI (The Graduate University for Advanced Studies)

Supported by the ANR national research program PACS (ANR-14-CE28-0002), and by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603).

Outline

Motivation: automotive industry

- Modern cars embed several processors and produce logs

Motivation: 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: 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: 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"

⇒ 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

Outline

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

- Naive algorithm for pattern matching

c	r	e	p	e	s	$\in ?L\Big(\{c \mid i \mid d\}^? r^* e\Big)$
c						

Untimed pattern matching: example

- Naive algorithm for pattern matching

c	r	e	p	e	s	$\in ? L\Big(\{c \vert i \vert d\}^? r^* e\Big)$
c	r					

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

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				$\checkmark$
	r	e				$\checkmark$
		e				$\checkmark$
			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				✓
	r	e				✓
		e				✓
			p			×

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				$\checkmark$
	r	e				$\checkmark$
		e				$\checkmark$
			p			$\times$
				e		

Untimed pattern matching: example

- Naive algorithm for pattern matching

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

Untimed pattern matching: example

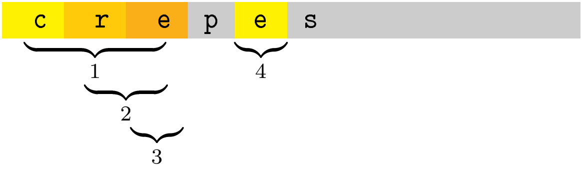
- Naive algorithm for pattern matching

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

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				✓
	r	e				✓
		e				✓
			p			×
				e		✓
					s	×

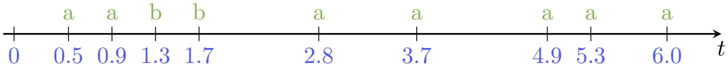


Timed pattern matching: timed word

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. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

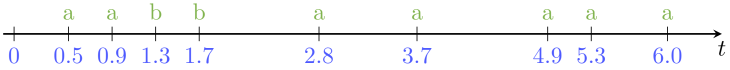
[WAH16] Masaki Waga, Takumi Akazaki, and Ichiro Hasuo. “A Boyer-Moore Type Algorithm for Timed Pattern Matching”. In: *FORMATS* (Aug. 24–26, 2016). Ed. by Martin Fränzle and Nicolas Markey. Vol. 9884. LNCS. Québec, QC, Canada: Springer, 2016, pp. 121–139. DOI: 10.1007/978-3-319-44878-7_8

Timed pattern matching: timed word

Timed word

[AD94]

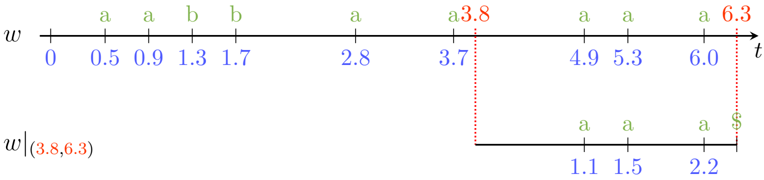
= sequence of actions and timestamps



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. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

[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

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."

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

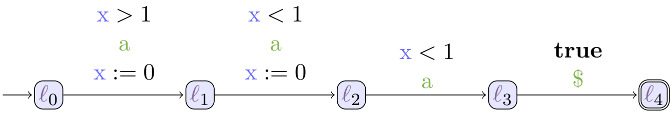
Timed pattern matching: timed automaton

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. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

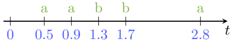
Timed pattern matching: principle

Timed pattern matching

- Inputs

A log

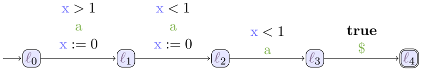
(timed word)



A property

usually a specification of faults

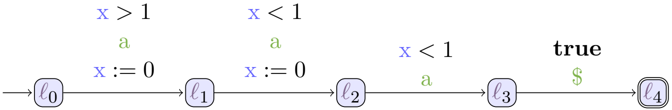
(timed automaton)



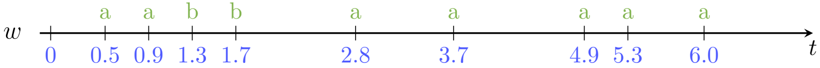
- Output
  - The set of time intervals where faults are detected
    $\Rightarrow$ Set of matching intervals $\{(t, t') \mid w|_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$

Timed pattern matching: example

Our property:

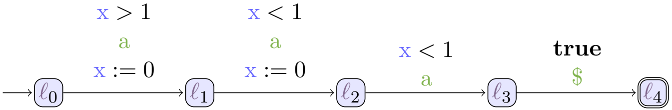


Our log:

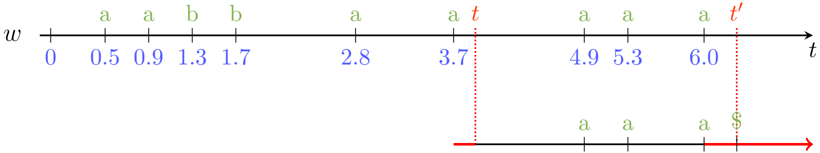


Timed pattern matching: example

Our property:

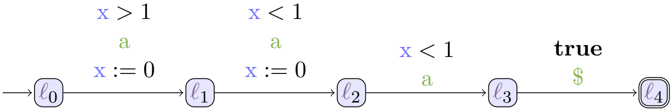


Our log:

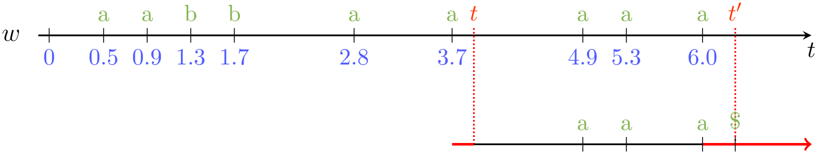


Timed pattern matching: example

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* (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

[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

[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

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"?

