

Wikiprint Book

Title: Project Name

Subject: Java Path Finder - summer-projects/2011-symexec-tree

Version: 2

Date: 02/23/2013 05:10:11 PM

Table of Contents

Project Name	3
Abstract	3
Contact	3
Repository	3
Description	3

Project Name

Memoized Symbolic Execution

Abstract

This project implements memoized symbolic execution, a new approach for more efficient application of forward symbolic execution, which is a well-studied technique for systematic exploration of program behaviors based on bounded execution paths. Our key insight is that application of symbolic execution often requires several successive runs of the technique on largely similar underlying problems, e.g., running it once to check a program to find a bug, fixing the bug, and running it again to check the modified program. The approach introduces a trie-based data structure that stores the key elements of a run of symbolic execution. Maintenance of the trie during successive runs allows re-use of previously computed results of symbolic execution without the need for re-computing them as is traditionally done. The approach holds benefits in various standard scenarios of using symbolic execution, e.g., with iterative deepening of exploration depth, to perform regression analysis, or to enhance coverage

Contact

student: Guowei Yang, UT Austin

mentor: Corina Pasareanu, CMU-SV/NASA Ames

co-mentor: Sarfraz Khurshid, UT Austin

Repository

The sources for this project are available from the jpf repository (jpf-memoise project)

Description

The project implements storing and retrieving the trie using JPF listeners and Java Serialization API.