Lab 2 Report

Lab Report #1
ECE 322
Yenosibina Makanjuola 1629453

Introduction (6 marks)

The purpose of this lab was to gain familiarity with black box testing techniques. This lab focuses on Extreme Point Combination (EPC), and weak n x 1. In EPC we test all combinations of min, max, min – 1, max +1, and include a point that falls within the boundary, this generates a total of 4^n + 1 test cases where n is the number of inputs. In weak n x 1 we test n points on each subdomain and one point outside the subdomain where n is the total number of inputs, this test also checks for one point inside every subdomain, and in total it generates b (n +1) + 1 test cases where b is the number of boundaries and n is the number of inputs . The completion of this lab exercise ensures that a person understands and is exposed to the various black box testing methods mentioned above. To gain familiarity with the black box testing methods mentioned above we were given two programs to test. The first program is a drone program that enables the government to perform peaceful reconnaissance missions without the risk of sending a highly trained pilot and using very expensive aircraft and the second program is a program that accepts Cartesian inputs in the form of (x, y) that specify where the car should move.

Part One –Drone program (32 marks)

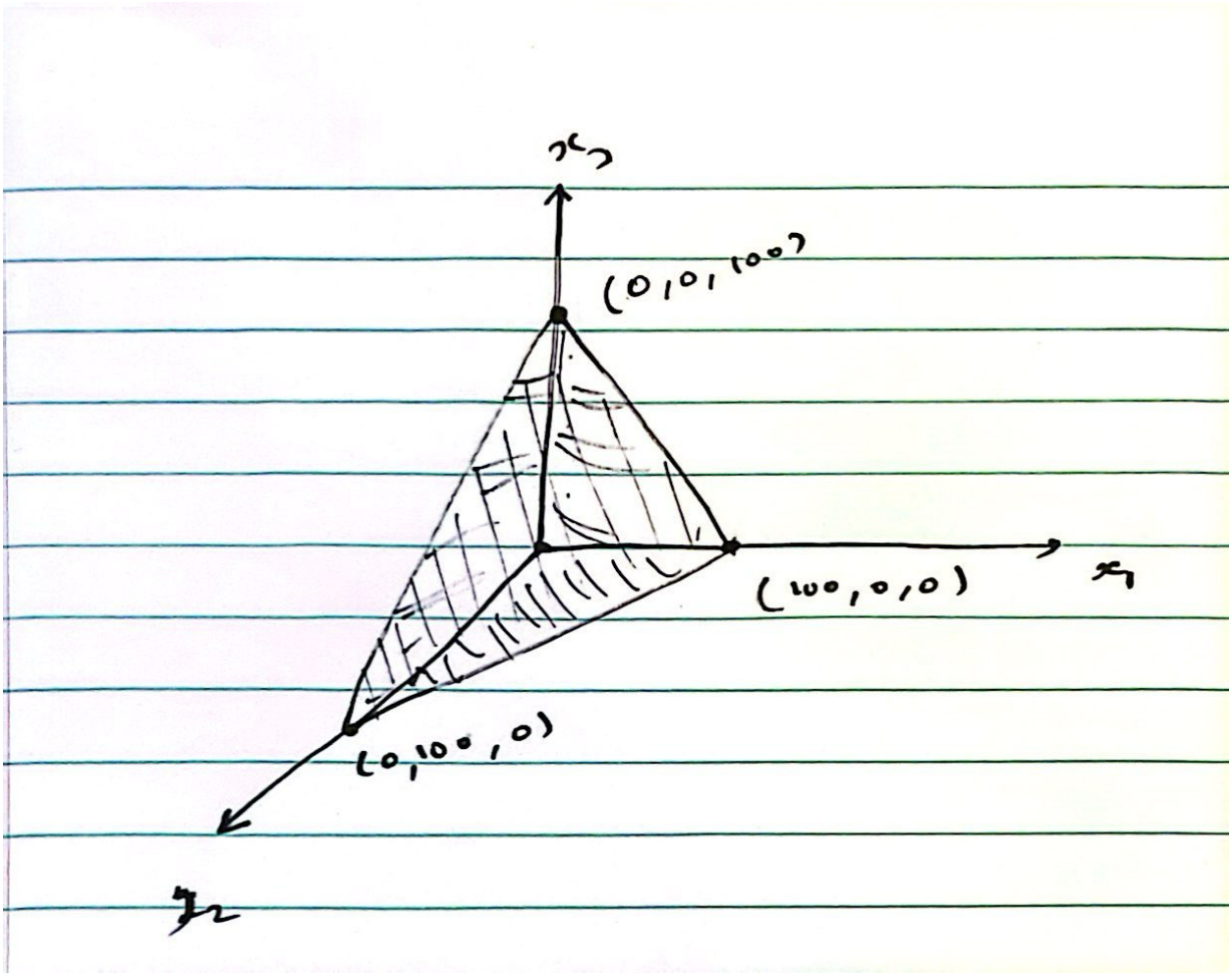
Q1 : An explanation of the application under test, they should describe the problem that is being tested:

