

# Liveness in L/U-Parametric Timed Automata

**Étienne André** and Didier Lime [[AL17](#)]

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

Highlights, 14 September 2017, London, England

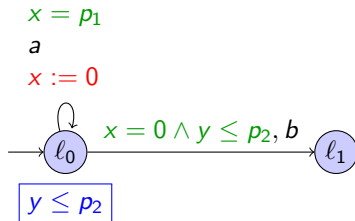
# Introduction

- ▶ Parametric timed automata (PTA) allow for flexible, abstract, and robust modelling;
- ▶ The answer to **parametric** model-checking is appealing;
- ▶ Many undecidability results exist for safety / reachability properties;
- ▶ And a few **decidable** subclasses:
  - ▶ L/U PTA [HRSV02];
  - ▶ IP-PTA [ALR16];
  - ▶ bounded integer PTA [JLR15].

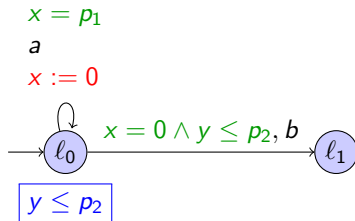
# Introduction

- ▶ Parametric timed automata (PTA) allow for flexible, abstract, and robust modelling;
- ▶ The answer to **parametric** model-checking is appealing;
- ▶ Many undecidability results exist for safety / reachability properties;
- ▶ And a few **decidable** subclasses:
  - ▶ L/U PTA [HRSV02];
  - ▶ IP-PTA [ALR16];
  - ▶ bounded integer PTA [JLR15].
- ▶ What about **liveness**?

# Parametric Timed Automata [AHV93]



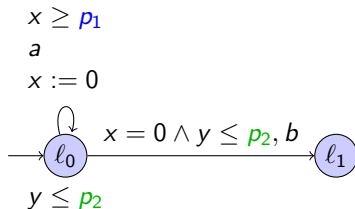
## Parametric Timed Automata [AHV93]



For  $p_1 = 1.2$  and  $p_2 = 4$ :

$$\begin{array}{ccccccc}
 l_0 & & l_0 & & l_0 & & l_1 & & l_1 \\
 x = 0 & \xrightarrow{1.2} & x = 1.2 & \xrightarrow{a} & x = 0 & \xrightarrow{b} & x = 0 & \xrightarrow{2.4} & x = 2.4 \\
 y = 0 & & y = 1.2 & & y = 1.2 & & y = 1.2 & & y = 3.6
 \end{array}$$

## L/U Parametric Timed Automata [HRSV02]



- ▶ Parameters are used either as **lower** bounds or as **upper** bounds, never both.
- ▶ **Monotonicity**: increasing upper bounds or decreasing lower bounds gives **more** behaviours.

# Liveness in (Parametric) Timed Automata

- ▶ Our **liveness** properties concern **maximal** paths:
  - ▶ Existence of an **infinite** maximal path (discrete **cycle**, denoted EC);
  - ▶ Existence of a **finite** maximal path (**deadlock**, denoted ED);
  - ▶ Existence of a maximal path preserving some property (CTL **EG** property).

# Liveness in (Parametric) Timed Automata

- ▶ Our **liveness** properties concern **maximal** paths:
  - ▶ Existence of an **infinite** maximal path (discrete **cycle**, denoted EC);
  - ▶ Existence of a **finite** maximal path (**deadlock**, denoted ED);
  - ▶ Existence of a maximal path preserving some property (CTL **EG** property).
- ▶ **Parametric** properties:
  - ▶  $\phi$ -emptiness: is the **set of parameter valuations** s.t.  $\phi$  holds **empty**?
  - ▶  $\phi$ -universality: is the set of parameter valuations s.t.  $\phi$  holds universal?



