FTSCS 2015

Friday 6th of November, 2015
Paris, France

What's Decidable About Parametric Timed Automata?

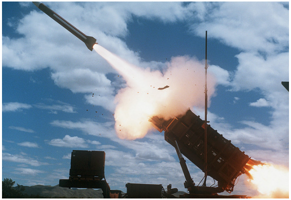
Étienne André

LIPN, Université Paris 13, Sorbonne Paris Cité, CNRS, France
École Centrale de Nantes, IRCCyN, CNRS, UMR 6597, France

Context: Model checking (1/2)

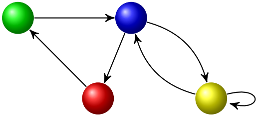
- Need for early bug detection
  - Bugs discovered when final testing: expensive

$\leadsto$ Need for a thorough modeling and verification phase



Context: Model checking (2/2)

- Model checking



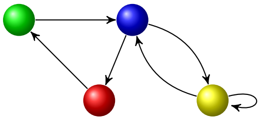
A model of the system

 is unreachable

A property to be satisfied

Context: Model checking (2/2)

- Model checking



$\stackrel{?}{\models}$

 is unreachable

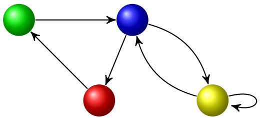
A model of the system

A property to be satisfied

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

Context: Model checking (2/2)

- Model checking



$\overset{?}{\models}$

 is unreachable

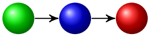
A model of the system

A property to be satisfied

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

Yes

No



Counterexample

Context: Timed model checking

- Timed systems are characterized by a set of timing constants
  - "The packet transmission lasts for 50 ms"
  - "The sensor reads the value every 10 s"

- Powerful model checking tools, e.g.:
  - UPPAAL [Larsen et al., 1997]
  - PAT [Sun et al., 2009]

Beyond timed model checking: parameter synthesis

- Verification for one set of constants does not usually guarantee the correctness for other values

- Challenges
  - Numerous verifications: is the system correct for any value within $[40; 60]$?
  - Optimization: until what value can we increase 10?
  - Robustness [Markey, 2011]: What happens if 50 is implemented with 49.99?

Beyond timed model checking: parameter synthesis

- Verification for one set of constants does not usually guarantee the correctness for other values

- Challenges
  - Numerous verifications: is the system correct for any value within $[40; 60]$?
  - Optimization: until what value can we increase 10?
  - Robustness [Markey, 2011]: What happens if 50 is implemented with 49.99?

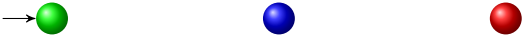
- Parameter synthesis
  - Consider that timing constants are unknown constants (parameters)
  - Find good values for the parameters

Outline

Outline

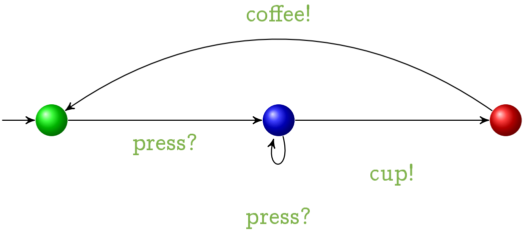
Timed automaton (TA)

- Finite state automaton (sets of locations)



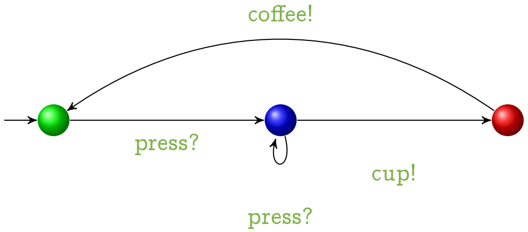
Timed automaton (TA)

- Finite state automaton (sets of locations and actions)



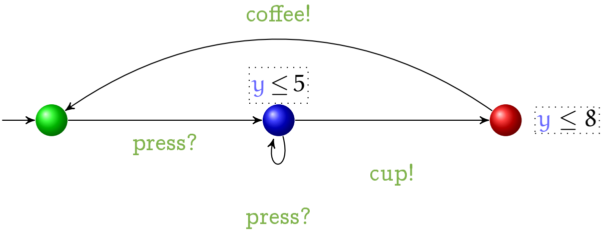
Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [Alur and Dill, 1994]
  - Real-valued variables evolving linearly at the same rate



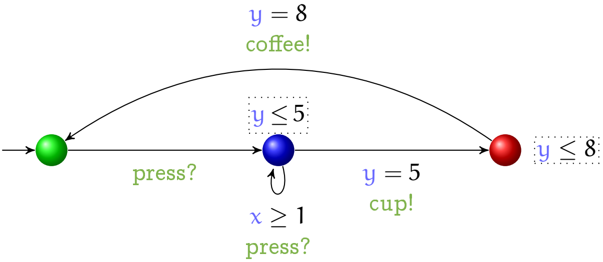
Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [Alur and Dill, 1994]
  - Real-valued variables evolving linearly at the same rate
- Features
  - Location invariant: property to be verified to stay at a location