Answer: The application under testing is a GPS-enabled automated pilot system. The application allows the government to perform peaceful reconnaissance missions without the risk of sending a highly trained pilot and using very expensive aircraft. In this application we are testing the maximum range the aircraft can travel before needing to turn back using Extreme Point Combination (EPC), and weak n x 1.

Q2: The error in the application should be identified, and some possible explanations given. (3 marks if missing)

Answer: In the application there was an error when checking for negative values for the input x2. In test cases 57 and 25 the application fails to check for invalid input for x2 we expected the program to give an invalid argument error, but it gave success in both cases. This issue could be as a result for the program not perform a proper check for the second input in the program

Q3: Subdomain drawing
Answer:



Q4: Discuss the effectiveness of the testing methods used
Answer:
For this program both EPC and weak n x 1 detected the same error which was the improper check of the second input in the program x2. Both black box testing techniques were able to discover the inability of the program to detect negative values for x2. In this scenario weak n x 1 is more effective because it took a lower number of test cases to detect the same error.

Q5 : Test Cases Table:

Answer:

1. For EPC(Extreme Point Combination) there are 4^n + 1 test cases. Therefore there are 65 test cases. For EPC we are going to test all combinations of max point , min point, min point -1, and max point + 1
    1. The max point for x1,x2,and x3 is 100
    2. The min point for x1,x2, and,x3 is 0

Id	Input x1	Input x2	Input x3	Description	Expected Result	Actual Result
1	0	0	0	Test case when all values are inputs are zero.	Success !	Success!
2	0	0	100	Test case when 2 inputs are at min value and one is at max value.	Success !	Success !
3	0	0	-1	Test case when two inputs are at min value and one is below the min value	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
4	0	0	101	Test case when two values are at min and one is above max	Failure!	Failure!
5	0	100	0	Test case when 2 inputs are at min value and one is at max value.	Success!	Success!
6	0	100	100	Test case when two inputs are at max and one is at min	Failure!	Failure!
7	0	100	-1	Test case when one value is at min the other is at max and the last is one value below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
8	0	100	101	Test case when one input is at min, one is at max, and one is above above max	Failure!	Failure!

9	0	-1	0	Test case when one input is below the min value and two of the inputs are at the min value	Error: Invalid Argument - negative value	Success!
10	0	-1	100	Test case when one input is below min, one is at the min, and one is at the max.	Error: Invalid Argument - negative value	Success!
11	0	-1	-1	Test case when two values are below min and one is at the min point.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
12	0	-1	101	Test case when one input is below min, one is at the min, and one is above the max	Error: Invalid Argument - negative value	Success!
13	0	101	0	Test case when two inputs are at the min value and one is above the max	Failure!	Failure!
14	0	101	100	Test case when one input is at the max, one at the min, and one above the max	Failure!	Failure!
15	0	101	-1	Test case when one input is above the max, one at the min, and one below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
16	0	101	101	Test case when two inputs are above the max and one is at the min	Failure!	Failure!
17	100	0	0	Test case when one input is at the max and two inputs are at the min.	Success!	Success!
18	100	0	100	Test case when two values are at the max and one is at the min	Failure!	Failure!
19	100	0	-1	Test case when one one input is at the min, one is	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value

