Lesson 3: White-box Testing

- Statement Coverage
- Branch Coverage
- Path Coverage

White-box Testing

White-box testing is a technique that involves testing the internal structure, logic, and code of the software. Testers design cases to validate functions, paths, and branches, aiming for high code coverage.

Techniques include:

1. Statement Coverage: Test each line of code.
2. Branch Coverage: Test all decision outcomes.
3. Path Coverage: Verify all possible code paths.

Whitebox Testing Example

#1. Statement coverage: ensuring every line of code is executed at least once.

Example 1: Consider the code below:

```
INPUT A & B
C = A + B
IF C>100
    PRINT "ITS DONE"
```

How many test cases to use for statement coverage?

Whitebox Testing Example

#1. Statement coverage: ensuring every line of code is executed at least once.

Example 1: Consider the code below:

```
INPUT A & B
C = A + B
IF C>100
    PRINT "ITS DONE"
```

How many test cases to use for statement coverage?

TestCase_01: A=40 and B=70

We would only need one test case to check all the lines of the code.

Whitebox Testing Example

#2. Branch coverage:

Branch in a programming language is like the "IF statement". An IF statement has two branches: True and False. So in Branch coverage (also called Decision coverage), we validate whether each branch is executed at least once.

In case of an "IF statement", there will be two test conditions:

- One to validate the true branch
- Other to validate the false branch.

Whitebox Testing Example

Example 2: Consider the code below:

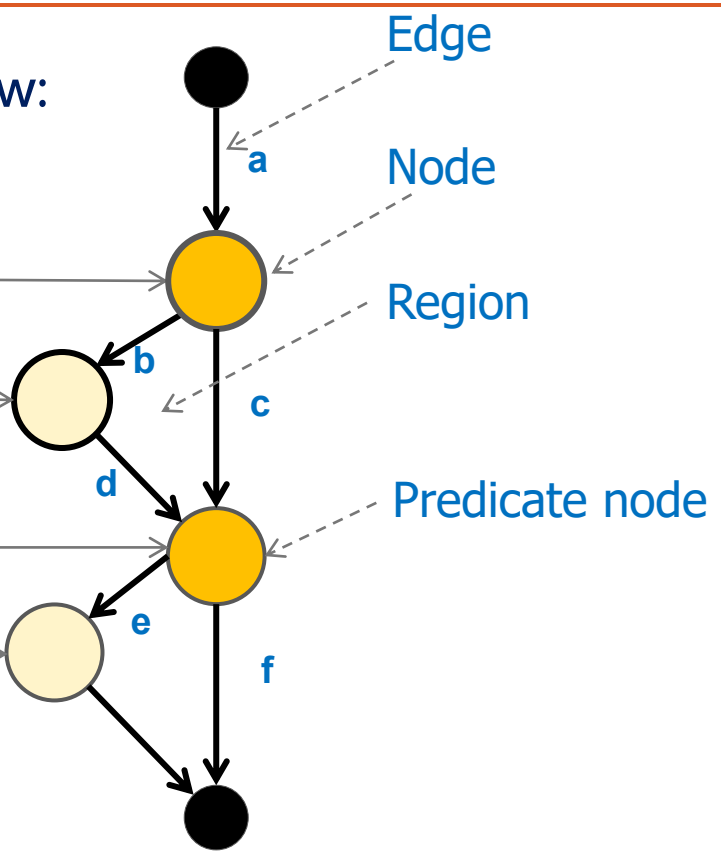
```
INPUT A & B
C = A + B
IF C>100
    PRINT “ITS DONE”
END IF
IF A>50
    PRINT “ITS PENDING”
END IF
```

How many test cases to use for branch coverage?

White Box Testing Example

Example 2: Consider the code below:

```
INPUT A & B
C = A + B
IF C>100
    PRINT "ITS DONE"
END IF
IF A>50
    PRINT "ITS PENDING"
END IF
```



- TestCase_01: A=60 and B=70 (abde)
- TestCase_02: A=30 and B=30 (acf)

Whitebox Testing Example

#3. Path coverage: tests all the paths of the program.

