INF 111 / CSE 121: Software Tools and Methods

Lecture Notes for Fall Quarter, 2007
Michele Rousseau
Set 11

Announcements

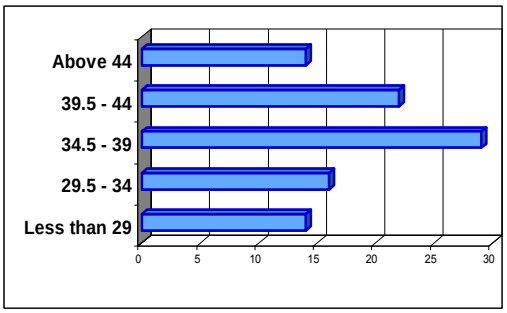
- **Quiz #2 – Monday 10/29/07**
  - Will not include the Ch 2 from "The Mythical Man-Month"
  - Will include – all other readings assigned since the last Quiz
  - Everything in lecture on Wed. 10/17 on and including everything on testing from 10/12 (Slide sets 7 through slide set 12)
  - Van Vliet Ch. 4 will not be included on this quiz
- **Lab 4 will be posted later today**
- **Read: The Mythical Man-Month – CH 2:**
  - "The Mythical Man-Month:

Quiz Results

- **Max → 49**
- **Min →14**
- **Median 37.5**

* Not including the 4 who did not turn it in

Breakdown of grades



Some perspective

- **Quiz is worth 5% of your grade**
- **If I miss one can I still get an A?**
- **I suffer from test anxiety – what can I do?**
  - http://www.studygs.net/tstprp8.htm
  - http://ub-counseling.buffalo.edu/stresstestanxiety.shtml
  - http://www.sdc.uwo.ca/learning/mcanx.html
  - http://www.kidshealth.org/teen/school_jobs/school/test_anxiety.html

How do I improve my performance on Quizzes – and the final?

- **If you have to miss lecture – get notes from your friends**
- **Review lecture slides (take notes)**
- **Do the reading**
- **Attend discussion section**
- **For a study group**
- **Ask questions**
  - In class
  - Email
  - Office hours
- **What if I aced it? → WTG!**

What if I want a Re-grade?

Submit by next Wednesday

- Include the quiz with explanation of which question(s) should be re-graded and ***why***
- If it is a simple clerical error – just tell us which points were miscalculated

- **Entire quiz will likely be reassessed**
- **Keep in mind that graders are human too**
- **If you are unsure about something feel free to come to office hours**

Previous on INF 111…

- **More on Testing**
  - Static Analysis
  - Formal Verification

Today's Lecture

- **More on Testing**
  - Test Adequacy
    - Coverage Based Testing

Fundamental Testing Questions

- **Test Criteria: What should we test?**
- **Test Oracle: Is the test correct?**
- **Test Adequacy: How much is enough?**
- **Test Process: Is our testing effective?**

How to make the most of limited resources?

Test Adequacy

Tells you when to stop testing

- **Coverage-Based Testing**
  - Coverage metrics
    - when sufficient percentage of the program structure has been exercised
- **Fault-Based Testing**
  - Empirical assurance
    - when failures/test curve flatten out
  - Error seeding
    - percentage of seeded faults found is proportional to the percentage of real faults found
- **Error-Based Testing**
  - faults found in common are representative of total population of faults
  - Equivalence Partitioning

Coverage-Based Testing

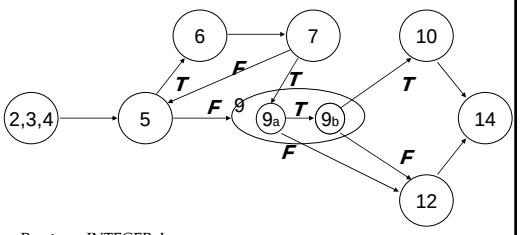
- **Flow Graphs**
  - Control Flow
    - Partial order of Statement Execution
  - Data Flow
    - Flow of values from Definition to Variables

Graph representation of control flow and data flow relationships

A Sample Program to Test

```
1   function P return INTEGER is
2   begin
3       X, Y: INTEGER;
4       READ(X); READ(Y);
5       while (X > 10) loop
6           X := X – 10;
7           exit when X = 10;
8       end loop;
9       if (Y < 20 and then X mod 2 = 0) then
10          Y := Y + 20;
11      else
12          Y := Y – 20;
13      end if;
14      return 2*X + Y;
15  end P;
```

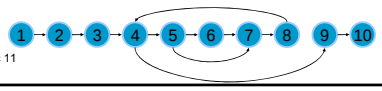

Prog P's Control Flow Graph (CFG)




```
1  function P return INTEGER is
2  begin
3      X, Y: INTEGER;
4      READ(X); READ(Y);
5      while (X > 10) loop
6          X := X – 10;
7          exit when X = 10;
8      end loop;
9      if (Y < 20 and then X mod 2 = 0) then
10         Y := Y + 20;
11     else
12         Y := Y – 20;
13     end if;
14     return 2*X + Y;
15 end P;
```

