Timed automata with parametric updates

ACSD 2018
Étienne André *, Didier Lime ** & Mathias Ramparison*

*LIPN, Université Paris 13
**LS2N, École Centrale de Nantes

June 27, 2018
Introduction

- Discovering a bug during a test of a system can be very expensive
- Can have dramatical consequences in critical embedded system: autonomous car, in aeronautics...
Introduction

- Discovering a bug during a test of a system can be very expensive
- Can have dramatical consequences in critical embedded system: autonomous car, in aeronautics...
- Need for formal verification to ensure ahead the good behavior of a system
Model checking

- Model of a system:

  ![Model diagram]

  - $l_1$ ➔ $l_3$ ➔ $l_2$

- A property of the system: $l_3$ is reachable

- Check whether the system satisfies the property
Example of timed automaton

A timed automaton [AD94] which models a coffee machine

serve:
   \( y = 8 \)

Locations: \( \{l_1, l_2, l_3\} \), clocks: \( \{x, y\} \), action: \{press, press again, prepare, serve\}

Guard(press again) = \{y \leq 5 \land x \geq 0\},
Guard(prepare) = \{y = 5\}, Guard(serve) = \{y = 8\}

Reset(press) = \{x, y := 0\}, Reset(press again) = \{x := 0\}
A timed automaton [AD94] which models a coffee machine

serve:
\[ y = 8 \]

press:
\[ x := 0 \]
\[ y := 0 \]
press again:
\[ y \leq 5, x > 1 \]
\[ x := 0 \]

prepare:

\[ l_1 \] press → \[ l_2 \] press again → \[ l_3 \] prepare

▶ A run : \((l_1, (0, 0)) \xrightarrow{\text{press } 2.1} (l_2, (0, 0)) \xrightarrow{\text{press again } 1.2} (l_2, (0, 1.2)) \xrightarrow{\text{prepare } 3.8} (l_3, (3.8, 5)) \xrightarrow{\text{serve } 3} (l_1, (6.8, 8))\]

▶ triple (location, (value of x, value of y)) and name \( \delta \) discrete transition “name” after a delay \( \delta \).
Common decision problems for timed automata

► **Reachability**: Is there a run such that the location $l$ is reachable?

**Unavoidability**: For all runs, is the location $l$ reachable?
Common decision problems for timed automata

- **Reachability**: Is there a run such that the location \( l \) is reachable?
- **Unavoidability**: For all runs, is the location \( l \) reachable?
- Proved decidable in PSPACE [AD94]. Strategy: construct a finite automaton using an abstraction of clock valuations (clock regions)
Model checking with unknown constants

- **What if all constants are not specified ahead?**
- Model of a system with parameters:

  ![Diagram of a system with parameters](image)

  - $p_1 \leq \text{clock}$
  - $p_2 = \text{clock}$
  - $l_3$

- A property of the system: $l_3$ is reachable
- Compute the values of $p_1, p_2$ such that the system satisfies the property
A parametric timed automaton [AHV93] which models a parametric coffee machine

A possible run if $p_1 = 2$, $p_2 = 3$: $(l_1, (0, 0)) \xrightarrow{\text{press}} (l_2, (0, 0)) \xrightarrow{\text{press again}} (l_2, (0, 1)) \xrightarrow{\text{prepare}} (l_3, (1, 2)) \xrightarrow{\text{serve}} (l_1, (2, 3))$

The same run is impossible if $p_1 = 5$, $p_2 = 2$, or $p_1 < 1$. 
Challenges for parametric timed automata

- **EF-emptiness (decision problem):** is the set of parameter valuations s.t. there exists a run reaching \( l \) in the instantiated TA empty?

- **EF-synthesis (computation problem):** Compute all parameter valuations s.t. there exists a run reaching \( l \) in the instantiated TA
Challenges for parametric timed automata

- **EF-emptiness (decision problem):** is the set of parameter valuations s.t. there exists a run reaching l in the instantiated TA empty?

- **EF-synthesis (computation problem):** Compute all parameter valuations s.t. there exists a run reaching l in the instantiated TA

- EF-emptiness problem: proved undecidable in general case [AHV93], unbounded integer-valued parameters, (un)bounded rational valued parameters and even with only one bounded parameter [Mil00]
Challenges for parametric timed automata

- **EF-emptiness (decision problem):** is the set of parameter valuations s.t. there exists a run reaching \( l \) in the instantiated TA empty?

- **EF-synthesis (computation problem):** Compute all parameter valuations s.t. there exists a run reaching \( l \) in the instantiated TA

- **EF-emptiness problem:** proved undecidable in general case [AHV93], unbounded integer-valued parameters, (un)bounded rational valued parameters and even with only one bounded parameter [Mil00]

- To recover decidability, we need to add restrictions on parameters, or restrain the PTA syntax
Where to start from?

