Towards Symbolic Model-Based Mutation Testing:
Combining Reachability and Refinement Checking

Bernhard K. Aichernig, Elisabeth Jöbstl

Institute for Software Technology
Graz University of Technology

ejoebstl@ist.tugraz.at
www.ist.tugraz.at

Trufal
Trust via Failed Falsification
Model-Based Testing

Requirements

model

conformance

SUT (black box)

test case generator

verdicts

test case executor
Model-Based Mutation Testing

Requirements

model

SUT (black box)

test case generator

verdicts

fault models

mutated models

conformance
Outlook

• Action Systems
• Conformance
• Refinement Checking
• Experimental Results
• Future Work & Conclusions
Action Systems

- Ralph-Johan Back
- Guarded commands
- Reactive systems
- Non-determinism

```prolog
type(int, X) :- X in 0..10.
var([v₁, v₂], int).
state_def([v₁, v₂]).
init([0,0]).

as :-
   actions ( 
     A₁ :: g₁ => v₁ := e₁ ,
     ... ,
     Aₘ(X) :: gₘ => (g => (v₁ := X ; v₂ := e₂))
   ),
   dood ( 
     A₁ [] ... [] Aₘ
   ).
```
Semantics of Actions

- Predicative Semantics
- UTP (Unifying Theories of Programming)

\[ l :: g \Rightarrow B \quad =_{df} \quad g \land B \land tr' = tr^\sim[l] \]

\[ l(X) :: g \Rightarrow B \quad =_{df} \quad \exists X : g \land B \land tr' = tr^\sim[l(X)] \]

\[ x := e \quad =_{df} \quad x' = e \land y' = y \land \ldots \land z' = z \]

\[ g \Rightarrow B \quad =_{df} \quad g \land B \]

\[ B(\vec{v}, \vec{v}') \land B(\vec{v}, \vec{v}') \quad =_{df} \quad \exists \vec{v}_0 : B(\vec{v}, \vec{v}_0) \land B(\vec{v}_0, \vec{v}') \]

\[ B \parallel B \quad =_{df} \quad B \lor B \]
Conformance

• UTP‘s Refinement:

\[ M \subseteq I \equiv_{df} \forall x, x', y, y', \ldots \in \alpha : I \Rightarrow M \quad \text{for all } M, I \text{ with alphabet } \alpha. \]

• We want to find counterexamples for refinement, i.e., cases where \( M^O \not\subseteq M^M \)

\[ \exists x, x', y, y', \ldots \in \alpha : M^M \land \neg M^O \]

• Unsafe state:

\[ u \in \{ s \mid \exists s' : M^M(s,s') \land \neg M^O(s,s') \} \]
Example: Non-Refinement

\[ \exists x, x', y, y', \ldots \in \alpha : M^M \land \neg M^O \]
(Non-)Refinement of Action Systems

- Consider reachability:

\[ \exists \bar{v}, \bar{v}', tr, tr' : (\bar{v} \in \text{reachable}(AS^O, tr) \land (A_1^M \lor \cdots \lor A_n^M) \land \neg A_1^O \land \cdots \land \neg A_m^O) \]

- Non-refinement:

\[ \forall i = 1^n \exists \bar{v}, \bar{v}', tr, tr' : (\bar{v} \in \text{reachable}(AS^O, tr) \land A_i^M \land \neg A_1^O \land \cdots \land \neg A_m^O) \]

- By application of distributive law:

\[ \exists \bar{v}, \bar{v}', tr, tr' : (\bar{v} \in \text{reachable}(AS^O, tr) \land A_1^M \land \cdots \land \neg A_m^O) \]

Is the unsafe state reachable from the initial state?
Searching Unsafe States

- **AS**
- **ASM**

1. **find mutated action** → **mutated action found**
   - yes: **non-refinement constraint**
   - no: **equivalent mutant**

2. **non-refinement constraint**
   - yes: **reachability & non-refinement**
   - no: **max. depth reached**

3. **reachability & non-refinement**
   - yes: **refined**
   - no: **unsafe state & trace**

4. **unsafe state & trace**
   - yes: **test case extraction**
   - no: **end**

**AS**: no, yes
**ASM**: no, yes
**Mutated Action Found**: yes, no
**Non-refinement Constraint**: yes, no
**Max. Depth Reached**: yes, no
**Refined**: yes, no
**Unsafe State & Trace**: yes, no
**Test Case Extraction**: no, yes
**Equivalent Mutant**: yes, no
**End**: yes, no
Finding a Mutated Action

- refinement check action per action of ASM

\[ \bigwedge_i A^M_i \land \lnot A^O_1 \land \ldots \land \lnot A^O_m \]

- fast
Reaching an Unsafe State

- Breadth-first search: transition relation as constraints
- For each state: check non-refinement constraint

Algorithm 1 gives details on the action-wise non-refinement check, which is depicted on the left-hand side of Figure 4 (box 1). We transform the whole do-od block of the original into the non-refinement constraint for the just translated action is then given to a constraint solver to check non-conformance. We use this for our implementation as we perform the non-refinement check action by action. Here, we first concentrate on finding a possibly unreachable state. Reachability is dealt with separately (see Section 4.3).

