ICECCS 2017

November 6th, 2017
Fukuoka, Japan

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

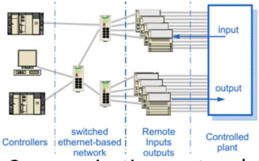
3 Exploration Orders for Parametric Zone Inclusion

4 Implementation and Experiments

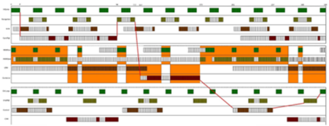
5 Conclusions

Parametric Verification of Real-Time Systems

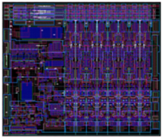
- Verification techniques used for critical systems, timed systems where a failure or a too late answer can lead to dramatic consequences! such as:



Communication protocols



Processor Scheduling



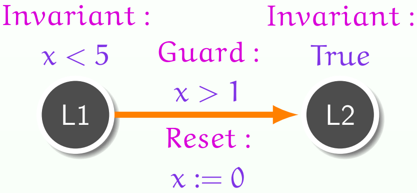
Asynchronous Circuits

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

$\Rightarrow$ 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 : $x > 1$

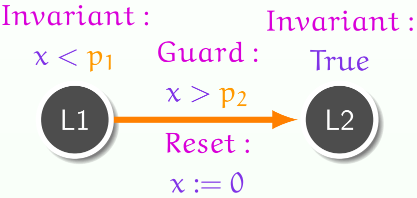
Invariant : True

L1

L2

Reset : $x := 0$

Timed Automata-TA



Invariant : $x < p_1$

Guard : $x > p_2$

Invariant : True

L1

L2

Reset : $x := 0$

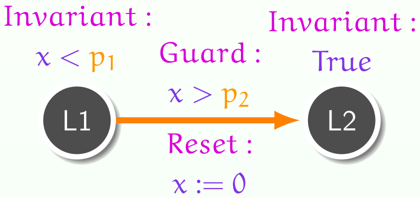
PTA

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]



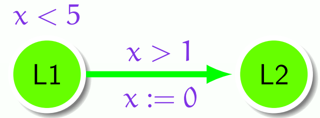
Invariant : $x < p_1$

Guard : $x > p_2$

Invariant : True

Reset : $x := 0$

PTA

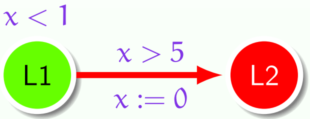


$x < 5$

$x > 1$

$x := 0$

With $p_1 = 5$ and $p_2 = 1$



$x < 1$

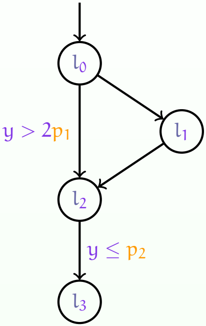
$x > 5$

$x := 0$

With $p_1 = 1$ and $p_2 = 5$

System Behaviour depends on the values of parameters

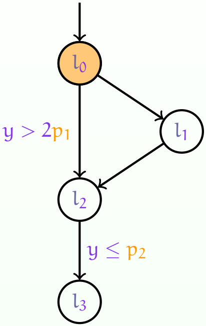
Parametric Zone Graph (PZG)



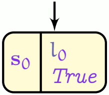
A PTA example

Example: a part of a parameterized version of the FDDI case study of [Herbreteau and Tran, 2015]

Parametric Zone Graph (PZG)



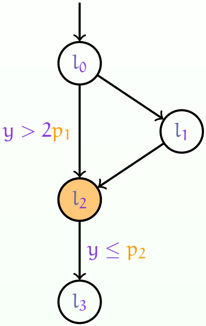
A PTA example



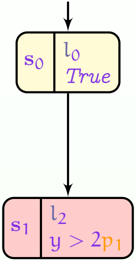
Parametric Zone Graph - PZG

- 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)



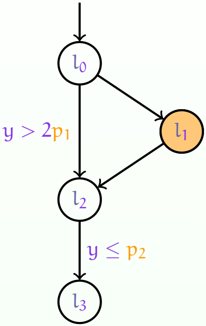
A PTA example



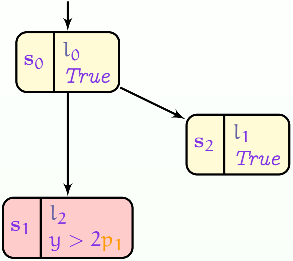
Parametric Zone Graph - PZG

- 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)



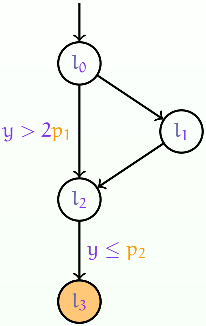
A PTA example



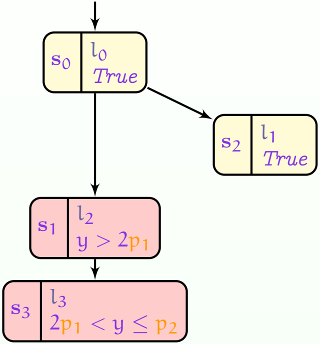
Parametric Zone Graph - PZG

- 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)



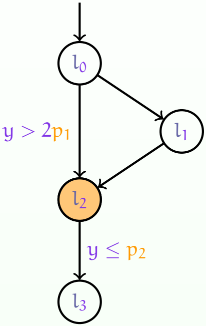
A PTA example



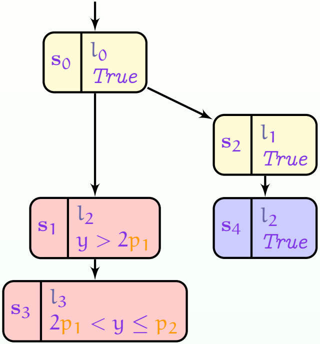
Parametric Zone Graph - PZG

- 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)



