Efficient Parameter Synthesis Using Optimized State Exploration Strategies

Hoang Gia NGUYEN

Joint work with: Étienne André, Laure Petrucci

LIPN, Université Paris 13, CNRS, France
Outline

1. Context
2. Parametric Zone Inclusion
3. Exploration Orders for Parametric Zone Inclusion
4. Implementation and Experiments
5. Conclusions
Outline

1 Context

2 Parametric Zone Inclusion

3 Exploration Orders for Parametric Zone Inclusion

4 Implementation and Experiments

5 Conclusions
Parametric Verification of Real-Time Systems

- Verification techniques used for **critical systems, timed systems** where a failure or a too late answer can lead to dramatic consequences! 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, but also **synthesize** good valuations of these timing constants
Parametric Timed Automata (PTA)

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

Invariant: \( x < 5 \)
Guard: \( x > 1 \)
Reset: \( x := 0 \)

Invariant: \( x > 1 \)
Guard: True
Reset: \( x := 0 \)

Timed Automata-TA

PTA

\( x \): Clock

\( p_1 / p_2 \): Parameters allow to represent unknown values
Parametric Timed Automata (PTA)

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

PTA

Invariant: $x < p_1$

Guard: $x > p_2$

Reset: $x := 0$

System Behaviour depends on the values of parameters
Parametric Zone Graph (PZG)

A PTA example

Example: a part of a parameterized version of the FDDI case study of [Herbreteau and Tran, 2015]
Parametric Zone Graph (PZG)

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint)
- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters
**Parametric Zone Graph (PZG)**

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint).
- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters.

**A PTA example**

- $l_0$: $y > 2p_1$
- $l_1$: True
- $l_2$: $y \leq p_2$
- $l_3$: $y \leq p_2$

**Parametric Zone Graph - PZG**

- $s_0$: $l_0$ $s_1$: $l_2$
- $s_0$: $y > 2p_1$
- $s_1$: $y \leq p_2$
**Parametric Zone Graph (PZG)**

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint).
- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters.

A PTA example

- \( l_0 \)
  - \( y > 2p_1 \)

- \( l_1 \)
- \( l_2 \)
  - \( y \leq p_2 \)

- \( l_3 \)

Parametric Zone Graph - PZG

- \( s_0 \)
  - \( l_0 \)
  - \( True \)

- \( s_1 \)
  - \( l_2 \)
  - \( y > 2p_1 \)

- \( s_2 \)
  - \( l_1 \)
  - \( True \)
Parametric Zone Graph (PZG)

Symbolic state: a symbolic state is a pair made of a location, and an attached parametric zone (constraint)

Parametric zone: is a set of valuations defined by conjunctions of constraints on clocks and parameters
Parametric Zone Graph (PZG)

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint)

- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters

A PTA example

Parametric Zone Graph - PZG
Parametric Zone Graph (PZG)

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint)

- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters

A PTA example

Parametric Zone Graph - PZG

\[
\begin{align*}
&y > 2p_1 \\
y \leq p_2
\end{align*}
\]
**Parametric Zone Graph (PZG)**

A PTA example

- **Symbolic state**: a symbolic state is a pair made of a location, and an attached parametric zone (constraint)
- **Parametric zone**: is a set of valuations defined by conjunctions of constraints on clocks and parameters
Outline

1 Context

2 Parametric Zone Inclusion

3 Exploration Orders for Parametric Zone Inclusion

4 Implementation and Experiments

5 Conclusions
Objective

- **Problem**: the order in which we select the states has a huge impact on the efficiency
- **Goal of this work**: perform reachability synthesis, i.e., find valuations for which a given location is reachable; to do this, we use the parametric zone graph
  - Find efficient exploration order strategies
Objective (cont.)

2 popular exploration orders for model checking algorithms

1. Depth-first search - DFS
2. Breadth-first search - BFS

Many authors (e.g., [Behrmann et al., 2000, Behrmann, 2005]) showed that using BFS is much more efficient than DFS for checking reachability properties in TAs

⇒ modify and optimize the breadth-first search (BFS)
**Parametric Zone Inclusion Illustration**

- **Parametric zone inclusion**: is an optimization technique relying on the parametric zone graph to speed up the parametric model checking.

