Black-box testing techniques

Durga Prasad Mohapatra
Professor
Dept. of CSE
NIT Rourkela

Boundary Value Analysis

Boundary Value Analysis

- Some typical programming errors occur:
  - at boundaries of equivalence classes
  - might be purely due to psychological factors.
- Programmers often fail to see:
  - special processing required at the boundaries of equivalence classes.

Boundary Value Analysis

- Programmers may improperly use < instead of <=
- Boundary value analysis:
  - select test cases at the boundaries of different equivalence classes.

Example

- For a function that computes the square root of an integer in the range of 1 and 5000:
  - test cases must include the values {0,1,5000,5001} along with the values obtained from Equivalence partitioning.

BOUNDARY VALUE ANALYSIS (BVA)

- BVA offers several methods to design test cases. Following are the few methods used:
- **1.** BOUNDARY VALUE CHECKING (BVC)
- **2.** ROBUSTNESS TESTING METHOD
- **3.** WORST-CASE TESTING METHOD
- **4.** ROBUST WORST-CASE TESTING METHOD

BOUNDARY VALUE CHECKING (BVC)

- In this method, the test cases are designed by holding one variable at its extreme value and other variables at their nominal values in the input domain.
- The variable at its extreme value can be selected at:

BOUNDARY VALUE CHECKING (BVC)

- (a) Minimum value (Min)
- (b) Value just above the minimum value (Min+)
- (c) Maximum value (Max)
- (d) Value just below the maximum value (Max−)

BOUNDARY VALUE CHECKING (BVC)

- Let us take the example of two variables, A and B.
- If we consider all the above combinations with nominal values, then following test cases (see Fig. 1) can be designed:

- 1. Anom, Bmin
- 2. Anom, Bmin+
- 3. Anom, Bmax
- 4. Anom, Bmax–
- 5. Amin, Bnom
- 6. Amin+, Bnom
- 7. Amax, Bnom
- 8. Amax–, Bnom
- 9. Anom, Bnom

BOUNDARY VALUE CHECKING (BVC)

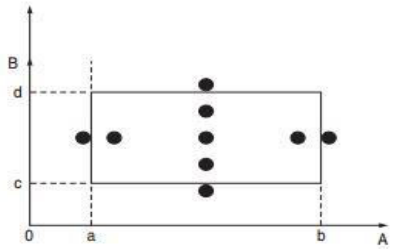

Fig 1: Boundary Value Checking

BOUNDARY VALUE CHECKING (BVC)

- It can be generalized that for n variables in a module, 4n + 1 test cases can be designed with boundary value checking method.

ROBUSTNESS TESTING METHOD

- The idea of BVC can be extended such that boundary values are exceeded as: □

1. A value just greater than the Maximum value (Max+)

2. □ value just less than Minimum value (Min−)

ROBUSTNESS TESTING METHOD

- When test cases are designed considering the above points in addition to BVC, it is called robustness testing.
- Let us take the previous example again. Add the following test cases to the list of 9 test cases designed in BVC:
- 10. Amax+, Bnom 11. Amin–, Bnom
- 12. Anom, Bmax+ 13. Anom, Bmin–

ROBUSTNESS TESTING METHOD

- It can be generalized that for n input variables in a module, 6n + 1 test cases can be designed with robustness testing.

WORST-CASE TESTING METHOD

- We can again extend the concept of BVC by assuming more than one variable on the boundary.
- It is called worst-case testing method.
- Again, take the previous example of two variables, A and B. We can add the following test cases to the list of 9 test cases designed in BVC as:

WORST-CASE TESTING METHOD

10. Amin, Bmin 11. Amin+, Bmin
12. Amin, Bmin+ 13. Amin+, Bmin+
14. Amax, Bmin 15. Amax–, Bmin
16. Amax, Bmin+ 17. Amax–, Bmin+
18. Amin, Bmax 19. Amin+, Bmax
20. Amin, Bmax– 21. Amin+,Bmax–
22. Amax, Bmax 23. Amax–, Bmax
24. Amax, Bmax– 25. Amax–,Bmax–

WORST-CASE TESTING METHOD

- It can be generalized that for n input variables in a module, 5^n test cases can be designed with worst-case testing.

