Cut-Offs in Parameterized Verification

Parosh Aziz Abdulla  Frédéric Haziza  Lukáš Holík
Parameterized System

Features

Analysis Abstraction

Completeness
Parameterized System
Parameterized System

Remarks
- Parameterized System
  - unbounded number of components
  - Parameterized Verification: verify correctness regardless of number of components
- Motivation: ubiquitous
  - unbounded number of processes
  - unbounded data structures
  - unbounded number of variables
Parameterized System

Unbounded Number of Processes
Parameterized System

Unbounded Number of Processes

Mutual Exclusion Protocols

- unbounded number of processes
- correctness:
  - lock taken by at most one process
Parameterized System

Unbounded Number of Processes

Cache Coherence Protocol

- unbounded number of processes
- correctness:
  - exclusive ownership: at most one process
Parameterized System

Unbounded Data Structures
Parameterized System

Unbounded Data Structures

Unbounded Channels

- unbounded FIFO channels
- correctness:
  - regardless of channels size
Parameterized System

Unbounded Data Structures

Petri Nets

- unbounded number of tokens
- correctness:
  - coverability
Parameterized System

Unbounded Data Structures

Petri Nets

- unbounded number of clocks
- multiply unbounded
Parameterized System

Unbounded Data Structures

- unbounded heap
- unboundedly many threads
- unbounded data domain
Parameterized System

Unbounded Number of Processes
Parameterized Systems

Processes:
- finite-state
- infinite-state
- dynamic

Unbounded Number of Processes

Topology:
- array
- tree
- graph
- multiset

Communication:
- rendez-vous
- broadcast
- global
Parameterized System

∀L

∃L

∀L

∃L

Burns’ Mutual Exclusion Protocol

critical section
Parameterized System

Configuration
Parameterized System

Local Transition
Parameterized System

Existential Global Transition
Parameterized System

Existential Global Transition
Parameterized System

Universal Global Transition
Parameterized System

Universal Global Transition
Initial Configurations

Parameterized System

- Infinite, but regular
Parameterized System

Bad Configurations

two or more processes in critical section
Parameterized System

Bad Configurations

- two or more processes in critical section

Set of bad configurations
- infinite, but
- upward closed
Parameterized System

Goal

Verify correctness regardless of # of processes

# processes = parameter of the system
Parameterized Systems

- Mutual Exclusion
  - Burns
  - Dijkstra
  - Szymanski

- Cache coherence
  - MOSI
  - German
Parameterized Systems

- Mutual Exclusion
- Cache coherence
- Petri Nets
- Trees
- Rings
- Burns
- Dijkstra
- Szymanski
- MOSI
- German
Parameterized System

Remarks

- Infinite-state system
  - unbounded number of processes
  - *Parameterized Verification*: verify correctness regardless of number of processes
- Problem undecidable in general
  - Challenge: find abstractions that work often
Parameterized System

Features

Analysis Abstraction

Completeness

✓
Parameterized System

Analysis Abstraction
Small model property

⇒ inspect small instances of the system

⇒ efficient method
Parameterized Systems
Parameterized Systems

$R_2$  $R_3$  $R_4$
Parameterized Systems

Goal
Safety
infinite family
Parameterized Systems

Small Model

- Efficient method
Intuition

- Bad configurations:
  - can be characterized by fixed number of witness processes

- Bad patterns:
  - appear in small system instances

Small Model
Parameterized System

Analysis

Abstraction

Abstraction modulo $k$
  - $k$: natural number

$\alpha_k: \mathbb{C} \rightarrow \forall$

$k=2$
Parameterized System
Analysis Abstraction

\[ \alpha_k : \mathcal{C} \rightarrow \mathcal{V} \quad k=2 \]
Parameterized System

Analysis Abstraction

$\alpha_k: C \rightarrow V$

$k=2$

Configurations

Views
For each $k$ : two procedures (in parallel):

- Bug detection
  - under-approximation
  - concrete domain
- Verification
  - over-approximation
  - abstract domain

- abstraction of initial configurations
- abstract post operator
Parameterized System

Analysis
Abstraction

\[ \mathcal{I} = \bigcup_{n \geq 0} \text{Init}_n \]

\[ \mathbf{V} = \alpha_2(\mathcal{I}) \]
Parameterized System

Analysis Abstraction

\[ \gamma_k : \text{set of } \forall \rightarrow \text{set of } \mathbb{C} \]

\[ \gamma_k(X) = \{ c \in \mathbb{C} | \alpha_k(c) \subseteq X \} \]
Parameterized System