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

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 $\mathcal{A}$	$\{(t, t') \in (\mathbb{R}_{>0})^2 \mid w\|_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$
parametric timed PM	timed word	PTA $\mathcal{A}$	$\{(t, t', v) \mid w\|_{(t,t')} \in \mathcal{L}(v(\mathcal{A}))\}$

Outline

Methodology

Main idea

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

- Toolkit: IMITATOR [And+12]

[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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. “IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems”. In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36

Methodology

Main idea

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

- Toolkit: IMITATOR [And+12]

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* (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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. "IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems". In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36

Methodology

Main idea

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

- Toolkit: IMITATOR [And+12]

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* (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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. "IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems". In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36

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, a is observed.
Then, within strictly less than p_2 time units, another a is observed.
Then, within strictly less than p_2 time units, another a is observed."

[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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

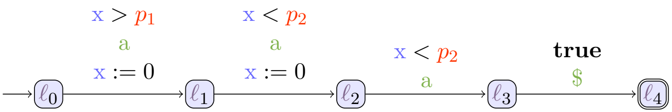
Property: parametric timed automaton

Expressing a parametric timed property on a log

Example

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

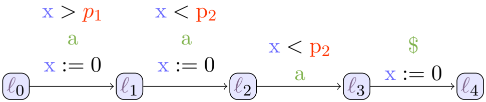
Our solution: use parametric timed automata [AHV93]



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC* (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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

Modifying the property pattern

Add some start and end gadgets for completeness of the method [AHW18]

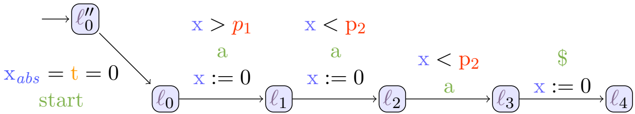


[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. “Offline timed pattern matching under uncertainty”. In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

Modifying the property pattern

Add some start and end gadgets for completeness of the method [AHW18]

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

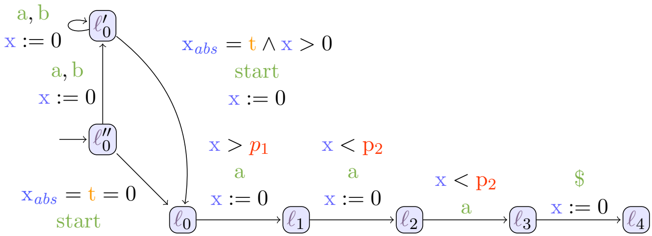


[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

Modifying the property pattern

Add some start and end gadgets for completeness of the method [AHW18]

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

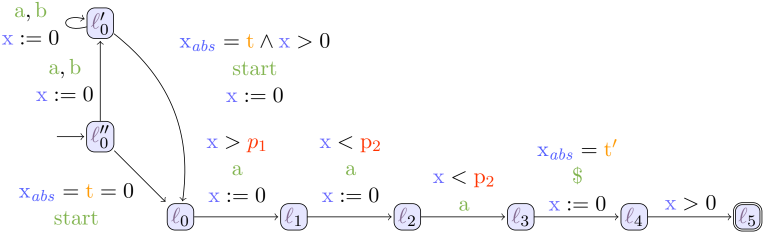


[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

Modifying the property pattern

Add some start and end gadgets for completeness of the method [AHW18]

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



[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. “Offline timed pattern matching under uncertainty”. In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

Transforming a log into a (parametric) timed automaton

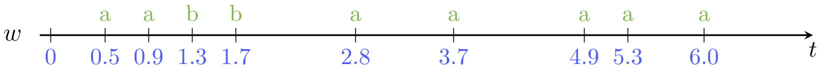
Essentially easy:

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:

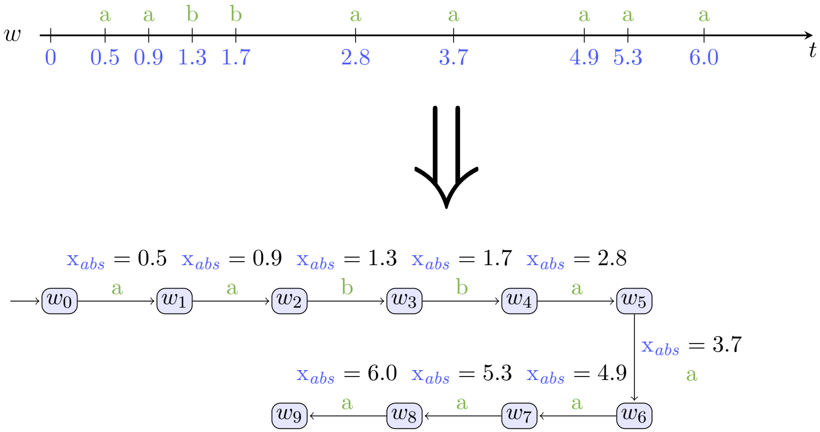
1. Add one clock never reset (absolute time)
2. Convert pairs (action, time) into transitions



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

[And19b] 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

[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

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!

