

Software Verification

Exercise Solution: Testing Criteria

Algorithm 1

```
Algorithm 1:  
String function(int x, int y)  
{  
    boolean z;  
  
    if (x < y)  
        z := true  
    else  
        z := false  
  
    if (z && x+y == 10)  
        result := "a"  
    else  
        result := "b"  
}
```

Branch and path coverage

(a) How many branches are present?

4

(b) Is it possible to test every branch?

Yes.

Provide a set of tests to exercise as many branches as possible.

function(4,6)

function(6,4)

(c) How many paths are present?

3

(d) Which paths remain untested by your tests of (b)?

z := true -> result := "b"

(e) Is it possible to test all the paths? Yes, since a path is a feasible execution path, it is

always possible to test all paths.

Add tests, if required, to do so.

function(1,2)

Predicate coverage

(a) Write down the predicates that occur in the code.

$x < y$

$z \ \&\& \ x+y == 10$

(b) Is it possible to obtain full predicate coverage?

Yes.

Provide a set of tests which will obtain the highest predicate coverage.

function(4,6)

function(6,4)

Note that this definition of predicate coverage is equivalent to branch coverage.

Clause coverage

(a) Write down the clauses appearing in the function.

$x < y$

z

$x+y == 10$

(b) Can we exercise full clause coverage?

Yes.

Write tests for maximal clause coverage.

function(4,6)

function(2,1)

Algorithm 2

Algorithm 2:

```
if x > 0 then
  y := x + x
  while y < 15 do
    y := y + 2
  end
else
  if x = 0 then
    y := 1
  else
    y := x * x
  end
end
```

Branch and path coverage

(a) How many branches are present?

6

(b) Is it possible to test every branch?

Yes.

Provide a set of tests to exercise as many branches as possible.

x := 1,
x := 0,
x := -1.

(c) How many paths are present?

10

(d) Which paths remain untested by your tests of (b)?

$y := x + x \rightarrow (y := y + 2)^n$, where $0 \leq n \leq 6$.

(e) Is it possible to test all the paths? Yes, since a path is a feasible execution path, it is always possible to test all paths.

Add tests, if required, to do so.

x := 2,
x := 3,

x := 4,
x := 5,
x := 6,
x := 7,
x := 8.

Predicate coverage

(a) Write down the predicates that occur in the code.

x > 0
y < 15
x = 0

(b) Is it possible to obtain full predicate coverage?

Yes.

Provide a set of tests which will obtain the highest predicate coverage.

x := 1,
x := 0,
x := -1.

Note that this definition of predicate coverage is equivalent to branch coverage.

Clause coverage

Note that the clauses are the same as the predicates for this example.

(a) Write down the clauses appearing in the function.

x > 0
y < 15
x = 0

(b) Can we exercise full clause coverage?

Yes.

Write tests for maximal clause coverage.

x := 1,
x := 0,
x := -1.