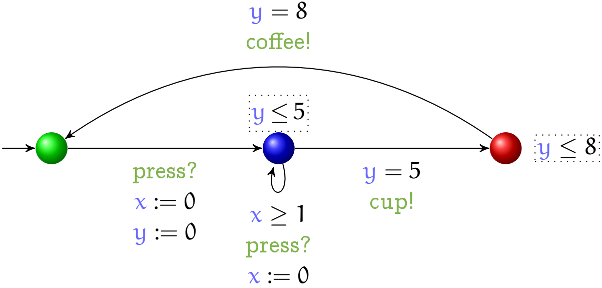
Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [Alur and Dill, 1994]
  - Real-valued variables evolving linearly at the same rate
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition



Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [Alur and Dill, 1994]
  - Real-valued variables evolving linearly at the same rate
- 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 at each transition



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

- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine

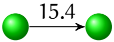
Concrete semantics of timed automata

- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

x	0
y	0

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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

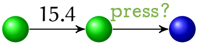


x	0	15.4
y	0	15.4

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

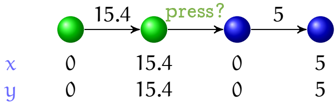
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0
y	0	15.4	0

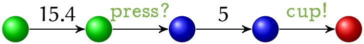
Concrete semantics of timed automata

- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



Concrete semantics of timed automata

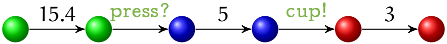
- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0	5	5
y	0	15.4	0	5	5

Concrete semantics of timed automata

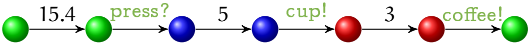
- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0	5	5	8
y	0	15.4	0	5	5	8

Concrete semantics of timed automata

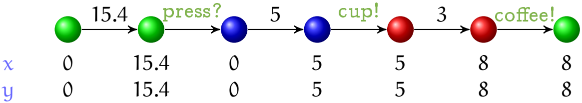
- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



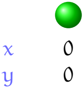
x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

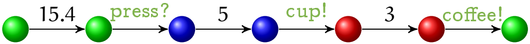


    - Coffee with 2 doses of sugar



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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



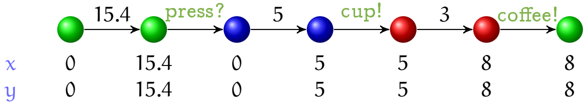
x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

    - Coffee with 2 doses of sugar

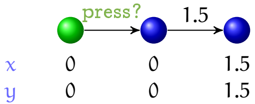
x	0	0
y	0	0

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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



    - Coffee with 2 doses of sugar



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

- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

$$\bullet \xrightarrow{15.4} \bullet \xrightarrow{\text{press?}} \bullet \xrightarrow{5} \bullet \xrightarrow{\text{cup!}} \bullet \xrightarrow{3} \bullet \xrightarrow{\text{coffee!}} \bullet$$

x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

    - Coffee with 2 doses of sugar

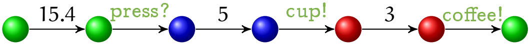
$$\bullet \xrightarrow{\text{press?}} \bullet \xrightarrow{1.5} \bullet \xrightarrow{\text{press?}} \bullet$$

x	0	0	1.5	0
y	0	0	1.5	1.5

Concrete semantics of timed automata

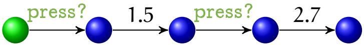
- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock

- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

    - Coffee with 2 doses of sugar



x	0	0	1.5	0	2.7
y	0	0	1.5	1.5	4.2

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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

$$\bullet \xrightarrow{15.4} \bullet \xrightarrow{press?} \bullet \xrightarrow{5} \bullet \xrightarrow{cup!} \bullet \xrightarrow{3} \bullet \xrightarrow{coffee!} \bullet$$

x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

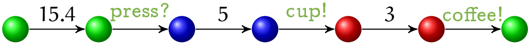
    - Coffee with 2 doses of sugar

$$\bullet \xrightarrow{press?} \bullet \xrightarrow{1.5} \bullet \xrightarrow{press?} \bullet \xrightarrow{2.7} \bullet \xrightarrow{press?} \bullet$$

x	0	0	1.5	0	2.7	0
y	0	0	1.5	1.5	4.2	4.2

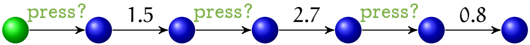
Concrete semantics of timed automata

- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

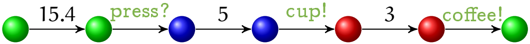
    - Coffee with 2 doses of sugar



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

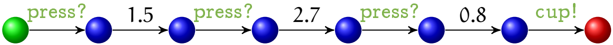
Concrete semantics of timed automata

- Concrete state of a TA: pair (l, w), where
  - l is a location,
  - w is a valuation of each clock
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

