CMSC838M: Advanced Topics in Software Testing

- Atif M. Memon (atif@cs.umd.edu)
- 4115 A.V. Williams Building
- Phone: 301-405-3071
- Office Hours
  - Tu.Th. (3:30PM-6:30PM)
- Don’t wait, Don’t hesitate, Do Communicate!!
Interests

• Testing GUIs.
• Extend it to other types of software interfaces (such as web).
• Student introductions and their interests!!
Goals of the Course

- Discuss *advanced* software testing techniques
- Two parts of the course
  - Review testing fundamentals
  - State-of-the-art & emerging techniques
- What do I expect from students?
MS and Ph.D. Qualifying

• Is the course is valid for PhD qualifying coursework?
  - Yes (Software Engineering/Programming Languages)

• Is the course is valid for MS qualifying coursework?
  - Yes (Software Engineering/Programming Languages)

• Is the course is valid for MS comps?
  - Yes (Both Midterm and Final exams count towards the MS comps.)
Assessment

- 25% Mid-term Exam
- 25% Final Exam
- 20% Topic Presentation
  - 40 minutes
- 5% Project Presentation
  - 10 minutes
- 25% Term Project
  - chosen by the student and approved. May be a team (2 students) project, depending on the scope of the project)
Exam Contents

- **Midterm**
  - Everything discussed in class

- **Final exam**
  - Everything discussed/presented after midterm
Student Presentations

• Students must present a paper (perhaps from the list of suggested papers) on one of the listed topics
• 40 minutes
• Contents
  - Problem definition/motivation
  - What are the challenges?
  - Background literature surveyed by the authors
  - Specific technique developed in this paper
  - Weaknesses of the technique
Student Projects

- Develop a new testing technique to test a particular type of software and experimentally demonstrate that it works (or does not work!!)
- Project Proposal Proposal Presentations
- Project Presentations

- Possible future research opportunities, publications, and conference talks
- Questions?
Testing: Our Experiences

Software to be tested
Testing: Our Experiences

Test Case

Software to be tested
Testing: Our Experiences

Test Case

Software to be tested

Output

Testing: Our Experiences
Testing: Our Experiences
When to Stop?

1. Software to be tested
2. Test Case
3. Output

- Enough?
  - No
  - Yes
When to Stop?

Test Case Generation

Test Case

Software to be tested

Output

Enough? Yes No

Verification

Test Coverage
A Real Testing Example

Sorting Program
A Real Testing Example

SPECS:
Takes a list of numbers; returns a sorted list.
A Real Testing Example

\{1,3,2\}

Sorting Program
A Real Testing Example

Test Cases

\{1,3,2\}
A Real Testing Example

Test Cases

\{1,3,2\}

Just a list.

Sorting Program
A Real Testing Example

Test Cases
{1,3,2}

Sorting Program

Output
{1, 2, 3}
# A Real Testing Example

**Test Cases**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,3,2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{1,2,3}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sorting Program**
# A Real Testing Example

*Test Cases*

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,3,2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{1,2,3}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A sorted list.

Sorting Program
A Real Testing Example

Test Cases

- \{1,3,2\}
- \{1,2,3\}

Sorting Program

\{1,2,3\}

Output
A Real Testing Example

**Test Cases**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,3,2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{1,2,3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{3,2,3}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sorting Program
A Real Testing Example

Test Cases

\begin{tabular}{|c|}
\hline
\{1,3,2\} \\
\{1,2,3\} \\
\{3,2,3\} \\
\hline
\end{tabular}

Repeated entry.

Sorting Program
A Real Testing Example

Test Cases

- \{1,3,2\}
- \{1,2,3\}
- \{3,2,3\}

Sorting Program

Output

\{2, 3, 3\}
A Real Testing Example

Test Cases

{1,3,2}

{1,2,3}

{3,2,3}

{}
A Real Testing Example

Test Cases

| {}   | 1,3,2 | 1,2,3 | 3,2,3 | {}   |

Empty list.
A Real Testing Example

Test Cases

{1,3,2}
{1,2,3}
{3,2,3}
{}

Sorting Program

Output

{ }
A Real Testing Example

<table>
<thead>
<tr>
<th>Test Cases</th>
<th>Sorting Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,3,2}</td>
<td></td>
</tr>
<tr>
<td>{1,2,3}</td>
<td></td>
</tr>
<tr>
<td>{3,2,3}</td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>{-1, -2}</td>
<td></td>
</tr>
</tbody>
</table>
# A Real Testing Example

## Test Cases

<table>
<thead>
<tr>
<th>Test Cases</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,3,2}</td>
<td></td>
</tr>
<tr>
<td>{1,2,3}</td>
<td></td>
</tr>
<tr>
<td>{3,2,3}</td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>{-1, -2}</td>
<td></td>
</tr>
</tbody>
</table>

*Negative numbers.*
A Real Testing Example

Test Cases

{1,3,2}
{1,2,3}
{3,2,3}
{}
{-1, -2}

Sorting Program

Output

{-2, -1}
Automated Testing

Test Case Generation

Test Case

Software to be tested

Output

Verification

Enough?

No

Test Coverage

Yes
Automated Testing

- Test Case Generator
- Test Case
- Software to be tested
- Output
- Coverage Evaluator
- Test Oracle
- Verifier
- Test Specs
Automated Testing

- Test Case Generator
- Test Case
- Software to be tested
- Output
- Coverage Evaluator
- Test Oracle
- Verifier
- Test Specs
Testing the New Version

- Original Test Cases
- Original Software
- Modified Software
- New Test Cases
Regression Testing

- Original Test Cases
- Original Software
- Modified Software
- New Test Cases
Regression Testing

Original Test Cases

Original Software

Modified Software

New Test Cases
Different Software

- Object-oriented
- Component-based
- Concurrent
- Distributed
- Graphical-user Interfaces
- Web