Analysis Abstraction

\[ \gamma_2( ) = \ldots \]

\[ \ldots \]
Parameterized System
\[ \gamma_2(x) = \ldots \]

Analysis
Abstraction
Parameterized System

Abstraction

\[ \gamma_2( ) \]
Parameter System

Analysis Abstraction

$A_{post_k}(X) := \alpha_k \left( \text{post} \left( \gamma_k(X) \right) \right)$
Parameter System
Analysis Abstraction

\[ V_k := \mu X . \alpha_k(I) \cup \text{Apost}_k(X) \]

\[ R \subseteq \gamma_k(V_k) \]

\[ \gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k) \]

\[ \text{Apost}_k(X) = \alpha_k(\text{post}(\gamma_k(X))) \]
\( \gamma_{k|m}(X) = \{ c \in \mathcal{C} | \alpha_k(c) \subseteq X, |c| \leq m \} \)
Parameterized System

Analysis

Abstraction

\[ \alpha_2 ( \text{post} ( \gamma_{2|3}(V))) \]
Parameterized System

Analysis Abstraction

\[ \text{Apost}_k(\mathbf{X}) := \alpha_k(\text{post}(\gamma_{k|k+1}(\mathbf{X}))) \]

\[ V_k := \mu \mathbf{X}. \alpha_k(I) \cup \alpha_k(\text{post}(\gamma_k(\mathbf{X}))) \]
for \( k = 1 \) to \( \infty \) do

if \( R_k \cap \text{Bad} \neq \emptyset \) then return \text{Unsafe}

\[ V = \mu X . a_k(I) \cup a_k \circ \text{post} \circ \gamma_k(X) \]

if \( \gamma_k(V) \cap \text{Bad} = \emptyset \) then return \text{Safe}
Parameterized System Analysis Abstraction

\[ R_k \cap \text{Bad} \neq \emptyset \]

\[ k = 1 \]

Increase precision

\[ k ++ \]

\[ V = \mu X. \alpha_k(I) \cup \alpha_k \circ \text{post} \circ \gamma_k(V) \]

\[ \gamma_k(V) \cap \text{Bad} = \emptyset \]

Unsafe

Safe
Parameterized System Analysis Abstraction

∀L
∃L
∀R

R_1 \cap \text{Bad} = \emptyset
\[ k = 1 \]
\[ \mathbf{V} = \mu \mathbf{X} . \ a_1(I) \cup a_1 \cdot \text{post} \cdot \gamma_1(\mathbf{X}) \]
\( k = 1 \)

if \( \gamma_1(V) \cap \text{Bad} = \emptyset \) then return Safe

\( \gamma_1(V) \) contains everything!
\( \forall L \exists L \)

\( k=2 \)  \( \mathbf{R}_2: \)

\( R_2 \cap \text{Bad} = \emptyset \)
Parameterized System
Analysis
Abstraction

∀L
∃L
∀R

k=2

V = \mu X . a_2(I) \cup a_2 \cdot \text{post} \cdot \gamma_{2|3}(X)

extensions of size 3 only

All pairs but
\[ k = 2 \]

\[
\text{if } \gamma_2(V) \cap \text{Bad} = \emptyset \text{ then return Safe}
\]

Collected all views

\[ k=2 \text{ is cutoff point} \]
Parameter System

Analysis Abstraction

Features
Parameterized System

Analysis

Abstraction

Features

Universal condition

Existential condition
Universal condition

Existential condition

Global variables

Broadcast

Creation/Deletion

Rendez-Vous
Global variables

universal condition

Existential condition

Creation/Deletion

Topologies

- Linear
- Ring
- Tree
- Multiset

Broadcast

Rendez-Vous
Parameterized System

Leader Election
Leader Election

Parameterized System
Universal condition

Existential condition

Non-atomic
global conditions
<table>
<thead>
<tr>
<th>Parameterized System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Abstract</td>
</tr>
<tr>
<td>Features</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>?</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>?</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### Parameterized System Analysis Abstraction Features

<table>
<thead>
<tr>
<th></th>
<th>✓</th>
<th></th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>✓</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parameterized System

Analysis

Abstract Features

∃

∀

Analysis Abstraction

Features

<table>
<thead>
<tr>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>O</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>?</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>✓</td>
<td>O</td>
<td>.</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Parameterized System