# Liveness in (Parametric) Timed Automata

- ▶ Our **liveness** properties concern **maximal** paths:
  - ▶ Existence of an **infinite** maximal path (discrete **cycle**, denoted EC);
  - ▶ Existence of a **finite** maximal path (**deadlock**, denoted ED);
  - ▶ Existence of a maximal path preserving some property (CTL **EG** property).
- ▶ **Parametric** properties:
  - ▶  $\phi$ -emptiness: is the **set of parameter valuations** s.t.  $\phi$  holds **empty**?
  - ▶  $\phi$ -universality: is the set of parameter valuations s.t.  $\phi$  holds universal?

# Results from the Literature

Class	PTA	L/U PTA
EC-emptiness	open	PSPACE-c. <sup>1</sup>
ED-emptiness	open	open
EG-emptiness	open	open

---

<sup>1</sup>Integer parameters [BL09].

## EC-emptiness is PSPACE-c for L/U PTAs

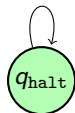
- ▶ There exists a **rational** parameter valuation s.t. there is a cycle iff there exists an **integer** valuation.
- ▶ Use the **monotonicity** property of L/U PTAs: **round** up for upper bounds, down for lower bounds to get a good **integer** valuation.

## EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;

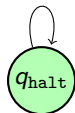
## EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;
- ▶ Halting:



## EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;
  
- ▶ Halting:



- ▶ There is a (discrete) cycle in the PTA iff the counter are bounded:
  - ▶ if the machine halts,  $q_{\text{halt}}$  is reachable  $\rightarrow$  cycle;
  - ▶ if the machine does not halt but the counters are bounded, there is a parameter valuation **small enough** to have a cycle among the instruction widgets;
  - ▶ if the counters are unbounded, for any valuation, the PTA will eventually **block** in the increment widget.

ED-emptiness is *undecidable* for L/U PTAs

- ▶ Reduce from the **halting** problem of 2-counter machines;

## ED-emptiness is *undecidable* for L/U PTAs

- ▶ Reduce from the **halting** problem of 2-counter machines;
- ▶ Change previous construction to “split” parameters and get an L/U PTA
  - ▶ Replace  $x \leq p$  with  $x \leq p^+$
  - ▶ Replace  $x \geq p$  with  $x \geq p^-$
  - ▶ Replace  $x = p$  with  $p^- \leq x \leq p^+$
- ▶ We use the deadlock property to **enforce**  $p^- = p^+$ .



# EG-emptiness is *undecidable* for L/U PTAs

- ▶ by reduction from the **halting** problem of 2-counter machines;
- ▶ similar to the ED-construction with a different encoding adapted from [BLS15];
- ▶ the main idea is to eliminate cycles by:
  - ▶ making sure all widgets execute in 1 t.u.;
  - ▶ add a global invariant limiting the **total execution time** so that it does not exceed some parameter  $p_2$ ;
  - ▶ then the PTA can only execute **at most**  $p_2$  instructions and  $p_2$  has to be **big enough** for executing a halting sequence.

# Results up to now

Class	PTA	L/U PTA
EC-emptiness	<i>Undec.</i>	<b>PSPACE-c.</b>
ED-emptiness	<i>Undec.</i>	<i>Undec.</i>
EG-emptiness	<i>Undec.</i>	<i>Undec.</i>

## Results up to now

Class	PTA	L/U PTA
EC-emptiness	<i>Undec.</i>	<b>PSPACE-c.</b>
ED-emptiness	<i>Undec.</i>	<i>Undec.</i>
EG-emptiness	<i>Undec.</i>	<i>Undec.</i>

- ▶ We can find some decidability by considering parameters are **bounded** (each takes its values in some bounded interval);
- ▶ Changes nothing for **PTAs**;
- ▶ We consider both (topologically) **closed** and **open** parameter domains.