Example 3: Consider the code below:

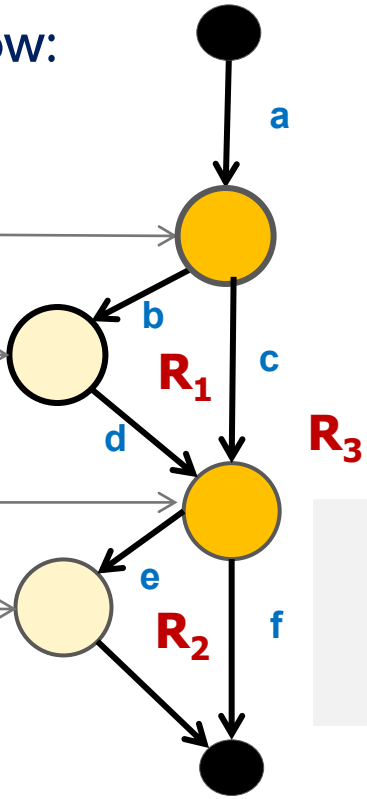
```
INPUT A & B
C = A + B
IF C>100
    PRINT "ITS DONE"
END IF
IF A>50
    PRINT "ITS PENDING"
END IF
```

How many test cases to use for path coverage?

White Box Testing Example

Example 3: Consider the code below:

```
INPUT A & B
C = A + B
IF C>100
    PRINT "ITS DONE"
END IF
IF A>50
    PRINT "ITS PENDING"
END IF
```



Path: 3

Number of region: 3

V (G) = P + 1

- **Test Case 1 (abde):** a = 60 and b = 70 -> ITS DONE ITS PENDING
- **Test Case 2 (abdf):** a = 20 and b = 90 -> ITS DONE
- **Test Case 3 (acf):** a = 40 and b = 10 -> Null

Whitebox Testing Example

Example 4: Write test cases to solve an equation ax + b = 0.

Consider this below code:

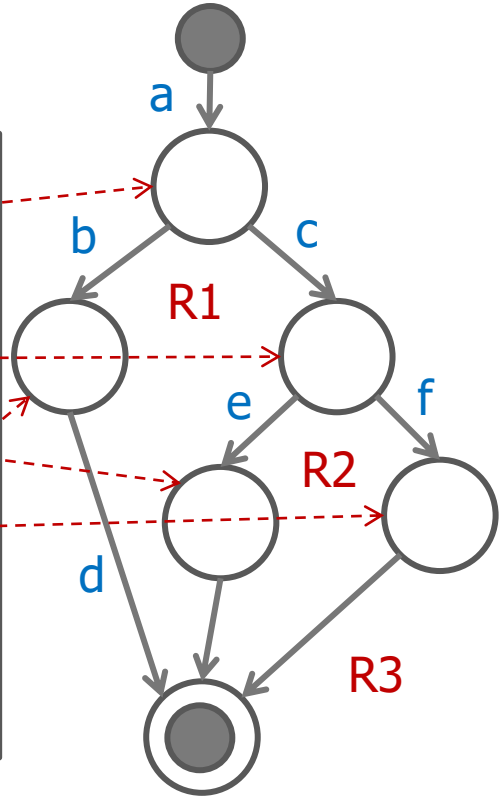
```
void solve_equation(double a, double b)
{
   if (a == 0)
      if (b == 0)
        cout << “Identity equation\n";
      else
        cout << “Contrary equation\n";
   else
      cout << “x = " << -b / a << endl;
}
```

Whitebox Testing Example

Example 4: Write test cases to solve an equation ax + b = 0.

Consider this below code:

```
void solve_equation(double a, double b)
{
   if (a == 0)
      if (b == 0)
        cout << “Identity equation\n";
      else
        cout << “Contrary equation\n";
   else
      cout << “x = " << -b / a << endl;
}
```

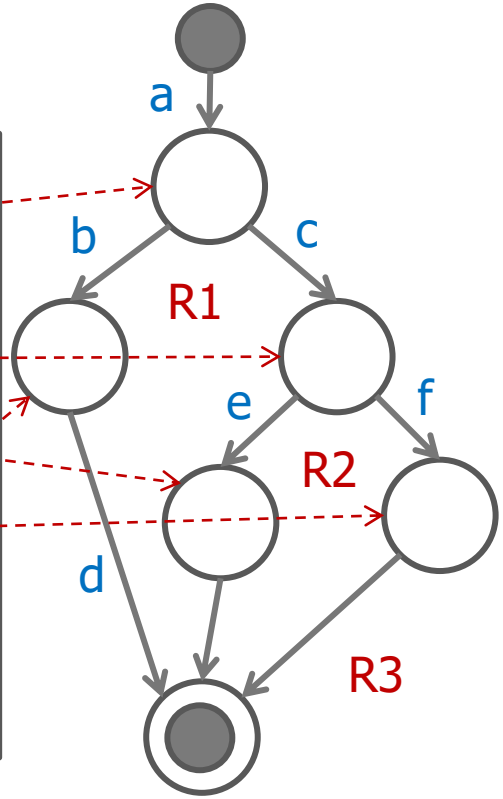


Whitebox Testing Example

Example 4: Write test cases to solve an equation ax + b = 0.