    - Coffee with 2 doses of sugar



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

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

- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar

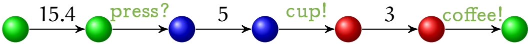
		15.4		press?		5		cup!		3		coffee!	
x	0		15.4		0		5		5		8		8
y	0		15.4		0		5		5		8		8

    - Coffee with 2 doses of sugar

		press?		1.5		press?		2.7		press?		0.8		cup!		3	
x	0		0		1.5		0		2.7		0		0.8		0.8		3.8
y	0		0		1.5		1.5		4.2		4.2		5		5		8

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
- Concrete run: alternating sequence of concrete states and actions or elapsing of time
  - Possible concrete runs for the coffee machine
    - Coffee with no sugar



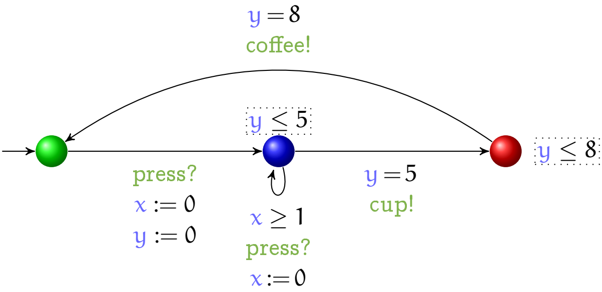
x	0	15.4	0	5	5	8	8
y	0	15.4	0	5	5	8	8

    - Coffee with 2 doses of sugar

x	0	0	1.5	0	2.7	0	0.8	0.8	3.8	3.8
y	0	0	1.5	1.5	4.2	4.2	5	5	8	8

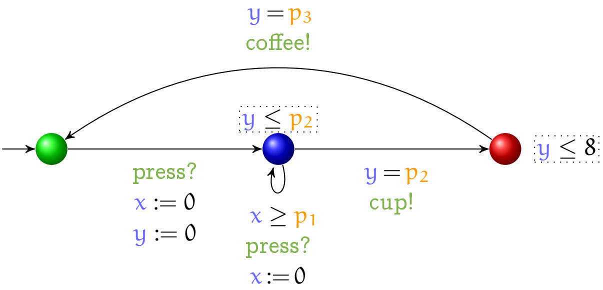
Parametric Timed Automaton (PTA)

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



Parametric Timed Automaton (PTA)

- Timed automaton (sets of locations, actions and clocks) augmented with a set P of parameters [Alur et al., 1993]
  - Unknown constants used in guards and invariants



Valuation of a PTA

- Given a PTA $\mathcal{A}$ and a parameter valuation v, we denote by $v(\mathcal{A})$ the (non-parametric) timed automaton where all parameters are valuated by v

Integers or rationals?

In PTAs, both the clocks and the parameters can be either integer-valued or rational-valued.
This gives three possibilities:

	Clocks	Parameters
Discrete time	$\mathbb{N}$	$\mathbb{N}$
Dense time	$\mathbb{R}^+$	$\mathbb{N}$
Dense time	$\mathbb{R}^+$	$\mathbb{Q}^+$

Integers or rationals?

In PTAs, both the clocks and the parameters can be either integer-valued or rational-valued.
This gives three possibilities:

	Clocks	Parameters
Discrete time	$\mathbb{N}$	$\mathbb{N}$
Dense time	$\mathbb{R}^+$	$\mathbb{N}$
Dense time	$\mathbb{R}^+$	$\mathbb{Q}^+$

…and this does have an impact on decidability.

Model checkers for PTAs

Tools taking PTAs as an input formalism

- HyTech [Henzinger et al., 1997] (hybrid automata)
- An extension of Uppaal [Hune et al., 2002]
- Roméo [Lime et al., 2009] (parametric time Petri nets)
- IMITATOR [André et al., 2012]

Case studies using PTAs

- Variants of train controllers [Alur et al., 1993, Hune et al., 2002]
- The root contention protocol [Hune et al., 2002]
- Philip's bounded retransmission protocol [Hune et al., 2002]
- An asynchronous circuit commercialized by ST-Microelectronics [Chevallier et al., 2009]
- A 4-phase handshake protocol [Knapik and Penczek, 2012]
- A distributed prospective architecture for the flight control system of the next generation of spacecrafts designed at ASTRIUM Space Transportation [Fribourg et al., 2012]
- Analysis of music scores [Fanchon and Jacquemard, 2013]
- The alternating bit protocol [Jovanović et al., 2015]
- (non-preemptive) schedulability problems [Jovanović et al., 2015]
- An unmanned aerial video system by Thales