				below the min, and onesies a the max		
20	100	0	101	Test case when one input is at the max, one is at the min, and one is above the max.	Failure!	Failure!
21	100	100	0	Test case when two inputs are at max, and one at min.	Failure!	Failure!
22	100	100	100	Test case where all inputs are at max	Failure!	Failure!
23	100	100	-1	Test case where two inputs are at max and one is below min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
24	100	100	101	Test case where two inputs are at max,and one is above max	Failure!	Failure!
25	100	-1	0	Test case where one input is at max, one is below min, and one is at min.	Error: Invalid Argument - negative value	Success!
26	100	-1	100	Test case where two values are at max and one is below min	Failure!	Failure!
27	100	-1	-1	Test case where one value is at max and two inputs are below min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
28	100	-1	101	Test case when one input is at max, one is above max, and one is below min	Error: Invalid Argument - negative value	Failure!
29	100	101	0	Test case when on input is at max, one is above max, and one is at min	Failure!	Failure!
30	100	101	100	Test case when two values are at the max, and one is above the max.	Failure!	Failure!

31	100	101	-1	Test case when on input is at max, one is above max, and one is below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
32	100	101	101	Test case when two inputs are above the max and one is at the max.	Failure!	Failure!
33	-1	0	0	Test case where two values are at the min and one is below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
34	-1	0	100	Test case where one value is below the min, one value is at the min, and one value is at the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
35	-1	0	-1	Test case where two inputs are below the min and one is at the min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
36	-1	0	101	Test case where one input is below the min, one is at the min, and one is above the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
37	-1	100	0	Test case where one input is below the min, one is at the max, and one is at the min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
38	-1	100	100	Test case where two inputs are at the max and one is below the min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
39	-1	100	-1	Test case where two inputs are below the min and one is at the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
40	-1	100	101	Test case where one input is below the min, one is at the max, and one is above the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value

41	-1	-1	0	Test case where two inputs are below the min and one is at the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
42	-1	-1	100	Test case where two inputs are below the min and one is at the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
43	-1	-1	-1	Test case where all inputs are below the min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
44	-1	-1	101	Test case where two inputs are below min and one is above max.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
45	-1	101	0	Test case where one input is below min, one is above max and one is at min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
46	-1	101	100	Test case where one input is below the min, one is above the max, and one is at the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
47	-1	101	-1	Test case where two inputs are at the min and one is above the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
48	-1	101	101	Test case where two inputs are above the max and one is below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
49	101	0	0	Test case where two inputs are at the min and one is above the max	Failure!	Failure!
50	101	0	100	Test case where one input is at the min, one is at the max, and one is above the max.	Failure!	Failure!
51	101	0	-1	Test case where one input is above the max, one is below	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value

				the min, and one is at the min		
52	101	0	101	Test case where one input is at the min and two are above the max	Failure!	Failure!
53	101	100	0	Test case where one is at the min, one is at the max, and one is above the max	Failure!	Failure!
54	101	100	100	Test case where two inputs are at the max and one is above the max	Failure!	Failure!
55	101	100	-1	Test case where one input is above the max, one is at the max, and one is below the min	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
56	101	100	101	Test case where two inputs are above the max and one is at the max	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
57	101	-1	0	Test case where one input is above the max, one is below the min, and one is at the min.	Error: Invalid Argument - negative value	Success!
58	101	-1	100	Test case where one input is above the max, one is below the min, and one is at the max.	Error: Invalid Argument - negative value	Failure!
59	101	-1	-1	Test case where one input is above the max, and two are below the min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
60	101	-1	101	Test case where two inputs are above max and one is below min.	Error: Invalid Argument - negative value	Failure!
61	101	101	0	Test case where two inputs are above max and one is at min.	Failure!	Failure!