[And19b]

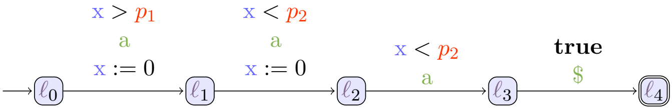
See formal result in paper [AHW18]

[And19b] 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

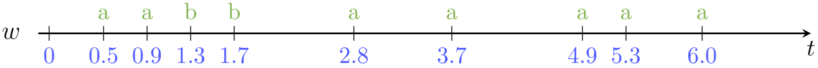
[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

Product and reachability synthesis: example

Our property:

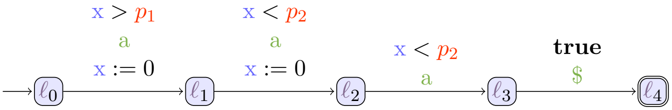


Our log:

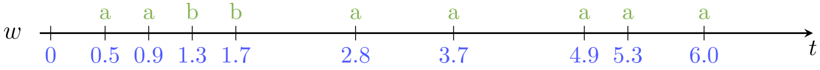


Product and reachability synthesis: example

Our property:



Our log:

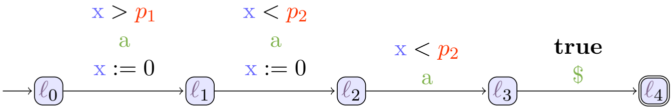


Set of matching intervals:

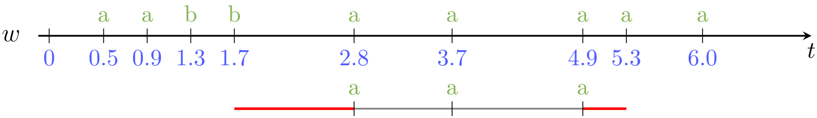
$$
\begin{aligned}
& 1.7 < t < 2.8 - p_1 \wedge 4.9 \leq t' < 5.3 \wedge p_2 > 1.2 \\
\vee\ & 2.8 < t < 3.7 - p_1 \wedge 5.3 \leq t' < 6 \wedge p_2 > 1.2 \\
\vee\ & 3.7 < t < 4.9 - p_1 \wedge t' \geq 6 \wedge p_2 > 0.7
\end{aligned}
$$

Product and reachability synthesis: example

Our property:



Our log:

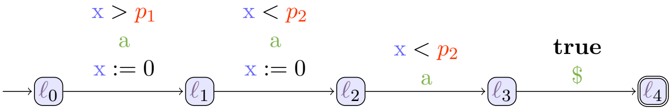


Set of matching intervals:

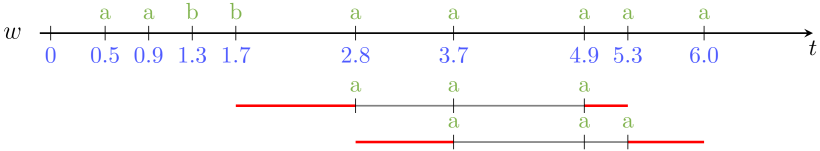
$$
\begin{aligned}
& 1.7 < t < 2.8 - p_1 \land 4.9 \leq t' < 5.3 \land p_2 > 1.2 \\
\lor \; & 2.8 < t < 3.7 - p_1 \land 5.3 \leq t' < 6 \land p_2 > 1.2 \\
\lor \; & 3.7 < t < 4.9 - p_1 \land t' \geq 6 \land p_2 > 0.7
\end{aligned}
$$

Product and reachability synthesis: example

Our property:



Our log:

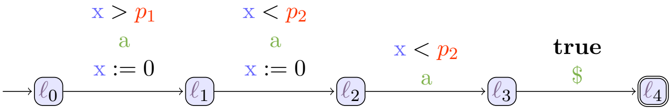


Set of matching intervals:

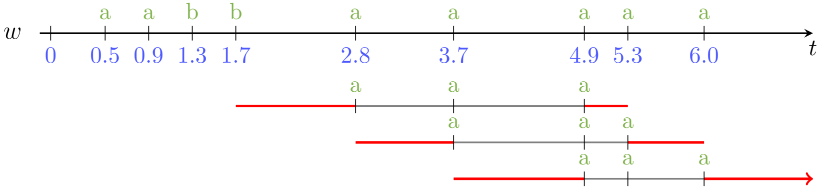
$$
\begin{aligned}
& 1.7 < t < 2.8 - p_1 \wedge 4.9 \leq t' < 5.3 \wedge p_2 > 1.2 \\
\vee\ & 2.8 < t < 3.7 - p_1 \wedge 5.3 \leq t' < 6 \wedge p_2 > 1.2 \\
\vee\ & 3.7 < t < 4.9 - p_1 \wedge t' \geq 6 \wedge p_2 > 0.7
\end{aligned}
$$

Product and reachability synthesis: example

Our property:



Our log:



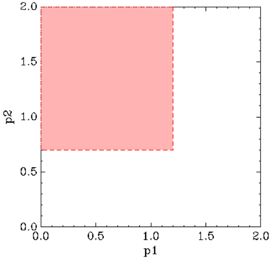
Set of matching intervals:

$$1.7 < t < 2.8 - p_1 \wedge 4.9 \leq t' < 5.3 \wedge p_2 > 1.2$$
$$\vee\ 2.8 < t < 3.7 - p_1 \wedge 5.3 \leq t' < 6 \wedge p_2 > 1.2$$
$$\vee\ 3.7 < t < 4.9 - p_1 \wedge t' \geq 6 \wedge p_2 > 0.7$$