Outline

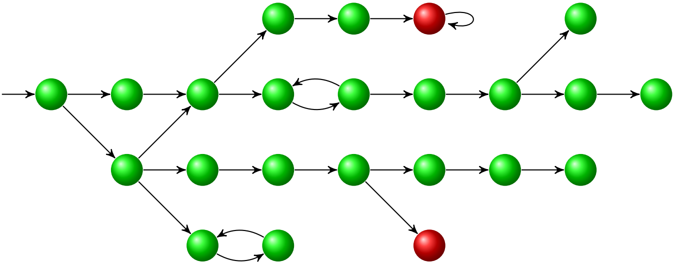
Decision problems (1/3)

Definition (reachability emptiness (EF-emptiness))

Input: a PTA $\mathcal{A}$ and a set of locations G

Problem: Is the set of parameter valuations v such that there exists a run of $v(\mathcal{A})$ reaching a location $l \in G$ empty?

$v(\mathcal{A})$:



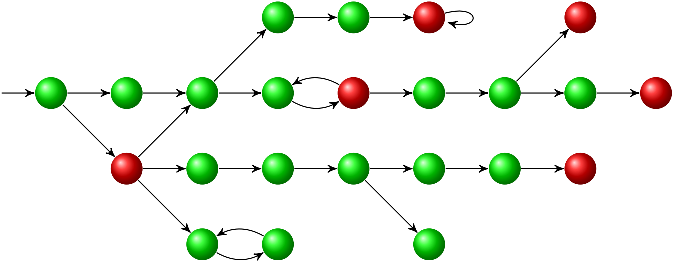
Decision problems (2/3)

Definition (unavoidability emptiness (AF-emptiness))

Input: a PTA $\mathcal{A}$ and a set of locations G

Problem: Is the set of parameter valuations v such that all runs of $v(\mathcal{A})$ eventually reach a location $l \in G$ empty?

$v(\mathcal{A})$:



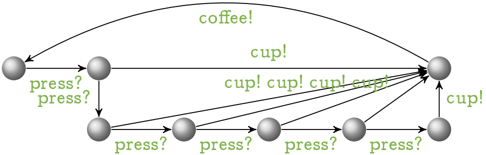
Decision problems (3/3)

Definition (language preservation emptiness)

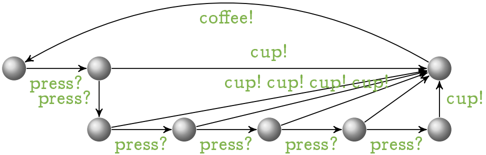
Input: a PTA $\mathcal{A}$ and a valuation v

Problem: Does there exist another valuation $v' \neq v$ such that the untimed languages of $v(\mathcal{A})$ and $v'(\mathcal{A})$ are the same?

$v(\mathcal{A})$:



$v'(\mathcal{A})$:



Motivation

Why surveying decidability?

Without decidability, no hope for exact synthesis

- “Find all parameter valuations such that…”

(although approximated results can still be output in general, and exact results “sometimes”)

Outline

Undecidability: reachability

Reachability emptiness

Reachability emptiness ("does there exist at least one parameter valuation reaching a given location l?") is undecidable for PTAs [Alur et al., 1993]

- even with a single real-valued parameter [Miller, 2000]
- even with only strict constraints [Doyen, 2007]
- even with a single integer-valued parameter [Beneš et al., 2015]

Undecidability: reachability

Reachability emptiness

Reachability emptiness ("does there exist at least one parameter valuation reaching a given location l?") is undecidable for PTAs [Alur et al., 1993]

- even with a single real-valued parameter [Miller, 2000]
- even with only strict constraints [Doyen, 2007]
- even with a single integer-valued parameter [Beneš et al., 2015]

Proof.

By reduction from the halting problem of a 2-counter machine, which is undecidable [Minsky, 1967] □

Undecidability: unavoidability

Unavoidability emptiness

Unavoidability emptiness ("does there exist at least one parameter valuation such that all runs reach a given location l?") is undecidable for PTAs, even with a single bounded parameter

[Jovanović et al., 2015]

Undecidability: language preservation

Language preservation

Language preservation emptiness is undecidable for PTAs, even with a single bounded parameter

[André and Markey, 2015]

Outline

Bounding the number of clocks

- All problems are undecidable for just 1 parameter, and even over bounded-time
- However, bounding clocks brings decidability
  - Parametric clocks: clocks compared with a parameter in at least one guard or one invariant
  - Non-parametric clocks: clocks never compared with a parameter

EF-emptiness: main results

EF-emptiness is decidable for

- 1 parametric clock and arbitrarily many non-parametric clocks and parameters over discrete time [Bundala and Ouaknine, 2014]
- 2 parametric clocks, arbitrarily many non-parametric clocks and 1 parameter over discrete time [Bundala and Ouaknine, 2014]
- 1 parametric clock, no non-parametric clock and arbitrarily many parameters over dense time [Miller, 2000]

