Reachability Testing of Semaphore-based Programs

Yu Lei, Univ. of Texas at Arlington
Richard Carver, George Mason Univ. Fairfax

International Computer Software and Applications Conference, 2004

Presentation by Daniel Schweizer
### An example program

S1.count = 2; S2.count = 1

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1.down</td>
<td>S1.down</td>
<td>S1.down</td>
</tr>
<tr>
<td></td>
<td>-- T1 in critical section</td>
<td>-- T2 in critical section</td>
<td>-- T3 in critical section</td>
</tr>
<tr>
<td></td>
<td>print(&quot;1&quot;)</td>
<td>print(&quot;2&quot;)</td>
<td>print(&quot;3&quot;)</td>
</tr>
<tr>
<td></td>
<td>S1.up</td>
<td>S2.up</td>
<td>S1.up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2.down</td>
<td>S2.down</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-- T2 in critical section</td>
<td>-- T3 in critical section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>print(&quot;2&quot;)</td>
<td>print(&quot;3&quot;)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S1.up</td>
<td>S1.up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2.up</td>
<td>S2.up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S1.up</td>
<td></td>
</tr>
</tbody>
</table>

Possible outputs:

1233, 1332, 2331, 3231, ...
An example program

- possible outputs (determined statically):
  1233, 1332, 2331, 3231, ...

- actual outputs (20 test runs):
  - 1332  3 x
  - 1233  17 x
  - but: 2331  0 x

- Problem: We did not observe all feasible executions when testing!
Definitions

- when a thread $T$ calls down or up on a semaphore $S$, a call event is performed by $T$
- when a down or up operation on a semaphore $S$ is completed, a completion event occurs on $S$
- an execution of a semaphore-based program is characterized by the sequence of call and completion events it exercises, called the CC-sequence of the execution
- if the operation of a call event $c$ is completed by a completion event $e$, then $c$ and $e$ form a completion pair $<c, e>$
Race

- $Q$: a CC-sequence exercised by an execution of a semaphore-based program CP
- $c, c'$: call events in $Q$ ($c \neq c'$)
- $e$: completion event in $Q$
- $<c, e>$ is a completion pair

- there is a **race** between $c'$ and $<c, e>$ in $Q$ if $c'$ and $e$ can form a completion pair in another execution $Q'$ of CP, provided that all the events that happen before $c'$ or $e$ in $Q$ are replayed in $Q'$
Q
("3123")

c_2

down
e_1

c_3

down
e_2

c_4

down
e_3

c_5

down
e_4

c_6

up
e_5

c_7

down
e_6

c_8

up
e_7

c_9

up
e_8

c_{10}

up
e_{10}
Race variant

- $Q$: a CC-sequence

- a **race variant** $V$ of $Q$ is a CC-sequence that is derived by changing the call partner of one or more completion events in $Q$, with the following constraints:
  - if we change the call partner of a completion event $e$,
    1) there must be a race between the new call partner of $e$ and the completion pair $<c, e>$ in $Q$
    2) we must remove all events that happen after $e$
Reachability-Test

Reachability-Test (CP: a semaphore-based program)

\[ \text{do} \]

variants = $\emptyset$
 collect a CC-sequence $Q_0$ by executing $CP$ non-deterministically
derive variants($Q_0$) -- the race variants of $Q_0$
variants = variants $\cup$ variants($Q_0$)

while variants not empty loop

withdraw a variant $V$ from variants
 collect a CC-sequence $Q$ using prefix-based testing with $V$
derive variants($Q$) -- the race variants of $Q$
variants = variants $\cup$ variants($Q$)

end

end
Results (1)

“Theorem: Let CP be a semaphore-based program. Assume that every execution of CP with input l terminates. Then, algorithm Reachability-Test terminates, and executes all feasible CC-sequences of CP with input l. “

- no proof
## Results (2)

<table>
<thead>
<tr>
<th>Program</th>
<th>Configuration</th>
<th># of Seqs</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>3P + 3C + 2S</td>
<td>324</td>
</tr>
<tr>
<td>BB</td>
<td>2P + 2C + 2S</td>
<td>12</td>
</tr>
<tr>
<td>RW</td>
<td>2R + 2W</td>
<td>608</td>
</tr>
<tr>
<td>RW</td>
<td>2R + 3W</td>
<td>12816</td>
</tr>
<tr>
<td>RW</td>
<td>3R + 2W</td>
<td>21744</td>
</tr>
<tr>
<td>DP</td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

Open questions:
- comparison with other methods?
- performance?
- what about larger programs?
Discussion