Piacente Cristian 866020

Homework H11 – Mutation Analysis

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Consider the following program and its mutated version in which we replace the multiplication operator (*) with the sum operator (+) at line 9:

```
1   public static double[] multiply(int[][] matrix, int[] vector) {
2       int rows = matrix.length;
3       int columns = matrix[0].length;

4       double[] result = new double[rows];

5       for (int row = 0; row < rows; row++) {
6           double sum = 0;
7           for (int column = 0; column < columns; column++) {
8               sum += matrix[row][column]
9                       * vector[column];
            }
10          result[row] = sum;
        }
11      return result;
    }
```

```
1   public static double[] mutatedMultiply(int[][] matrix, int[] vector) {
2       int rows = matrix.length;
3       int columns = matrix[0].length;

4       double[] result = new double[rows];

5       for (int row = 0; row < rows; row++) {
6           double sum = 0;
7           for (int column = 0; column < columns; column++) {
8               sum += matrix[row][column]
9                       + vector[column];
            }
10          result[row] = sum;
        }
11      return result;
    }
```

Consider the following test case:

```
int [][] matrix = new int[3][3];
matrix[0][0] = 2;
matrix[0][1] = 0;
matrix[0][2] = 2;
matrix[1][0] = 2;
matrix[1][1] = 1;
matrix[1][2] = 3;
matrix[2][0] = 0;
matrix[2][1] = 2;
matrix[2][2] = 4;

int[] vector = new int[3];
vector[0] = 2;
vector[1] = 0;
vector[2] = 2;
```

Perform weak mutation analysis and specify at which point of the execution the mutant is marked as dead and why (consider single statements as segments).

The answer should

Piacente Cristian 866020

Homework H11 – Mutation Analysis

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

- indicate the sequence of statements up to the statement at which the execution kills the mutant. The sequence of statements should be a sequence of statement-numbers that refer to the code, for example, write 1 2 3 4 to indicate the execution of the first four statements of the program
- explain why the mutant is killed by indicating the difference in the states of the original program and the mutant that allows for killing the mutant

We can perform weak mutation analysis by comparing **intermediate states** after the execution of each segment (single statement) in both programs (the original one and the mutant).

Let's perform the analysis: each row of the table shows both program states after executing a single statement.

Statement	Original program state	Mutant state	Mutant is alive
1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2}	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2}	YES
2	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3	YES
3	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3	YES
4	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0}	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0}	YES
5	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0	YES
6	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0	YES
7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0 column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0 column = 0	YES
8	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0 (it's not 2.0 because the whole right-hand expression is calculated before changing the variable value) column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 0.0 column = 0	YES
9	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 0	YES

7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	YES
8	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	YES
9	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 1	YES
7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 2	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 2	YES
8	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 2	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 4.0 column = 2	YES
9	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 8.0 column = 2	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 8.0 column = 2	YES
7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 8.0 column = 3	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {0.0,0.0,0.0} row = 0 sum = 8.0 column = 3	YES
10	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 0 sum = 8.0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 0 sum = 8.0	YES

5	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1	YES
6	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0	YES
7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0 column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0 column = 0	YES
8	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0 column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 0.0 column = 0	YES
9	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 0	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 0	YES
7	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 1	YES
8	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 1	YES
9	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 4.0 column = 1	matrix = {{2,0,2},{2,1,3},{0,2,4}} vector = {2,0,2} rows = 3 columns = 3 result = {8.0,0.0,0.0} row = 1 sum = 5.0 column = 1	NO

Piacente Cristian 866020

Homework H11 – Mutation Analysis

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

The sequence of statements up to the statement at which the execution kills the mutant is the following:
1 2 3 4 5 6 7 8 9 7 8 9 7 8 9 7 10 5 6 7 8 9 7 8 9.

The mutant is marked as dead after executing the last statement of the sequence, because when multiplying {2,1,3} with {2,0,2} an intermediate result is 2*2 + <u>1*0</u> for the original program, but 2+2 + <u>1+0</u> for the mutant.

EXTRA
I don’t get why line 8 is separated from line 9, actually it’s just one statement so line 8 can’t change the state.