CPA-SymExec
Efficient Symbolic Execution in CPAchecker

Dirk Beyer and Thomas Lemberger

OVERVIEW

CPA-SymExec is a symbolic-execution engine for C programs, implemented in CPAchecker. It tackles the path-explosion problem of symbolic execution with counterexample-guided abstraction refinement (CEGAR). In the context of symbolic execution, it provides:
- Generation of executable test cases for condition coverage
- Concrete, symbolic and executable program traces
- Interactive, visual analysis reports based on HTML

For examples of these, have a look at the demo or the YouTube video.

SYMBOLIC EXECUTION [4]

- Idea: Testing with symbolic values.
- Path constraints restrict these.

```
unsigned char a = 7;
unsigned char b = 7;
unsigned char c = b + 1;
while (a < 100)
  if (c == b) error();
```

⇒ Issue with scalability: path explosion.
Because of high precision, amount of states may grow exponentially and loops may be unrolled infinitely.

SYMBOLIC EXECUTION WITH CEGAR [1]

- Abstraction: ("Precision")
Which symbolic memory and path constraints to track.
- Counterexample check:
Traditional symbolic execution over found counterexample.
- Abstraction Refinement:
Trial & Error based on Craig interpolation [3]; “Information ‘x’ needed to show counterexample infeasible?” ⇒ track ‘x’

EXPERIMENTAL RESULTS

Task: Find calls to error function.
4 cores (3.4 GHz), 900 s CPU time, 15 GB memory,

<table>
<thead>
<tr>
<th>Tool</th>
<th>correct TRUE</th>
<th>incorrect FALSE</th>
<th>correct TRUE</th>
<th>incorrect FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPA-SymExec</td>
<td>2137</td>
<td>545</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>KLEE</td>
<td>446</td>
<td>899</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>Symbiotic</td>
<td>1201</td>
<td>848</td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

REFERENCES