- Almost everything is undecidable for PTAs [And17]—especially EF-emptiness, AF-emptiness (is there a parameter valuation such that all runs reach a given location).
- Therefore, we go back to TAs.
- The reachability problem is PSPACE-complete for timed automata with updates to rational constants [BDFP04].

Figure: An updatable TA
Contributions

- New formalism with parametric updates of clocks: update-to-parameter TA (U2P-TA)
- Undecidability result for EF-emptiness and universality (are all parameter valuations such that there is a run reaching a given location) and AF-emptiness and universality (are all runs reaching a given location) for rational-valued parameters
- Decidability result for the same problems (in PSPACE) for integer-valued parameters, and synthesis of parameters
Update-to-parameter TA (U2P-TA): TA extended with updates to rational-valued parameters.

Parametric clock updates: $y := p_1$, $x := p_2$.
Bounded parameters $p_1, p_2$ i.e. $p_1, p_2 \in [a, b]$ with $a, b \in \mathbb{N}$. 
Theorem

*The EF-emptiness problem is undecidable for bounded rational-valued U2P-TAs*

Proof sketch: we prove that a bounded PTA can be simulated by a bounded U2P-TA.
U2P-TA

Figure: A PTA $A$

Figure: A U2P-TA obtained from $A$
Duplicate $x$.

**Figure:** A PTA $A$

$\begin{align*}
&l_0 \xrightarrow{x := 0} l_1 \quad x = p, x \geq 1 \xrightarrow{} l_2
\end{align*}$

**Figure:** A U2P-TA obtained from $A$

$\begin{align*}
&l_0 \xrightarrow{x := 0, x_p := px} l_1 \quad x \geq 1 \xrightarrow{} l_2
\end{align*}$
Compare $x_p$ with $C_{MAX}$ (maximum value between constants and parameters appearing in guards) where $x$ is compared to $p$.

![Figure: A PTA $A$](image)

![Figure: A U2P-TA obtained from $A$](image)
U2P-TA

As we can simulate (w.r.t. reachability) any **bounded rational-valued** U2P-TA using an **unbounded rational-valued** U2P-TA:

**Theorem**

*The EF-emptiness problem is undecidable for unbounded rational-valued U2P-TAs*

![Diagram](attachment:image.png)

**Figure:** A gadget that ensures a parameter $p$ is bounded by $min$ and $max$
U2P-TAs with integer-valued parameters over dense time.
Integer-valued U2P-TA

U2P-TAs with integer-valued parameters over dense time.

**Theorem**

*EF-synthesis is computable for unbounded integer-valued U2P-TAs.*
Corollary

The EF-emptiness problem is PSPACE-complete for unbounded integer-valued U2P-TAs and unlike integer-valued PTAs for which EF-emptiness is undecidable [AHV93,BBLS15].
Corollary

the EF-emptiness problem is PSPACE-complete for unbounded integer-valued U2P-TAs

and unlike integer-valued PTAs for which EF-emptiness is undecidable [AHV93,BBLS15].

Proof sketch: using equivalence between parameter valuations if $> K_{MAX}$ (the maximum constant value), we enumerate parameter valuations $\leq K_{MAX} + 1$ as they are bounded integers.
$v$ and $v'$ are equivalent.
$v$ and $v'$ are equivalent.
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

\[ p_1 \]

\[ p_2 \]

$K_{\text{MAX}}$
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

$K_{\text{MAX}}$
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

$p_1$ $p_2$

$v$ $K_{\text{MAX}}$ $K_{\text{MAX}}$
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

LaTeX code: \begin{itemize}
\item $p_1$
\item $p_2$
\item $K_{\text{MAX}}$
\item $K_{\text{MAX}} + 1$
\end{itemize}
Enumeration below $K_{\text{MAX}} + 1$. 

$K_{\text{MAX}}$
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

![Diagram showing a grid with axes labeled $p_1$, $p_2$, and $K_{\text{MAX}}$, with a point marked at a coordinate.](image)
Integer-valued U2P-TA

Enumeration below $K_{\text{MAX}} + 1$. 

\[ p_1 \]

\[ p_2 \]

\[ K_{\text{MAX}} \]
Integer-valued U2P-TA

Enumeration below $K_{MAX} + 1$. 

$p_2$

$p_1$

$K_{MAX}$

$K_{MAX}$
Integer-valued U2P-TA

Enumeration below $K_{MAX} + 1$. 

![Diagram with point $v$ at coordinates $(K_{MAX}, K_{MAX})$]
Conclusion

- Two new subclasses of PTAs: rational-valued U2P-TAs for which the EF-emptiness problem is undecidable, and integer-valued U2P-TAs for which it is decidable.
- In fact we have the same results for EF-universality, AF-emptiness/universality.
- We also can perform parameter synthesis.

