Software Testing

Assessment type: Assignment 01	Marks: 10
Deadline: 5, October, 2020	Assignment Type: Group based

Name: Abadullah Mubarak Khan (BSE181026)

Waji ul hassan (BSE181024)

1. **Choose any problem statement of your choice and do following.**
    a. **Explain the case study of the problem**

    Despite the hard work of the programmer, sometimes they make mistakes or sometimes they forget to include all the possibilities of the question for which they are writing the program, which is very humanly in nature. And for those mistakes, a testing unit is always there. The job of this unit is to figure out what mistake has the programmer made or what did he/she forget to include. In some of the big organizations, they have huge teams of testing for their products because they have an important consideration in mind of the consequences of a software error. Most of the software usually need to stick to a single rule, i.e. to make sure that what is expected, it does. To use all the available resources in better sense, computers should also be helpful in "the art of software testing" to an improved extent, than is currently the case today. One of the issues today is if humans can make errors in coding then they can also make errors in software testing. The solution of this is not to remove human beings from the process of software testing but to use today's software development art form and make computers also participate. This thesis will present research aimed at classifying, examining, and improving the basic concept of boundary value analysis through automated software testing. To describe it further, we should first talk about software testing, boundary value analysis, functional dependencies and then possibility of creating a method that will solve the problem. 8 Software testing is defined as a formal process in which a software unit, several integrated software units or an entire package are examined by running the programs on a computer. All the associated tests are performed according to approved test procedures on approved test cases (Galin, 2004). Software testing is to test or check whether the software is executing perfectly and the necessary requirements are fulfilled. It is a human tendency to make errors and for elimination of these, tests are done on the product being developed to find out the problem in the software, that is why software testing is necessary. Software Testing is very expensive method in terms of time and money but it is the only way to find out bugs in the system. Due to time and budget constraints, it is impossible for us to perform exhausting testing for every set of test data, especially when there are enormous pools of input combinations. It requires an easy way or special techniques which will select test cases intelligently from the pool of cases, so that every test scenario is covered. So, optimal testing is

necessary to save time and money. The hard part in this procedure of testing is to generate the test cases. A test case is a condition put on each input parameter and those sets of conditions will give the tester an output which will help in the testing objective. A good testing technique is the one which covers every aspect of the requirement and the objective of testing. In the Late 90's test cases were derived manually but for some products this procedure takes time much more than the time of required for the development. So, the method of automated testing is introduced to test the product 9 automatically using program. Detecting and removing errors in the earlier phases will even reduce the cost of whole development.

b. Identify the functions (at least 3) from the case study and there must me at least one function that takes 3 parameters

Functions:

Level 1: Divide the parameters in three sets:

- D: dependent parameters,
- B: boundary-determining parameters,
- I: independent parameters.

Level2: For every parameter d D, create a set of its determining parameters. Each element of this set shall also be an element of set B.

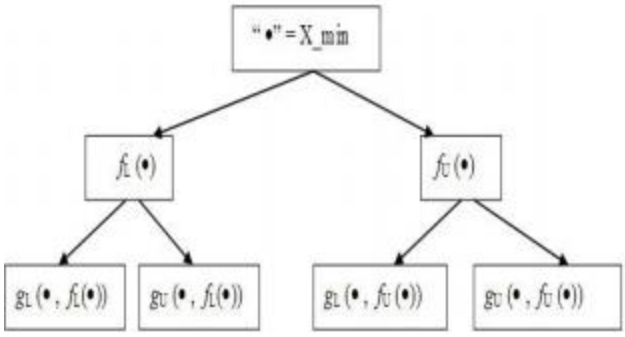
Level3: For every parameter d D, create separate sets of all possible boundary value ranges for d. Mark these sets d1 to dx. These sets must contain these bounds (min, min+, nom, max−, max) based on its boundary-determining parameters.

C Black box Testing

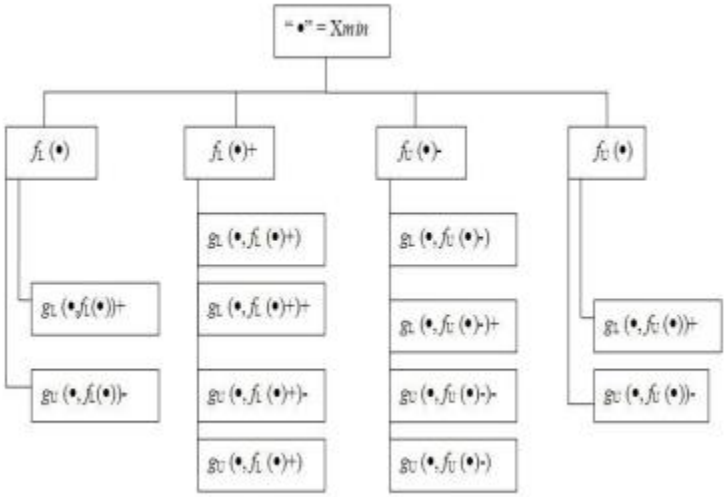
i. Using worst case BVA, identify test cases of **each function** and list down all test cases

(a)