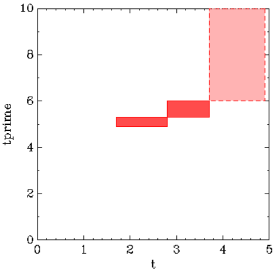
Exemple: graphical representation

$$\begin{aligned} & 1.7 < t < 2.8 - p_1 \wedge 4.9 \leq t' < 5.3 \wedge p_2 > 1.2 \\ \vee\ & 2.8 < t < 3.7 - p_1 \wedge 5.3 \leq t' < 6 \wedge p_2 > 1.2 \\ \vee\ & 3.7 < t < 4.9 - p_1 \wedge t' \geq 6 \wedge p_2 > 0.7 \end{aligned}$$

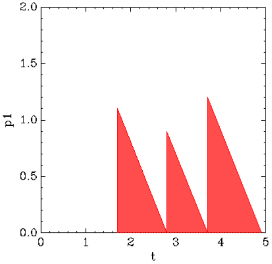
Projections in 2 dimensions:



On p_1 and p_2



On t and t'



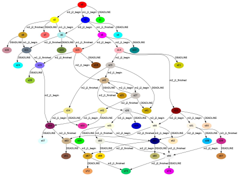
On t and p_1

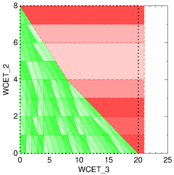
Outline

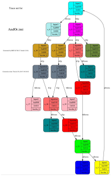
IMITATOR

- A tool for modeling and verifying timed concurrent systems with unknown constants modeled with parametric timed automata
  - Communication through (strong) broadcast synchronization
  - Rational-valued shared discrete variables
  - Stopwatches, to model schedulability problems with preemption

- Synthesis algorithms
  - (non-Zeno) parametric model checking (using a subset of TCTL)
  - Language and trace preservation, and robustness analysis
  - Parametric deadlock-freeness checking







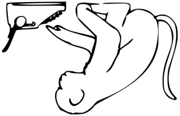
IMITATOR

Under continuous development since 2008 [And+12]

A library of benchmarks [And19a]

- Communication protocols
- Schedulability problems
- Asynchronous circuits
- ...and more

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



[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. “IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems”. In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6

[And19a] Étienne André. “A benchmark library for parametric timed model checking”. In: *FTSCS* (Nov. 16, 2018). Ed. by Cyrille Artho and Peter Csaba Ölveczky. Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5

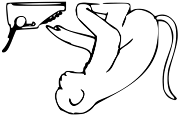
IMITATOR

Under continuous development since 2008 [And+12]

A library of benchmarks [And19a]

- Communication protocols
- Schedulability problems
- Asynchronous circuits
- ...and more

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



Try it!

www.imitator.fr

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. "IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems". In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6

[And19a] Étienne André. "A benchmark library for parametric timed model checking". In: *FTSCS* (Nov. 16, 2018). Ed. by Cyrille Artho and Peter Csaba Ölveczky. Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5

Some success stories

- Modeled and verified an asynchronous memory circuit by ST-Microelectronics
- Parametric schedulability analysis of a prospective architecture for the flight control system of the next generation of spacecrafts designed at ASTRIUM Space Transportation [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

[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

[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

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

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
- PTPM_{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.”

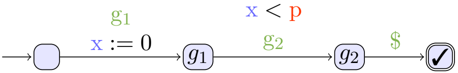
This condition is related to the requirement ϕ_5^{AT} proposed in [HAF14] (the nominal value for p in [HAF14] is 2).

[HAF14] Bardh Hoxha, Houssam Abbas, and Georgios E. Fainekos. “Benchmarks for Temporal Logic Requirements for Automotive Systems”. In: *ARCH@CPSWeek*. Ed. by Goran Frehse and Matthias Althoff. Vol. 34. EPiC Series in Computing. Berlin, Germany and Seattle, WA, USA: 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*. 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

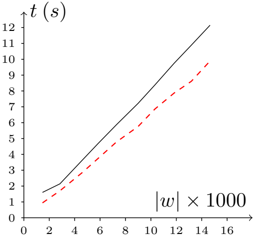
Case study 1: GEAR (experiments)

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



Experiments data:

Model		PTPM				PTPM_{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



PTPM_{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

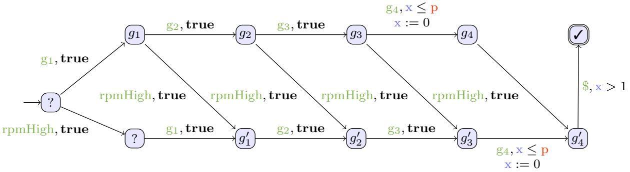
“If a gear changes from 1 to 2, 3, and 4 in this order in p seconds and engine RPM becomes large during this gear change, then the velocity of the car must be sufficiently large in one second.”

This condition models the requirement ϕ_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*. Ed. by Goran Frehse and Matthias Althoff. Vol. 34. EPiC Series in Computing. Berlin, Germany and Seattle, WA, USA: EasyChair, 2014, pp. 25–30

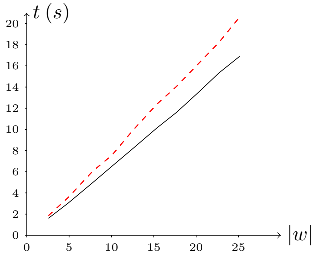
Case study 2: ACCEL (experiments)

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."