Future work:
- Find syntactic restrictions in order to find a decidability result for rational parameter valuations
- Adapt our formalism to hybrid systems, in which clocks can evolve at different rates
Conclusion

► Two new subclasses of PTAs: rational-valued U2P-TAs for which the $EF$-emptiness problem is *undecidable*, and integer-valued U2P-TAs for which it is *decidable*.

► In fact we have the same results for $EF$-universality, $AF$-emptiness/universality.

► We also can perform *parameter synthesis*.

*Future work:*

► Find syntactic restrictions in order to find a decidability result for rational parameter valuations

► Adapt our formalism to hybrid systems, in which clocks can evolve at different rates
Rajeev Alur and David L. Dill.
A theory of timed automata.

Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi.
Parametric real-time reasoning.

Étienne André.
What’s decidable about parametric timed automata?

Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit.
Updatable timed automata.

Joseph S. Miller.
Decidability and complexity results for timed automata and semi-linear hybrid automata.

Rajeev Alur and David L. Dill.
A theory of timed automata.

Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi.
Parametric real-time reasoning.

Étienne André.
What’s decidable about parametric timed automata?

Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit.
Updatable timed automata.

Joseph S. Miller.
Decidability and complexity results for timed automata and semi-linear hybrid automata.
Clock regions

- The corner point: $R_1 = \{(4, 4)\}$
- The vertical line: $R_2 = \{(x, y) \mid x = 2, \, 0 < y < 1\}$
- The horizontal line: $R_3 = \{(x, y) \mid y = 3, \, 1 < x < 2\}$
- The diagonal: $R_4 = \{(x, y) \mid x = y - 3, \, 4 < y < 5\}$
- The upward triangle: $R_5 = \{(x, y) \mid 0 < x < y - 1, \, 1 < y < 2\}$
- The downward triangle: $R_6 = \{(x, y) \mid y + 1 < x < 4, \, 2 < y < 3\}$
Clock regions

Two clocks $x, y$, max constants $c_x = 2, c_y = 1$.
Time successors of the blue region
\[ \{0 < y < 1, 0 < y < x - 1\} \] different of itself: four regions in green:
\[ \{0 < y < 1, x = 2\}, \{0 < y < 1, x > 2\}, \{y = 1, x > 2\} \]
and \[ \{y > 1, x > 2\} \]
Using regions for parametric timed automata?

In $l_1$: $(x, y) = (0, p)$

But after letting some time elapse, depending on the value of $0 < p < 1$ we reach different regions:

- region $y = 1$, $0 < x < p$ if $1 > p > \frac{1}{2}$
Using regions for parametric timed automata?

In \( l_1 \): \((x, y) = (0, p)\)

But after letting some time elapse, depending on the value of \(0 < p < 1\) we access different regions:

- region \( y = 1, x = p \) if \( p = \frac{1}{2} \)
Using regions for parametric timed automata?

In $l_1$: $(x, y) = (0, p)$

But after letting some time elapse, depending on the value of $0 < p < 1$ we access different regions:

- region $p < y < 1$, $x = p$ if $p < \frac{1}{2}$
Timed automata with parametric updates

Introduction
Timed automata
Example of timed automaton
Common decision problems for timed automata
Parametric timed automata
Model checking with unknown constants
Challenges for parametric timed automata
Contributions
U2P-TA
Integer-valued U2P-TA
Conclusion
References
Timed automata with parametric updates

Introduction

Timed automata
Example of timed automaton
Common decision problems for timed automata

Parametric timed automata
Model checking with unknown constants
Challenges for parametric timed automata

Contributions
U2P-TA
Integer-valued U2P-TA

Conclusion

References
Timed automata
with parametric
updates

Introduction

Timed automata

Example of timed automaton

Common decision problems
for timed automata

Parametric timed
automata

Model checking with
unknown constants

Challenges for parametric
timed automata

Contributions

U2P-TA

Integer-valued U2P-TA

Conclusion

References
Example

\(x = 2\)
\(\text{comA}\)
\(x := 0\)

(a) Committee A

\(y = 3\)
\(\text{comB}\)
\(y := 0\)

(b) Committee B

(c) A PhD student’s defense workflow

Figure: A motivating example of integer-valued U2P-TA
Graphical visualization in two dimensions of the parameter synthesis of with $p_m = 6$ (left) and $p_m = 9$ (right) provided by IMITATOR. Constraints are:

$$p_A \leq 2 \land p_B \leq p_A + 1$$

$$\lor$$

$$p_B \geq 2 \land p_B \leq 3 \land p_B \geq p_A + 1$$

with $p_m = 6$

$$p_B \geq 2 \land p_A \leq 2 \land p_A + 1 \geq p_B$$

with $p_m = 9$