Analyse data flow testing using any coverage – statement, branch, decision etc.

Following example uses statement coverage:

Associations:

**Code (def, uses, variable)**

1. read x (1, (2, t), x)
2. if (x > 0) (1, (2, f), x)
3. a = x + 1 (1, 3, x)
4. if (x <= 0) { (1, (4, t), x)
5. if (x < 1) (1, (4, f), x)
6. x = x + 1; goto (5) (1, (5, t), x)

else (1, (5, f), x)

1. a = x + 1 (1, 6, x)
2. print a; (1, 7, x)

(3, 7, a)

(3, 8, a)

(6, 6, x)

(6, 7, x)

(6, (5, t), x)

(6, (5, f), x)

(7, 8, a)

Eg:

Set x = -1

Path = 1, 2, 4, 5, 6, 5, 6, 5, 7, 8

Output = 2

Set x = 1

Path = 1, 2, 3, 8

Output = 2

Def-clear path:

A definition-clear path (def-clear) p with respect to x is a sub-path where x is not defined at any of the nodes in p

A du-path is a simple path where the initial node of the path is the only defining node of x in the path.

Associations:

(1, (2, t), x), (1, (2, f), x), (1, 3, x), (1, (4, t), x), (1, (4, f), x), (1, (5, t), x), (1, (5, f), x), (1, 6, x), (1, 7, x), (3, 7, a), (3, 8, a), (6, 6, x), (6, 7, x), (6, (5, t), x), (6, (5, f), x), (7, 8, a)

All defs coverage (ADC):

(1, (2, t), x), (6, 6, x), (3, 8, a), (7, 8, a)

All uses coverage (AUC):

(1, (2, t), x), (1, (2, f), x), (1, 3, x), (1, (4, t), x), (1, (4, f), x), (1, (5, t), x), (1, (5, f), x), (1, 6, x), (1, 7, x), (3, 7 a), (3, 8, a), (6, 6, x), (6, 7, x), (6, (5, t), x), (6, (5, f), x), (7, 8, a)

All c-use coverage:

(1, 3, x), (1, 6, x), (1, 7, x), (3, 7 a), (3, 8, a), (6, 6, x), (6, 7, x), (7, 8, a)

All c-use some p-use coverage:

(1, 3, x), (1, 6, x), (1, 7, x), (3, 8, a), (6, 6, x), (6, 7, x), (7, 8, a)

All p-use coverage:

(1, (2, t), x), (1, (2, f), x), (1, (4, t), x), (1, (4, f), x), (1, (5, t), x), (1, (5, f), x), (6, (5, t), x), (6, (5, f), x)

All p-use some c-use coverage:

(1, (2, t), x), (1, (2, f), x), (1, (4, t), x), (1, (4, f), x), (1, (5, t), x), (1, (5, f), x), (3, 8, a), (6, (5, t), x), (6, (5, f), x), (7, 8, a)

All du path coverage (ADUPC):

All paths from one definition to its uses. Similar to all use coverage – but all uses can give only one predicate, but all du path considers all paths