EF-emptiness is open for

- 2 parametric clocks, arbitrarily many non-parametric clocks and > 1 parameters over discrete time
- 1 parametric clock, 1 or 2 non-parametric clocks and any parameters over dense time
- 2 parametric clock, 0, 1 or 2 non-parametric clocks and any parameters over dense time

EF-emptiness: full state-of-the-art

$\mathbb{T}$	$\mathbb{P}$	Guards	Invariants	P-clocks	NP-clocks	Params	Decidability	Main ref.
$\mathbb{N}$	$\mathbb{N}$	$x \leq\geq p\|d^+$		1	any	any	NEXPTIME-compl.	[Bundala and Ouaknine, 2014]
$\mathbb{N}$	$\mathbb{N}$	$x \in I$	None	1	any	any	non-elementary	[Alur et al., 1993]
$\mathbb{N}$	$\mathbb{N}$	$x \leq\geq p\|d^+$		2	any	1	PSPACE$^{\text{NEXP}}$-hard	[Bundala and Ouaknine, 2014]
$\mathbb{N}$	$\mathbb{N}$	any		2	any	> 1	open	
$\mathbb{N}$	$\mathbb{N}$	$x \sim p\|d$	None	3	0	1	undecidable	[Beneš et al., 2015]
$\mathbb{N}$	$\mathbb{N}$	$x <> p$		any	any	any	open	
$\mathbb{N}$	$\mathbb{N}$ bounded	$x \sim plt$	$x \preceq plt$	any	any	any	decidable	[Jovanović et al., 2015] (conseq.)
$\mathbb{R}^+$	$\mathbb{N}$	$x \in I$	None	1	0	any	non-elementary	[Alur et al., 1993] (conseq.)
$\mathbb{R}^+$	$\mathbb{N}$	$x \sim p\|d$	$x \preceq p$	1	any	any	NEXPTIME	[Beneš et al., 2015]
$\mathbb{R}^+$	$\mathbb{N}$	$x \leq\geq p\|d^+$		2	any	1	PSPACE$^{\text{NEXP}}$-hard	[Bundala and Ouaknine, 2014]
$\mathbb{R}^+$	$\mathbb{N}$	any		2	any	> 1	open	
$\mathbb{R}^+$	$\mathbb{N}$	$x \sim p\|d$	None	3	0	1	undecidable	[Beneš et al., 2015]
$\mathbb{R}^+$	$\mathbb{N}$	$x \sim plt$	$x \preceq plt$	3	0	2	undecidable	[Jovanović et al., 2015]
$\mathbb{Q}^+/\mathbb{R}^+$	$\mathbb{N}$	$x <> p$		any	any	any	open	
$\mathbb{R}^+$	$\mathbb{N}$ bounded	$x \sim plt$	$x \preceq plt$	any	any	any	PSPACE-complete	[Jovanović et al., 2015]
$\mathbb{R}^+$	$\mathbb{R}^+$	$x \in I$	None	1	0	any	non-elementary	[Alur et al., 1993]
$\mathbb{R}^+$	$\mathbb{Q}^+$	$x \sim p\|d$		1	0	any	NP-complete	[Miller, 2000]
$\mathbb{R}^+$	$\mathbb{Q}^+$	$x \sim p\|d$		1	0	bounded	PTIME	[Miller, 2000]
$\mathbb{R}^+$	$\mathbb{R}^+$	any		1	1 or 2	1	open	
$\mathbb{R}^+$	$\mathbb{Q}^+$	$x \sim p\|d$		1	3	1	undecidable	[Miller, 2000]
$\mathbb{R}^+$	$\mathbb{R}^+$	any		2	any	any	open	
$\mathbb{R}^+$	$\mathbb{R}^+$	$x \in I$	None	3	0	6	undecidable	[Alur et al., 1993]
$\mathbb{R}^+$	$\mathbb{Q}^+$	$x \sim p\|d$		3	0	1	undecidable	[Miller, 2000]
$\mathbb{R}^+$	$\mathbb{R}^+_{[1;2]}$	$x \sim p\|d$		1	3	1	undecidable	[Miller, 2000]
$\mathbb{R}^+$	$\mathbb{R}^+_{[1;2]}$	$x \sim p\|d$		3	0	1	undecidable	[Miller, 2000]
$\mathbb{Q}^+/\mathbb{R}^+$	$\mathbb{Q}^+/\mathbb{R}^+$	$x <> p$		< 2	< 3	< 2	open	
$\mathbb{Q}^+/\mathbb{R}^+$	$\mathbb{Q}^+/\mathbb{R}^+$	$x <> p$		2	3	2	undecidable	[Doyen, 2007]

Outline

Restricting the use of parameters

So far:

- ☹ Bounding the number of parameters...
- ☹ Bounding time...
- ☹ Considering discrete-time...

does not bring decidability

Bounding the number of clocks brings decidability

- ☹ ...but can we still model something interesting with 1 clock?