62	101	101	100	Test case where two inputs are above max and one is at max.	Failure!	Failure!
63	101	101	-1	Test case where two inputs are above max and one is below min.	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
64	101	101	101	Test case where all inputs are above max	Failure!	Failure!
65	3	6	9	Test case were all inputs are positive and the sum of all inputs are less than 100	Success!	Success!

2. For weak n by 1 test there are b(n +1) + 1 test cases. Therefore there are 17 test cases

Id	Input x1	Input x2	Input x3	Description	Expected Result	Actual Result
1	0	3	5	On the boundary x1 = 0	Success!	Success!
2	0	6	4	On the boundary x1 = 0	Success!	Success!
3	0	55	30	On the boundary x1 = 0	Success!	Success!
4	-1	23	24	Outside the boundary x1 = 0	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
5	22	0	23	On the boundary x2 = 0	Success!	Success!
6	45	0	23	On the boundary x2 = 0	Success!	Success!
7	66	0	11	On the boundary x2 = 0	Success!	Success!

8	24	-1	25	Outside the boundary x2 = 0	Error: Invalid Argument - negative value	Success!
9	57	23	0	On the boundary x3 = 0	Success!	Success!
10	14	16	0	On the boundary x3 = 0	Success!	Success!
11	17	19	0	On the boundary x3 = 0	Success!	Success!
12	19	29	-1	Outside the boundary x3 = 0	Error: Invalid Argument - negative value	Error: Invalid Argument - negative value
13	40	30	30	On the boundary x1 + x2 + x3 =100	Success!	Success!
14	25	25	50	On the boundary x1 + x2 + x3 = 100	Success!	Success!
15	50	40	10	On the boundary x1 + x2 + x3 = 100	Success!	Success!
16	40	30	32	Outside the boundary x1 + x2 + x3 = 0	Failure!	Failure!
17	20	30	35	Inside every boundary	Success!	Success!

Part 2 – Remote Car Program (32 marks)

Q1: explanation of the application under test, and a description of the problem
Answer: The application under testing is a program that accepts Cartesian inputs in the form of (x, y) that specify where the car should move. In this application we are testing whether the instructions given to the car are understood by the car when the car falls withing a certain radius using Extreme Point Combination (EPC), and weak n x 1. For this problem we were asked to use an approximate subdomain to test the problem.s

Q2: Line segment equations

Answer:

a. y = x + 1
b. y = x -1
c. y = -x -1
d. y = -x + 1

Q3: A discussion regarding the results. You should comment on the effectiveness of the subdomain approximation, and comment on how this affected a) your test case choices and b) the test case results. This should answer the questions in the lab assignment regarding how the approximation would be affected with more lines and how that would affect the number of test cases.

Answer:

a. It affected my test cases choices because I had to pick values that fall only on the subdomain individually and when making choices of points outside the boundary these points must be reasonably far from the subdomain since it is an approximation because it would give wrong results due to the approximation.
b. The subdomain approximation is not very accurate as it does recognize a good number of inputs that fall within the range r when using weak n x 1 testing. Therefore, due to the approximation test cases 3, 6, 9, and 12 failed.
c. The domain approximation using a square is not effective. If we used an octagon for the approximation instead of a square the domain approximation would consider more points in the boundary. The complexity of the problem would be affected because there would be more subdomains needed to be considered therefore increasing the number of test cases. The number of test cases for EPC would still be the same but that of weak n x 1 would be 25.

Q5: Test Cases Table :

Answer:

1. For EPC (Extreme Point Combination) there are 4^n + 1 test cases. Therefore, there are 17 test cases. For EPC we are going to test all combinations of max point, min point, min point -1, and max point + 1.
    1. The max point for x and y is 1.
    2. The min point for x and y is -1.

