What Tests Can Tell Us About Program Safety

Patrick Lam
University of Waterloo
Waterloo, Canada
patrick.lam@uwaterloo.ca

Abstract

Unit tests are ubiquitous in modern programming practice. Fundamentally, they encode expected program behaviour on selected inputs. Today, developers will compile tests and run them (in the best case, continuously, i.e. upon each commit) to gain some confidence that their code is behaving as expected.

Because tests encode expected behaviour, I believe that they can be a rich, and currently underused, source of information for program analysis tools. Tests themselves are generally simple straight-line code with concrete inputs, but they exercise arbitrary code in the System Under Test. Analyzing the test is easy, but analyzing what the system does as a result of the test is potentially complicated, and can inspire novel static and dynamic analysis techniques.

While concolic execution already leverages test inputs quite successfully (to find new inputs), this work aims for potential applications beyond test generation. For instance, writing assertions in the main code is challenging. This work could allow developers to write concrete tests, using a tool to lift these tests to more general assertions. Analyzing test cases could also help with reaching a more foundational understanding of how program behaviour may change when given different inputs.

Biography. Patrick Lam is an Associate Professor of Electrical and Computer Engineering at the University of Waterloo. His research interests focus on ways to use static and dynamic analysis techniques to help developers write better code. He is particularly interested in test code as well as program transformation.

Workshop on Compiler-Driven Performance, November 15, 2022, Toronto, CA
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn