Parametric model checking timed automata under non-Zenoness assumption

Hoang Gia NGUYEN

Supervisors: Laure Petrucci and Étienne André

LIPN, Université Paris 13, Sorbonne Paris Cité, CNRS, France
Outline

1 Context
   - Parametric Verification of Real-Time Systems
   - Parametric Timed Automata (PTA)

2 Zenoness
   - Zenoness Introduction
   - Zenoness in Parametric Timed Model Checking

3 CUB-PTA
   - CUB-TA Introduction
   - CUB-PTA Introduction
   - CUB-PTA Detection
   - CUB-PTA Transformation
   - Non-Zenoness Parametric Model Checking

4 Implementation and Experiments

5 Conclusions
Outline

1 Context
   - Parametric Verification of Real-Time Systems
   - Parametric Timed Automata (PTA)

2 Zenoness
   - Zenoness Introduction
   - Zenoness in Parametric Timed Model Checking

3 CUB-PTA
   - CUB-TA Introduction
   - CUB-PTA Introduction
   - CUB-PTA Detection
   - CUB-PTA Transformation
   - Non-Zenoness Parametric Model Checking

4 Implementation and Experiments

5 Conclusions
Parametric Verification of Real-Time Systems

- Verification techniques used for critical systems, timed systems where changes of time value is vital! such as:
  
  1. Systems incompletely specified, some timing delays may not be known yet, or may change
  2. Verifying system for numerous values of constants requires a very long time, or even infinite

⇒ Use parameterised techniques, by using parameters instead of constants, then one can check many values at the same time
Parametric Timed Automata (PTA)

PTA is a formalism to model and verify concurrent real-time systems [Alur et al., 1993]

\( x \): Clock

\( p \): Parameters allow to represent unknown values (e.g. a transmission delay or a timeout)

\( K_0 \): Initial parameter constraint (e.g. \( p_1 \leq p_2 \) or \( p_1 > p_2 \)
Outline

1. Context
   - Parametric Verification of Real-Time Systems
   - Parametric Timed Automata (PTA)

2. Zenoness
   - Zenoness Introduction
   - Zenoness in Parametric Timed Model Checking

3. CUB-PTA
   - CUB-TA Introduction
   - CUB-PTA Introduction
   - CUB-PTA Detection
   - CUB-PTA Transformation
   - Non-Zenoness Parametric Model Checking

4. Implementation and Experiments

5. Conclusions
Zenoness Introduction

**Definition**

An infinite number of discrete actions in a finite time

Get a half way of $A \rightarrow B \left(\frac{1}{2}\right)$, then get half the remaining distance $\frac{1}{2} \rightarrow B \left(\frac{1}{4}\right)$, then again and again $\rightarrow$ never reach $B!$ ($A$ and $B$ can be the parameters).

$\Rightarrow$ Infeasible in reality!
Zenoness in parametric timed model checking

2 types of Zeno run (infinite):

1. Run has a clock such that time cannot elapse

2. Run has a clock bounded by a parameter or a constant

Existing an infinite run in a finite time is not feasible!
Zenoness in parametric timed model checking (cont.)

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Existing loop in product of Büchi automata (negated LTL formula), etc. Zeno run in counter-example is spurious</td>
</tr>
<tr>
<td>2. Zeno run cannot be checked directly on PTA model or its symbolic semantic!</td>
</tr>
</tbody>
</table>

⇒ Important to find and avoid Zeno loops in checking result!
Outline

1. Context
   - Parametric Verification of Real-Time Systems
   - Parametric Timed Automata (PTA)

2. Zenoness
   - Zenoness Introduction
   - Zenoness in Parametric Timed Model Checking

3. CUB-PTA
   - CUB-TA Introduction
   - CUB-PTA Introduction
   - CUB-PTA Detection
   - CUB-PTA Transformation
   - Non-Zenoness Parametric Model Checking

4. Implementation and Experiments

5. Conclusions
CUB-TA Introduction

CUB stands for "Clock Upper Bound", an approach derived from the paper [Wang et al., 2015] for solving the non-Zenoness problem on Timed Safety Automata (TA)

1. Zeno loops can be checked directly on CUB-TA's Zone Graph
2. More efficient than other current existing approaches
3. No need to introduce any new clock

⇒ We define a CUB approach for PTA
CUB-TA Introduction (cont.)

CUB-TA Definition

- A path is **non-decreasing upper bound** iff for each edge from location \( l \) to \( l' \) with guard \( g \), for each clock \( x \), the upper bound \( l_x \) is less than or equal to \( g_x \) and \( l'_x \) (if \( x \) is not reset)

- A TA \( A \) is a **CUB-TA**, iff every clock has a **non-decreasing upper bound** along any path before it is reset

---

**TA containing a non-decreasing upper bound path example**

**TA containing a decreasing upper bound path example**

---

CUB-TA Transformation examples
CUB-PTA Introduction

CUB-PTA Definition

A PTA $A$ is a **CUB-PTA**, iff there exists a constraint $A.K_0$ on parameters that guarantees every clock has a non-decreasing upper bound along any path before it is reset, for all parameter valuations satisfying the initial constraint $A.K_0$.

There are 2 cases:

$A.K_0 = p_1 \leq p_2$ : non-decreasing upper bound path! $\Rightarrow$ CUB-PTA

$A.K_0 = p_1 > p_2$ : decreasing upper bound path! $\Rightarrow$ not CUB-PTA

$\Rightarrow$ No transformation exists such that a CUB-PTA can cover all cases!

But a list of CUB-PTAs can
A disjunctive CUB-PTA is a list of CUB-PTAs.

With a CUB-PTA or disjunctive CUB-PTA, we can synthesize parameter valuations of non-Zeno runs on its symbolic semantic Parametric Zone Graph - PZG (similar to Zone Graph of TA and not always finite).
CUB-PTA Detection

CUB-PTA detection aims at non-Zenoness synthesizing a partial or complete result without modification on the given model.

\[ A.K_0 = p1 \leq p2 \land p1 \leq p1 \]
\[ \iff \text{CUB-PTA with } A.K_0 = p1 \leq p2 \]
(Partial result)
Missing result: \( A.K_0 = p1 > p2 \)

**Main idea**

Given PTA \( A \), for each clock \( x \) on each edge with guard \( g \) from location \( l \) to \( l' \) we enforce a constraint with upper bound \( l_x \) less than or equal to \( g_x \) and \( l'_x \) (if \( x \) is not reset). If a conjunction of all constraints \( A.K_0 \) contains some valuations then the given PTA is \( CUB-PTA \).
CUB-PTA Transformation

An arbitrary PTA can be transformed into a disjunctive CUB-PTA (with a new initial location), while preserving the symbolic runs.
CUB-PTA Transformation (cont.)

Main idea

Given a PTA $A$:

1. Infer all possible parameter relations $A.K_0$s
2. Each copy of $A$ will be transformed due to each $A.K_0$ by:
   1. Splitting the location* into new locations with different upper bounds
   2. Copying all incoming and outgoing edges of old location to the new location
   3. Removing all decreasing upper bound edges
3. Add a new initial location connecting to all initial locations of the copies of $A$

*location*: a location containing an outgoing edge implies a decreasing upper bound
CUB-PTA Transformation Example

\[ l_1 \]

Graphical representation:

- Node 1 labeled \( l_1 \)
- Edges:
  - \( x \leq p_1 \)
  - \( x \leq p_2 \)

Infer all possible bounds on the fly. For each \( A.K_0 \), split the location into different upper bounds.

Copy all incoming and outgoing edges of old location to new location.

Remove all decreasing upper bound edges.
CUB-PTA Transformation Example

Infer all possible $A.K_0$s on the fly. For each $A.K_0$, split the location into different upper bounds.
CUB-PTA Transformation Example

Infer all possible \( A.K_0 \)s on the fly.
For each \( A.K_0 \), split the location* into different upper bounds

Copy all incoming and outgoing edges of old location to new location

Remove all decreasing upper bound edges
Non-Zenoness Parametric Model Checking

With CUB-PTA Parametric Non-Zenoness can be checked directly on the Parametric Zone Graph - PZG

Main idea

A CUB-PTA $\mathcal{A}$ contains a non-Zeno run iff:

1. There exists parameter valuation such that $PZG(\mathcal{A})$ has a SCC containing an edge from location $l$ to $l'$ where time can elapse

2. For every clock $x$ in $\mathcal{A}$, if $x$ is bounded by a constant or a parameter for some location in the SCC, there exists an edge in the SCC where $x$ is reset

SCC: Strongly Connected Component
Non-Zenoness Parametric Model Checking (cont.)

