Software Development II
Unit 5: Whitebox Testing
Test Coverage

Markus Roggenbach

March 2020

You will learn

- There there is a ‘plethora’ of coverage criteria.
- Three coverage criteria and what the rationale is behind them.

Coverage

Criteria

Going back to early work by E.F. Miller in 1977, there have been established many coverage criteria for test suites:

$$C_0,\ C_1,\ C_{1P},\ C_2,\ C_d,\ C_{MCC},\ C_ik,\ C_{stat},\ C_\infty.$$

Here, only a small selection.

Computational Problem – ClipLower

ClipLower:

Input: integers $lower$, x

Output: x, if $x \geq lower$

$lower$, otherwise

Program: clip

```
public static int clip (int lower, int x) {
        if (x < lower) {
                x = lower;
        }
        return x;
}
```

C_0

“every node”

Find a set of test cases such that, when executed, every node of the program graph is traversed at least once.

A minimal Test Suite w.r.t. C_0 for `clip` and ClipLower

Test Name	lower	x	expected output
Test 1	1	0	1

Note:

- Selection of “lower” and “x”: based on `clip`.
- Determination of “expected output”: based on ClipLower.

Definition

A test suite is called minimal for a test approach if there is no other test suite for this test approach that includes fewer test cases.

C_p

"every branch"

Find a set of test cases such that, when executed, it covers all branches of the program.

A minimal Test Suite w.r.t. C_p for `clip` and ClipLower

Test Name	lower	x	expected output
Test 1	1	0	1
Test 2	2	4	4

Program: `clipModified`

```
public static int clipModified (int lower, int x) {
        x = lower - 1;
        if (x < lower) {
                x = lower;
        }
        return x;
}
```

A minimal C_p test suite will detect the error.

A minimal C_0 test suite can't detect the error.

C_ik

“every loop up to k times”

Find a set of test cases such that, when executed, every cylcle is followed 0 times, 1 times, ..., and k times.

Computational Problem – Multiply

Multiply:

Input: integers a, b

Output: $a * b$

Program: Erna

```
public static int Erna (int a, int b) {
    int d = 1;
    int i = 0;
    while (i <= a) {
        d = d + b;
        i++;
    }
    return d;
}
```

Remark:

Erna fails to be a correct implementation of **Multiply**.

A minimal Test Suite w.r.t. C_i0 for `Erna` and Multiply

Test Name	a	b	expected output
Test 0	-1	3	-3

A minimal Test Suite w.r.t. C_i1 for `Erna` and Multiply

Test Name	a	b	expected output
Test 0	-1	3	-3
Test 1	0	3	0

A minimal Test Suite w.r.t. C_i2 for `Erna` and Multiply

Test Name	a	b	expected output
Test 0	-1	3	-3
Test 1	0	3	0
Test 2	1	4	4

What you have learned in this unit

Definitions

- Coverage criteria C_0, C_p and $C_i(k)$.
- Definition of minimal test suite for a coverage criterion.

You should be able to construct

- (Minimal) test suits for C_0, C_p and $C_i(k)$.
- UML activity diagram for whitebox testing