Lecture 19: Structural Testing

Kenneth M. Anderson
Software Methods and Tools
CSCI 3308 - Fall Semester, 2003
(Happy Halloween!)

Today's Lecture

- Discuss Structural Testing
  - Terminology
  - Techniques
  - Examples

Structural Testing

- Structural Testing supplies another criteria to answer the question:
  - "How many test cases are enough?"
- Recall that functional testing's criteria was "Test all functions"
- Structural Testing's criteria is "Test all code"
  - Structural Testing is also known as white box testing, because now we look at a program's source code to help create test cases

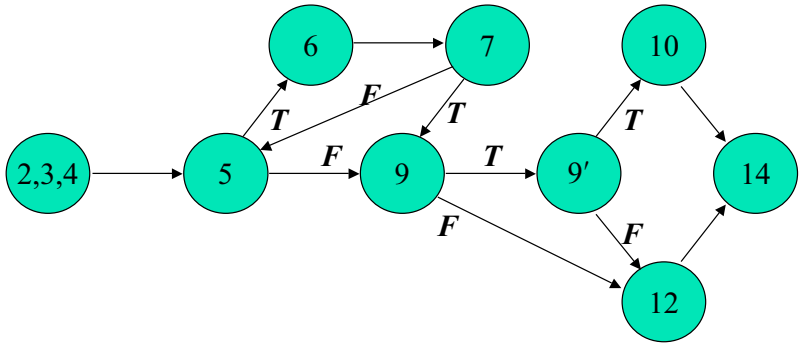
Control Flow Graphs (CFGs)

- Structural Testing is based on CFGs
- Control flow graphs capture the various ways in which a program can execute
  - A node in a CFG represents a program statement
  - An edge in the CFG represents the ability for a program to flow from its current statement to the statement at the other end of the edge
    - If an edge is associated with a conditional, label the edge with the conditional's value, either true or false

A Sample Ada Program

```
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;
```


P's Control Flow Graph (CFG)



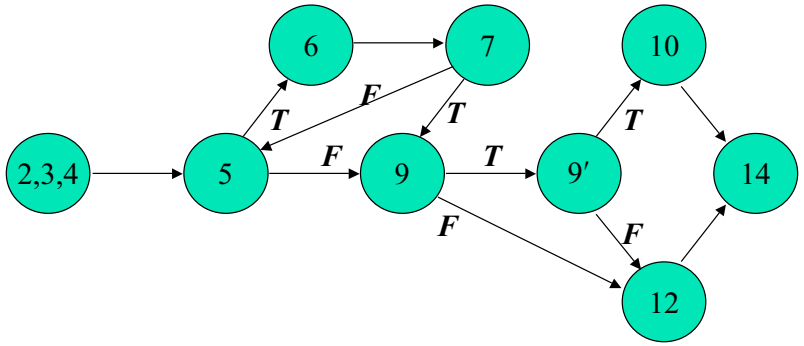
Types of Coverage

- Statement Coverage
  - Every statement is executed at least once
- Edge Coverage
  - Every edge is traversed at least once
- Condition Coverage
  - For binary logical operators (&&, ||), the individual components are evaluated in every possible combination of true and false
- Relational Coverage
  - For relational operators (<, >, <=, >=) the equal condition is treated as a separate branch
- Path Coverage
  - Every possible path is executed at least once

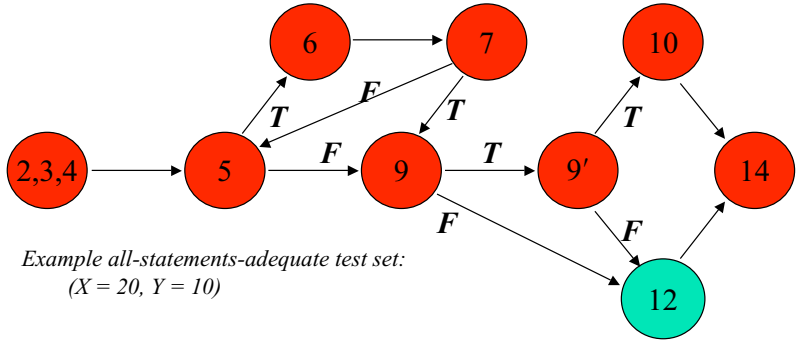
White-box Testing Criteria

- Statement Coverage
  - Execute each statement at least once
  - Pick test case and plot its path through the CFG
  - Keep picking test cases until all statements are covered