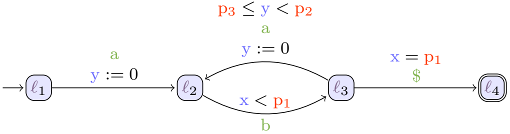
Experiments data:

Model		PTPM				PTPM$_{opt}$	
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



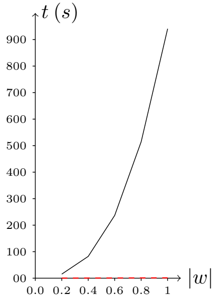
Case study 3: Blowup

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

- 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

Summary

- 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

- An online algorithm
  - We believe our algorithm is in fact essentially online
    - No need for the whole log to start the analysis
    - The word could be fed to IMITATOR in an incremental manner
  - But the speed may need to be improved further

Outline

(see dedicated slides)

[WAH19]

[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

Outline

Conclusion and perspectives

Two methods:

- Parametric timed pattern matching [AHW18]
- Symbolic monitoring [WAH19]

Perspectives

- Go more efficient
- Go more symbolic (e. g., BDDs?)
- Graphical representation and interpretation
  - How to interpret dozens of thousands of matches?

[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

[WAH19] Masaki Waga, Étienne André, and Ichiro Hasuo. "Symbolic Monitoring against Specifications Parametric in Time and Data". In: *CAV, Part*

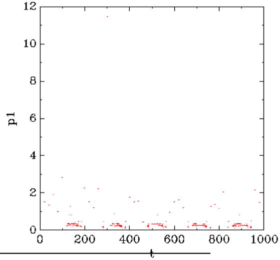
Conclusion and perspectives

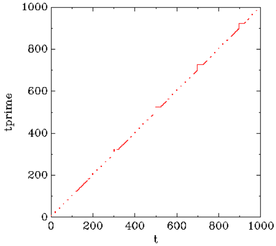
Two methods:

- Parametric timed pattern matching [AHW18]
- Symbolic monitoring [WAH19]

Perspectives

- Go more efficient
- Go more symbolic (e. g., BDDs?)
- Graphical representation and interpretation
  - How to interpret dozens of thousands of matches?





[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010

[WAH19] Masaki Waga, Étienne André, and Ichiro Hasuo. "Symbolic Monitoring against Specifications Parametric in Time and Data". In: *CAV, Part*

Bibliography

References I

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

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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242.

Étienne André, Ichiro Hasuo, and Masaki Waga. “Offline timed pattern matching under uncertainty”. In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010.

Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. “IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems”. In: *FM* (Aug. 27–31, 2012). Ed. by Dimitra Giannakopoulou and Dominique Méry. Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6.

Étienne André. “A benchmark library for parametric timed model checking”. In: *FTSCS* (Nov. 16, 2018). Ed. by Cyrille Artho and Peter Csaba Ölveczky. Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5.

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.

References II

Yashwanth Annpureddy, Che Liu, Georgios E. Fainekos, and Sriram Sankaranarayanan. "S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems". In: *TACAS*. 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.

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

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.

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.

Bardh Hoxha, Houssam Abbas, and Georgios E. Fainekos. "Benchmarks for Temporal Logic Requirements for Automotive Systems". In: *ARCH@CPSWeek*. Ed. by Goran Frehse and Matthias Althoff. Vol. 34. EPiC Series in Computing. Berlin, Germany and Seattle, WA, USA: EasyChair, 2014, pp. 25–30.

References III

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.

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.

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.

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.

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.

References IV

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.

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.

Additional information

Untimed pattern matching

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 $\mathcal{A}$	$\{(t,t') \in (\mathbb{R}_{>0})^2 \mid w\|_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$
parametric timed PM	timed word	PTA $\mathcal{A}$	$\{(t,t',v) \mid w\|_{(t,t')} \in \mathcal{L}(v(\mathcal{A}))\}$

Timed pattern matching

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 $\mathcal{A}$	$\{(t,t') \in (\mathbb{R}_{>0})^2 \mid w\|_{(t,t')} \in \mathcal{L}(\mathcal{A})\}$
parametric timed PM	timed word	PTA $\mathcal{A}$	$\{(t,t',v) \mid w\|_{(t,t')} \in \mathcal{L}(v(\mathcal{A}))\}$

Outline

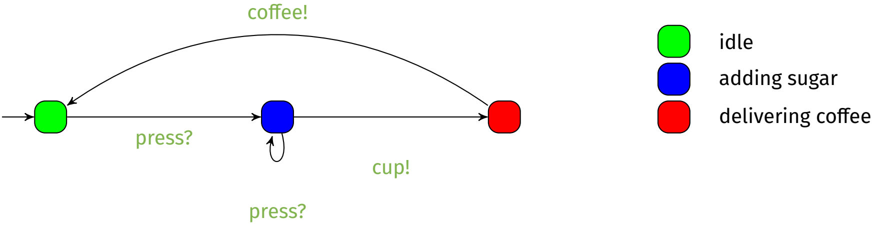
Timed automaton (TA)

- Finite state automaton (sets of locations)

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

Timed automaton (TA)

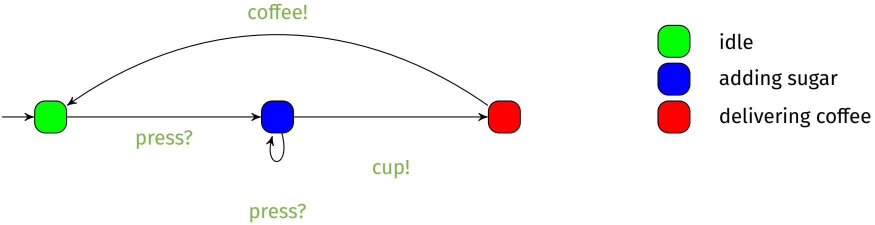
- Finite state automaton (sets of locations and actions)



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

Timed automaton (TA)