For each state: check non-refinement constraint.

For each state: check non-refinement constraint.

The non-refinement condition presented in Theorem 4.1 is a disjunction of constraints of which each non-refinement condition is a disjunction of constraints of which each side of a sequential composition is always deterministic and existential quantification can be eliminated. In this way, the left-hand side of a sequential composition is always deterministic and existential quantification can be eliminated.

Our car alarm system example (cf. Listing 1) already satisfies this normal form. Otherwise, each action lead to quantified constraint satisfaction problems (QCSPs) that are not supported by common constraint solvers. Fortunately, we can resolve this problem by a normal form that requires that non-deterministic choice is always the outermost operator and not allowed in nested expressions. In this way, the left-hand side of a sequential composition is always deterministic and existential quantification can be eliminated.

The non-refinement condition for the just translated action is then given to a constraint solver to check non-conformance. We use this for our implementation as we perform the non-refinement check action by action. Here, we first concentrate on finding a possibly unreachable state. Reachability is dealt with separately (see Section 4.3).
Empirical Evaluation: Car Alarm System (CAS)

AlarmSystem_StateMachine

- **OpenAndUnlocked**
  - Open
  - Close
  - Unlock

- **ClosedAndUnlocked**
  - Lock
  - Unlock
  - 20
  - Unlock

- **ClosedAndLocked**
  - 30
  - Lock
  - Unlock

- **OpenAndLocked**
  - Close
  - Open

- **Armed**
  - Show Armed /entry
  - Show Unarmed /exit
  - Unlock

- **SilentAndOpen**
  - Open

- **Alarm**
  - Activate Alarms /entry
  - Deactivate Alarms /exit
  - FlashAndSound
  - Flash
  - 30 / Deactivate Sound
  - 300
Empirical Evaluation: CAS

% definitions: types, variables, state, initial state
as :-
  actions (  
    'after'(Wait_time)::(true) => (  
      ((Wait_time $\neq$ 20 $\lor\ aState$ $\neq$ 3) =>  
        (aState := 2; fromClosedAndLocked_OR_fromSilentAndOpen := 1))  
    [],  
      ((Wait_time $\neq$ 30 $\lor\ aState$ $\neq$ 1 $\lor\ fromArmed$ $\neq$ 4) =>  
        (aState := 0; fromAlarm := 4; fromArmed := 0))  
    [],  
      ((Wait_time $\neq$ 270 $\lor\ aState$ $\neq$ 0 $\lor\ fromAlarm$ $\neq$ 2) =>  
        (aState := 7; fromAlarm := 1; fromArmed := 0))  
    ),  
    'Lock'::(true) => (  
      ((aState $\neq$ 6 $\lor\ fromAlarm$ $\neq$ 0) => (aState := 5))  
    [],  
      ((aState $\neq$ 4 $\lor\ fromArmed$ $\neq$ 1) => (aState := 3; fromArmed := 0))  
    ),  
    ...  
  ),  
  dood ( 'Lock' [] [X:int]:'after'(X) [] ... ).
Empirical Evaluation: Mutations

- Manual mutations:
  - guard true: 34 mutants
  - comparison operator inversion: 52 mutants
  - increment integer constant: 116 mutants

\[ 206 \text{ mutants} + 1 \text{ unaltered (original)} = 207 \text{ mutants} - 12 \text{ mutants (constraint solver problems)} = 195 \text{ mutants} \]
<table>
<thead>
<tr>
<th>CAS version</th>
<th>Refinement checker</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>find mutated</td>
<td>reach &amp;</td>
<td>total</td>
<td></td>
</tr>
<tr>
<td></td>
<td>action</td>
<td>non-refine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20/30/270</td>
<td>total</td>
<td>16</td>
<td>90</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>0.08</td>
<td>0.46</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>0.30</td>
<td>2.80</td>
<td>3.10</td>
</tr>
<tr>
<td>*10</td>
<td>total</td>
<td>15</td>
<td>86</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>0.08</td>
<td>0.44</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>0.27</td>
<td>2.80</td>
<td>3.07</td>
</tr>
<tr>
<td>*100</td>
<td>total</td>
<td>16</td>
<td>90</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>0.08</td>
<td>0.46</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>0.27</td>
<td>2.77</td>
<td>3.04</td>
</tr>
<tr>
<td>*1000</td>
<td>total</td>
<td>15</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>0.08</td>
<td>0.44</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>0.27</td>
<td>2.69</td>
<td>2.96</td>
</tr>
</tbody>
</table>
Refinement Checker (symbolic)

### Time for 195 mutants

<table>
<thead>
<tr>
<th></th>
<th>find mutated action</th>
<th>reach &amp; non-refine</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>106</td>
<td>101</td>
<td>106</td>
</tr>
<tr>
<td>CAS *10</td>
<td>90</td>
<td>86</td>
<td>90</td>
</tr>
<tr>
<td>CAS *100</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>CAS *1000</td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Minimum time for 1 mutant