All-Statements Coverage of P

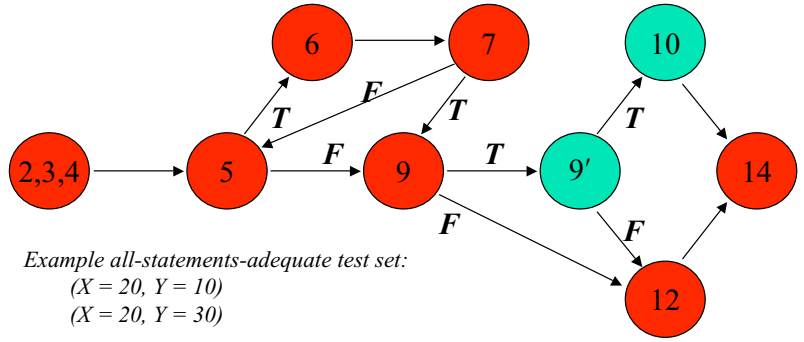


Test Case 1: X=20, Y = 10



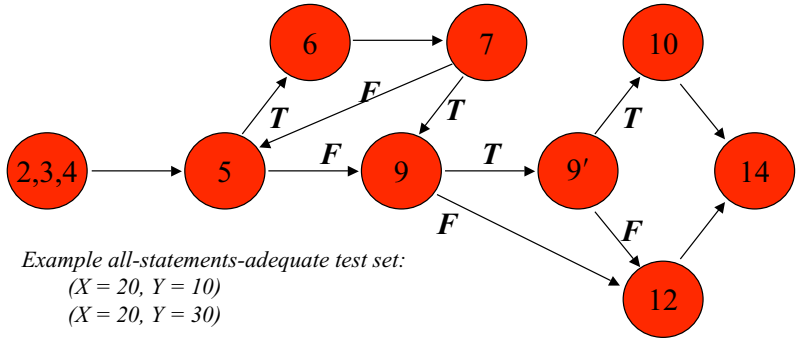
Example all-statements-adequate test set:
(X = 20, Y = 10)

Test Case 2: X = 20, Y = 30



Example all-statements-adequate test set:
(X = 20, Y = 10)
(X = 20, Y = 30)

Combined: Complete Coverage

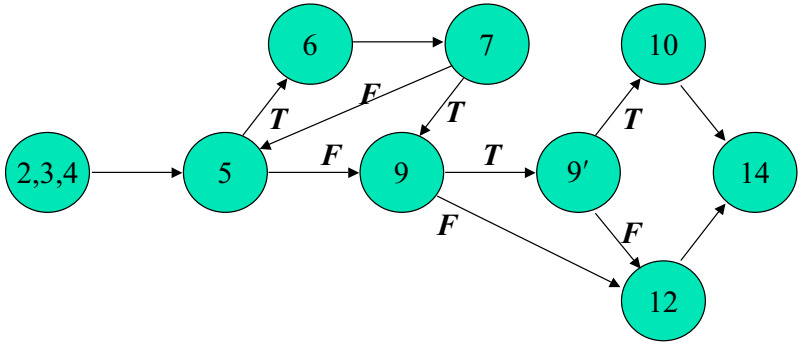


Example all-statements-adequate test set:
(X = 20, Y = 10)
(X = 20, Y = 30)

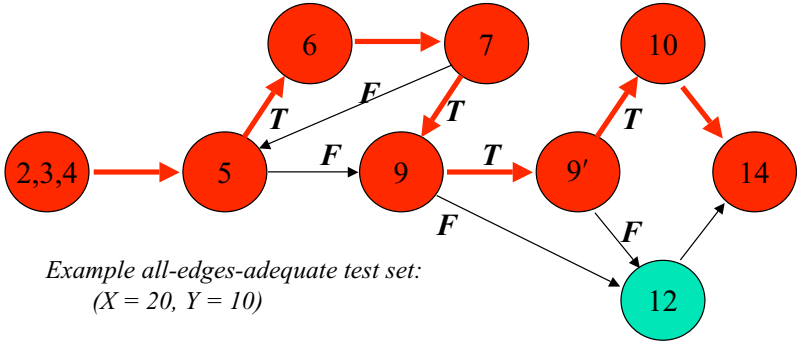
White-box Testing Criteria

- Edge Coverage
  - Traverse each edge at least once
  - Pick test case and plot its path through CFG
  - Keep picking test cases until all edges are covered
- Also known as Branch Coverage
  - We must traverse each conditional (such as an if statement) along its true and false edge