Restricting the use of parameters

So far:

- ☹ Bounding the number of parameters…
- ☹ Bounding time…
- ☹ Considering discrete-time…

does not bring decidability

Bounding the number of clocks brings decidability

- ☹ …but can we still model something interesting with 1 clock?

Idea: restrict the syntactic use of the parameters

Lower-bound and upper-bound parameters

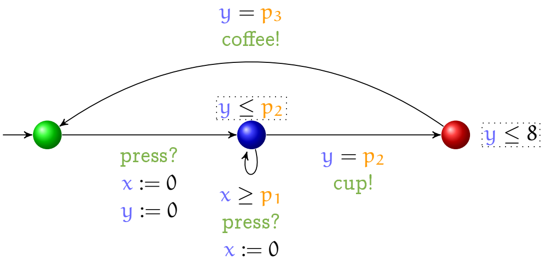
- A lower-bound parameter can only be compared with a clock from below
  - Examples: $p_l \leq x$, $p_l < x$
- An upper-bound parameter can only be compared with a clock from above
  - Examples: $p_u \geq x$, $p_u > x$

Lower-bound and upper-bound parameters

- A lower-bound parameter can only be compared with a clock from below
  - Examples: $p_l \leq x$, $p_l < x$
- An upper-bound parameter can only be compared with a clock from above
  - Examples: $p_u \geq x$, $p_u > x$

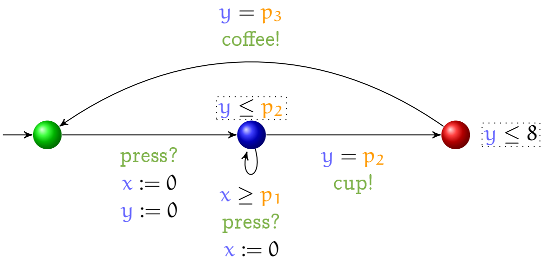
- An L/U-PTA contains only lower-bound and upper-bound parameters
- An L-PTA contains only lower-bound parameters
- A U-PTA contains only upper-bound parameters

Examples



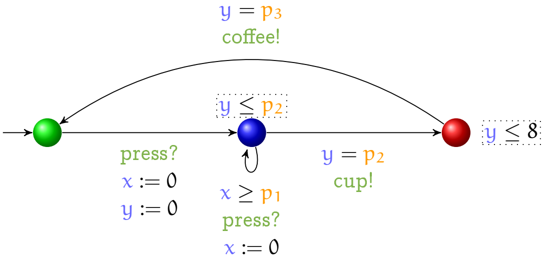
Examples

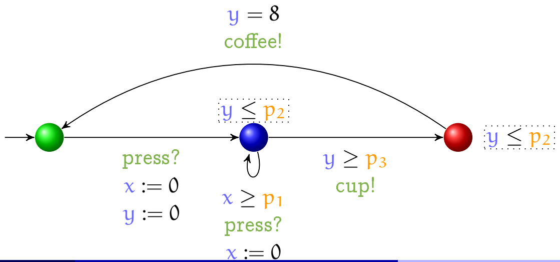
Not an L/U-PTA



Examples

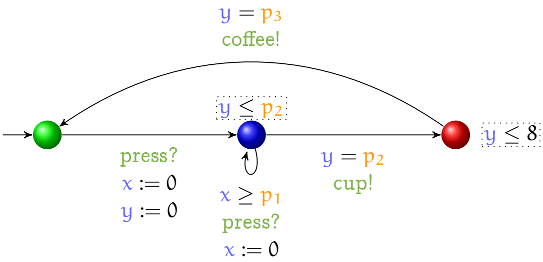
Not an L/U-PTA



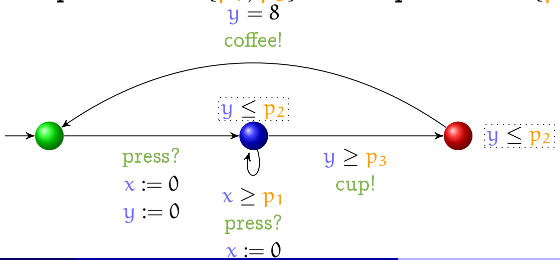


Examples

Not an L/U-PTA



An L/U-PTA with L-parameters $\{p_1, p_3\}$ and U-parameter $\{p_2\}$



Modeling with L/U-PTAs

- Many case studies in the literature are L/U-PTAs
  - Including case studies proposed before the definition of L/U-PTAs
- All models with intervals are L/U-PTAs
  - Asynchronous circuits with (parametric) bi-bounded delays
  - Train controllers with intervals
  - Real-time systems with (parametric) bi-bounded periods

Decidability for L/U-PTAs

Reachability emptiness

Reachability emptiness ("does there exist at least one parameter valuation reaching a given location l?") is decidable for L/U-PTAs

[Hune et al., 2002, Bozzelli and La Torre, 2009]