<table>
<thead>
<tr>
<th></th>
<th>find mutated action</th>
<th>reach &amp; non-refine</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>CAS *10</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>CAS *100</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>CAS *1000</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Average time for 1 mutant

<table>
<thead>
<tr>
<th></th>
<th>find mutated action</th>
<th>reach &amp; non-refine</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>0.54</td>
<td>0.52</td>
<td>0.54</td>
</tr>
<tr>
<td>CAS *10</td>
<td>0.46</td>
<td>0.44</td>
<td>0.46</td>
</tr>
<tr>
<td>CAS *100</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>CAS *1000</td>
<td>0.08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Maximum time for 1 mutant

<table>
<thead>
<tr>
<th></th>
<th>find mutated action</th>
<th>reach &amp; non-refine</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>3.1</td>
<td>2.8</td>
<td>3.07</td>
</tr>
<tr>
<td>CAS *10</td>
<td>2.8</td>
<td>2.8</td>
<td>2.77</td>
</tr>
<tr>
<td>CAS *100</td>
<td>0.3</td>
<td>0.27</td>
<td>0.27</td>
</tr>
<tr>
<td>CAS *1000</td>
<td>2.96</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Explicit ioco checking (Ulysses)

Action System Model

IOLTS\textsuperscript{S}

Mutated Action Systems

IOLTS\textsuperscript{M}

discriminating test case

for every mutant

ioco ?
| CAS version | Refinement checker | Ulysses |  |
|-------------|--------------------|---------|
|             | find mutated       | reach & non-refine | total | in/out | out |  |
| 20/30/270   | total              | 16 | 90 | 106 | 98 | 65 |  |
|             | average            | 0.08 | 0.46 | 0.54 | 0.50 | 0.34 |  |
|             | min                | 0.01 | 0.02 | 0.03 | 0.05 | 0.05 |  |
|             | max                | 0.30 | 2.80 | 3.10 | 6.30 | 5.33 |  |
| *10         | total              | 15 | 86 | 101 | 8.8 h | 7.9 h |  |
|             | average            | 0.08 | 0.44 | 0.52 | 2.7 min | 2.4 min |  |
|             | min                | 0.01 | 0.02 | 0.03 | 0.45 | 0.36 |  |
|             | max                | 0.27 | 2.80 | 3.07 | 2.6 h | 2.6 h |  |
| *100        | total              | 16 | 90 | 106 | - | - |  |
|             | average            | 0.08 | 0.46 | 0.54 | - | - |  |
|             | min                | 0.01 | 0.02 | 0.03 | - | - |  |
|             | max                | 0.27 | 2.77 | 3.04 | - | - |  |
| *1000       | total              | 15 | 85 | 100 | - | - |  |
|             | average            | 0.08 | 0.44 | 0.52 | - | - |  |
|             | min                | 0.01 | 0.02 | 0.03 | - | - |  |
|             | max                | 0.27 | 2.69 | 2.96 | - | - |  |
Ulysses (explicit)

<table>
<thead>
<tr>
<th>Time for 195 mutants</th>
<th>Minimum time for 1 mutant</th>
</tr>
</thead>
<tbody>
<tr>
<td>in/out</td>
<td>out</td>
</tr>
<tr>
<td>31680</td>
<td>28440</td>
</tr>
<tr>
<td>0.45</td>
<td>0.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Average time for 1 mutant</th>
<th>Maximum time for 1 mutant</th>
</tr>
</thead>
<tbody>
<tr>
<td>in/out</td>
<td>out</td>
</tr>
<tr>
<td>162</td>
<td>144</td>
</tr>
<tr>
<td>9360</td>
<td>9360</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>seconds</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>CAS *10</td>
</tr>
<tr>
<td>CAS 20/30/270</td>
<td>CAS *10</td>
</tr>
</tbody>
</table>
Future Work

• Other constraint solvers, e.g. MINION
• SMT solvers (ongoing diploma thesis)
• Trace to unsafe state $\rightarrow$ adaptive test case
• More experiments with different systems
Conclusions

- symbolic vs. explicit

**Time for 195 mutants**

<table>
<thead>
<tr>
<th></th>
<th>find mutated action</th>
<th>reach &amp; non-refine</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>106</td>
<td>101</td>
<td>100</td>
</tr>
<tr>
<td>CAS *10</td>
<td>90</td>
<td>86</td>
<td>90</td>
</tr>
<tr>
<td>CAS *100</td>
<td>106</td>
<td>101</td>
<td>100</td>
</tr>
<tr>
<td>CAS *1000</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

- **Time for 195 mutants**

<table>
<thead>
<tr>
<th></th>
<th>in/out</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS 20/30/270</td>
<td>98</td>
<td>65</td>
</tr>
<tr>
<td>CAS *10</td>
<td>31680</td>
<td>28440</td>
</tr>
</tbody>
</table>

Elisabeth Jöbstl
Tallinn, March 25th 2012
MBT 2012
Conclusions

- symbolic vs. explicit

Thank you for your attention!