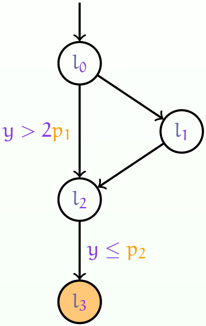
A PTA example



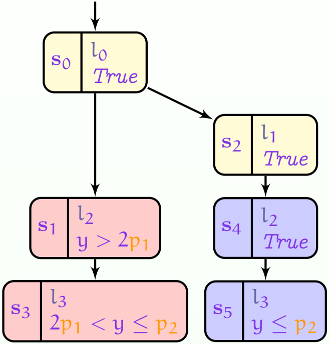
Parametric Zone Graph - PZG

- 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)



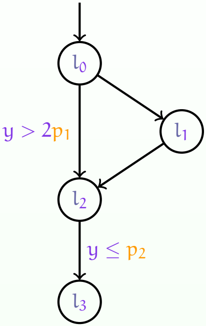
A PTA example



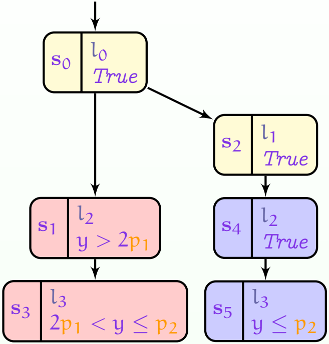
Parametric Zone Graph - PZG

- 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)



A PTA example



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

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.)

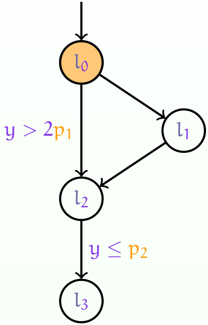
2 popular exploration orders for model checking algorithms

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

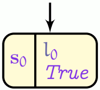
- Many authors (e. g., [Behrmann et al., 2000, Behrmann, 2005]) showed that using BFS is much more efficient than DFS for checking reachability properties in TAs

⇒ modify and optimize the breadth-first search (BFS)

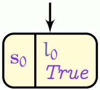
Parametric Zone Inclusion Illustration



A PTA example



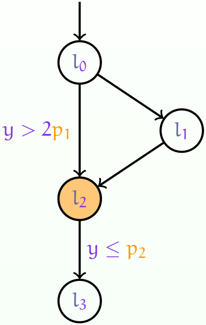
Without parametric zone inclusion



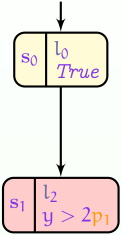
With parametric zone inclusion

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

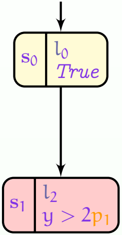
Parametric Zone Inclusion Illustration



A PTA example



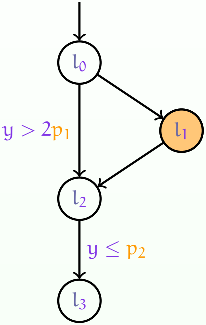
Without parametric zone inclusion



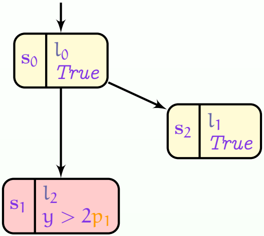
With parametric zone inclusion

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

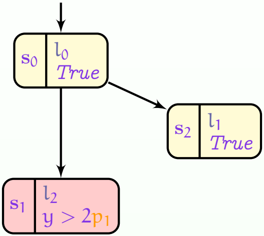
Parametric Zone Inclusion Illustration



A PTA example



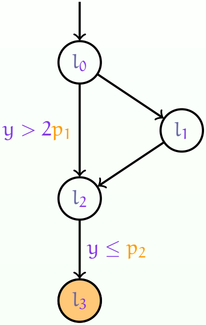
Without parametric zone inclusion



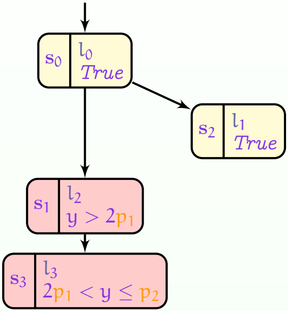
With parametric zone inclusion

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

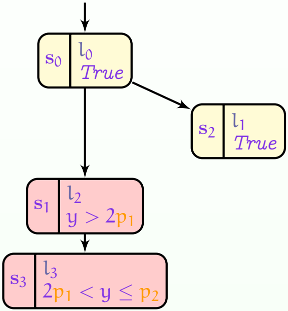
Parametric Zone Inclusion Illustration



A PTA example



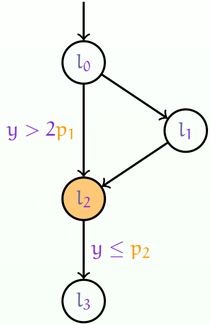
Without parametric zone inclusion



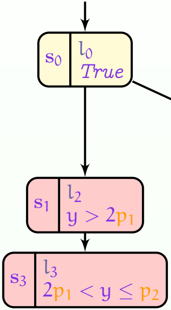
With parametric zone inclusion

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

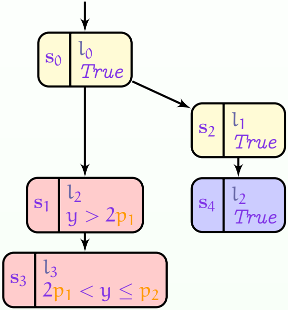
Parametric Zone Inclusion Illustration



A PTA example



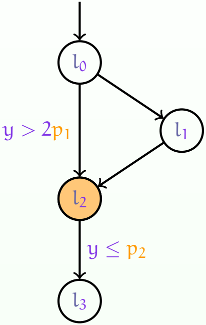
Without parametric zone inclusion



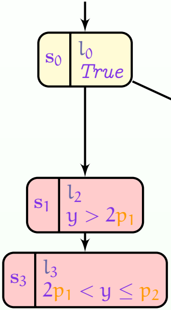
With parametric zone inclusion

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

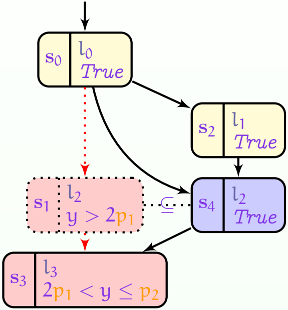
Parametric Zone Inclusion Illustration



A PTA example



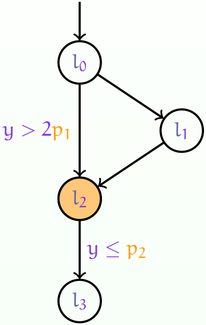
Without parametric zone inclusion



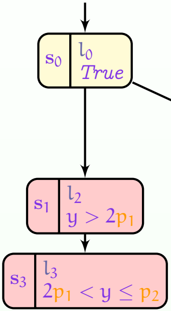
With parametric zone inclusion

- 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

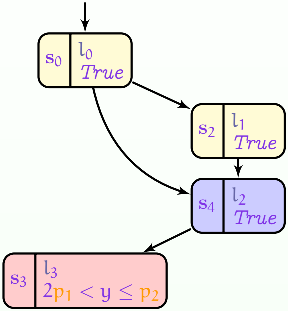
Parametric Zone Inclusion Illustration



A PTA example



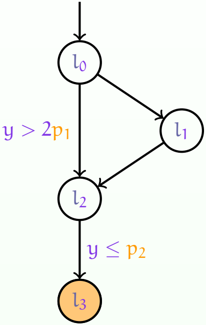
Without parametric zone inclusion



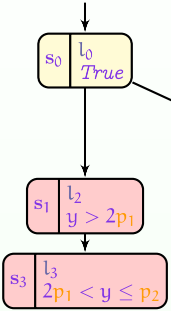
With parametric zone inclusion

- 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

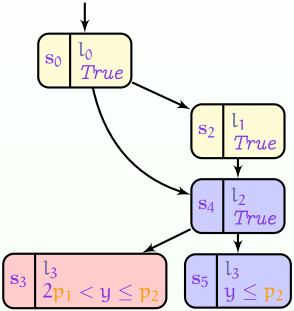
Parametric Zone Inclusion Illustration



A PTA example



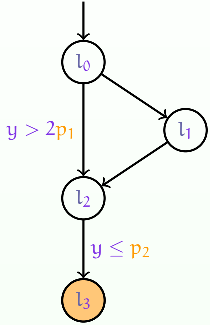
Without parametric zone inclusion



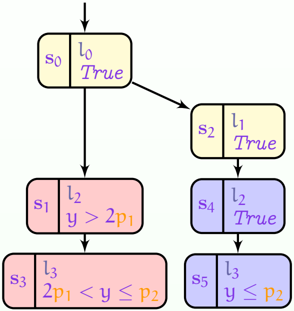
With parametric zone inclusion

- 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

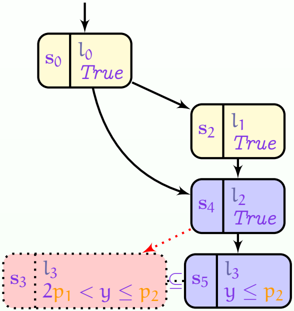
Parametric Zone Inclusion Illustration



A PTA example



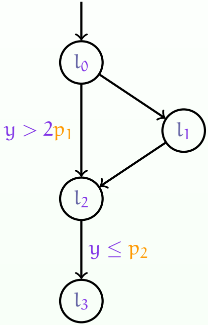
Without parametric zone inclusion



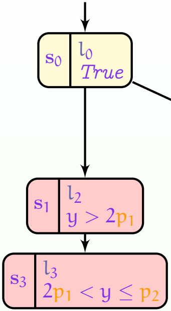
With parametric zone inclusion

- 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 Illustration

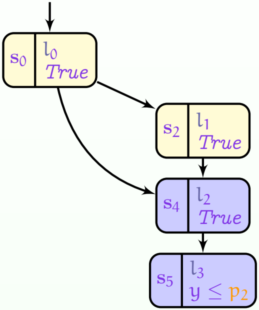


A PTA example



Without parametric zone inclusion

Order: $s_0 \rightarrow s_5$ States: 6

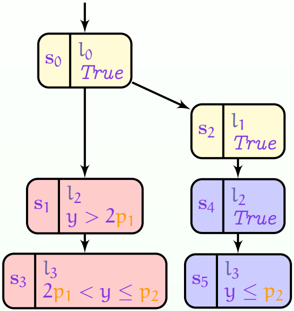


With parametric zone inclusion

Order: $s_0 \rightarrow s_5$ States: 4

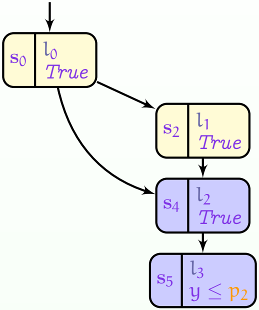
- Problem: inefficient phenomenon happen is when a larger zone is explored after exploring smaller zones (red states)

Parametric Zone Inclusion Illustration



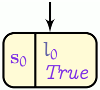
Without parametric zone inclusion

Order: $s_0 \longrightarrow s_5$ States: 6



With parametric zone inclusion

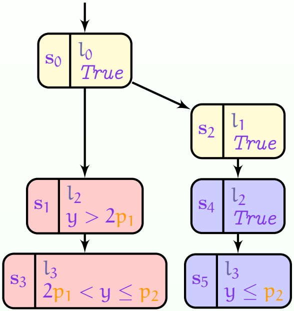
Order: $s_0 \longrightarrow s_5$ States: 4



Ideal exploration order

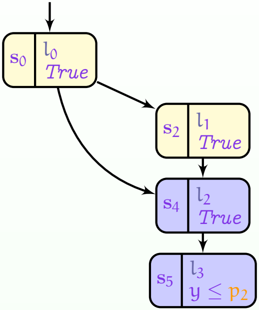
- Question: how to reduce inefficient phenomenon or useless computation?
- → Find an exploration order to explore the biggest zone first!