Decidability for L/U-PTAs

Reachability emptiness

Reachability emptiness ("does there exist at least one parameter valuation reaching a given location l?") is decidable for L/U-PTAs

[Hune et al., 2002, Bozzelli and La Torre, 2009]

Reachability universality

Reachability universality ("do all parameter valuations reach a given location l?") is decidable for L/U-PTAs

[Hune et al., 2002, Bozzelli and La Torre, 2009]

Undecidability for L/U-PTAs

Unavoidability emptiness

Unavoidability emptiness ("does there exist at least one parameter valuation such that all runs reach a given location l?") is undecidable for L/U-PTAs, even with a single bounded parameter

[Jovanović et al., 2015]

Undecidability for L/U-PTAs

Unavoidability emptiness

Unavoidability emptiness ("does there exist at least one parameter valuation such that all runs reach a given location l?") is undecidable for L/U-PTAs, even with a single bounded parameter

[Jovanović et al., 2015]

Language preservation

Language preservation emptiness is undecidable for L/U-PTAs, even with a single bounded parameter

[André and Markey, 2015]

Quite bad news

Synthesis

Although the emptiness and universality are decidable, the EF-synthesis ("synthesize all parameter valuations such that a given location l is reachable") is intractable for L/U-PTAs

[Jovanović et al., 2015]

This rules out the possibility to do parameter synthesis for L/U-PTAs.

Outline

Summary

Parametric timed automata

- ☺ Powerful and expressive formalism
- ☺ Used to solve many case studies
- ☺ Models robustness, uncertainty, unknown drift
- ☹ All non-trivial problems undecidable

L/U-PTAs

- ☺ Emptiness decidable
- ☺ Remain quite expressive
- ☹ Synthesis intractable

Perspectives

Open decidability issues

- Case of PTAs with 2 clocks over discrete time
- Classes of L-PTAs and U-PTAs
  - Less expressive than L/U-PTAs
  - All problems (decision and synthesis) are open

Perspectives

Open decidability issues

- Case of PTAs with 2 clocks over discrete time
- Classes of L-PTAs and U-PTAs
  - Less expressive than L/U-PTAs
  - All problems (decision and synthesis) are open

A unified syntax for PTAs and a comparison of expressiveness

- Nearly every paper defines a different syntax: are they all just as expressive?
- And what is the expressiveness of PTAs?

Perspectives

Open decidability issues

- Case of PTAs with 2 clocks over discrete time
- Classes of L-PTAs and U-PTAs
  - Less expressive than L/U-PTAs
  - All problems (decision and synthesis) are open

A unified syntax for PTAs and a comparison of expressiveness

- Nearly every paper defines a different syntax: are they all just as expressive?
- And what is the expressiveness of PTAs?

Is decidability really required?

- Use approximated synthesis
- Devise efficient (semi-)algorithms
- Propose restrictions dedicated to real-time scheduling

Bibliography

References I

Alur, R. and Dill, D. L. (1994).
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235.

Alur, R., Henzinger, T. A., and Vardi, M. Y. (1993).
Parametric real-time reasoning.
In *STOC*, pages 592–601. ACM.

André, É., Fribourg, L., Kühne, U., and Soulat, R. (2012).
IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.
In *FM*, volume 7436 of *Lecture Notes in Computer Science*, pages 33–36. Springer.

André, É. and Markey, N. (2015).
Language preservation problems in parametric timed automata.
In *FORMATS*, volume 9268 of *Lecture Notes in Computer Science*, pages 27–43. Springer.

Beneš, N., Bezděk, P., Larsen, K. G., and Srba, J. (2015).
Language emptiness of continuous-time parametric timed automata.
In *ICALP, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 69–81. Springer.

References II

Bozzelli, L. and La Torre, S. (2009).
Decision problems for lower/upper bound parametric timed automata.
Formal Methods in System Design, 35(2):121–151.

Bundala, D. and Ouaknine, J. (2014).
Advances in parametric real-time reasoning.
In *MFCS*, volume 8634 of *Lecture Notes in Computer Science*, pages 123–134. Springer.

Chevallier, R., Encrenaz-Tiphène, E., Fribourg, L., and Xu, W. (2009).
Timed verification of the generic architecture of a memory circuit using parametric timed automata.
Formal Methods in System Design, 34(1):59–81.

Doyen, L. (2007).
Robust parametric reachability for timed automata.
Information Processing Letters, 102(5):208–213.

Fanchon, L. and Jacquemard, F. (2013).
Formal timing analysis of mixed music scores.
In *International Computer Music Conference*.

References III

Fribourg, L., Lesens, D., Moro, P., and Soulat, R. (2012).
Robustness analysis for scheduling problems using the inverse method.
In *TIME*, pages 73–80. IEEE Computer Society Press.

