Presentation on *Software Testing: A research Travelogue* (2000 - 2014)

Presented by: Ji Gao

1Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

August 26, 2018
Overview

1 Introduction

2 Research Topics
   1. Automated test input generation
   2. Testing strategies
   3. Regression Testing
   4. Empirical studies

3 Challenges and Opportunities
1. Introduction

2. Research Topics
   - 1. Automated test input generation
   - 2. Testing strategies
   - 3. Regression Testing
   - 4. Empirical studies

3. Challenges and Opportunities
Testing: Most popular and practised way to assess software quality
Many research progress in past years
Outline

1. Introduction

2. Research Topics
   1. Automated test input generation
   2. Testing strategies
   3. Regression Testing
   4. Empirical studies

3. Challenges and Opportunities
Outline

1. Introduction

2. Research Topics
   1. Automated test input generation
   2. Testing strategies
   3. Regression Testing
   4. Empirical studies

3. Challenges and Opportunities
Automated test input generation

Formally, generating a set of input values for a program or program component, typically with the aim of achieving some coverage goal or reaching a particular state.
Symbolic Execution
Automated test input generation

- Use symbols to replace variable in the code.
- Express the execution of the program as a bunch of paths.
- Test the validity of paths, and its corresponding range of input symbols.
- Symbolic state as a mapping $S : \mathcal{M} \rightarrow \mathcal{E}$, $\mathcal{M}$ is the set of memories, and $\mathcal{E}$ is the set of symbolic values.
- Execution: $S' = S \oplus [m \rightarrow e']$
White box fuzzing

- Concolic testing
- In the execution of a certain path, try to cover other path, generating new test cases by flipping the original input.
Machine learning models can be expressed as symbolic operation. However, the optimization process gives a huge tree that can’t be checked.
Search-Based Testing
Automated test input generation

- Use heuristic search-based optimization techniques (for example, hill climbing) to solve optimization problems.
- Have a variety of forms and wide applications.
- Challenge and opportunities:
  - Oracle problem
  - Combining SBST with symbolic testing
  - Co-evolutionary computation
  - Hyper-heuristic software engineering
Random testing
Automated test input generation

- Randomly generate test inputs
- Or not that randomly
- Adaptive Random Testing: Generate input that is the most ’distant’ from previous inputs
Combined techniques
Automated test input generation

- Combine testing with other verification skills: static verification, abstraction, ...
# Outline

1. **Introduction**

2. **Research Topics**
   - 1. Automated test input generation
   - 2. Testing strategies
   - 3. Regression Testing
   - 4. Empirical studies

3. **Challenges and Opportunities**
Testing strategies

Strategies of testing that can lead to a better result
Combinatorial testing

Testing strategies

- Combinatorial testing: Model the problem as a finite set of parameters. Find a combination of test inputs that covers all inputs.
- Weighted, biased, guided way of CIT?
Model-based Testing

Testing strategies

- Based on the specialty of every model, develop testing suites.
Use large data to help testing
Outline

1. Introduction

2. Research Topics
   1. Automated test input generation
   2. Testing strategies
   3. Regression Testing
   4. Empirical studies

3. Challenges and Opportunities
Regression Testing

- Verify whether a software previously tested still performs correctly after changed
- Attempt to select/modify a smaller test suite to complete the quest
Outline

1. Introduction

2. Research Topics
   - 1. Automated test input generation
   - 2. Testing strategies
   - 3. Regression Testing
   - 4. Empirical studies

3. Challenges and Opportunities
Emperical studies

- Experiments
- Frameworks
- Continuous integration
Outline

1 Introduction

2 Research Topics
   - 1. Automated test input generation
   - 2. Testing strategies
   - 3. Regression Testing
   - 4. Empirical studies

3 Challenges and Opportunities
Testing modern systems

- Heterogeneous software systems
- Rich environments
- High configurability
Oracle problem

- Inferring invariants from programs
- Rich environments
- High configuability
Other opportunities

- Probabilistic program analysis
- Testing non-functional properties
- Domain-based testing
- Cloud and crowd computation