Summary of

"Reconstruction of Core Dumps"

Nikhil Satish Pai Nikhil Anand

29th October 2015

Important Keywords:

***ii1. RECORE** - "Reconstruction core dumps" is an evolutionary search based test generation to reconstruct failures from saved core dumps for Java programs.*

***ii2. Memory Dumps or Core dumps** - A persistent snapshot which records all the values of the stack and heap memory of the program at the moment of failure.*

***ii3. Failure reproduction** - First step in debugging a failure in a program to determine the input combination which caused the failure.*

***ii4. Test case generation** - Generating input combinations which will cause the input program to throw the same exceptions and have the same stack trace at the time of throwing the exception.*

Brief notes:

iii1.Motivational Statement

Whenever a software program fails the developers must be able to reproduce the bug in order to locate the failure cause. A common method of diagnostics is the core dump. These only provide a snapshot of the final state of the program and does not provide any data regarding the origin of the problem. The given paper attempts to automatically generate a test case which will reproduce the core dump with a search based test generation without incurring additional overhead during execution

iii2.Future Work:

The paper only recreates test cases for core dumps from Java programs, this can be expanded to include C/C++ also.

Implementing a combination of search based and constraint based techniques to generate test cases and measure their performance.

iii4.Baseline Results :

The performance of RECORE is compared with an automated debugger BUGEX[ref], on 7 test cases, Three bugs in the Java JODATIME API, The vending Machine bug, two bugs in Commons Codec Base64 and Commons Math Sparse Iterator Bug.

Except for 2 bugs in the JODATIME API, All other generated test cases reconstruct the failure from scratch, using primitive values only which is achieved with zero overhead at runtime.

iii3.Related work:

[28] suggest a large scale study to with hundreds of users to measure the usefulness of RECORE and how well it integrates with other debugging tools.

[14] suggest synthesizing GUI events as inputs. Such generation techniques could also be integrated into RECORE.

Area of Improvement:

The current baseline only has 7 test which is very small and needs to be increased.
There are several parameters like weights, timeouts and thresholds which are used for the study, the effect on the runtime when these are varied needs to be investigated further.

Relation to Original paper :

The original paper suggest a technique combining search based and constraint based approaches to achieve high code coverage. In the current paper only uses symbolic execution to produce the required test case. The authors suggest using the techniques proposed in the original paper to improve the performance of the test case generator.

Reference to the Paper:

Jeremias Rossler, Andreas Zeller, Gordon Fraser, Cristian Zamfir, George Candea, "Reconstructing Core Dumps", *ICST*, 2013, 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation (ICST 2013), 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation (ICST 2013) 2013, pp. 114-123, doi:10.1109/ICST.2013.18.