Test Generators

17-939A
What Makes Good Research in Software Engineering?

Paul.Li@cs.cmu.edu

Purpose of Presentation

• Present my view of the development of Test Generators
• Elicit opinions regarding my views
• Elicit corrections regarding dates and events
• Elicit ideas regarding proceeding towards the comprehensive view of the maturation of Test Generators and the research strategies that contributed to its maturation.

Outline

• Purpose of Test Generators
• Uses
• Limitations
• Conclusion

Why Test Generators

• Testing is an effective means of ensuring the quality of software.
• There are three technical activities involved:
  – Creating
  – Executing
  – Evaluating Test cases
• Creating test cases is the most essential, from which all other activities follow.
What is a Test Generator

• An automated software that tool that creates test cases from… inputs. (Marciniak, 94)
• Inputs, in my opinion, are descriptions of the system.
  – Specifications
  – Code

Paul’s View on Test Generators

• There are two distinct paths
  – Code Based with concentrates on coverage criteria and the different methods of going about achieving that coverage.
  – Specification Based which takes the view of the program and creates test cases based on that view.

Paul’s View on Specification Based Test Generators

• The development/advancement of Specification Based Test Generators depend largely on the development of the Specification.
  – When a method of description become mature enough, a test generator based on that description is born.

Specification Based

• Formal Method Specifications
  – Zed
• Object-Oriented Models
  – UML
• Programming Language Semantics
  – Compilers
Formal Methods Specification Based

- Original Research Area into External Exploration:
  - First definition of Formal Specifications in 1987.

- Test Generator:

Object-Oriented Models

- Original Research Area into External Exploration:
  - UML 1.1 1991

- Test Generator:
  - Kim et al. : Testing by transforming UML into Finite State Machines, 1999

Programming Language Specification Based

- Original Research Area into External Exploration:
  - Stanford: XPL system, 1970

- Test Generator
  - Hanford: PL/1 Compiler, 1972
  - Houssias: Algol 68 Compiler : 1976

Development Timeline

- XPL System, 1970
- Grammar Test Generators, 1972
- STL IEEE, 1991
- Object Oriented Design, 1991
- Object State Testing, 1994
- Model-Based Specifications, 1993
Inter Area Development

- As testing criteria (coverage) becomes more complete, the development in each area is altered to meet that criteria or have arguments for their sufficiency in the view of the existing criteria.

Research Methodology

- **Question**: Feasibility: If the new method of program description can be used to produce a test generator?
- **Result**: System: A system is produced that uses the program program specification.
- **Validation**: Evaluation: Based on the existing criteria (or one specially specified by this model) the system produces the desired result.

Code Based

- There is more of a development process
- Everything is based on the amount of code/decisions cover.
- A clear development path exists where development generates new problems.
- Better solutions to existing problems are formed.

Development History

- Clarke 1976
- Ramanamoorthy, 1976
- Path Oriented
- Laski and Korel, 1983
- Branch Oriented
- Better Test Cases
- Coverage Possible?
- Better Coverage
- DeMillo, 1991
- Constraint Based
- Korel, 1990
- Goal Oriented (Dynamic)
- Rapps, Weyuker 1985
- Data Flow Analysis
- Roper et al., 1995
- Genetic Algorithms
Research Methodology

• **Question: Methods, Feasibility:** How do we get better coverage? Is the criteria feasible? How do we go about getting the best coverage using the least possible number of test cases?

• **Result: System, Models:** A system is produced that uses the generates test data. A analytical or usage model gives feasibility of coverage.

• **Validation: Evaluation, Analysis:** Based on the existing criteria the system produces the desired result. Analysis of the program gives result.

Uncertain Road Ahead

• In the Specification Based section there are many outliers, were there are early appearances of test generators; not necessarily until the external exploration phase.

• How to discern better general time periods for the Code Based section?

The End – Thank You

Please Email me with ideas/suggestions:

Paul.Li@cs.cmu.edu