# EG-emptiness is **decidable** for closed bounded L/U PTA

1. Test if there is an infinite path preserving  $\phi$  in the TA obtained by setting:
  - ▶ **lower** bounds to their **minimum** value,
  - ▶ and upper bounds to their maximal values.i.e. verify CTL property “EG ( $\phi \wedge$  EX true)” on the **region graph** of the TA.
2. if yes we are done

# EG-emptiness is **decidable** for closed bounded L/U PTA

1. Test if there is an infinite path preserving  $\phi$  in the TA obtained by setting:
  - ▶ **lower** bounds to their **minimum** value,
  - ▶ and upper bounds to their maximal values.
 i.e. verify CTL property “EG ( $\phi \wedge$  EX true)” on the **region graph** of the TA.
2. if yes we are done
3. otherwise all paths preserving  $\phi$  are finite: explore them symbolically, using the **symbolic polyhedral abstraction** of **linear hybrid automata**;
4. test all symbolic states on those paths for **deadlocks**:
  - ▶ consider all states that can reach some guard (classic **past** operator)
  - ▶ check if those states **cover** the whole symbolic state (polyhedral union and inclusion).

# EG-emptiness is *undecidable* for open bounded L/U PTA

- ▶ Reduce from the **halting** problem of **2-counter machines**
- ▶ Make sure all widgets execute in  $[p_2^-, p_2^+]$  t.u. (instead of 1);
- ▶ use the **open** parameter domain to enforce  $p_2^- > 0$ ;
- ▶ add a **global invariant** so that the whole PTA can only execute for 1 t.u. to **eliminate cycles**;
- ▶ the machine **halts** iff there exists a parameter valuation s.t.  $p_1^- = p_1^+$  and  $p_2^- = p_2^+$  and there is a **deadlock** in the PTA.

# Final Results

Class	PTA	L/U PTA	closed b. L/U	open b. L/U
EC-empt.	<i>Undec.</i>	<b>PSPACE-c.</b>	<b>PSPACE-c.</b>	open
ED-empt.	<i>Undec.</i>	<i>Undec.</i>	<i>Undec.</i>	<i>Undec.</i>
EG-empt.	<i>Undec.</i>	<i>Undec.</i>	<b>Dec.</b>	<i>Undec.</i>





- ▶ The other results follow directly from the previous constructions;
- ▶ We conjecture that EC-emptiness for open bounded L/U PTAs is **decidable** with techniques similar to [\[San11\]](#).

# Conclusion and Perspectives





- ▶ Summary:
  - ▶ We have exhibited a very thin border of decidability for **liveness** properties;
  - ▶ It depends on the **boundedness** of the parameters and the topological **closure** of their initial domain.
  
- ▶ Future work:
  - ▶ Prove that EC-emptiness for open bounded LU PTAs is **decidable**;
  - ▶ Complete the results for the **universality** problems;
  - ▶ Find the **complexity** of EG-emptiness for closed bounded L/U PTA.



# References I

-  Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi.  
Parametric real-time reasoning.  
In *STOC*, pages 592–601. ACM, 1993.
-  Étienne André and Didier Lime.  
Liveness in L/U-parametric timed automata.  
In *ACSD*, pages 9–18. IEEE, 2017.  
To appear.
-  Étienne André, Didier Lime, and Olivier H. Roux.  
Decision problems for parametric timed automata.  
In *ICFEM*, volume 10009 of *LNCS*, pages 400–416. Springer, 2016.
-  Nikola Beneš, Peter Bezděk, Kim G. Larsen, and Jiří Srba.  
Language emptiness of continuous-time parametric timed automata.  
In *ICALP, Part II*, volume 9135 of *LNCS*, pages 69–81. Springer, 2015.

## References II

-  Laura Bozzelli and Salvatore La Torre.  
Decision problems for lower/upper bound parametric timed automata.  
*Formal Methods in System Design*, 35(2):121–151, 2009.
-  Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager.  
Linear parametric model checking of timed automata.  
*JLAP*, 52-53:183–220, 2002.
-  Aleksandra Jovanović, Didier Lime, and Olivier H. Roux.  
Integer parameter synthesis for timed automata.  
*IEEE TSE*, 41(5):445–461, 2015.
-  Ocan Sankur.  
Untimed language preservation in timed systems.  
In *MFCS*, volume 6907 of *LNCS*, pages 556–567. Springer, 2011.

# EC-emptiness is PSPACE-c for L/U PTAs

- ▶ There exists a **rational** parameter valuation s.t. there is a cycle iff there exists an **integer** valuation.
- ▶ Use the **monotonicity** property of L/U PTAs: **round** up for upper bounds, down for lower bounds to get a good **integer** valuation.

# EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;

# EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;
- ▶ States of the machines are encoded by locations  $q_i$ ;
- ▶ Counters are encoded by clocks  $y, z$  and one parameter  $p$ : when clock  $x$  is null,

$$y = 1 - c_1 p$$

$$z = 1 - c_2 p$$

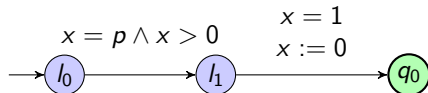
# EC-emptiness is *undecidable* for PTAs

- ▶ Reduce from the **counter boundedness** problem of **2-counter machines**
  - ▶ Finite-state machine + 2 non-negative integer counters;
  - ▶ **increment** some counter and go to some state;
  - ▶ **if** some counter is **zero** then **decrement** it and go to some state; otherwise go to some other state;
- ▶ States of the machines are encoded by locations  $q_i$ ;
- ▶ Counters are encoded by clocks  $y, z$  and one parameter  $p$ : when clock  $x$  is null,

$$y = 1 - c_1 p$$

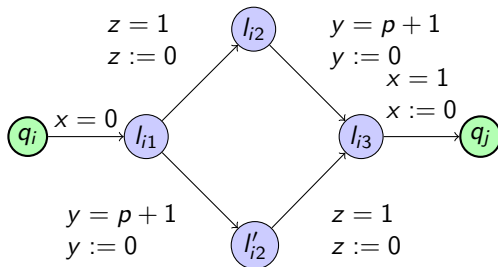
$$z = 1 - c_2 p$$

- ▶ Initialisation:



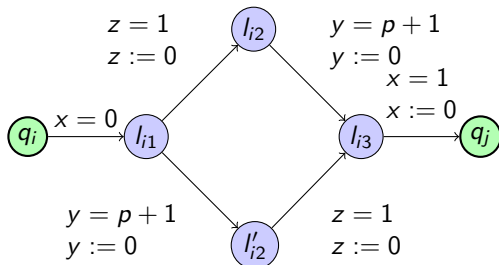
EC-emptiness is *undecidable* for PTAs

## ► Increment:



EC-emptiness is *undecidable* for PTAs

## ► Increment:

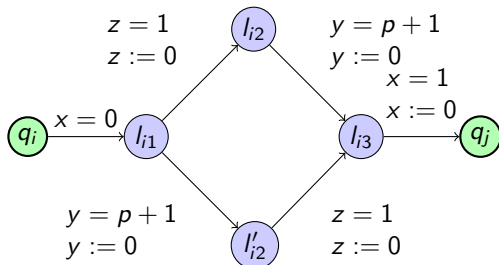


$$\begin{array}{ccc}
 q_i & l_{i1} & l_{i2} \\
 x = 0 & x = 0 & x = c_2 p \\
 y = 1 - c_1 p & y = 1 - c_1 p & y = 1 - (c_1 - c_2) p \\
 z = 1 - c_2 p & z = 1 - c_2 p & z = 0 \\
 & \xrightarrow{0} & \xrightarrow{c_2 p} \\
 & & \\
 & l_{i3} & q_j \\
 & x = (c_1 + 1) p & x = 0 \\
 & y = 0 & y = 1 - (c_1 + 1) p \\
 & z = (c_1 - c_2 + 1) p & z = 1 - c_2 p \\
 & & \xrightarrow{1 - (c_1 + 1) p}
 \end{array}$$