Consider this below code:

```
void solve_equation(double a, double b)
{
   if (a == 0)
      if (b == 0)
        cout << "Identity equation\n";
      else
        cout << "Contrary equation\n";
   else
      cout << "x = " << -b / a << endl;
}
```



- **Test Case 1 (abd):** a = 5 and b = -10 -> x=10/5
- **Test Case 2 (ace):** a = 0 and b = 0 -> Identity equation
- **Test Case 3 (acf):** a = 0 and b = 5 -> Contrary equation

Whitebox Testing Example

Example 5: Write test cases to test the program that sum the digits of the positive integer n until the sum of all digits is less than 10.

For example:

- 123456 → 1 + 2 + 3 + 4 + 5 + 6 = 21 → 2 + 1 → 3 < 10
- 857 → 8 + 5 + 7 = 20 → 2 + 0 = 2 < 10

Can you code a function to solve the above requirements?

Whitebox Testing Example

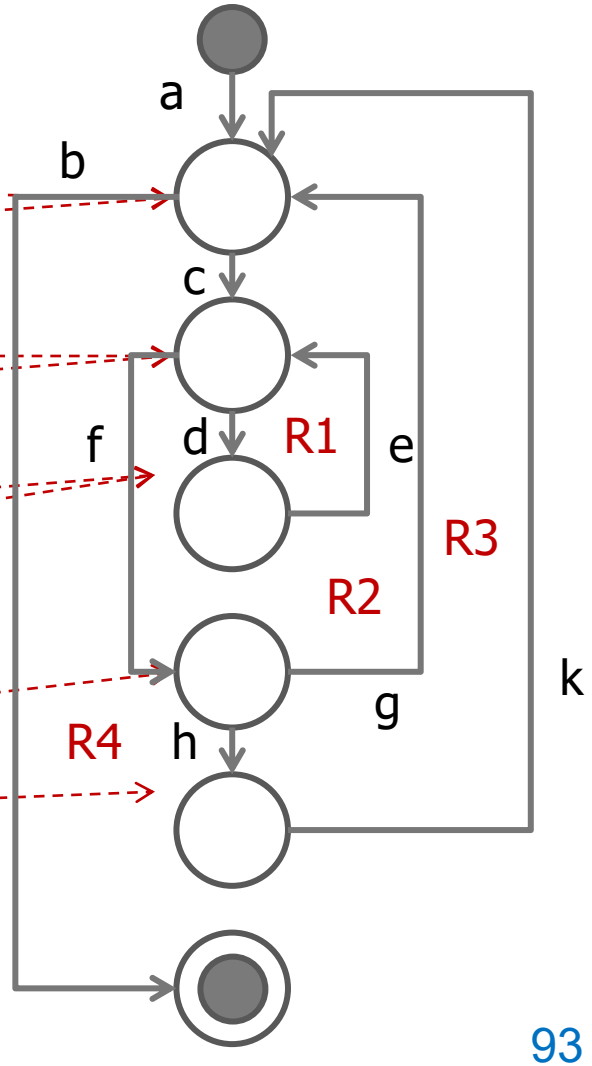
Example 5:

```
int sum(int n)
{
   int total= n;
   while (n>=10)
   {
      total = 0;
      while (n!=0)
      {
         total +=n % 10;
         n = n / 10;
      }

      if (total >= 10)
         n = total;
   }
   return total;
}
```

Whitebox Testing Example

Example 5:

```
int sum(int n)
{
    int total= n;
    while (n>=10)
    {
        total = 0;
        while (n!=0)
        {
            total +=n % 10;
            n = n / 10;
        }

        if (total >= 10)
            n = total;
    }
    return total;
}
```



Whitebox Testing Example

Example 5:

#1) Paths: 4

#2) Test cases:

1. a,b (n=5)
2. a,c,d,e,f,g,b (n-12)
3. a,c,d,e,f,h,k,g,b (n=987)
4. a,c,d,e,f,-,g,b (n=55955)