Software Testing

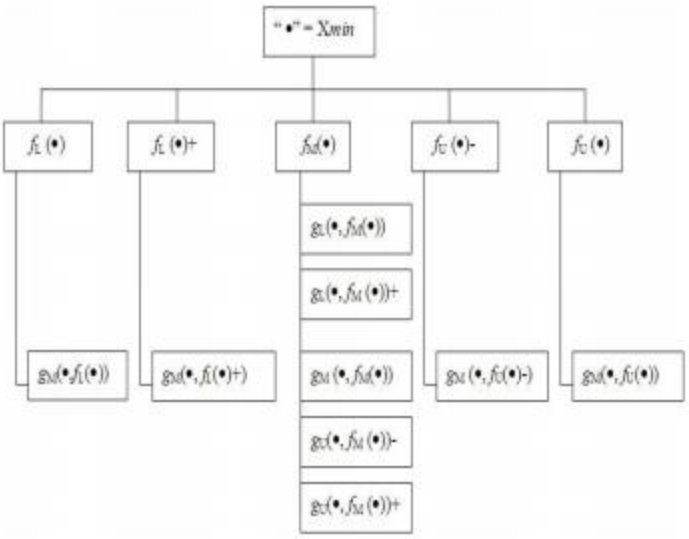


(b)



©

Software Testing



ii. **Implement function**

Each level is an upgrade of the other. So, it's decided only to implement the last level i.e. Level 3. The reason for this choice is that this level is almost like the traditional boundary value analysis except that in traditional boundary value analysis, all except 1 has the middle value but in Level 3 of Function Tree Algorithm has the middle value in at-least one of these parameters. Now, let assume that these three input parameters are X, Y and Z. Also, the following assumptions are made relating to the functions to be constrained:

(1) Xmin will be the lower bound and Xmax will be the upper bound for X. For Y, the upper bound and the lower bound will be the functions of X. So, Ymax = fU(X) …….. equation 3.1 Ymin = fL(X) …… equation 3.2 (2) For the third parameter, the upper bound and the lower bound will be functions of X and Y. So, Zmin = gL(X, Y) …….. equation 3.3 Zmax = gU(X, Y) …….. equation 3.4

(3) All the three parameters: X, Y and Z must satisfy the following constraint function: h(X, Y, Z) ≥ 0 or h(X, Y, Z) = 0 ……. equation 3.5

iii. **Implement 50% test cases of each function**

(a)

Software Testing

```
Enter the total number of dependent variables
2
Enter the 1 variable in order of Functional Dependency
a
Enter the 2 variable in order of Functional Dependency
b
Enter the min and max value respectively for variable a
14 67
Enter the equation for F(b)
Enter the constant term of variable b
56
Enter the coefficient of a^1
6
Enter the coefficient of a^2
-7
Enter the coefficient of a^3
0

Equation for F(b)=56+6a^1+-7a^2+0a^3
Test Case #1 40.5 -30965
Test Case #2 40.5 -30964
Test Case #3 40.5 -16098.5
Test Case #4 40.5 -1233
Test Case #5 40.5 -1232
Test Case #6 14 -16098.5
Test Case #7 15 -16098.5
Test Case #8 40.5 -16098.5
Test Case #9 66 -16098.5
Test Case #10 67 -16098.5
```

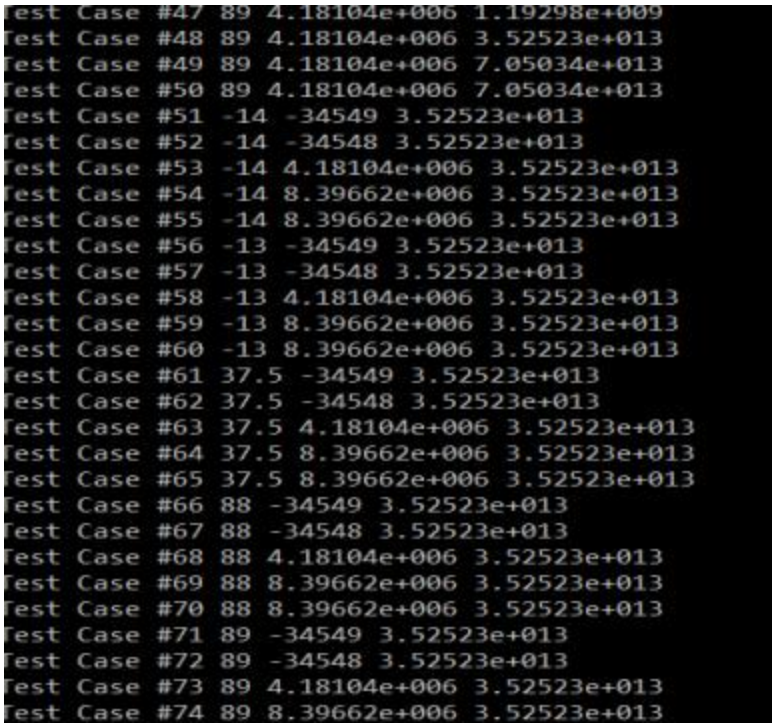
(b)

```
Enter the min and max value respectively for variable a
-14
89
Enter the equation for F(b)
Enter the constant term of variable b
3
Enter the coefficient of a^1
4
Enter the coefficient of a^2
-8
Enter the coefficient of a^3
12
Enter the equation for F(c)
Enter the constant term of variable c
7
Enter the coefficient of a^1
18
Enter the coefficient of a^2
-3
Enter the coefficient of a^3
0
Enter the coefficient of b^1
1
Enter the coefficient of b^2
4
Enter the coefficient of b^3
0

Equation for F(b)=3+4a^1+-8a^2+12a^3
Equation for F(c)=7+18a^1+-3a^2+0a^3+1b^1+4b^2+0b^3
```

©

Software Testing



b. Create account of github and commit all **code+word document+pdf document** on github. **No zip file (on email and github) will be acceptable**

c. Submit word and pdf document on team assignment

Notice

Do not use following case studies

1. Triangle problem
2. Next data problem
3. Commission problem

Checklist	Marks
Neatness	
Word document	
PDF document	
Git repository	
Team work	
Daily work (from 25 Nov-to 30 Nov)	
Source code	
Neatness	