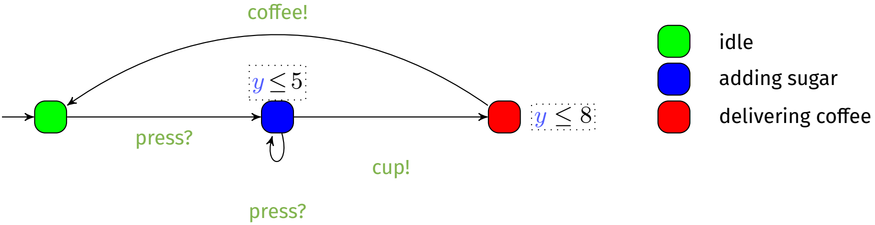
- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate



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

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - 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

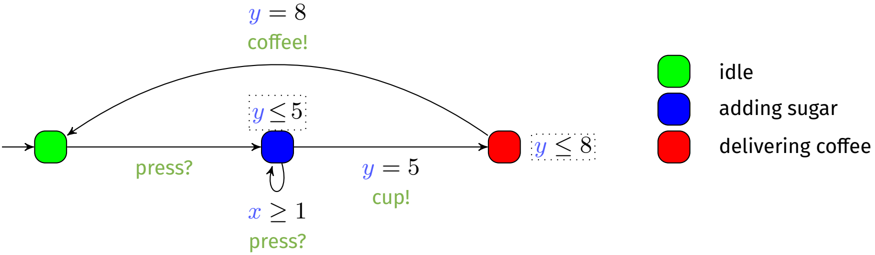


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - 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

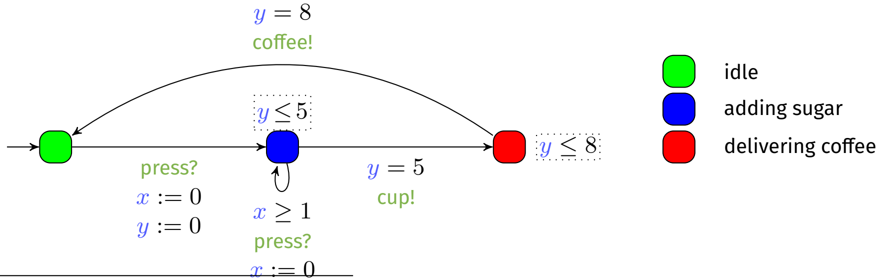


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

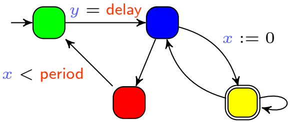
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition
  - Clock reset: some of the clocks can be set to 0 along transitions



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

Outline

timed model checking



A model of the system

$\overset{?}{\models}$

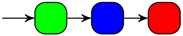
 is unreachable

A property to be satisfied

- Question: does the model of the system satisfy the property?

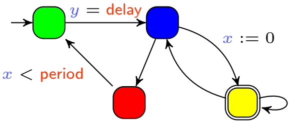
Yes

No



Counterexample

Parametric timed model checking



A model of the system

$\overset{?}{\models}$

 is unreachable

A property to be satisfied

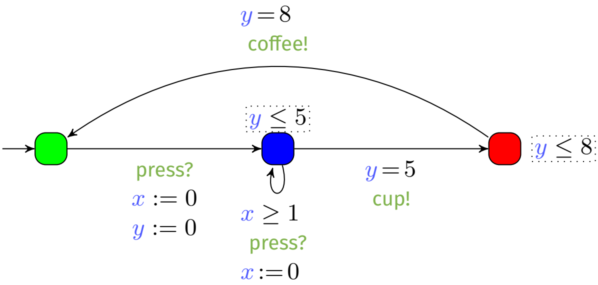
- Question: for what values of the parameters does the model of the system satisfy the property?

Yes if...

$$2\text{delay} > \text{period} \\ \land\ \text{period} < 20.46$$

Parametric Timed Automaton (PTA)

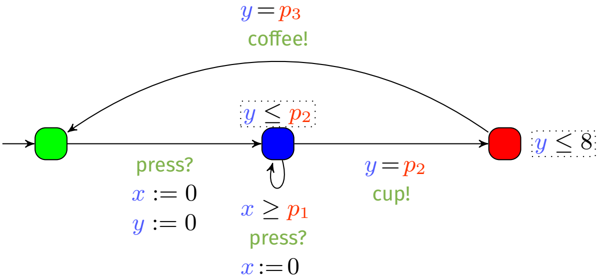
- Timed automaton (sets of locations, actions and clocks)



[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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

Parametric Timed Automaton (PTA)

- Timed automaton (sets of locations, actions and clocks) augmented with a set P of parameters [AHV93]
  - Unknown constants compared to a clock in guards and invariants



[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. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

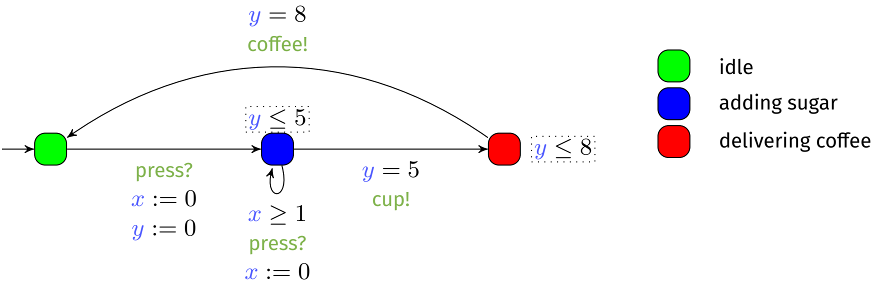
Concrete semantics of timed automata

- Concrete state of a TA: pair (ℓ, w), where
  - ℓ is a location,
  - w is a valuation of each clock

