

Structural Testing (2)

Linear Code Sequence and Jump (LCSAJ) Coverage

Un LCSAJ este o cale (execuție) a unui program formată dintr-o secvență de cod (Linear Code Sequence) urmată de un salt (Jump) al controlului programului.

Un LCSAJ este definit ca un triplet (X, Y, Z), unde

- X este startul secvenței liniare
- Y este sfarsitul secvenței liniare
- Z este linia de cod unde este transferat controlul după sfârșitul secvenței liniare

Exemplu:

(calculul x^y , $x > 0$, $y \geq 0$)

```
1 begin
2   int x, y, z;
3   read(x, y);
4   z = 1;
5   while (y > 0) {
6       z = z*x;
7       y = y - 1;
8   }
9   write(z);
10 end
```

| LCSAJ | Start | End | Jump to |
|-------|-------|-----|---------|
| 1 | 1 | 8 | 5 |
| 2 | 5 | 8 | 5 |
| 3 | 5 | 5 | 9 |
| 4 | 1 | 5 | 9 |
| 5 | 9 | 9 | Exit |

Considerăm $T = \{t1, t2\}$, unde $t1 = (x = 3, y = 0)$, $t2 = (x = 3, y = 2)$

$t1: (1, 5, 9) \rightarrow (9, 9, \text{exit})$

$t2: (1, 8, 5) \rightarrow (5, 8, 5) \rightarrow (5, 5, 9) \rightarrow (9, 9, \text{exit})$

T acoperă toate cele 5 LCSAJ

Un set de teste care realizează o acoperire la nivel de decizie nu realizează în mod necesar o acoperire la nivel de LCSAJ

Exemplu:

```
1 begin
2   int x, y, z;
3   read(x, y);
4   p = 0;
5   if (x < 0)
6     p = x;
7   if (y < 0)
8     p = p + 1;
9   else
10    p = p + 2;
11 end
```

| LCSAJ | Start | End | Jump to |
|-------|-------|-----|---------|
| 1 | 1 | 9 | exit |
| 2 | 1 | 5 | 7 |
| 3 | 7 | 9 | exit |
| 4 | 7 | 7 | 10 |
| 5 | 1 | 7 | 10 |
| 6 | 10 | 10 | Exit |

Considerăm $T = \{t1, t2\}$, unde $t1 = (x = -1, y = -1)$, $t2 = (x = 0, y = 0)$

Pentru $t1$ ambele condiții sunt satisfăcute, $t1: (1, 9, \text{exit})$

Pentru $t2$ ambele condiții sunt false, $t2: (1, 5, 7) \rightarrow (7, 7, 10) \rightarrow (10, 10, \text{exit})$

Cele două LCSAJ rămase pot fi parcurse de $t3 = (x = -1, y = 0)$ și $t4 = (x = 0, y = -1)$,

$t3: (1, 7, 10) \rightarrow (10, 10, \text{exit})$

$t4: (1, 5, 7) \rightarrow (7, 9, \text{exit})$