CUB-PTA

(Definition)

Emptiness non-Zenoness check: False!
Approximation: Under-approximation
(no result is given for \( p_1 > p_2 \))

PZG of the CUB-PTA

\[ \forall x \leq p_1 \land p_2 \geq 0 \land p_1 \geq 0 \land p_2 \geq p_1 \land x \geq 0 \]

Time elapsed

Zeno run!
Non-Zenoness Parametric Model-Checking (cont.)

Disjunctive CUB-PTA

Emptiness non-Zeno check:
False!
Approximation:
Exact!

PZG of the disjunctive CUB-PTA
Outline

1 Context
2 Zenoness
3 CUB-PTA
4 Implementation and Experiments
5 Conclusions
Implementation in **IMITATOR** [André, Fribourg, Kühne, Soulat, 2012]

- About **3,000** lines of new **OCaml** code for our non-Zenoness parameter synthesis algorithm
- Thank to the **Parma Polyhedra Library (PPL)** library for solving linear inequality systems
### Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>Name</th>
<th>synthCycle</th>
<th>CUBdetect</th>
<th>CUBtrans</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time (s)</td>
<td>Result</td>
<td>Appr.</td>
<td>time (s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Result</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Appr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CUB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>time (s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>time (s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Result</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>3 / 3 / 28</td>
<td>TO</td>
<td>invalid</td>
<td>0.013</td>
</tr>
<tr>
<td>Fischer</td>
<td>2 / 4 / 13</td>
<td>TO</td>
<td>invalid</td>
<td>0.003</td>
</tr>
<tr>
<td>RCP</td>
<td>6 / 5 / 48</td>
<td>TO</td>
<td>Some</td>
<td>0.013</td>
</tr>
<tr>
<td>WFAS</td>
<td>4 / 2 / 10</td>
<td>TO</td>
<td>Some</td>
<td>0.009</td>
</tr>
<tr>
<td>AndOr</td>
<td>4 / 4 / 27</td>
<td>TO</td>
<td>Some</td>
<td>0.012</td>
</tr>
<tr>
<td>Flip-flop</td>
<td>5 / 2 / 52</td>
<td>0.058</td>
<td>exact</td>
<td>0.002</td>
</tr>
<tr>
<td>Sched5</td>
<td>21 / 2 / 153</td>
<td>190</td>
<td>exact</td>
<td>0.051</td>
</tr>
<tr>
<td>simop</td>
<td>8 / 2 / 46</td>
<td>TO</td>
<td>invalid</td>
<td>0.012</td>
</tr>
<tr>
<td>coffee</td>
<td>2 / 3 / 4</td>
<td>TO</td>
<td>Some</td>
<td>0.000</td>
</tr>
<tr>
<td>CUBPTA1</td>
<td>1 / 3 / 2</td>
<td>0.006</td>
<td>invalid</td>
<td>0.000</td>
</tr>
<tr>
<td>JLR13</td>
<td>2 / 2 / 2</td>
<td>TO</td>
<td>invalid</td>
<td>0.000</td>
</tr>
</tbody>
</table>

- **synthCycle** (without non-Zenoness assumption): Synthesizes all parameter valuations of loops
- **CUBdetect**: Detects a given PTA is CUB-PTA then synthesizes parameter valuations of non-Zeno runs
- **CUBtrans**: Transforms a given PTA into CUB-PTA then synthesizes parameter valuations of non-Zeno runs
Outline

1 Context
2 Zenoness
3 CUB-PTA
4 Implementation and Experiments
5 Conclusions
Conclusions

Contributions:

- Proposed and implemented new Zeno-free parametric model synthesizing approaches in IMITATOR [André, Fribourg, Kühne, Soulat, 2012] tool
- Gave an overall view of our algorithms’ performance and complexity, a set of case studies for non-Zenoness parametric model checking study

Paper submitted:

- Étienne André, Hoang Gia Nguyen, Laure Petrucci, Jun Sun Parametric model checking timed automata under non-Zenoness assumption

Future work:

- Implement other techniques such as yet to be defined parametric extensions of strongly non-Zeno TAs [Tripakis et al., 2005] or guessing zone graph [Herbreteau et al., 2012] could turn to be more efficient and should be investigated
Bibliography
References I


Licensing
Source of the graphics used I

Title: Ocaml logo  
Author: Amir Chaudhry  
Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg  
License: CC BY-SA 4.0

Title: IMITATOR logo (Typing Monkey)  
Author: Kater Begemot  
Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg  
License: CC BY-SA 3.0

Title: PPL logo  
Author: Unknown  
Source: http://bugseng.com/files/ext/images/site/ppl_mm_8.png  
License: GCC