All-Edges Coverage of P

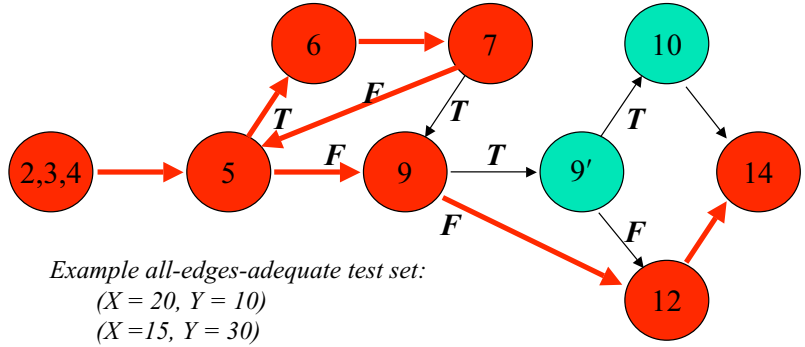


Test Case 1: X = 20, Y = 10



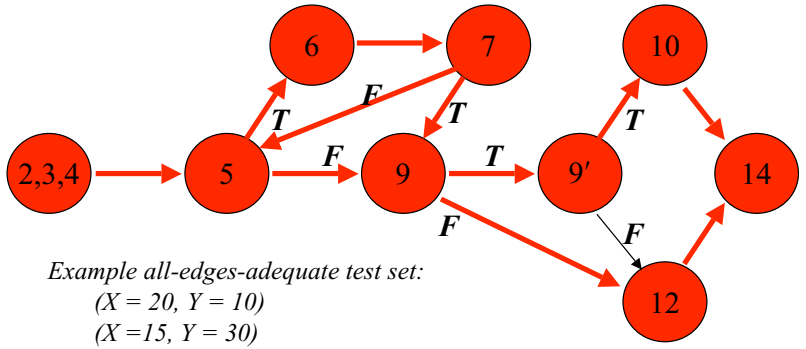
Example all-edges-adequate test set:
(X = 20, Y = 10)

Test Case 2: X = 15, Y = 30



Example all-edges-adequate test set:
(X = 20, Y = 10)
(X =15, Y = 30)

Combined: Complete Coverage

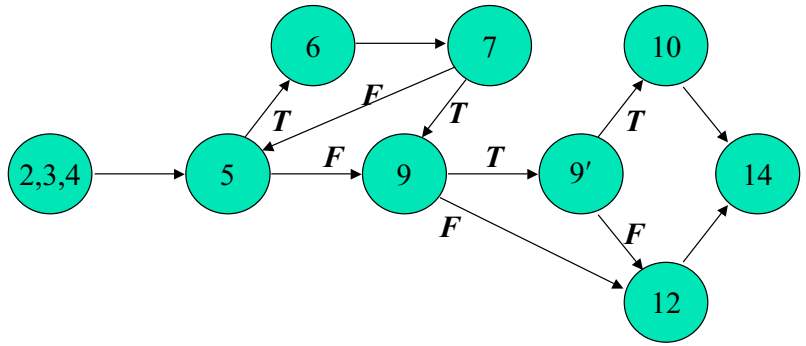


Example all-edges-adequate test set:
(X = 20, Y = 10)
(X =15, Y = 30)

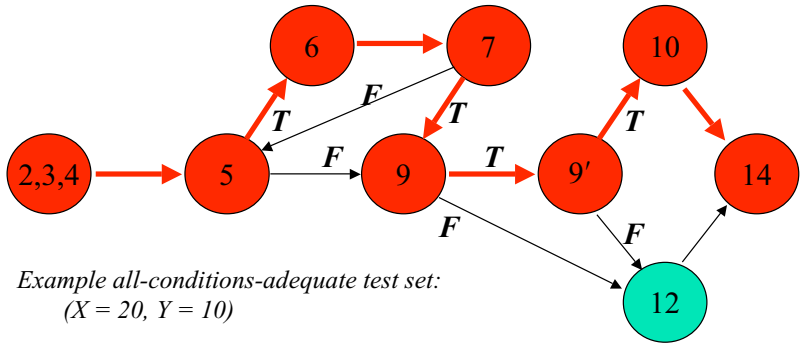
White-box Testing Criteria

- Condition Coverage
  - Traverse all edges at least once but
    - in binary logical operators (also known as short circuit operators), all possible combinations of true and false must be tested
  - Pick test case and plot its path through CFG; keep creating test cases until all conditions are covered

