A Survey of Symbolic Executions Techniques

Hallet Adrien Sens Loan

October 9, 2018

Abstract

1 Introduction

1.1 (Attempting) A definition

The first occurences of symbolic execution described the then-new method as a middle ground [3] between the two most-used method of its time. On one hand, program testing (*e.g.: unit testing*) can not always detect a fault in a program and producing a correct test sample and proving that it indeed is correct is not that easy. On the other hand, program proving can indeed ensure that a program is correct from its entry point to the result but it heavily relys on the proof definitions by the programmer and the formal definition of the problem.
Nowadays, symbolic execution is both described as (part of) the core of many modern techniques to software testing [4] and an effective way to create tests suites with extensive coverage. [1]

1.2 The concept

2 History

3 Methods

3.1 Concolic execution

The name "concolic" is a portmanteau of the words "concrete" and "symbolic", the idea of this testing method is to mix symbolic execution alongside concrete ones.

This technique concept was first introduced on 2005 [2]. Since then the idea was further extended and combined with other testing techniques.

4 Tools and languages

5 Conclusions

References

[1] Cristian Cadar and Koushik Sen. Symbolic execution for software testing: Three decades later. 56:82–90, 02 2013.

[2] Patrice Godefroid, Nils Klarlund, and Koushik Sen. Dart: directed automated random testing. In *ACM Sigplan Notices*, volume 40, pages 213–223. ACM, 2005.

[3] J.C. King. A new approach to program testing. 10:228–233, 06 1975.

[4] David Trabish, Andrea Mattavelli, Noam Rinetzky, and Cristian Cadar. Chopped symbolic execution. In *International Conference on Software Engineering (ICSE 2018)*, 5 2018.