Statement coverage technique:

Branch Coverage Technique:

1.It tests both true & False conditions.

2.It will check for decision coverage like If or Loop statements.

3.It measures whether each possible branch (true or false) in control structures has been executed.

4.100% branch coverage guarantees 100% statement coverage but not vice versa.Flow

Branch Coverage (% ) = Total no. of branches/Number of branches teste\*100.

Eg:

Read A,B

C = A-B

IF (C<0) THEN

PRINT “CORRECT VALUE”,

ELSE IF(C>0)

PRINT “WRONG VALUE”.

ELSE

PRINT “DO NOTHING”.

Test case1: A = 50, B=40;Else if ( true)

Testcase2: A=40; B =50 , if ( true)

Ex2:

1.def categorize\_age(age):

2. if (age < 0):

3. category = "Invalid age"

4. elif age <= 12:

5. category = "Child"

6. elif age <= 19:

7. category = "Teenager"

8.elif age <= 64:

9. category = "Adult"

10. else:

11. category = "Senior"

12. return category

1.If(age<0)( true and false)

2.(age<=12)

3.age<=19

4.age<=64

5.else

Testcase1: categorize\_age(0):

Executed lines:1,2,3,12

Result = Invalid age

Testcase2: categorize\_age(10):

Executed lines:1,4,5, 12

Result = Child

Testcase3: categorize\_age(16):

Executed lines:1,6,7, 12

Result = Teenager

Testcase4: categorize\_age(50):

Executed lines:1,8,9, 12

Result = Adult

Testcase5: categorize\_age(65):

Executed lines:1,10,11, 12

Result = Senior

Advantages of Branch Coverage:

1.It gives maximum coverage compared to Statement coverage

2.It assures each branch from decision point is executed.

Disadvantages of Branch Coverage:

1.It will not check for boolean conditions.

2.This metric ignores the branches within Boolean expression, which occur due to Short-circuit operators.

Decision Coverage: ( Contional coverage)

Which reports the true or false outcomes of each boolean expression of the source code

The goal of this techniques is to cover and validate all the accessible source code by chcking and ensuring that each branch of every possible decion points is executed atleat once.

Decision coverage = no. of decicion outcomes tested / total no. of decision outcomes

Eg:

def example\_function(x, y):

if (x > 0 and y > 0):

return "Both positive"

else:

return "Not both positive"

Decision point : (x > 0 and y > 0):

Possible outcomes:

(x > 0 and y > 0): example\_function(1,1) == true

(x > 0 and y < 0): example\_function(1,1)== false

(x < 0 and y > 0): example\_function(1,1)== false

(x < 0 and y < 0): example\_function(1,1)== false

Combition:

(x > 0 and y > 0): True

(x > 0 and y > 0): false

Decicion coverage (% ) = 2 total outcomes/2outcomes tested\*100 = 100%

Advantages of decision coverage :

1. Throuogh testing of code path
2. Better condition testing
3. Higher defect dection.

Diadvantages of decision coverage:

1. Increased Complexity and efforts
2. Time consuming

Path Coverage Technique:

Path testing used to verify the flow of execution of programs in such a way that every path must be executed atleat once.

Here we write floe graphs and test all the independent paths by using Cyclomatica complexity metrics.

A computer keyboard with a white arrow pointing to a black button

Description automatically generated

Flow graph notations.

Arrow indicated edges.(E)

Circles indicated the nodes to preform actions ( expressions and statements) (N)

Cyclomatic complexity:

It is used to measure independent path with minimum number of possibilities.

Higher the cyclomatic complexity, high risk and difficult to understand.

Cyclomatic complexity = E-N+2

Ex:

1.def categorize\_age(age):

2. if (age < 0):

3. category = "Invalid age"

4. elif age <= 12:

5. category = "Child"

6. elif age <= 19:

7. category = "Teenager"

8.elif age <= 64:

9. category = "Adult"

10. else:

11. category = "Senior"

12. return category

Flow chart Method:

CC = (E-N)+2

CE=(14 -11)+2 =5

Minimum no. of testcases required = 5

Testcase1: (age < 0):

Testcase2 : elif age <= 12

Testcase3: elif age <= 19:

Testcase4: elif age <= 64:

Testcase5: . else:

Ex2:

def example\_function(x, y):

if (x > 0 and y > 0):

return "Both positive"

else:

return "Not both positive"

cc = (3-4)+2 = 1

A diagram of a diagram

Description automatically generated

Decision point method:

For a single function with a 1 decision point, then we can directly calculate cyclomatic complexity by adding 1 to decision point.

CC= number of decision points+1

Cc=1+1 =2

Loop coverage testing:

Loop testing evaluates the loops(for, while, do-while), ensures loops repeat for all the defined no. of cycles, and ensures that the terminating condition is working properly or not.

For example , if the developer know the loop is repeating only 5 times, and the 6th time if that loop is repeating, then it is error.

We will use automation testing for loop coverage testing .

Objectives of Loop Coverage:

1.To fix the infinite loop repetition problem.

2.To know the performance.

3.to identify the loop initialization problem.

4.to determine the initialized variables.

Ex:

Condition =true;

While(loop<=10){

-----------------------

If (loop==5&&condition){

------------------  
break;

}

Else{

--------------------

}

}

Ex:

For(i=5;i>=5;i++)

{

Print(i)

}

Advantages of Loop testing:

----------------------------------

1.Loop testing limits the no. of iteartions of loop

2.Loop testing ensures that the program does not go into infinite loop process.

3.Loop testing ensures that initilization of every used variable inside the loop.

4. Loop testing helps to identify the different problems inside the loop.

5.Loop testing helpd in determination of capacity.

Disadavantages of Loop Testing:

1.loop testing is not useful to detect the bugs..

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_---------------------------------------------

Control flow Technique:

It a software testing technique that helps to ensure that the different paths of the program can be tested.

Can include different levels of testing such as statement coverage, branch coverage, condition coverage, path coverage.

Control flow testing is usually performed using a control flow graphs,

Notation:

Nodes:

Edges:

Decision points: ( condition nodes,represents the point in the program where a decision is made based on a condition( if-else)

junction nodes: