Software Testing a Verification System

Mahmoud A. Bokhari  Markus Wagner
Outlines

• Introduction
  – Verification Systems
  – Axioms
  – Beam Search
• Objectives
• Methods
• Case Study: KeY System
• Experimental Results
Verification Systems

Programs & Requirements

Verification Systems

output
Verification Systems

Programs & Requirements → Deductive Verification Systems → output

Axioms
Axioms

• Axioms
  – Set of rules.
  – Help to understand the software.
  – Encode semantics.
Beam Search

- Heuristic.
- Explores promising nodes.
Beam Search

• Heuristic.
• Explores promising nodes.
Outlines

• Introduction
  – Verification Systems
  – Axioms
  – Beam Search
• Objectives
• Methods
• Case Study: KeY System
• Experimental Results
Objectives

Maximise the axiomatisation coverage.

Uncover hidden defects within the axiomatisation base.
Outlines

• Introduction
  – Verification Systems
  – Axioms
  – Beam Search

• Objectives

• Methods

• Case Study: KeY System

• Experimental Results
Test Case Generation

P
+
Requirements
+
WL

\[
\text{Foreach } \alpha x \text{ in PL} \\
\text{WL} := \text{WL} - \alpha x_i
\]

Deductive Verification Systems

\[R\]

Reduction

\[\text{Promising List generation} <\text{beam search}>\]

Mandatory Set

\[\text{M}\]
Test Case Generation

\[
\text{P} \quad \text{+ Requirements} \quad \text{+ WL}
\]

\[
\text{Deductive Verification Systems}
\]

\[
\text{Foreach } ax \text{ in PL } \quad \text{WL := WL} - ax_i
\]

\[
\text{Reduction}
\]

\[
\text{Promising List generation} \quad \text{<beam search>}
\]

\[
\text{Mandatory Set} \quad \text{M}
\]

New test case
Test Case Generation

\[ P \]
\[ + \]
\[ Requirements \]
\[ + \]
\[ WL \]

\[ \text{Foreach } ax \text{ in } PL \]
\[ WL := WL - ax_i \]
\[ WL := WL + ax_i \]

Deductive Verification Systems
Test Case Generation

\[
P + \text{Requirements} + WL
\]

Deductive Verification Systems

Foreach \( ax \) in PL
\[
WL := WL - ax_i
\]
\[
WL := WL + ax_i
\]
\[
WL := WL - ax_{i+1}
\]
The main idea is to guide the search process.

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Replacement sets</th>
<th>Successful</th>
<th>Total Successful</th>
<th>Unsuccessful</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>{A3,A5}</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A100</td>
<td>{A4}</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>{A10, A6}</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Beam Search Algorithm

Verify (Test Case)

Get (Mandatory Set M)

Check \( m_i \) in GT

- **proved**
  - Guidance Table

- **not proved**
  - Add \( m_i \) to Discard list
  - Add \( m_i \) to Promising list

- **New axiom found**

  - Last
  - Stop

Check \( m_i \) successful count

- **found**
  - Check \( m_i \) is the last in M

- **not found**

  - Add \( m_i \) to Discard list
  - Add \( m_i \) to Promising list

  - **count > 0**

else

  - Guidance Table

else

  - Count > 0

Add \( m_i \) to Discard list

Add \( m_i \) to Promising list

Check \( m_i \) is the last in M

else

  - Last

Stop
Outlines

• Introduction
  – Verification Systems
  – Axioms
  – Beam Search
• Objectives
• Methods
• Case Study: KeY System
• Experimental Results
Case Study: KeY System

Programs + Requirements

KeY

1520 Axioms ≈ 2^{1520}
possible solutions
Outlines

• Introduction
  – Verification Systems
  – Axioms
  – Beam Search

• Objectives

• Methods

• Case Study: KeY System

• Experimental Results
# Experiment Settings

## Axiomatization Coverage

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Cases</strong></td>
<td>319</td>
</tr>
<tr>
<td><strong>Reduction Phase</strong></td>
<td>Enabled</td>
</tr>
<tr>
<td><strong>Time Limit</strong></td>
<td>24 hours per test case</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td>Intel Xeon E5430</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td>Debian GNU/Linux 5.0.8 Java SE RE 1.7.0</td>
</tr>
</tbody>
</table>
Results

Axiomatisation Coverage

Number of axioms

Random Depth-First Search
Random Breadth-First Search
BeamSearch

Approaches

Total covered
First mandatory set

50%
Thank You ...

Any Question ...