Id	x	y	Description	Expected Result	Actual Result
1	-1.1	-1.1	Test case where both inputs are below the min.	Out of range!	Out of range!
2	-1.1	-1	Test case where one input is below min and the other is at min	Out of range!	Out of range!
3	-1.1	1	Test case where one input is below min and the other is at max	Out of range!	Out of range!
4	-1.1	1.1	Test case where one input is below the min and the other is one above the max	Out of range!	Out of range!
5	-1	-1.1	Test case where one input is at the min and the other is one below the min	Out of range!	Out of range!
6	-1	-1	Test case where both inputs are at min.	Out of range!	Out of range!
7	-1	1	Test case where one input is at min and the other is at max.	Out of range!	Out of range!
8	-1	1.1	Test case where one input is at the min and the other is above the max	Out of range!	Out of range!
9	1	-1.1	Test case where one input is at the max and the other is below the min	Out of range!	Out of range!
10	1	-1	Test case where one input is at the min and the other is at the max	Out of range!	Out of range!
11	1	1	Test case where both inputs are at the max	Out of range!	Out of range!
12	1	1.1	Test case where on input is at the max and the other is above the max	Out of range!	Out of range!

13	1.1	-1.1	Test case where one input is below the min and the other is below the min.	Out of range!	Out of range!
14	1.1	-1	Test case where one input is above the max and one is at the min	Out of range!	Out of range!
15	1.1	1	Test case where one input is at max and the other is above the max.	Out of range!	Out of range!
16	1.1	1.1	Test case where both inputs are above the max.	Out of range!	Out of range!
17	0	0	Test case where both inputs are within valid ranges	Ok.	Ok.

2. For weak n by 1 test there are b (n +1) + 1 test cases. Therefore, there are 13 test cases
   a. Boundary y = x + 1
   b. Boundary y = x -1
   c. Boundary y = -x -1
   d. Boundary y = -x + 1

Id	x	y	Description	Expected Result	Actual Result
1	-0.6	0.4	On boundary y = x + 1	Ok.	Ok.
2	-0.4	0.6	On boundary y = x + 1	Ok.	Ok.
3	-0.6	0.5	Outside boundary y = x + 1	Out of range!	Ok.
4	0.4	-0.6	On the boundary y = x - 1	Ok.	Ok.
5	0.2	-0.8	On the boundary y = x - 1	Ok.	Ok .
6	0.6	-0.5	Outside the boundary y = x - 1	Out of range!	Ok.
7	0.2	0.8	On the boundary y = -x + 1	Ok.	Ok.
8	0.6	0.4	On the boundary y = -x + 1	Ok.	Ok.
9	0.5	0.6	Outside the boundary y = -x + 1	Out of range!	Ok.
10	-0.6	-0.4	On the boundary y = -x -1	Ok.	Ok.
11	-0.4	-0.6	On the boundary y = -x -1	Ok.	Ok.
12	-0.4	-0. 7	Outside the boundary y = -x -1	Out of range!	Ok.

13	0	0	Inside all the boundaries	Ok.	Ok.

Conclusion and discussion (6 marks)

In this lab we were able to detect use both weak n x 1 testing and Extreme Point combination testing to find errors in both the drone program and the remote car program. In the drone program both EPC and weak n x 1 testing were both effective in testing the program because both testing methods detected the boundary error at input x2 where negative values were not being properly detected, but weak n x 1 testing was able to do this with a smaller number of test cases making it more efficient. In the remote car problem EPC is more effective than weak n x 1 testing because of the approximation being used, the boundary approximation being used didn't cover many points that were in the boundary therefore making the weak n x 1 method not very effective for this because it caused errors in the program due to the approximation meanwhile there were no errors. The main difficulty that arose from using weak n x 1 testing was finding the boundaries for the application being tested while the main difficulty for EPC was finding all the possible combinations for min, max, min -1, and max + 1.