Henzinger, T. A., Ho, P.-H., and Wong-Toi, H. (1997).
HyTech: A model checker for hybrid systems.
Software Tools for Technology Transfer, 1:110–122.

Hune, T., Romijn, J., Stoelinga, M., and Vaandrager, F. W. (2002).
Linear parametric model checking of timed automata.
Journal of Logic and Algebraic Programming, 52-53:183–220.

Jovanović, A., Lime, D., and Roux, O. H. (2015).
Integer parameter synthesis for timed automata.
IEEE Transactions on Software Engineering, 41(5):445–461.

Knapik, M. and Penczek, W. (2012).
Bounded model checking for parametric timed automata.
Transactions on Petri Nets and Other Models of Concurrency, 5:141–159.

Larsen, K. G., Pettersson, P., and Yi, W. (1997).
UPPAAL in a nutshell.
International Journal on Software Tools for Technology Transfer, 1(1-2):134–152.

References IV

Lime, D., Roux, O. H., Seidner, C., and Traonouez, L.-M. (2009).
Romeo: A parametric model-checker for Petri nets with stopwatches.
In *TACAS*, volume 5505 of *Lecture Notes in Computer Science*, pages 54–57. Springer.

Markey, N. (2011).
Robustness in real-time systems.
In *SIES*, pages 28–34. IEEE Computer Society Press.

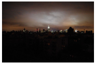
Miller, J. S. (2000).
Decidability and complexity results for timed automata and semi-linear hybrid automata.
In *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 296–309. Springer.

Minsky, M. L. (1967).
Computation: finite and infinite machines.
Prentice-Hall, Inc., NJ, USA.

Sun, J., Liu, Y., Dong, J. S., and Pang, J. (2009).
PAT: Towards flexible verification under fairness.
In *CAV*, volume 5643 of *Lecture Notes in Computer Science*, pages 709–714. Springer.

Additional explanation

Explanation for the 4 pictures in the beginning



Allusion to the Northeast blackout (USA, 2003)
Computer bug
Consequences: 11 fatalities, huge cost
(Picture actually from the Sandy Hurricane, 2012)

Error screen on the earliest versions of Macintosh

Allusion to the sinking of the Sleipner A offshore platform (Norway, 1991)
No fatalities
Computer bug: inaccurate finite element analysis modeling
(Picture actually from the Deepwater Horizon Offshore Drilling Platform)

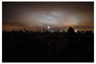
Allusion to the MIM-104 Patriot Missile Failure (Iraq, 1991)
28 fatalities, hundreds of injured
Computer bug: software error (clock drift)
(Picture of an actual MIM-104 Patriot Missile, though not the one of 1991)

Decision problems for L/U-PTAs

Problem	$\mathbb{P}$	Complexity	Main ref.
EF-emptiness	$\mathbb{R}^+$	PSPACE	[Hune et al., 2002]
AG-emptiness	$\mathbb{R}^+$	PSPACE	[Hune et al., 2002]
AF-emptiness	$\mathbb{R}^+$	undecidable	[Jovanović et al., 2015]
EG-emptiness	$\mathbb{R}^+$	open	
BüEF-emptiness	$\mathbb{N}$	PSPACE-complete	[Bozzelli and La Torre, 2009]
BüEF-universality	$\mathbb{N}$	PSPACE-complete	[Bozzelli and La Torre, 2009]
BüEF-finiteness	$\mathbb{N}$	PSPACE-complete	[Bozzelli and La Torre, 2009]
constrained BüEF-emptiness	$\mathbb{N}$	undecidable	[Bozzelli and La Torre, 2009]
constrained BüEF-universality	$\mathbb{N}$	undecidable	[Bozzelli and La Torre, 2009]
L/U-constrained BüEF-emptiness	$\mathbb{N}$	PSPACE-complete	[Bozzelli and La Torre, 2009]
L/U-constrained BüEF-universality	$\mathbb{N}$	PSPACE-complete	[Bozzelli and La Torre, 2009]
Language preservation	$\mathbb{N}$	undecidable	[André and Markey, 2015]
Language preservation	$\mathbb{R}^+$	undecidable	[André and Markey, 2015]

Licensing

Source of the graphics used I



Title: Hurricane Sandy Blackout New York Skyline
Author: David Shankbone
Source: https://commons.wikimedia.org/wiki/File:Hurricane_Sandy_Blackout_New_York_Skyline.JPG
License: CC BY 3.0

Title: Sad mac
Author: Przemub
Source: https://commons.wikimedia.org/wiki/File:Sad_mac.png
License: Public domain

Title: Deepwater Horizon Offshore Drilling Platform on Fire
Author: ideum
Source: https://secure.flickr.com/photos/ideum/4711481781/
License: CC BY-SA 2.0

Title: DA-SC-88-01663
Author: imcomkorea
Source: https://secure.flickr.com/photos/imcomkorea/3017886760/
License: CC BY-NC-ND 2.0

Source of the graphics used II

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