Systematic State Space Exploration for Event-driven Multi-threaded Programs

Pallavi Maiya & Aditya Kanade
Dept. of Computer Science & Automation, IISc

Event-driven Multi-threaded Programs

Event model – Queue the events as they arrive and execute handlers.
Non-determinism: Order of arrival of events

Thread model – Concurrent execution of tasks on different threads.
Non-determinism: Interleaving of operations executed on different threads

Most of the existing techniques focus on the thread model.

Research Objective: Develop techniques to detect concurrency bugs in event-driven multi-threaded programs.

DroidRacer Workflow and Results

- Acyclic graph representation of happens-before constraints.
- Nodes: operations in trace
- Edges: happens-before relation
- Saturate the graph with happens-before rules
- Report conflicting memory operations with no happens-before relation as race.

- Debugging assistance
- Method stack, high level events

- Classification of reported data races
  - Races across threads – 85 potential races
  - Races across handlers on the same thread
  - Cross-post races: 423 potential races
  - Co-enabled event races: 156 potential races
  - Delayed post races: 49 potential races

Tested on 15 Android applications including Facebook, Twitter, MyTracks, K-9 Mail…

Race Detection for Android Programs

Android programs are event-driven and multi-threaded.

Our Contributions

Formalized concurrency semantics of Android applications.

Defined happens-before relation reasoning about causal ordering across threads and across event handlers.
- Algorithm to detect both single-threaded & multi-threaded data races.

DroidRacer – a dynamic race detector.
- Performs systematic UI testing.
- Identified potential races in popular applications.

Systematic State Space Exploration

Scheduling non-determinism gives rise to a huge state space for multi-threaded programs.

Finding concurrency bugs requires systematic state space exploration techniques like model checking.
Partial Order Reduction minimizes redundant explorations by model checkers.

Example Races in an Android app

System

Application process

Event model

- LAUNCH ACTIVITY
- DESTROY ACTIVITY

Thread model

- bind thread (bt)
- Main thread (mt)

Experimentally evaluated on DroidRacer.

Threaded Programs

POR for Event-driven Multi-threaded Programs

- Existing POR techniques primarily for multi-threaded programs.
- Based on equivalence called Mazurszkiwicz traces induced by a notion of independence between operations.

Our Contributions

- Dependence relation suitable for event-driven programs.
- A new notion of similarity between sequences called dependence-covering sequences.
- A new backtracking set called dependence-covering sets, which preserve deadlock cycles and assertion violations.
- Preliminary experimental evaluation showing the scalability of dependence-covering sets compared to persistent sets, for event-driven programs.

Experimental Evaluation

<table>
<thead>
<tr>
<th>Android Apps</th>
<th>DPOR Sequences explored</th>
<th>Time taken</th>
<th>EM-DPOR Sequences explored</th>
<th>Time taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remind Me</td>
<td>24</td>
<td>0.18s</td>
<td>3</td>
<td>0.05s</td>
</tr>
<tr>
<td>My Tracks</td>
<td>1610684</td>
<td>TIMEOUT</td>
<td>405013</td>
<td>101m</td>
</tr>
<tr>
<td>Music Player</td>
<td>1508413</td>
<td>TIMEOUT</td>
<td>266</td>
<td>4.15s</td>
</tr>
<tr>
<td>Character Recognition</td>
<td>1284788</td>
<td>199m</td>
<td>756</td>
<td>6.58s</td>
</tr>
<tr>
<td>Aard Dictionary</td>
<td>359961</td>
<td>TIMEOUT</td>
<td>14</td>
<td>1.4s</td>
</tr>
</tbody>
</table>

| Time taken | 4 hours. |

DPOR – an algorithm to compute Persistent sets.
EM-DPOR – an algorithm to compute dependence-covering sets.

Related Publications and Tool Webpage

- Pallavi Maiya, Aditya Kanade, Rupak Majumdar. Race Detection for Android Applications. PLDI ’14
- Pallavi Maiya, Rahul Gupta, Aditya Kanade, Rupak Majumdar. Partial Order Reduction for Event-driven Multi-threaded Programs. TACAS 16
- DroidRacer tool page: http://www.iisc.ernet.net/droidracer