Parametric Zone Inclusion Illustration



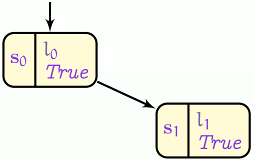
Without parametric zone inclusion

Order: $s_0 \longrightarrow s_5$ States: 6



With parametric zone inclusion

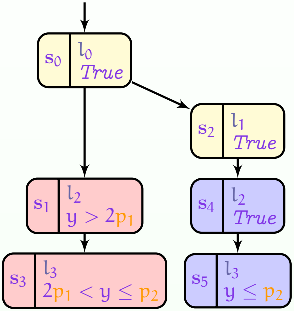
Order: $s_0 \longrightarrow s_5$ States: 4



Ideal exploration order

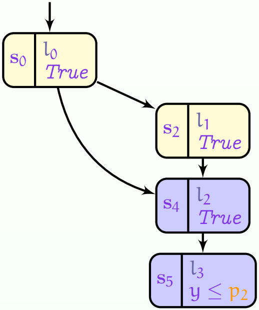
- Question: how to reduce inefficient phenomenon or useless computation?
- → Find an exploration order to explore the biggest zone first!

Parametric Zone Inclusion Illustration



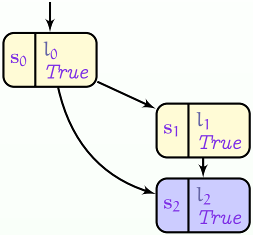
Without parametric zone inclusion

Order: $s_0 \longrightarrow s_5$ States: 6



With parametric zone inclusion

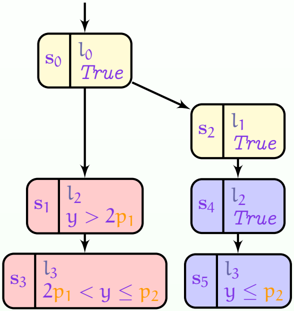
Order: $s_0 \longrightarrow s_5$ States: 4



Ideal exploration order

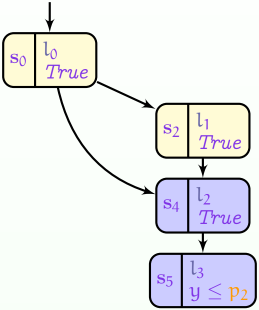
- Question: how to reduce inefficient phenomenon or useless computation?
- → Find an exploration order to explore the biggest zone first!

Parametric Zone Inclusion Illustration



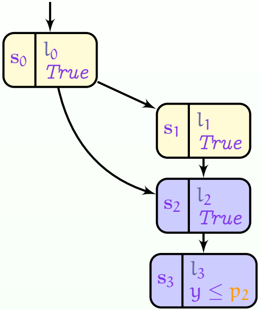
Without parametric zone inclusion

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?
- → 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

Exploration Orders Introduction

Our contribution: 2 new exploration orders for PTAs

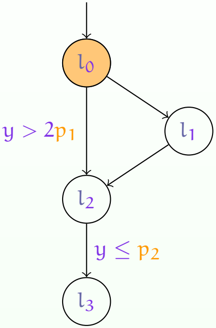
1. Parametric Ranking Strategy
   - This strategy assigns a priority value to each state, then it explores the state with highest priority first
   - Inspired by the “ranking system” strategy [Herbreteau and Tran, 2015].
2. Parametric Priority Strategy
   - A new strategy using an insertion mechanism within an ordered list of parametric zones

Parametric Ranking Strategy

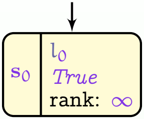
The main idea:

- Explore the state having the highest rank
- Ranking:
  1. A new explored state starts with rank 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

