

CSC 509 Lecture Notes Week 6

Survey of Test Coverage and Adequacy

I. Administrative Matters

A. This coming Wed, brief individual meetings.

1. I'll provide some concrete feedback on your project so far.
2. The schedule is posted at [509/admin/...](#)

II. Future Papers this Paper is Clairvoyant About

- A. *"Software Testing Research: Achievements, Challenges, Dreams"*, FOSE 2007
- B. *"A Survey of Combinatorial Testing"*, CSUR 2011
- C. *"Software Testing with an Operational Profile: OP Definition"*, CSUR 2014

III. Future and Contemporary Papers It Missed

- A.** *"How Did Software Get So Reliable without Proof?", 1996*

- B.** *"Search-Based Software Engineering: Trends, Techniques and Applications", 2012*

IV. Some Quotables from the Paper.

A. Dijkstra's 1972 claim that *"program testing can be used to show the presence of bugs, but never their absence"*.

B. The following is still (sadly) true:

"... it is common in current software testing practice that the test processes ... stop when money or time runs out"

... (sadly) true, cont'd

as evidenced by MS research paper from week 1:

"... for many projects it may be impractical to achieve complete coverage", aka, adequacy.

V. The gist of the paper.

A. Learn to appreciate that

1. There has been very considerable thought given to test adequacy criteria.
2. It's backed up by well established research.
3. Because it's hard to measure in practice, strong test adequacy remains an elusive goal.

B. Organizational structure of the paper:

1. Introduction

1.1. The Notion of Test Adequacy

1.2. Uses of Adequacy Criteria

1.3. Categories of Adequacy Criteria

1.4. Organization of Article

Organizational Structure, cont'd

2. Structural Testing

2.1. Program-Based, aka White-Box

2.2. Specification-Based, aka Black-Box

2.3. Summary

Organizational Structure, cont'd

3. Fault-Based

3.1. Error Seeding

3.2. Mutation

3.3. Mutation Variants

3.4. Perturbation

3.5. REPLAY Model

3.6. Spec Mutation

3.7. Summary

Organizational Structure, cont'd

4. Error-Based, Domain Analysis

4.1. Spec-Based Input Partitioning

4.2. Program-Based Input Partitioning

4.3. Boundary Analysis

4.4. Functional Analysis

4.5. Summary

Organizational Structure, cont'd

5. Comparison of Preceding

5.1. Fault-Detecting Ability

5.2. Software Reliability

5.3. Cost

5.4. Summary

Organizational Structure, cont'd

6. Axiomatic Assessment

6.1. Wyeuker

6.2. Parrish and Zweben

6.3. Zuh and Hall

6.4. Semantics

Organizational Structure, cont'd

7. Conclusion

VI. A Single Figure Worth 1000 Words

A. Figure 7 on Page 509 in Section 5.

B. Question -- "How did Programs get so reliable with barely adequate testing?"

VII. Some Technical Questions about the Paper

- A.** How does the concept of a testing oracle fit into the subject matter of the paper?

- B.** How do the concepts of white-box and black-box testing fit into the subject matter of the paper?

VIII. What You Should Know from CSC 30X

- A. The list of topics at the beginning of Section 1.1.
- B. This is as far I get in 30{7,9}.
- C. I.e.,
 1. *statement coverage*
 2. *branch coverage*
 3. *path coverage*
 4. *mutation adequacy*

IX. How Things Have Changed Since 1997

A. Early in the intro section of the paper they say:

"In contrast to the constant attention ... by academics, the software industry has been slow to accept test adequacy measurement."

X. How things haven't changed.

A. This area of research is pretty much done.

B. A rare bit of stability in our field.

XI. Practicalities -- When, Where, and How Much is Testing Theory is Necessary?

- A.** Consider in particular the mathematically formalized definitions on pages 368 - 369.
- B.** Are they useful for the practicing programmer?
 - 1.** As underlying concepts, absolutely.
 - 2.** As formalized mathematically definitions, very little if at all.

C. To make some use out of them,

1. The notion of the *stopping rule* is what your boss tells you have to achieve for coverage before you can ship your code.
2. The notion of the *adequacy measurement criteria* is (supposedly) what your boss uses to justify the choice of the coverage.

XII. Practicalities -- the Rise of Coverage Tools

- A.** There's no readily available concrete data from 20 years ago, but good (public domain) test coverage tools were really hard to come by.

- B.** Today they abound, which is really excellent!

XIII. What We're Continuing to Learn

- A.** Tools are king. However solid the theory may be, there gotsta be good usable tools before people pay attention to it.

- B.** It's cool that mutation testing doesn't require a coverage measure, but unfortunately it's way harder to develop a usable mutation tool than it is to develop a usable coverage tool.

What We're Continuing to Learn, cont'd

- C.** A combination of black box and white box testing makes the most sense, as opposed to use one or the other exclusively.

- D.** It's nice to have theory behind us, but pure theoretical beauty is not beautiful enough.