A PTA example

Without parametric zone inclusion

With parametric zone inclusion
**Parametric Zone Inclusion Illustration**

A PTA example

- Without parametric zone inclusion
- With parametric zone inclusion

- **Parametric zone inclusion**: is an optimization technique relying on the parametric zone graph to speed up the parametric model checking
**Parametric Zone Inclusion Illustration**

A PTA example

**Without parametric zone inclusion**

- $s_0 \xrightarrow{l_0} True$
- $s_1 \xrightarrow{l_2} y > 2p_1$
- $s_2 \xrightarrow{l_1} True$

**With parametric zone inclusion**

- $s_0 \xrightarrow{l_0} True$
- $s_1 \xrightarrow{l_2} y > 2p_1$
- $s_2 \xrightarrow{l_1} True$

**Parametric zone inclusion**: is an optimization technique relying on the parametric zone graph to speed up the parametric model checking.
Parametric Zone Inclusion: is an optimization technique relying on the parametric zone graph to speed up the parametric model checking
Parametric Zone Inclusion: is an optimization technique relying on the parametric zone graph to speed up the parametric model checking.

A PTA example

Without parametric zone inclusion

With parametric zone inclusion
### Parametric Zone Inclusion Illustration

A PTA example:

- States:
  - $s_0$: $l_0$, True
  - $s_1$: $l_2$, $y > 2p_1$
  - $s_2$: $l_2$, True
  - $s_3$: $l_3$, $2p_1 < y \leq p_2$
  - $s_4$: $l_2$, True

#### Without parametric zone inclusion

- Transition:
  - $s_0 \rightarrow s_1$ (y > 2$p_1$)
  - $s_1 \rightarrow s_3$ ($2p_1 < y \leq p_2$)
  - $s_2 \rightarrow s_4$ (True)

#### With parametric zone inclusion

- Transition:
  - $s_0 \rightarrow s_1$ (y > 2$p_1$)
  - $s_1 \rightarrow s_3$ ($2p_1 < y \leq p_2$)
  - $s_2 \rightarrow s_4$ (True)

---

**Parametric zone including**: given two reachable states $s_1 = (l_1, C_1)$ and $s_2 = (l_2, C_2)$, whenever $l_1 = l_2$ and $C_1 \subseteq C_2$, it is safe to replace $s_1$ with $s_2$ in the analysis.
Parametric zone including: given two reachable states $s_1 = (l_1, C_1)$ and $s_2 = (l_2, C_2)$, whenever $l_1 = l_2$ and $C_1 \subseteq C_2$, it is safe to replace $s_1$ with $s_2$ in the analysis.
Parametric zone inclusion: given two reachable states $s_1 = (l_1, C_1)$ and $s_2 = (l_2, C_2)$, whenever $l_1 = l_2$ and $C_1 \subseteq C_2$, it is safe to replace $s_1$ with $s_2$ in the analysis.
Parametric zone including: given two reachable states $s_1 = (l_1, C_1)$ and $s_2 = (l_2, C_2)$, whenever $l_1 = l_2$ and $C_1 \subseteq C_2$, it is safe to replace $s_1$ with $s_2$ in the analysis.
Problem: inefficient phenomenon happen is when a larger zone is explored after exploring smaller zones (red states)
Parametric Zone Inclusion Illustration

Without parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 6

With parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 4

Ideal exploration order

Question: how to reduce inefficient phenomenon or useless computation?

→ Find an exploration order to explore the biggest zone first!
Parametric Zone Inclusion Illustration

Without parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 6

With parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 4

Ideal exploration order

Question: how to reduce inefficient phenomenon or useless computation?

→ Find an exploration order to explore the biggest zone first!
Parametric Zone Inclusion Illustration

Without parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 6

With parametric zone inclusion
Order: $s_0 \rightarrow s_5$ States: 4

Ideal exploration order

Question: how to reduce inefficient phenomenon or useless computation?

→ Find an exploration order to explore the biggest zone first!
**Parametric Zone Inclusion Illustration**

Without parametric zone inclusion
Order: \( s_0 \rightarrow s_5 \) States: 6

With parametric zone inclusion
Order: \( s_0 \rightarrow s_5 \) States: 4

Ideal exploration order
Order: \( s_0 \rightarrow s_3 \) States: 4

- **Question:** how to reduce inefficient phenomenon or useless computation?
- \( \rightarrow \) Find an exploration order to explore the biggest zone first!
Outline

1. Context

2. Parametric Zone Inclusion

3. Exploration Orders for Parametric Zone Inclusion

4. Implementation and Experiments

5. Conclusions
Our contribution: 2 new exploration orders for PTAs

1. Parametric Ranking Strategy
   - This strategy assigns a priority value to each state, then it explores the state with highest priority first
   - Inspired by the “ranking system” strategy [Herbreteau and Tran, 2015].

