Synthesizing Adaptive Test Strategies from Temporal Logic Specifications

Roderick Bloem, Robert Könighofer, Ingo Pill, Franz Röck
Institute of Applied Information Processing and Communications
Graz University of Technology, Austria
2016-10-04
Outline

- Motivation
- Our Approach
- Fault Models
- Experimental Results
- Conclusion
Motivation

Testing is a Game

Specification → Test Case → System Impl. → Oracle → Pass/Fail/?

Test Strategy → System

System

Inputs

 Outputs

Inputs

 Outputs
Motivating Example

1. The lights must never be green simultaneously.
2. If a car is waiting, $f$ eventually turns true.
3. If no car is waiting, $h$ eventually becomes true.
4. A picture is taken if a car does a head start.

$$\Phi = G(\neg f \lor \neg h) \land G(c \rightarrow Ff) \land G(\neg c \rightarrow Fh) \land G[(\neg f \land X(c \land f \land X \neg c)) \leftrightarrow XXp]$$
“Good” Tests

- **Challenge**: what are *good* test cases?
  - Many coverage metrics have been proposed
  - Fault based: Tests should reveal certain faults
    - Assume “almost”-correct system under test (SUT)
    - Simple faults (flip, stuck-at-0, …) at single outputs
    - Faults can be permanent or transient
    - Tests must cause a specification violation for these faults
      → Tests will also reveal other faults
Goal

- From temporal logic specifications
- Test goals: certain faults must result in specification violation
- **Enforces** test goals for every implementation using **adaptive** test strategies
Test Case Generation Approach

Input: $I$
Output: $O$
Output' (not observable): $I'$, $O'$

$\delta (I,I',O,O')$ ... fault model
$\Phi_{corr} (I',O')$ ... specification of correct system behavior
$\Phi_{obs} (I,O)$ ... observable behavior w. r. t. the specification

$(\delta \land \Phi_{corr}) \rightarrow \neg \Phi_{obs}$
Fault models

- **Frequency**
  - Permanent fault (globally)
  - From some point on permanent (eventually globally)
  - ...
  - Occurs only once (eventually)

- **Fault description**
  - Bit flip \((o_i \leftrightarrow \neg o'_i)\)
  - Stuck at zero/one \((o_i = 0/1)\)
  - Delayed signal \((X(o_i) \leftrightarrow o'_i)\)
  - ...

...
Motivating Example – Test Strategy

Permanent stuck-at-0 fault of \( p \)

Stuck-at-0 fault of \( p \) that occurs from some point in time onwards
TABLE I
Results for the AMBA bus arbiter. The suffix “k” multiplies by $10^3$.

<table>
<thead>
<tr>
<th>Fault</th>
<th>( \alpha ) (( \alpha | = \alpha ))</th>
<th>Decide Next</th>
<th>Start Access</th>
<th>Grant Bus</th>
<th>Full Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>freq</td>
<td>(</td>
<td>T</td>
<td>)</td>
<td>sec</td>
</tr>
<tr>
<td>hmaster0</td>
<td>FG 2</td>
<td>359</td>
<td>147</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>hgrant0</td>
<td>F 2</td>
<td>18</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>hgrant1</td>
<td>-</td>
<td>856</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>hmasterlock start</td>
<td>-</td>
<td>803</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>locked decide</td>
<td>G 2</td>
<td>736</td>
<td>peak: 5,74 MB</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>decide</td>
<td>G 2</td>
<td>689</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>hmaster0 (( \alpha | = \alpha ))</td>
<td>FG 2</td>
<td>1,237</td>
<td>56</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>hgrant0</td>
<td>F 2</td>
<td>6,775</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>hgrant1</td>
<td>F 2</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>hmasterlock start</td>
<td>G 2</td>
<td>9,64</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>locked decide</td>
<td>GF 2</td>
<td>800</td>
<td>peak: 783 MB</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>decide</td>
<td>GF 2</td>
<td>1,011</td>
<td>-</td>
<td>-</td>
<td>GF 2</td>
</tr>
<tr>
<td>hmaster0 (( \alpha | = \alpha ))</td>
<td>G 2</td>
<td>22k</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>hgrant0</td>
<td>F 2</td>
<td>29</td>
<td>-</td>
<td>-</td>
<td>F 2</td>
</tr>
<tr>
<td>hgrant1</td>
<td>F 2</td>
<td>38</td>
<td>-</td>
<td>-</td>
<td>F 2</td>
</tr>
<tr>
<td>hmasterlock start</td>
<td>G 2</td>
<td>3,385</td>
<td>-</td>
<td>-</td>
<td>G 2</td>
</tr>
<tr>
<td>locked decide</td>
<td>GF 2</td>
<td>1,525</td>
<td>peak: 6,176 MB</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>decide</td>
<td>F 3</td>
<td>61</td>
<td>-</td>
<td>-</td>
<td>GF 2</td>
</tr>
</tbody>
</table>

Timeout (> 6 days for first output)
Door locked with a PIN

**TABLE II**

Results for the door specification.

| Fault      | \( o_i \) | freq | \(|T|\) | sec | MB |
|------------|-----------|------|--------|-----|----|
| stuck-at-0 |           |      |        |     |    |
| doorclosed | GF        | 25   | 22,341 | 347 |    |
| doorlocked | FG        | 29   | 2,425  | 285 |    |
| stuck-at-1 |           |      |        |     |    |
| doorclosed | GF        | 45   | 23,290 | 1,000 |  |
| doorlocked | FG        | 52   | 3,100  | 148 |    |
Conclusion

- Automatic generation of adaptive test strategies from temporal logic specifications
- Independent from implementation details
- No complete information necessary
- Discovers faults that are described in the fault model
Thank you for your attention 😊