EC-emptiness is *undecidable* for PTAs

## ► Increment:

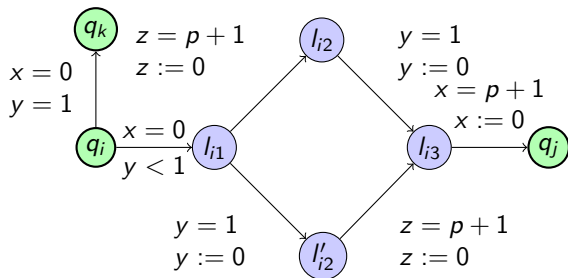


$$\begin{array}{ccc}
 q_i & l_{i1} & l_{i2} \\
 x = 0 & x = 0 & x = c_2 p \\
 y = 1 - c_1 p & y = 1 - c_1 p & y = 1 - (c_1 - c_2) p \\
 z = 1 - c_2 p & z = 1 - c_2 p & z = 0 \\
 & \xrightarrow{0} & \xrightarrow{c_2 p} \\
 & & \\
 l_{i3} & & q_j \\
 x = (c_1 + 1) p & \xrightarrow{1 - (c_1 + 1) p} & x = 0 \\
 y = 0 & & y = 1 - (c_1 + 1) p \\
 z = (c_1 - c_2 + 1) p & & z = 1 - c_2 p \\
 & & \xrightarrow{(c_1 - c_2 + 1) p}
 \end{array}$$

► implies  $p \leq \frac{1}{c_1 + 1}$  otherwise it **blocks** at  $l_{i3}$ .

# EC-emptiness is *undecidable* for PTAs

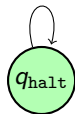
- ▶ Zero-test and decrement:



- ▶  $c_1 = 0$  iff  $y = 1$ .
- ▶ Decrement is similar to increment.

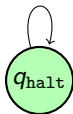
# EC-emptiness is *undecidable* for PTAs

- ▶ Halting:



# EC-emptiness is *undecidable* for PTAs

▶ Halting:



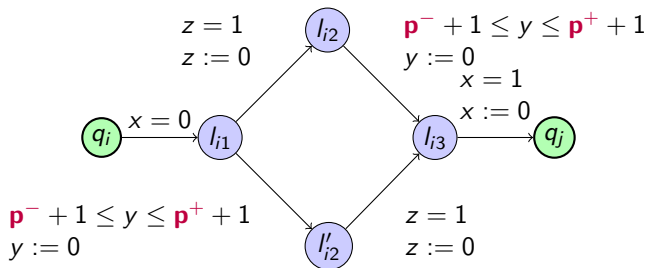
- ▶ There is a (discrete) cycle in the PTA iff the counter are bounded:
- ▶ if the machine halts,  $q_{\text{halt}}$  is reachable  $\rightarrow$  cycle;
  - ▶ if the machine does not halt but the counters are bounded, there is a parameter valuation **small enough** to have a cycle among the instruction widgets;
  - ▶ if the counters are unbounded, for any valuation, the PTA will eventually **block** in the increment widget.

## ED-emptiness is *undecidable* for L/U PTAs

- ▶ Reduce from the **halting** problem of 2-counter machines;

ED-emptiness is *undecidable* for L/U PTAs

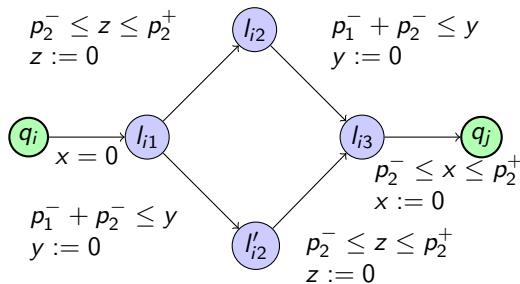
- ▶ Reduce from the **halting** problem of 2-counter machines;
- ▶ Change previous construction to “split” parameters and get an L/U PTA:



- ▶ We use the deadlock property to **enforce**  $p^- = p^+$ .

# EG-emptiness is *undecidable* for open bounded L/U PTA

- ▶ Reduce from the **halting** problem of **2-counter machines**
- ▶ Make sure all widgets execute in  $[p_2^-, p_2^+]$  t.u. (instead of 1);



- ▶ use the **open** parameter domain to enforce  $p_2^- > 0$ ;
- ▶ add a **global invariant** so that the whole PTA can only execute for 1 t.u. to **eliminate cycles**;
- ▶ the machine **halts** iff there exists a parameter valuation s.t.  $p_1^- = p_1^+$  and  $p_2^- = p_2^+$  and there is a **deadlock** in the PTA.