2. Parametric Priority Strategy
   - A new strategy using an insertion mechanism within an ordered list of parametric zones
Parametric Ranking Strategy

The main idea:

- Explore the state having the highest rank

Ranking:

1. A new explored state starts with rank $\text{infinity}$ (if its constraint is $True$) or zero (otherwise)
2. The rank of the larger parametric zone is set higher than the highest rank of the small parametric zone and those in its subtree (with the same location)
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy
Parametric Ranking Strategy

A PTA example

PZG with parametric ranking strategy

\[ y > 2p_1 \]

\[ y \leq p_2 \]

\[ l_0 \]

\[ l_1 \]

\[ l_2 \]

\[ l_3 \]

\[ s_0 \]

\[ s_2 \]

\[ s_3 \]

\[ s_4 \]

True

True

True

True

rank: $\infty$

rank: $0$

rank: $\infty$

rank: $\infty$

True

y \leq p_2

\[ y \leq p_2 \]
### Drawback of Parametric Ranking Strategy

There is no likely improvement if there are no *True* zones in a model, compared to using the BFS exploration order.

- **A PTA example**
  - $y > 2p_1$
  - $y \leq p_2$
  - $l_3$

- **Our PTA example**
  - $y > p_1$
  - $y > 2p_1$
  - $y \leq p_2$
  - $l_3$
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!

A PTA example

Parametric ranking strategy
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with **zero rank**
- **Inefficient phenomenon detected late!**
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Drawback of Parametric Ranking Strategy

- Different zone sizes are assigned with zero rank
- Inefficient phenomenon detected late!
Parametric Priority Strategy

The main idea:

- A new explored state is inserted into an ordered waiting list $\mathcal{W}$ by ascending zone size, then the state at the head of the list will be explored first

- The waiting list $\mathcal{W}$ structure:

  1. Two main parts in $\mathcal{W}$
     1. The first (at the head) is the true zones part
     2. The other is the non-true zone part composed of several parts each containing ordered comparable zones

<table>
<thead>
<tr>
<th>True</th>
<th>True</th>
<th>...</th>
<th>$x &lt; 2p_1$</th>
<th>$x &lt; p_1$</th>
<th>...</th>
<th>$x \geq p_2$</th>
<th>$x \geq 2p_2$</th>
</tr>
</thead>
</table>

True zones | non-True zones 1 | non-True zones n
Parametric Priority Strategy

\[ y > p_1 \]
\[ y > 2p_1 \]
\[ y \leq p_2 \]

Our PTA example

\[ \mathcal{W}: s_0 \]

Parametric priority strategy
Parametric Priority Strategy

Our PTA example

Parametric priority strategy
Parametric Priority Strategy

Our PTA example

Parametric priority strategy
Parametric Priority Strategy

Our PTA example

Parametric priority strategy
Parametric Priority Strategy

Our PTA example

\[ y > 2p_1 \]

\[ y \leq p_2 \]

\[ y > p_1 \]

\[ y > 2p_1 \]

Parametric priority strategy

\[ \mathcal{W}: s_3, s_1 \]
Parametric Priority Strategy

Our PTA example

Parametric priority strategy
Parametric Priority Strategy

Our PTA example

Parametric priority strategy

\[ y > 2p_1 \]

\[ y > p_1 \]

\[ y \leq p_2 \]

\[ y > p_1 \]

\[ p_1 < y \leq p_2 \]
Strategies Comparison

Parametric ranking strategy

⇒ Parametric priority strategy has less inefficient phenomenon
Outline

1. Context
2. Parametric Zone Inclusion
3. Exploration Orders for Parametric Zone Inclusion
4. Implementation and Experiments
5. Conclusions
Implementation

- Implementation in IMITATOR [André, Fribourg, Kühne, Soulat, 2012]
  - A software tool for parametric verification and robustness analysis of real-time systems
  - Thanks to the Parma Polyhedra Library (PPL) library for solving linear inequality systems

1http://www.imitator.fr/
Experiments

Search orders:

- **BFS**: Traditional *breadth-first search*
- **LayerBFS**: *Layer breadth-first search* is an extension of breadth-first search BFS, which explores states layer by layer (same depth in the parametric zone graph")
- **RS**: BFS with *Parametric ranking strategy*
- **PRIOR**: BFS with *Parametric priority strategy*

Semi-algorithms for reachability synthesis:

- **EFsynth** (*exact synthesis*): *EF-synthesis problem*, “find all parameter valuations for which a given location is reachable”
- **EFc-ex** (*partial synthesis*): *EF-counter-example synthesis problem*, “find at least some parameter valuations for which a given location is reachable, and stop as soon as some valuations are found”
# Experiments for Exact Synthesis: EFsynth

