

Wikiprint Book

Title: Java Memory Model

Subject: Java Path Finder - summer-projects/2010-jmm

Version: 9

Date: 02/23/2013 10:35:41 AM

Table of Contents

Java Memory Model	3
Abstract	3
More Details	3
Contact	3
Program & Timeline	3
Repository	3
Description	3
Project Blog	3

Java Memory Model

Abstract

Java Racefinder ([JRF](#)) is an extension of Java PathFinder that precisely detects data races as defined in the Java Memory Model (JMM) in Java byte code. Detecting and eliminating data races is important because only programs that are free of these races are guaranteed to exhibit sequentially consistent (SC) behavior, and JPF considers only SC executions. The goal of this project is to extend JRF to implement a modular data race detection by incorporating the JPF symbolic execution engine and ASSUME-GUARANTEE reasoning. Environment abstraction with symbolic execution frees the developer from having to directly specify inputs into the system under test and provides a general testing environment. The system under test will be divided into separate modules and annotated individually with preconditions to guarantee race freedom. Those preconditions will be verified for correctness in a general environment and then will be used as an assumption to guarantee the race freedom in a specific environment which is a composition of modules.

More Details

- A [simple example](#) in a slide deck with more details from [KyungHee Kim](#) from her presentation at NASA Ames in May 2010

Contact

Student: [KyungHee Kim](#)

- `khkim "at" cise "dot" ufl "dot" edu`

Mentor: [Eric Mercer](#)

- `egm "at" cs "dot" byu "dot" edu` (EGM)

- **Co-mentor:** Neha Rungta `*neharungta "at" gmail "dot" com` (NR)

Program & Timeline

This project is funded by a NASA Ames Internship. It follows the GSoC timeline (start 05/24, finish 08/16). Specific milestones follow:

- **5/25 - 5/28** Discussion with JPF experts at NASA (1week)
- **5/29 - 6/19** Design of overall approach & Preliminary design document (2weeks)
 - define the interface/structure of the annotation to each model class to augment the environment information
 - define the structure of the precondition for a data race freedom
 - search for appropriate test suites
- **6/20 - 7/17** Coding & Testing (4weeks)
 1. parse & analyze the model class annotation about the environment
 2. generate the entry stub (main routine) using (1)
 3. calculate summary function h during a symbolic execution of (2) using JPF
 4. compute a race free precondition for each class from the result of (3) and save as an additional annotation
- **7/18 - 8/7** Applying to various test suites and modify the design document & codes as necessary (3weeks)
- **8/8 - 8/16** Final document (1week)

Repository

The sources for this project are available from a Mercurial repository on <http://bitbucket.org/>

Description

Project Blog

17-Jun-2010 Modified abstract and added talks

I updated the abstract to convey the focus of the work, and I uploaded the new presentations that matches the abstract --- egm

17-May-2010 Added Abstract

Added an abstract and time line from KyungHee. --- egm

07-May-2010 project page created

Added the initial information to the page. --- nr