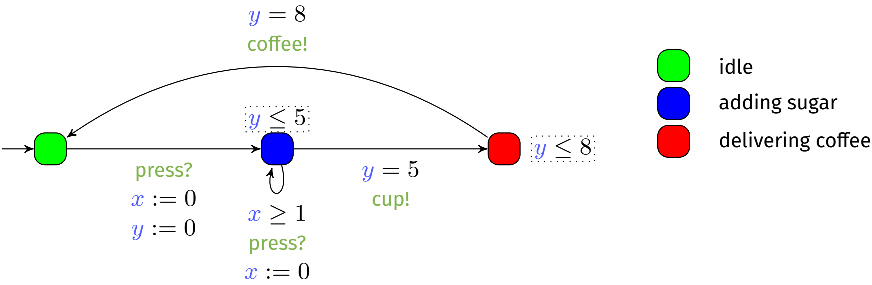
  Example: (, $\binom{x=1.2}{y=3.7}$)

- Concrete run: alternating sequence of concrete states and actions or time elapse

The most critical system: The coffee machine



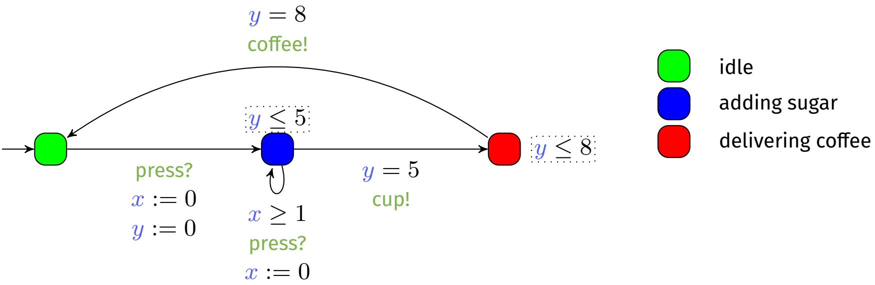
The most critical system: The coffee machine



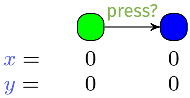
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

$x =$ 0
$y =$ 0

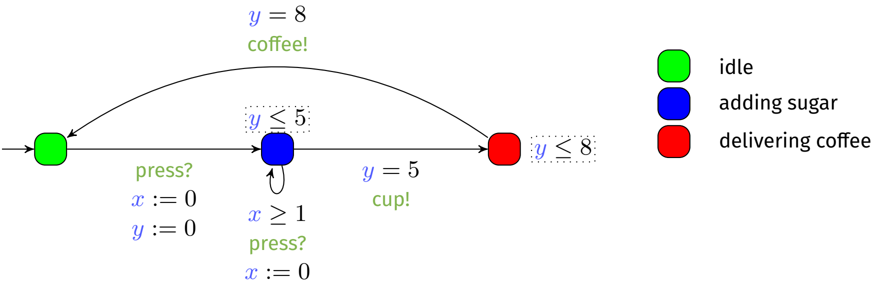
The most critical system: The coffee machine



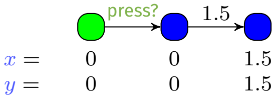
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



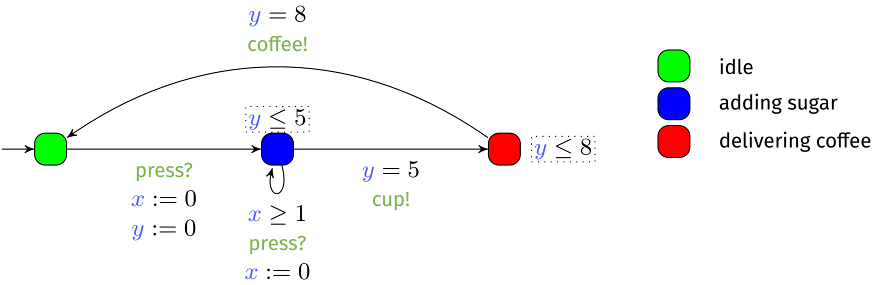
The most critical system: The coffee machine



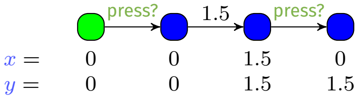
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



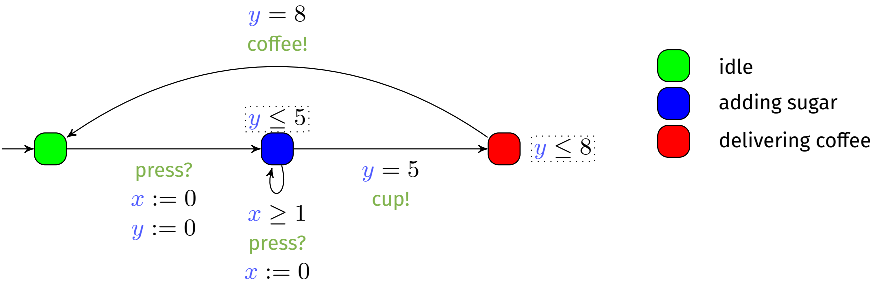
The most critical system: The coffee machine



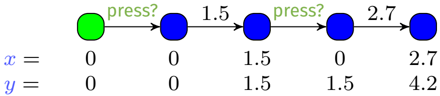
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