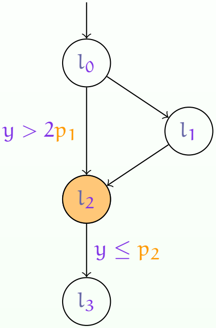


A PTA example

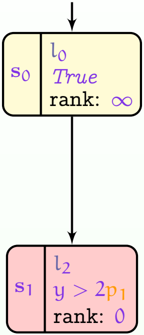


PZG with parametric ranking strategy

Parametric Ranking Strategy

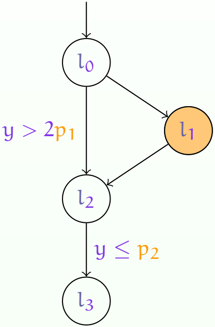


A PTA example

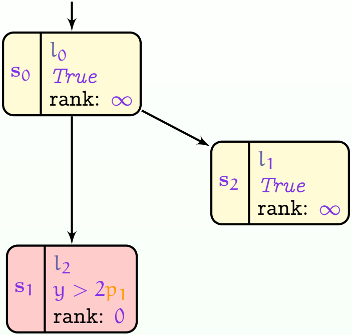


PZG with parametric ranking strategy

Parametric Ranking Strategy

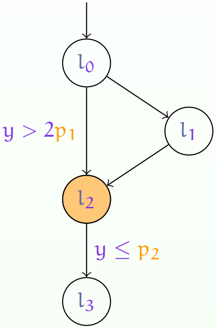


A PTA example

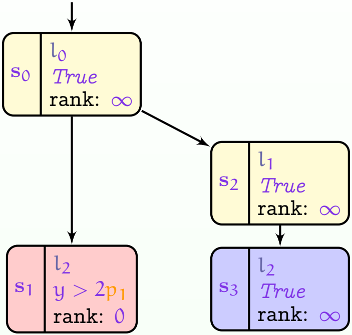


PZG with parametric ranking strategy

Parametric Ranking Strategy

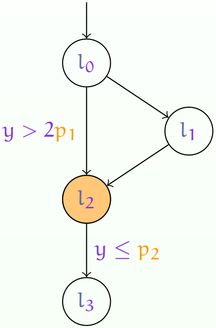


A PTA example

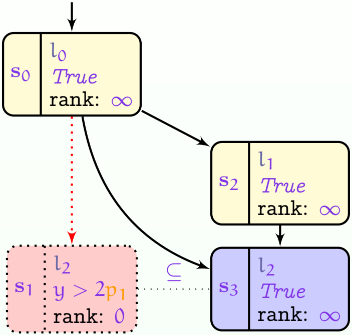


PZG with parametric ranking strategy

Parametric Ranking Strategy

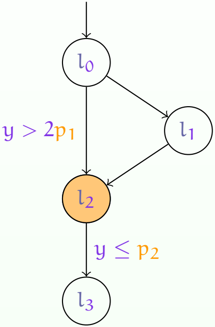


A PTA example

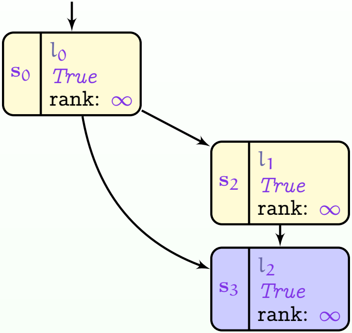

s_0 l_0 *True* rank: ∞
s_2 l_1 *True* rank: ∞
s_1 l_2 $y > 2p_1$ rank: 0
$\subseteq$
s_3 l_2 *True* rank: ∞

PZG with parametric ranking strategy

Parametric Ranking Strategy

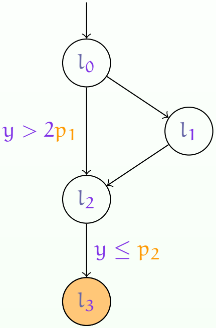


A PTA example

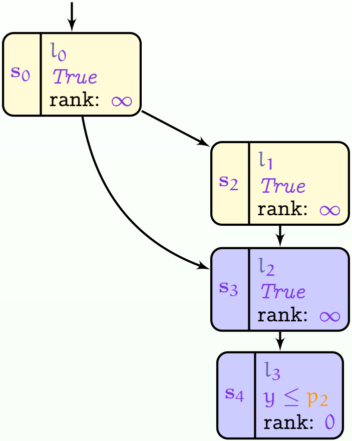


PZG with parametric ranking strategy

Parametric Ranking Strategy

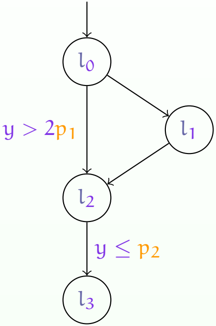


A PTA example

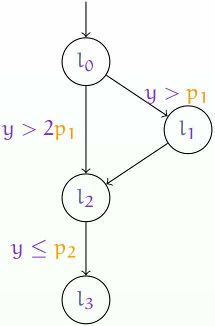

s_0 l_0 *True* rank: ∞
s_2 l_1 *True* rank: ∞
s_3 l_2 *True* rank: ∞
s_4 l_3 $y \leq p_2$ rank: 0

PZG with parametric ranking strategy

Drawback of Parametric Ranking Strategy



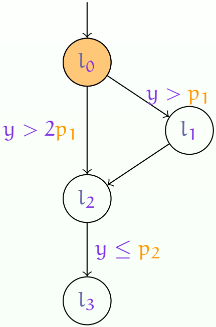
A PTA example



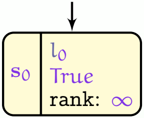
Our PTA example

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

Drawback of Parametric Ranking Strategy



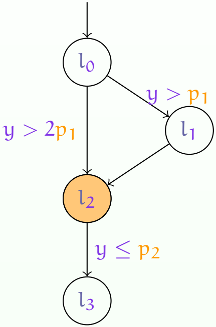
A PTA example



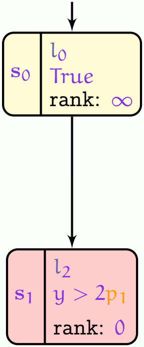
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



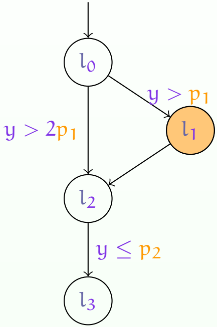
A PTA example



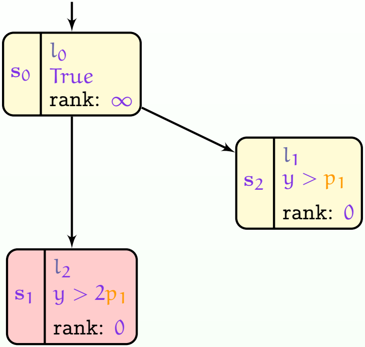
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



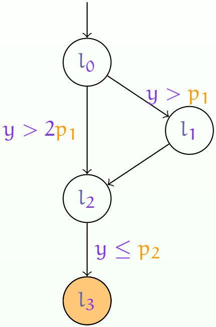
A PTA example



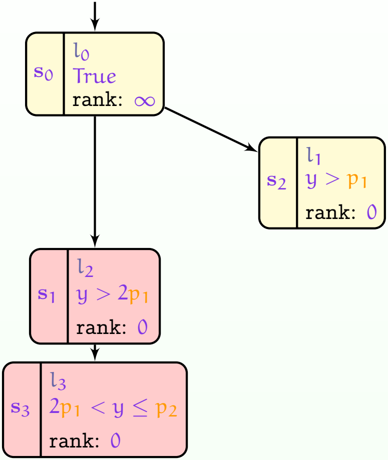
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



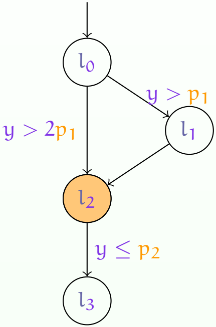
A PTA example


s_0 | l_0 True rank: ∞
s_2 | l_1 $y > p_1$ rank: 0
s_1 | l_2 $y > 2p_1$ rank: 0
s_3 | l_3 $2p_1 < y \leq p_2$ rank: 0

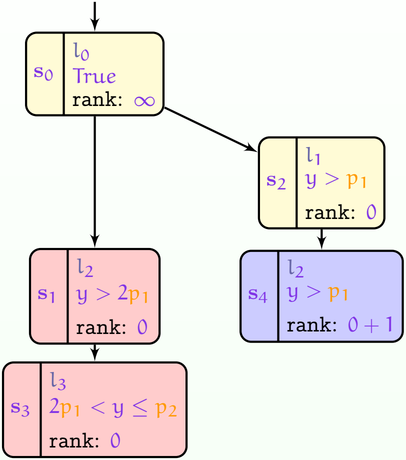
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



