Cut-Offs in Parameterized Verification

Parosh Aziz Abdulla  Frédéric Haziza  Lukáš Holík
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
• 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 System

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

Topology:
- array
- tree
- graph
- multiset

Unbounded Number of Processes

Communication:
- rendez-vous
- broadcast
- global

Hierarchy
Parameterized System

Burns' Mutual Exclusion Protocol

∀L ∃L ∀R

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
Parameterized System

Initial Configurations

- $\text{Init}_1$
- $\text{Init}_2$
- $\text{Init}_3$
- $\text{Init}_4$
- $\text{Init}_5$

Set of initial configurations
- 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

Bad Configurations
Parameterized System

Goal

Verify correctness regardless of # of processes

# processes = parameter of the system
Parameterized Systems

- Mutual Exclusion
- Cache coherence

- Burns
- Dijkstra
- Szymanski
- 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

Analysis Abstraction

Features

Completeness
Parameterized System

Analysis
Abstraction
Small model property

→ inspect small instances of the system

→ efficient method
Parameterized Systems
Parameterized Systems

$R_2$

$\forall L \exists L \forall R$
Parameterized Systems

$R_2$  $R_3$  $R_4$
Parameterized Systems

\[ R_2 \quad R_3 \quad R_4 \quad R_5 \quad R_6 \quad \ldots \]

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
Parameterized System

Analysis
Abstraction

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

$\alpha_k: C \rightarrow V$

$k=2$
Parameterized System

Analysis Abstraction

\[ \alpha_k: \mathbb{C} \rightarrow \mathbb{V} \]

k=2
Parameter System

Analysis Abstraction

\[ \alpha_k : C \rightarrow V \quad k=2 \]

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

\[ V = a_2(I) \]
Concrete

Conf

Conf

post
Parameter System

Analysis Abstraction

Concrete

Confs

post

Confs

Abstract

Views

$\gamma_k$

Confs

post

$\alpha_k$

Views

Confs
Parameterized System

Analysis Abstraction

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

$\gamma_k(X) = \{ c \in \mathbb{C} | \alpha_k(c) \subseteq X \}$
\[ \gamma_2(\ ) = \ldots \]
Parameterization System

Analysis Abstraction

\[ \gamma_2( ) = \ldots \]

\[ \ldots \]
Parameterized System

Analysis Abstraction

\[ \gamma_2(\, ) \ni \]
Parameterized System

Analysis

Abstraction

Apost

k

(X)

:=

\alpha_k (post (\gamma_k (X)))
\[ \mathcal{V}_k := \mu X. \alpha_k(I) \cup \text{Apost}_k(X) \]

\[ \mathcal{R} \subseteq \gamma_k(V_k) \]

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

\[ \text{Apost}_k(X) = \alpha_k(\text{post}(\gamma_k(X))) \]
Parameterized System

Analysis

Abstraction

\[ a_k\left( \text{post}\left( \gamma_k(V) \right) \right) \]

only of size k+1

\[ \gamma_{k|m}(X) = \{ c \in \mathcal{C} | a_k(c) \subseteq X, \ |c| \leq m \} \]
$a_2 \left( \text{post} \left( \gamma_{2|3}(\mathbf{V}) \right) \right)$
Parameterized System Analysis Abstraction

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

\[ \mathbf{V}_k := \mu \mathbf{X} . a_k(I) \cup a_k( \text{post} ( \gamma_k(\mathbf{X}))) \]
for $k = 1$ to $\infty$ do
  if $R_k \cap \text{Bad} \neq \emptyset$ then return Unsafe
  $V = \mu X \cdot a_k(I) \cup a_k \cdot \text{post} \cdot \gamma_k(X)$
  if $\gamma_k(V) \cap \text{Bad} = \emptyset$ then return Safe
Parameterized System Analysis Abstraction

$\mathcal{R}_k \cap \text{Bad} \neq \emptyset$

$k = 1$

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

Unsafe

Increase precision

$k ++$

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

Safe
∀L \exists L \forall R \exists L

\[ k=1 \quad R_1: \quad \text{Diagram with colored squares} \]

\[ R_1 \cap \text{Bad} = \emptyset \]
\( k=1 \)

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

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

\( \gamma_1(V) \) contains everything!
$\forall L \exists L \forall R \exists R_{2} \cap \text{Bad} = \emptyset$
$k = 2$

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

All pairs but
$k = 2$

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

Collected all views

$k=2$ is cutoff point
Parameterized System
Analysis Abstraction

Features
Universal condition

Existential condition
Global variables

Universal condition

Existential condition

Broadcast

Creation/Deletion

Rendez-Vous

Parameterized System

Analysis Abstraction

Features
Global variables

Parameterized System

Analysis

Broadcast

Abstraction

Features

Universal condition

Existential condition

Global variables

Creation/Deletion

Broadcast

Rendez-Vous

Topologies

• Linear
• Ring
• Tree
• Multiset
Leader Election
Parameterized System

Leader Election

- Upward propagation
- The root initiates the downward propagation
- Downward propagation: Election of a candidate
- A process candidates
Universal condition