ROBUST WORST-CASE TESTING METHOD

- In the previous method, the extreme values of a variable considered are of BVC only.
- The worst case can be further extended if we consider robustness also, that is,
- in worst case testing if we consider the extreme values of the variables as in robustness testing method covered in Robustness Testing

ROBUST WORST-CASE TESTING METHOD

- Again take the example of two variables, A and B. We can add the following test cases to the list of 25 test cases designed in previous section.
- 26. Amin-, Bmin-
- 27. Amin-, Bmin
- 28. Amin, Bmin-
- 29. Amin-, Bmin+
- 30. Amin+, Bmin-
- 31. Amin-, Bmax

- 32. Amax, Bmin-
- 33. Amin-, Bmax-
- 34. Amax-, Bmin-
- 35. Amax+,Bmax+
- 36. Amax+, Bmin
- 37. Amin, Bmin+
- 38. Amax+, Bmin+
- 39. Amax+,Bmax+
- 40. Amax+,Bmax
- 41. Amax, Bmax+
- 42. Amax+,Bmax-
- 43. Amax-, Bmax+
- 44. Amax+,Bnom
- 45. Anom, Bmax+
- 46. Amin-,Bnom
- 47. Anom, Bmin-
- 48. Amax+,Bmin-
- 49. Amin-, Bmax+

Example

A program reads an integer number within the range [1,100] and determines whether it is a prime number or not. Design test cases for this program using BVC, robust testing, and worst-case testing methods.

Test cases using BVC

- Since there is one variable, the total number of
- test cases will be 4n + 1 = 5.
- In our example, the set of minimum and maximum values is shown below:

- Min value = 1
- Min+ value = 2
- Max value = 100
- Max– value = 99
- Nominal value = 50–55

- Using these values, test cases can be designed as shown below:

Test Case ID	Integer Variable	Expected Output
1	1	Not a prime number
2	2	Prime number
3	100	Not a prime number
4	99	Not a prime number
5	53	Prime number

Test cases using robust testing

- Since there is one variable, the total number of test cases will be 6n + 1 = 7. The set of boundary values is shown below:

- Min value = 1
- Min– value = 0
- Min+ value = 2
- Max value = 100
- Max– value = 99
- Max+ value = 101
- Nominal value = 50–55

- Using these values, test cases can be designed as shown below:

Test Case ID	Integer Variable	Expected Output
1	0	Invalid input
2	1	Not a prime number
3	2	Prime number
4	100	Not a prime number
5	99	Not a prime number
6	101	Invalid input
7	53	Prime number

Test cases using worst-case testing

- Since there is one variable, the total number of test cases will be $5^n = 5$.
- Therefore, the number of test cases will be same as BVC.

Example

- A program computes a^b where a lies in the range [1,10] and b within [1,5].
- Design test cases for this program using BVC, robust testing, and worst-case testing methods.

Test cases using BVC

- Since there are two variables, a and b, the total number of test cases will be 4n + 1 = 9. The set of boundary values is shown below:

	a	b
Min value	1	1
Min+ value	2	2
Max value	10	5
Max– value	9	4
Nominal value	5	3

Using these values, test cases can be designed as shown below:

Test Case ID	a	b	Expected Output
1	1	3	1
2	2	3	8
3	10	3	1000
4	9	3	729
5	5	1	5
6	5	2	25
7	5	4	625
8	5	5	3125
9	5	3	125

Test cases using robust testing

- Since there are two variables, a and b, the total number of test cases will be 6n + 1 = 13.
- The set of boundary values is shown below:

	a	b
Min value	1	1
Min– value	0	0
Min+ value	2	2
Max value	10	5
Max+ value	11	6
Max– value	9	4
Nominal value	5	3

Using these values, test cases can be designed as shown below:

Test Case ID	a	b	Expected output
1	0	3	Invalid input
2	1	3	1
3	2	3	8
4	10	3	1000
5	11	3	Invalid input
6	9	3	729
7	5	0	Invalid input
8	5	1	5
9	5	2	25
10	5	4	625
11	5	5	3125
12	5	6	Invalid input
13	5	3	125

Test cases using worst-case testing

- Since there are two variables, a and b, the total number of test cases will be $5^n = 25$.
- The set of boundary values is shown below:

	a	b
Min value	1	1
Min+ value	2	2
Max value	10	5
Max– value	9	4
Nominal value	5	3

There may be more than one variable at extreme values in this case. Therefore, test cases can be designed as shown below :

Test Case ID	a	b	Expected Output
1	1	1	1
2	1	2	1
3	1	3	3
4	1	4	1
5	1	5	1
6	2	1	2
7	2	2	4
8	2	3	8
9	2	4	16
10	2	5	32
11	5	1	5
12	5	2	25
13	5	3	125
14	5	4	625
15	5	5	3125
16	9	1	9
17	9	2	81
18	9	3	729
19	9	4	6561
20	9	5	59049
21	10	1	10
22	10	2	100
23	10	3	1000
24	10	4	10000
25	10	5	100000

Summary

- We discussed black-box test case design using:
  - boundary value analysis
- Explained BVA with some examples.

References

1. Rajib Mall, Fundamentals of Software Engineering, (Chapter – 10), Fifth Edition, PHI Learning Pvt. Ltd., 2018.
2. Naresh Chauhan, Software Testing: Principles and Practices, (Chapter – 4), Second Edition, Oxford University Press, 2016.

Thank You