<table>
<thead>
<tr>
<th>Benchmark Models</th>
<th>Existing Search Orders</th>
<th>Our Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LayerBFS incl (s)</td>
<td>BFS incl (s)</td>
</tr>
<tr>
<td>AndOr</td>
<td>2.512</td>
<td>2.41</td>
</tr>
<tr>
<td>flipflop-P</td>
<td>121.108</td>
<td>102.42</td>
</tr>
<tr>
<td>BRP</td>
<td>377.913</td>
<td>370.74</td>
</tr>
<tr>
<td>Thales-3</td>
<td>627.956</td>
<td>759.987</td>
</tr>
<tr>
<td>Sched2.100.2</td>
<td>148.169</td>
<td>T.O</td>
</tr>
<tr>
<td>Sched2.50.2</td>
<td>28.137</td>
<td>217.399</td>
</tr>
<tr>
<td>FDDI-4</td>
<td>1.315</td>
<td>1.1</td>
</tr>
<tr>
<td>Fischer-3</td>
<td>0.521</td>
<td>0.48</td>
</tr>
<tr>
<td>Lynch-5</td>
<td>7.359</td>
<td>7.817</td>
</tr>
<tr>
<td>F4</td>
<td>21.813</td>
<td>37.558</td>
</tr>
<tr>
<td>Pipeline-KP12-3-3</td>
<td>T.O</td>
<td>T.O</td>
</tr>
<tr>
<td>RCP</td>
<td>1.105</td>
<td>1.099</td>
</tr>
<tr>
<td>critical-region-4</td>
<td>T.O</td>
<td>T.O</td>
</tr>
<tr>
<td>blowup</td>
<td>31.635</td>
<td>1.345</td>
</tr>
<tr>
<td>Normalized Average</td>
<td>3.47236</td>
<td>3.7417</td>
</tr>
</tbody>
</table>

- **RS** and **PRIOR** are slightly faster in EFsynth

Additional experiments with merging and bidirectional inclusion: see paper
# Experiments for Partial Synthesis \text{EFc-ex}

<table>
<thead>
<tr>
<th>Benchmark Models</th>
<th>Existing Search Orders</th>
<th>Our Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Layer</td>
<td>BFS incl (s)</td>
</tr>
<tr>
<td>AndOr</td>
<td>0.012</td>
<td>0.011</td>
</tr>
<tr>
<td>flipflop-P</td>
<td>0.061</td>
<td>0.059</td>
</tr>
<tr>
<td>BRP</td>
<td>2.874</td>
<td>2.944</td>
</tr>
<tr>
<td>Thales-3</td>
<td>16.638</td>
<td>19.968</td>
</tr>
<tr>
<td>Sched2.100.2</td>
<td>0.008</td>
<td>0.004</td>
</tr>
<tr>
<td>Sched2.50.2</td>
<td>0.028</td>
<td>0.023</td>
</tr>
<tr>
<td>FDDI-4</td>
<td>0.377</td>
<td>0.291</td>
</tr>
<tr>
<td>Fischer-3</td>
<td>0.097</td>
<td>0.097</td>
</tr>
<tr>
<td>Lynch-5</td>
<td>7.408</td>
<td>7.912</td>
</tr>
<tr>
<td>F4</td>
<td>4.086</td>
<td>6.543</td>
</tr>
<tr>
<td>Pipeline-KP12-3-3</td>
<td>21.927</td>
<td>18.229</td>
</tr>
<tr>
<td>RCP</td>
<td>0.51</td>
<td>0.454</td>
</tr>
<tr>
<td>spsmall</td>
<td>5.862</td>
<td>6.242</td>
</tr>
<tr>
<td>critical-region-4</td>
<td>1.008</td>
<td>0.821</td>
</tr>
<tr>
<td>blowup</td>
<td>32.893</td>
<td>1.346</td>
</tr>
<tr>
<td>Normalized Average</td>
<td>6.03173</td>
<td>5.28516</td>
</tr>
</tbody>
</table>

- : best time
- : 2nd best time
- T.O: time out (3600s)

- \text{RS} and \text{PRIOR} dominate other algorithms in \text{EFc-ex}

Additional experiments with merging and bidirectional inclusion: see paper
Experiment Summary

- RS and PRIOR are better in general

Additional experiments with merging and bidirectional inclusion: see paper
Outline

1. Context
2. Parametric Zone Inclusion
3. Exploration Orders for Parametric Zone Inclusion
4. Implementation and Experiments
5. Conclusions
Conclusions

Contributions:

- Proposed two new exploration order strategies for the parameter synthesis problems
- Implemented and evaluated in IMITATOR
- Give an overview of the impact of exploration orders in different parameter synthesis problems.

Future work:

- The waiting strategy of [Herbreteau and Tran, 2015] and the exact acceleration technique [Hendriks and Larsen, 2002] could serve as a basis for future parametric strategies
- Taking advantage of recent multi-core technology for DFS, by adapting the non-parametric algorithm of [Laarman et al., 2013]
Bibliography
Parametric real-time reasoning.
In *STOC*, pages 592–601. ACM.

IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.
In *FM*, volume 7436 of *Lecture Notes in Computer Science*. Springer.

Distributed reachability analysis in timed automata.

Distributing timed model checking – how the search order matters.

Exact acceleration of real-time model checking.

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