Existential condition

Non-atomic global conditions
Parameterized System

Analysis Abstraction

Features

\exists L

for-loop

for-loop
Parameterized System

Analysis

Abstraction

Features
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<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>
<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>
<tr>
<td>?</td>
<td>?</td>
<td>✓</td>
<td>0</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>
<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>
<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>
<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>0</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>
<tr>
<td>Parameterized System</td>
<td>Analysis</td>
<td>Abstraction</td>
<td>Features</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>

<p>| . | . | . | . | . | . | . | . |
| ✓ | 0 | ✓ | ✓ | ✓ | X | ? | . |
| . | . | . | . | . | . | . | . |
| ? | ? | ✓ | . | 0 | _ | _ | . |
| . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . |</p>
<table>
<thead>
<tr>
<th>Parameterized System</th>
<th>Analysis Abstraction</th>
<th>Features</th>
</tr>
</thead>
</table>

<p>| | | | | | | | |</p>
<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>o</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>✓</th>
<th>0</th>
<th>✓</th>
<th>✓</th>
<th>✗</th>
<th>?</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<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>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>

∃ₐ \forall \exists_L
### Analysis System

- **Parameterized**
- **Abstraction**
- **Features**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

### Diagram

- [Diagram with buttons and symbols]
Parameterized System

Analysis

Abstraction

Features

<table>
<thead>
<tr>
<th>0</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Analysis Abstraction

Features

Completeness
Parameterized System Analysis Abstraction Features Completeness
Well Quasi-Ordering (WQO)
Parameterized System Analysis Abstraction Features Completeness

Downward-closed set

if $c_1 \in D$ and $c_2 \preceq c_1$ then $c_1 \in D$
Upward-closed set

\[ U \]

If \( c_1 \in U \) and \( c_1 \preceq c_2 \) then \( c_2 \in D \)
$V_k := \mu X \cdot a_k(I) \cup \text{Apost}_k(X)$

$R \subseteq \gamma_k(V_k)$

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

if

- $R$ downward-closed
- $R$ inductive

then

- $R = \gamma_k(V_k)$ 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 \(\prec\) if \(\rightarrow\) is monotonic

Class of
- Lossy channel systems
- Petri Nets
- Parameterized Systems \((\text{no } \forall)\)
- ...
Theorem

if System is WQO and Safe

Procedure guaranteed to terminate

Procedure is complete

Applications: Petri Nets
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Time</th>
<th>k</th>
<th></th>
<th>V</th>
<th>γ_k+ε(V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demo (toy example)</td>
<td>0.01s</td>
<td>2</td>
<td>17</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Burns</td>
<td>0.01s</td>
<td>2</td>
<td>34</td>
<td>186</td>
<td></td>
</tr>
<tr>
<td>Dijkstra</td>
<td>0.07s</td>
<td>2</td>
<td>93</td>
<td>695</td>
<td></td>
</tr>
<tr>
<td>Szymanski</td>
<td>0.02s</td>
<td>2</td>
<td>48</td>
<td>264</td>
<td></td>
</tr>
<tr>
<td>Multiset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOSI Coherency</td>
<td>0.01s</td>
<td>1</td>
<td>10</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>German’s Coherency</td>
<td>15.3s</td>
<td>6</td>
<td>1890</td>
<td>15567</td>
<td></td>
</tr>
<tr>
<td>Petri Net</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German (simplified)</td>
<td>0.03s</td>
<td>2</td>
<td>43</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>BH250</td>
<td>2.85s</td>
<td>2</td>
<td>503</td>
<td>503</td>
<td></td>
</tr>
<tr>
<td>MOESI Coherency</td>
<td>0.01s</td>
<td>1</td>
<td>13</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Critical Section</td>
<td>0.01s</td>
<td>5</td>
<td>27</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Kanban</td>
<td>?</td>
<td>≥ 20</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Tree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percolate</td>
<td>0.05s</td>
<td>2</td>
<td>34</td>
<td>933</td>
<td></td>
</tr>
<tr>
<td>Tree Arbiter</td>
<td>0.7s</td>
<td>2</td>
<td>88</td>
<td>7680</td>
<td></td>
</tr>
<tr>
<td>Leader Election</td>
<td>0.1s</td>
<td>2</td>
<td>74</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>Ring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token Passing</td>
<td>0.01s</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Future Challenges

Shape Analysis
Parameterized System Analysis

Future Work

Contexts

CEGAR

Shape analysis
Future Work

Parameterized System Analysis

Features

Completeness

Future Work

Contexts

CEGAR

Shape analysis

push (a);

push (b);

push (c);

pop ();

push (a);

push (d);
Parameterized System Analysis

Contexts

GAR

Shape analysis

\[ t := S \]
\[ n \text{.next} := t \]
\[ \text{CAS}(S, t, n) \]

\[ t := S \]
\[ x := t \text{.next} \]
\[ \text{CAS}(S, t, x) \]

\[ t := S \]
\[ x := t \text{.next} \]
\[ \text{CAS}(S, t, x) \]
Future Work

- Contexts
- CEGAR
- Shape analysis
Parameterized System Analysis Abstraction Features Completeness

Future Challenges

Shape Analysis

Infinite-State Processes

Unbounded Data Structures

Weak Memory Models