Runtime Verification

Lecture 9 : Specification Free Monitoring

http://www.runtime-verification.org/course

May 29, 2008

This nineth **and last** lecture introduces four algorithms for analyzing the execution of a concurrent Java program for various forms of data races and deadlocks. The four algorithms can detect (i) traditional low-level data races, (ii) high-level data races, (iii) stale value computations, an (iv) deadlocks. A traditional low-level data race occurs when two (or more) threads access a variable without any mechanism for guaranteeing mutual exclusion (and at least one thread writes to the variable). A high-level data race occurs when data that should be treated in one atomic lock-protected block is accessed in separate blocks where locks are released in between. State value computation refers to the situation when a thread reads a variable shared amongst threads into a local variable and continues to operate on this local variable, while the original shared variable gets updated, and hence the local variable now holds a stale (out of date) value. A deadlock refers to the situation where two or more threads attempt to acquire locks in a circular manner in such a way, that they wait for each other, and therefore make no progress. All four algorithms are based on dynamic analysis of a single execution trace. The algorithms are powerful since they only attempt to detect *potentials* for these errors. This can be orders of magnitudes more effective than just detecting when the errors actually do occur.

Installation

The course does not require any experimentation with any of these tools. However, the website does refer to the Racer data race detection tool for Java in case you want to experiment with it.

Reading

1. *Racer: Effective Race Detection Using AspectJ*, Eric Bodden and Klaus Havelund.

Optional Reading

1. *Eraser: A Dynamic Data Race Detector for Multithreaded Programs* (the first low-level data race detection algorithm for C and C++), Stefan Savage, Michael

Burrows, Greg Nelson, Patrick Sobalvarro, and Thomas Anderson.

2. *High-level Data Races*, Cyrille Artho, Klaus Havelund, and Armin Biere.
3. *Using Block-local Atomicity to Detect Stale-value Concurrency Errors*, Cyrille Artho, Klaus Havelund, and Armin Biere.
4. *Dynamic Deadlock Analysis of Multi-Threaded Programs*, Saddek Bensalem and Klaus Havelund.

Assignments

There are no assignments for this last lecture.