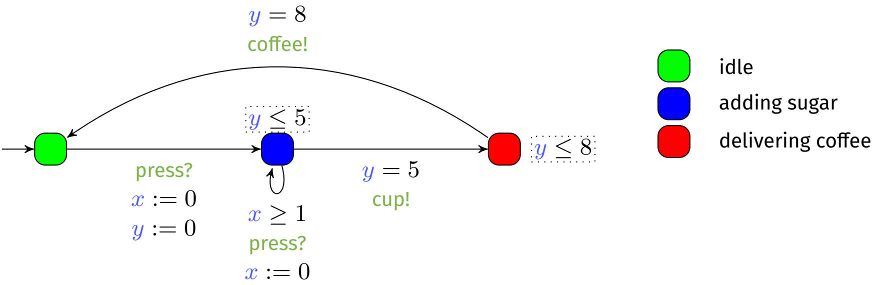
The most critical system: The coffee machine



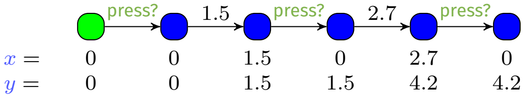
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



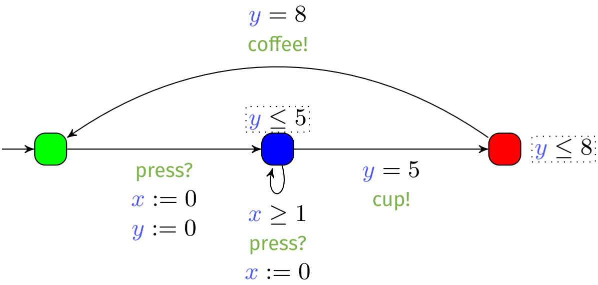
The most critical system: The coffee machine

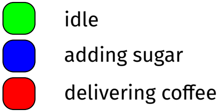


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

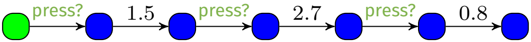


The most critical system: The coffee machine



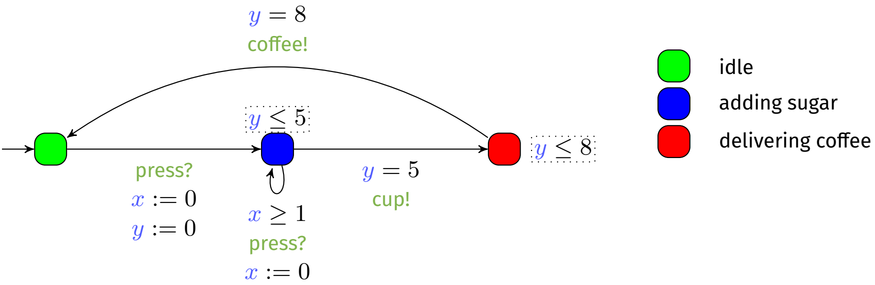


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

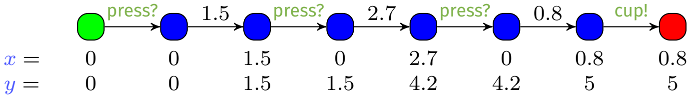


$x =$	0	0	1.5	0	2.7	0	0.8
$y =$	0	0	1.5	1.5	4.2	4.2	5

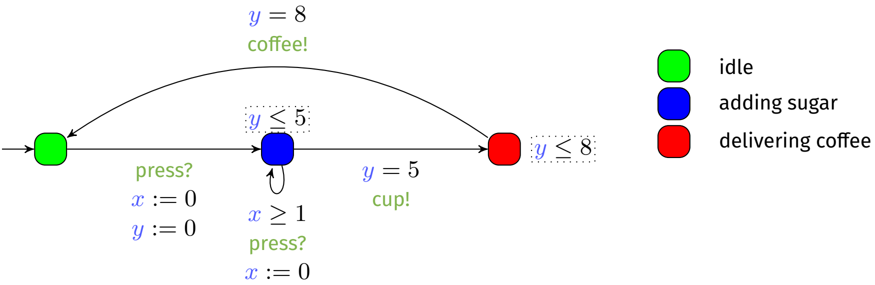
The most critical system: The coffee machine



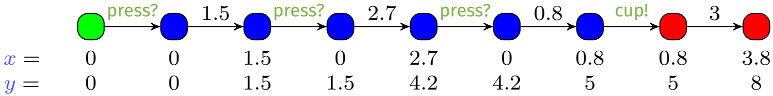
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



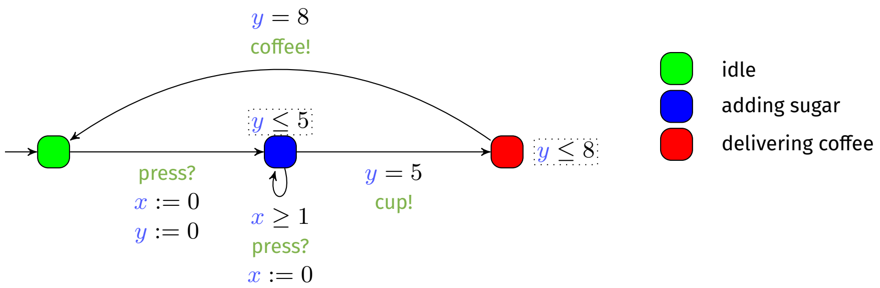
The most critical system: The coffee machine



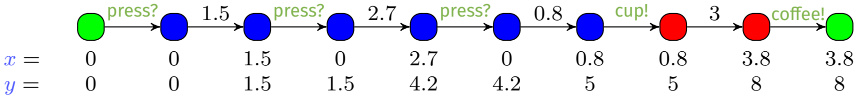
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



The most critical system: The coffee machine



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



Licensing

Source of the graphics used I

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

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

License of this document

Author: **Étienne André**