Analysis

Abstract

Features
<table>
<thead>
<tr>
<th>Parameter System</th>
<th>Analysis</th>
<th>Abstract</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
</tr>
</tbody>
</table>
Parameterized System Analysis Abstraction Features Completeness
Well Quasi-Ordering (WQO)

\((A, \preceq)\)

\(a_0 \preceq a_1 \preceq a_2 \preceq a_i \preceq a_j\)
if \( c_1 \in D \) and \( c_2 \preceq c_1 \) then \( c_2 \in D \)
Parameterized System Analysis Abstraction Features Completeness

\[ V_k := \mu X . \alpha_k(I) \cup \text{Apost}_k(X) \]

\[ R \subseteq \gamma_k(V_k) \]

\[ \gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k) \]

if

- \( D \) downward-closed
- \( D \) inductive

then

- \( D \supseteq \gamma_k(V_k) \) for some \( k \)

\[ \mathcal{R} = \gamma_k(V_k) \text{ for some } k \]
A system \((C, \rightarrow)\) is WQO w.r.t a WQO \(\preceq\)
if \(\rightarrow\) is monotonic
A system \((C, \rightarrow)\) is WQO w.r.t a WQO \(\preceq\)
if \(\rightarrow\) is monotonic

Class of
- Lossy channel systems
- Petri Nets
- Parameterized Systems \((\exists \forall)\)
- ...

Parameterized System Analysis Abstraction Feature Completeness
Theorem: if System is WQO and Safe

Procedure guaranteed to terminate

Procedure is complete

Applications: Petri Nets
Parameterized System

Features

Analysis Abstraction

Completeness
### Parameterized System Analysis

#### Abstraction Features Completeness

| Protocol                    | Time | $k$ | $|V|$ | $\gamma_{k+\ell}(V)$ |
|-----------------------------|------|-----|------|----------------------|
| **Array**                   |      |     |      |                      |
| Demo (toy example)          | 0.01s| 2   | 17   | 53                   |
| Burns                       | 0.01s| 2   | 34   | 186                  |
| Dijkstra                    | 0.07s| 2   | 93   | 695                  |
| Szymanski                   | 0.02s| 2   | 48   | 264                  |
| **Multiset**                |      |     |      |                      |
| MOSI Coherency              | 0.01s| 1   | 10   | 23                   |
| German’s Coherency          | 15.3s| 6   | 1890 | 15567                |
| **Petri Net**               |      |     |      |                      |
| German (simplified)         | 0.03s| 2   | 43   | 96                   |
| BH250                       | 2.85s| 2   | 503  | 503                  |
| MOESI Coherency             | 0.01s| 1   | 13   | 20                   |
| Critical Section            | 0.01s| 5   | 27   | 46                   |
| Kanban                      | ?    | >20 | ?    | ?                    |
| **Tree**                    |      |     |      |                      |
| Percolate                   | 0.05s| 2   | 34   | 933                  |
| Tree Arbiter                | 0.7s | 2   | 88   | 7680                 |
| Leader Election             | 0.1s | 2   | 74   | 362                  |
| **Ring**                    |      |     |      |                      |
| Token Passing               | 0.01s| 2   | 2    | 2                    |
Future Challenges

Shape Analysis
Parameterized System Analysis

Future Work

Contexts

CEGAR

Shape analysis
Parameterized System Analysis

Completeness

Future Work

Shape analysis

Lock-free Data Structures

Treiber's stack  M&S's queue
Parameterized System Analysis

Completeness

Future Work

Contexts

CEGAR

Shape analysis

push ( a );

push ( b );

push ( c );

pop ( );

push ( a );

push ( d );
Parameterized System Analysis

Completeness

Future Work

Shape analysis

\[
t := S \\
n . next := t \\
CAS(S \cdot t \cdot n)
\]

\[
v' \cdot n \\
t \cdot t \\
x \cdot x \cdot t \\
S \cdot x \cdot t \\
? \cdot # \\
t := S \\
x := t . next \\
CAS(S \cdot t \cdot x) \\
t := S \\
x := t . next \\
CAS(S \cdot t \cdot x)
\]
Future Work

Contexts

CEGAR

Shape analysis
Parameterized System
Analysis Abstraction
Feature Completeness

Future Challenges

Shape Analysis
Infinite-State Processes
Unbounded Data Structures
Weak Memory Models