All-Conditions Coverage of P

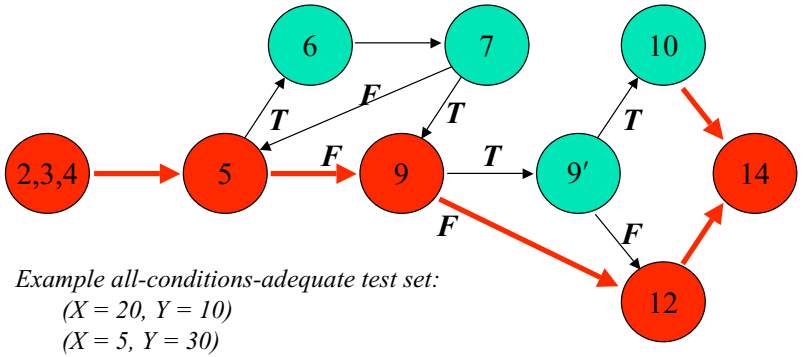


Test Case 1: X = 20, Y = 10



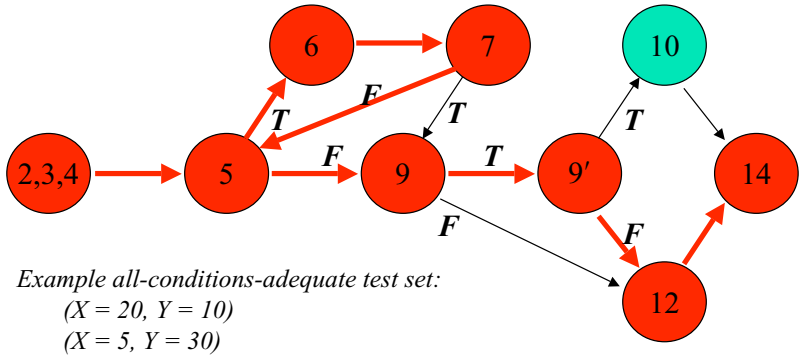
Example all-conditions-adequate test set:
(X = 20, Y = 10)

Test Case 2: X = 5, Y = 30



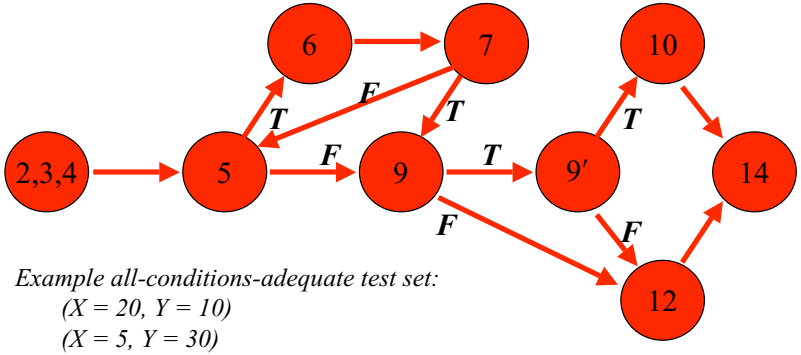
Example all-conditions-adequate test set:
(X = 20, Y = 10)
(X = 5, Y = 30)

Test Case 3: X=21, Y = 10



Example all-conditions-adequate test set:
(X = 20, Y = 10)
(X = 5, Y = 30)
(X = 21, Y = 10)

Combined: Complete Coverage



Example all-conditions-adequate test set:
(X = 20, Y = 10)
(X = 5, Y = 30)
(X = 21, Y = 10)

Relational Coverage

- Relational Coverage
  - This is a form of edge coverage in which any relational operator (<, >, <=, and >=) has its equal condition treated as a separate branch
- So, “if (x < y)” should be treated as
  - if (x < y)
    - …
  - else if (x == y)
    - …
  - else if (x > y)
    - ...

Relational Coverage, continued

- Relational coverage is thus a stronger form of edge coverage
- It is saying that for each conditional you should have at least three test cases
  - x < y, x > y, x == y
- Combine this approach with conditional coverage and you have the strongest form of edge coverage possible

Path Coverage

- Path Coverage
  - Traverse each path at least once
- Problem
  - Way too many paths, even in simple programs
- Approach
  - Use heuristics
    - e.g. for each loop take loop zero, one, and multiple times

Example

- How many paths does the following program fragment have?

```
a << cin; b = 0;
while (a > 0) {
      a--; b++;
}
if (b > 5) {
      printf("b > 5");
} else {
      printf("b <= 5");
}
```

- For any particular value of a, there is only one path possible
- but since a is entered by user, there are an infinite number of possible paths!

Path Coverage, continued

- In general
  - for loops
    - traversing a loop zero, one, two, … times is each a different path, so a loop has a potentially infinite number of paths
  - for conditionals
    - traverse true and false branches
    - for a program consisting of only if statements
      - if x is the number of if statements, there are a total of 2^x paths!
- As such, path coverage is an infeasible testing criteria in the general case; so use heuristics to approximate it, as discussed previously