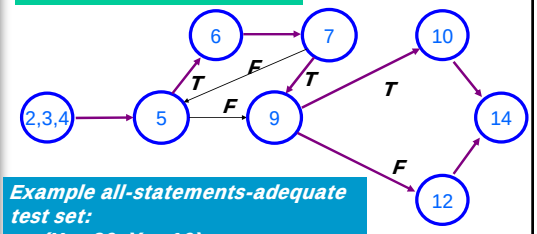

All-Statements Coverage

- **Select test cases such that every <u>node</u> in the graph is visited**
  - Also called node coverage
    - Guarantees that every statement in the source code is executed at least once
- **Selects minimal number of test cases**



All-Statements Coverage of P

At least 2 test cases needed

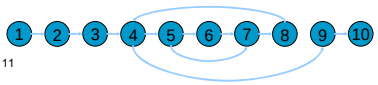


Example all-statements-adequate test set:
(X = 20, Y = 10)
<2,3,4,5,6,7,9,10,14>
(X = 20, Y = 30)
<2,3,4,5,6,7,9,12,14>

16

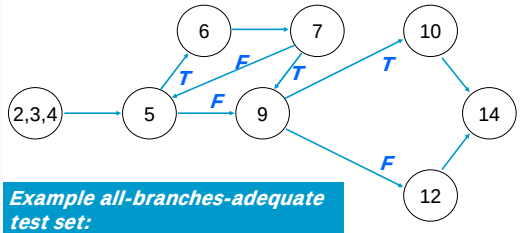
All-Branches Coverage

- **Select test cases such that every <u>branch</u> in the graph is visited**
  - Guarantees that every branch in the source code is executed at least once
- **More thorough than All-Statements coverage**
  - More likely to reveal logical errors



All-Branches Coverage of P

*At **least** 2 test cases needed*



Example all-branches-adequate test set:
(X = 20, Y = 10)
<2,3,4,5,6,7,9,10,14>
(X = 15, Y = 30)
<2,3,4,5,6,7,5,9,12,14>

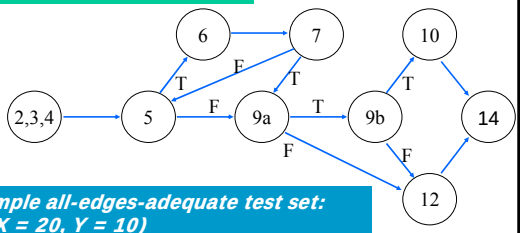
18

All-Edges Coverage

- **Select test cases such that every edge in the graph is visited**
  - Takes complex statements into consideration – treats them as separate nodes
- **More thorough than All-Branches coverage**
  - More likely to reveal logical errors

All-Edges Coverage of P

At least 3 test cases needed



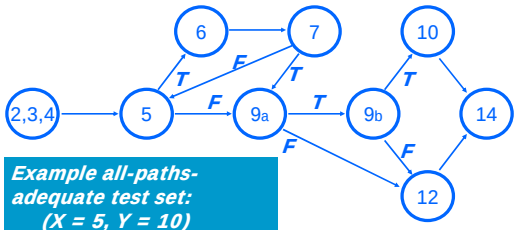
Example all-edges-adequate test set:
(X = 20, Y = 10)
<2,3,4,5,6,7,9a,9b,10,14>
(X = 5, Y = 30)
<2,3,4,5,9a,12,14>
(X = 21, Y = 10)
<2,3,4,5,6,7,5,6,7,5,9a,9b,12,14>

All-Paths Coverage

- **Path coverage**
  - Select test cases such that every path in the graph is visited
  - Loops are a problem
    - 0, 1, average, max iterations
- **Most thorough…**
  …but is it feasible?

All-Paths Coverage of P

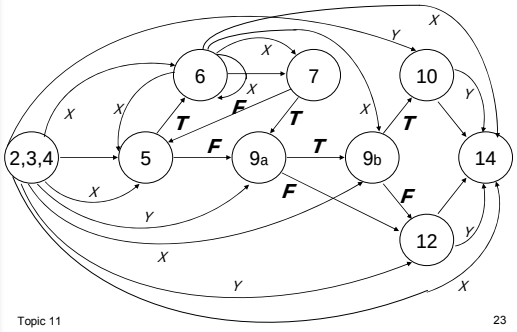
Infinitely many test cases needed



Example all-paths-adequate test set:
(X = 5, Y = 10)
(X = 15, Y = 10)
(X = 25, Y = 10)
(X = 35, Y = 10)
...

22

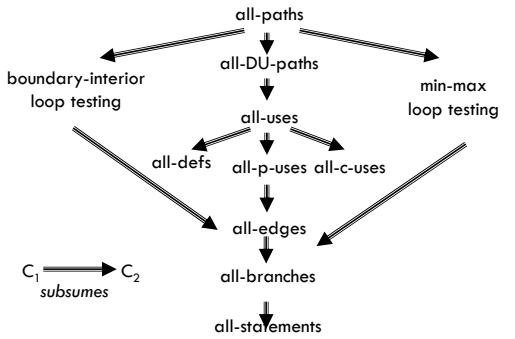
P's Control and Data Flow Graph



Subsumption of Criteria

- **C1 subsumes C2 if any C1-adequate test T is also C2-adequate**
  - But some T1 satisfying C1 may detect fewer faults than some T2 satisfying C2

Structural Subsumption Hierarchy



Topic 11

Subsumes => Stronger than