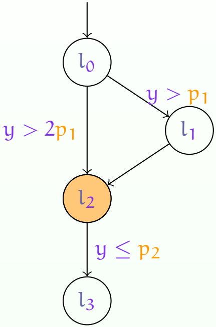
A PTA example


s_0 | l_0 True rank: ∞
s_2 | l_1 $y > p_1$ rank: 0
s_1 | l_2 $y > 2p_1$ rank: 0
s_4 | l_2 $y > p_1$ rank: $0 + 1$
s_3 | l_3 $2p_1 < y \leq p_2$ rank: 0

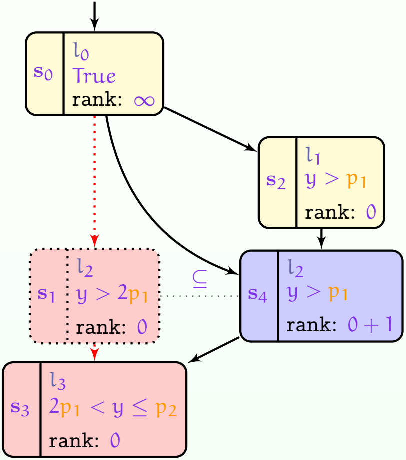
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



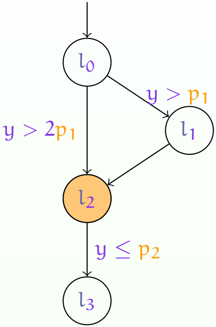
A PTA example



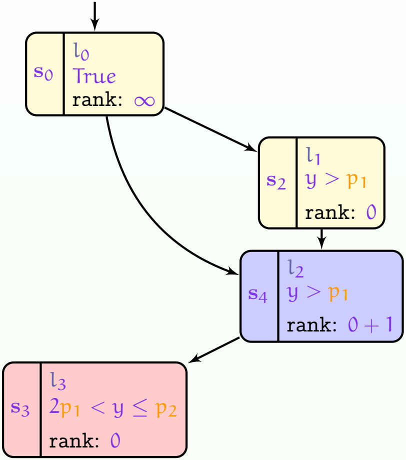
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



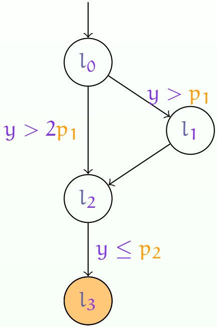
A PTA example


s_0 | l_0 | True | rank: ∞
s_2 | l_1 | $y > p_1$ | rank: 0
s_4 | l_2 | $y > p_1$ | rank: $0 + 1$
s_3 | l_3 | $2p_1 < y \leq p_2$ | rank: 0

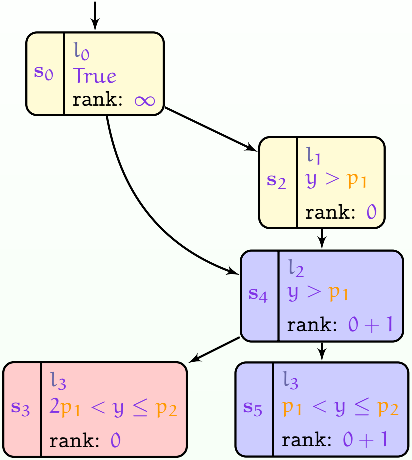
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



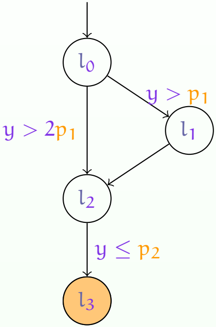
A PTA example



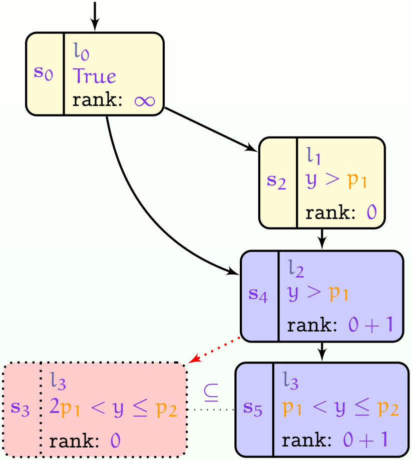
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



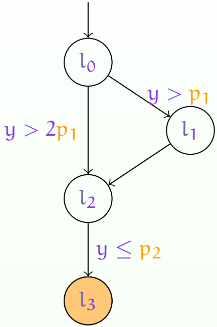
A PTA example



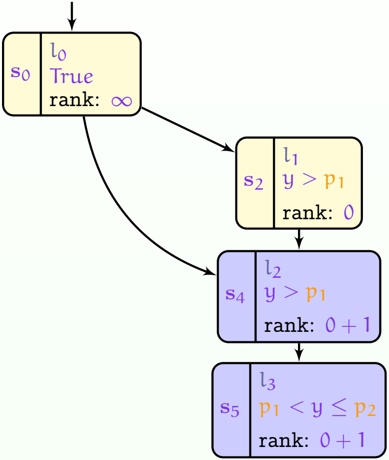
Parametric ranking strategy

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

Drawback of Parametric Ranking Strategy



A PTA example



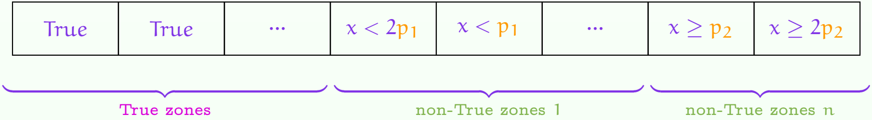
Parametric ranking strategy

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

Parametric Priority Strategy

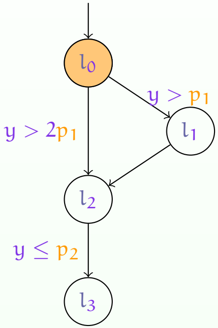
The main idea:

- 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

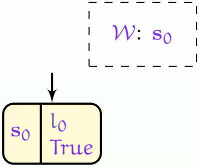


True	True	...	$x < 2p_1$	$x < p_1$	...	$x \geq p_2$	$x \geq 2p_2$
True zones			non-True zones 1			non-True zones n	

Parametric Priority Strategy

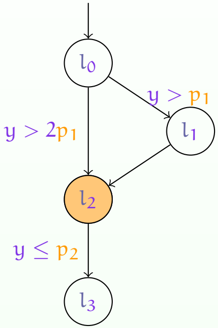


Our PTA example

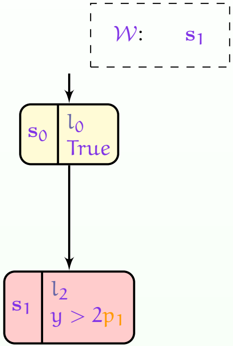


Parametric priority strategy

Parametric Priority Strategy

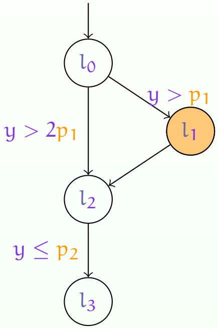


Our PTA example

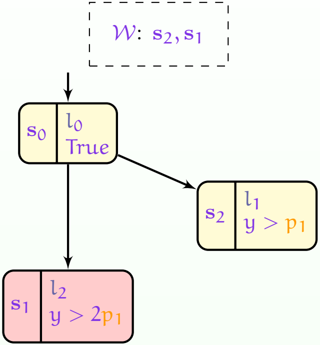


Parametric priority strategy

Parametric Priority Strategy



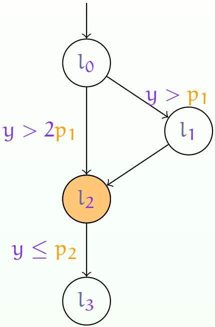
Our PTA example



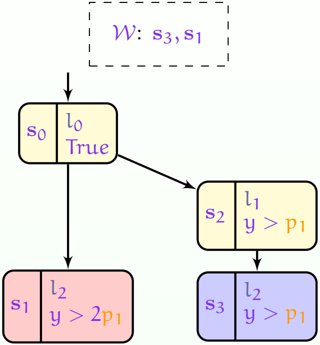
$\mathcal{W}$: $\mathbf{s}_2, \mathbf{s}_1$
$\mathbf{s}_0$ | l_0 True
$\mathbf{s}_2$ | l_1 $y > p_1$
$\mathbf{s}_1$ | l_2 $y > 2p_1$

Parametric priority strategy

Parametric Priority Strategy

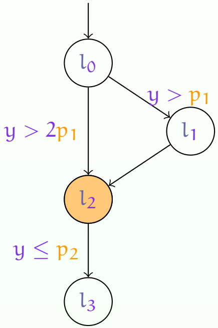


Our PTA example

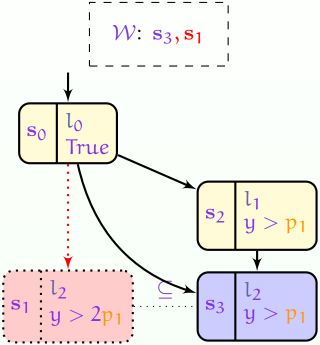

$\mathcal{W}$: $\mathbf{s}_3, \mathbf{s}_1$
$\mathbf{s}_0$ | l_0 True
$\mathbf{s}_2$ | l_1 $y > p_1$
$\mathbf{s}_1$ | l_2 $y > 2p_1$
$\mathbf{s}_3$ | l_2 $y > p_1$

Parametric priority strategy

Parametric Priority Strategy

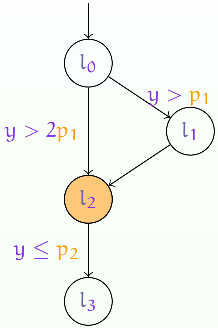


Our PTA example

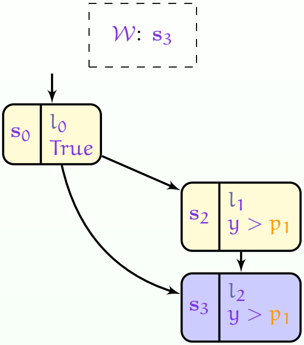

$\mathcal{W}$: $\mathbf{s}_3, \mathbf{s}_1$
$\mathbf{s}_0$ l_0 True
$\mathbf{s}_2$ l_1 $y > p_1$
$\mathbf{s}_1$ l_2 $y > 2p_1$
$\subseteq$
$\mathbf{s}_3$ l_2 $y > p_1$

Parametric priority strategy

Parametric Priority Strategy

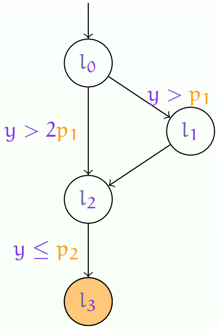


Our PTA example

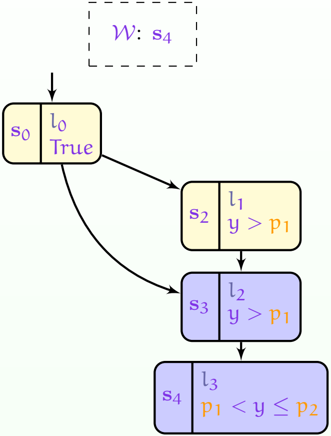


Parametric priority strategy

Parametric Priority Strategy

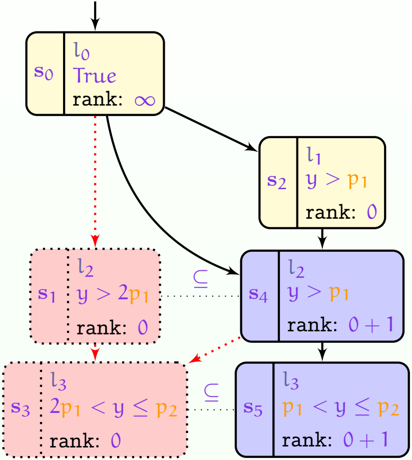


Our PTA example

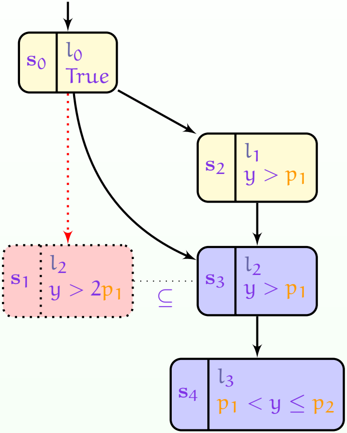


Parametric priority strategy

Strategies Comparison


s_0 l_0 True rank: ∞
s_2 l_1 $y > p_1$ rank: 0
s_1 l_2 $y > 2p_1$ rank: 0
$\subseteq$
s_4 l_2 $y > p_1$ rank: $0+1$
s_3 l_3 $2p_1 < y \leq p_2$ rank: 0
$\subseteq$
s_5 l_3 $p_1 < y \leq p_2$ rank: $0+1$

Parametric ranking strategy



Parametric priority 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

OCaml

PPL

- Implementation in IMITATOR [André, Fribourg, Kühne, Soulat, 2012] [1]
  - 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

[1] http://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

EFsynth				
Benchmark Models	Existing Search Orders		Our Contribution	
	LayerBFS incl (s)	BFS incl (s)	RS incl (s)	PRIOR incl (s)
AndOr	2.512	2.41	1.708	1.714
flipflop-P	121.108	102.42	139.822	140.193
BRP	377.913	370.74	174.038	160.079
Thales-3	627.956	759.987	636.823	597.57
Sched2.100.2	148.169	T.O	249.373	259.895
Sched2.50.2	28.137	217.399	36.81	35.26
FDDI-4	1.315	1.1	1.455	1.285
Fischer-3	0.521	0.48	1.172	1.316
Lynch-5	7.359	7.817	8.859	7.867
F4	21.813	37.558	108.629	96.983
Pipeline-KP12-3-3	T.O	T.O	T.O	T.O
RCP	1.105	1.099	0.093	0.095
spsmall	10.132	9.595	11.114	10.232
critical-region-4	T.O	T.O	T.O	T.O
blowup	31.635	1.345	1.493	1.134
Normalized Average	3.47236	3.7417	2.85594	2.81208

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

- RS and PRIOR are slightly faster in EFsynth

Additional experiments with merging and bidirectional inclusion: see paper

Experiments for Partial Synthesis EFc-ex

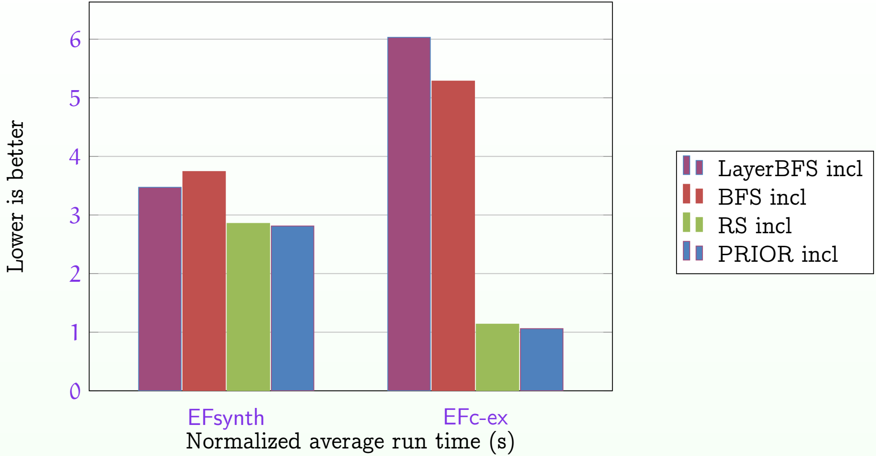
EFc-ex				
Benchmark Models	Existing Search Orders		Our Contribution	
	LayerBFS incl (s)	BFS incl (s)	RS incl (s)	PRIOR incl (s)
AndOr	0.012	0.011	0.008	0.008
flipflop-P	0.061	0.059	0.029	0.028
BRP	2.874	2.944	0.198	0.188
Thales-3	16.638	19.968	0.237	0.232
Sched2.100.2	0.008	0.004	0.005	0.004
Sched2.50.2	0.028	0.023	0.016	0.015
FDDI-4	0.377	0.291	0.091	0.078
Fischer-3	0.097	0.097	0.057	0.059
Lynch-5	7.408	7.912	8.847	7.829
F4	4.086	6.543	0.364	0.311
Pipeline-KP12-3-3	21.927	18.229	0.042	0.042
RCP	0.51	0.454	0.024	0.02
spsmall	5.862	6.242	0.143	0.143
critical-region-4	1.008	0.821	0.044	0.043
blowup	32.893	1.346	1.337	1.003
Normalized Average	6.03173	5.28516	1.13675	1.06026

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

- RS and PRIOR dominate other algorithms in 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

References I

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*. Springer.

Behrmann, G. (2005).
Distributed reachability analysis in timed automata.
STTT, 7(1):19–30.

Behrmann, G., Hune, T., and Vaandrager, F. W. (2000).
Distributing timed model checking – how the search order matters.
In *CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 216–231. Springer.

Hendriks, M. and Larsen, K. G. (2002).
Exact acceleration of real-time model checking.
Electr. Notes Theor. Comput. Sci., 65(6):120–139.

References II

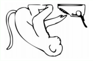
Herbreteau, F. and Tran, T. (2015).
Improving search order for reachability testing in timed automata.
In *FORMATS*, volume 9268 of *Lecture Notes in Computer Science*, pages 124–139. Springer.

Laarman, A., Olesen, M. C., Dalsgaard, A. E., Larsen, K. G., and Van De Pol, J. (2013).
Multi-core emptiness checking of timed Büchi automata using inclusion abstraction.
In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 968–983. Springer.

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