CSE 4321: Software Testing and Maintenance
Fall 2020

Midterm Exam

10/15/2020

1. This is an open book/notes exam. No discussion/collaboration is allowed
2. This exam consists of 4 problems and has a total of 80 points.
3. This exam is 80 minutes long. Use your time wisely by working on the problems you consider easier first.
4. If you use the back of the exam sheets or if you use additional sheets, please indicate so. Write your name on the additional sheets.

Name: ANDREW DUONG

First Last

Problem 1: (3 + 3 + 3 + 3 + 3 = 15 points)

Consider the following faulty program (written in Java):

```
public int countPositive (int [] x) {
// effects: if x == null throw NullPointerException
//          else return the number of positive elements
//          in x
    int count = 0;
    for (int i = 0; i < x.length; i ++) {
        if (x[i] >= 0) {
            count ++;
        }
    }
    return count;
}
```

Answer the following questions:

(a) Identify the fault.
(b) If possible, identify a test case that does not execute the fault. Otherwise, explain why.
(c) If possible, identify a test case that executes the fault, but does not result in an error state. Otherwise, explain why.
(d) If possible, identify a test case that results in an error, but not a failure. Otherwise, explain why.
(e) If possible, identify a test case that results in a failure. Otherwise, explain why.

a) The fault is the if statement also counting 0 as positive.

b) Let x be NULL.

c) Let x = [1, 3, −2, 1]

d) That is not possible because if we put 0 into x[] to trigger the error state, then it will be a failure.

e) Let x = [1, 0, −2, 1]

Problem 2: (20 points)

Consider a system that consists of 3 parameters, P1, P2, and P3. P1 has three values: 0, 1 and 2. P2 has two values: *a* and *b*. P3 has three values: *x*, *y* and *z*. Apply algorithm IPO to build a pairwise test set for this system. The parameters should be covered in the given order, i.e., P1, P2, and P3. That is, you should build a pairwise test set T for P1 and P2 first, and then extend T to cover P3. Use "-" to represent *don't care* values, i.e., values that do not affect coverage. Clearly indicate your tie-breaking rules that may be needed in the test generation process. You must show intermediate steps to obtain full credits.

Step 1: Horizontal Extension and Vertical Extension for P3

P1	P2	P3
0	a	x
0	b	y
1	a	z
1	b	x
2	a	y
2	b	z
0	-	z
1	-	y
2	-	x

P1	P3
~~0~~	~~x~~
~~0~~	~~y~~
~~0~~	~~z~~
~~1~~	~~x~~
~~1~~	~~y~~
~~1~~	~~z~~
~~2~~	~~x~~
~~2~~	~~y~~
~~2~~	~~z~~

P2	P3
~~a~~	~~x~~
~~b~~	~~x~~
~~a~~	~~y~~
~~b~~	~~y~~
~~a~~	~~z~~
~~b~~	~~z~~

x : 2	y : 1	z : 1
x : 1	y : 2	z : 1
x : 1	y : 0	z : 2

Step 2: Final

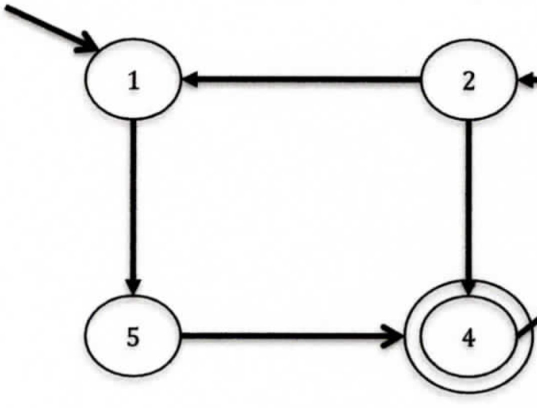
P1	P2	P3
0	a	x
0	b	y
1	a	z
1	b	x
2	a	y
2	b	z
0	-	z
1	-	y
2	-	x

P1	P3
~~0~~	~~x~~
~~0~~	~~y~~
~~0~~	~~z~~
~~1~~	~~x~~
~~1~~	~~y~~
~~1~~	~~z~~
~~2~~	~~x~~
~~2~~	~~y~~
~~2~~	~~z~~

P2	P3
~~a~~	~~x~~
~~b~~	~~x~~
~~a~~	~~y~~
~~b~~	~~y~~
~~a~~	~~z~~
~~b~~	~~z~~

Problem 3: (10 + 10 = 20 points)

Consider the following graph, where node 1 is the initial node, and node 4 is the final node. Note that a final node is allowed to have outgoing edges.



(a) Identify all the prime paths in the graph.
(b) Identify a test path set (i.e., one or more test paths) that achieves prime path coverage.

a)

length = 0
[1]
[2]
[3]
~~[4]~~!
[5]

length = 1
[1, 5]
[2, 1]
~~[2, 4]~~!
[3, 2]
[4, 3]
~~[5, 4]~~!

length = 2
~~[1, 5, 4]~~!
[2, 1, 5]
[3, 2, 1]
~~[3, 2, 4]~~!
[4, 3, 2]
[5, 4, 3]
[2, 4, 3]

length = 3
[1, 5, 4, 3]
~~[2, 1, 5, 4]~~!
[3, 2, 1, 5]
[3, 2, 4, 3]*
[4, 3, 2, 1]
[4, 3, 2, 4]!
[5, 4, 3, 2]
[2, 4, 3, 2]*

length = 4
[1, 5, 4, 3, 2]
[2, 1, 5, 4, 3]
[3, 2, 1, 5, 4]!
[4, 3, 2, 1, 5]
[5, 4, 3, 2, 1]

length = 5
[1, 5, 4, 3, 2, 1]*
[2, 1, 5, 4, 3, 2]*
[3, 2, 1, 5, 4, 3]*
[4, 3, 2, 1, 5, 4]*
[5, 4, 3, 2, 1, 5]*

Prime Path Coverage

[3, 2, 4, 3]*
[4, 3, 2, 4]!
[2, 4, 3, 2]*
[3, 2, 1, 5, 4]!
[1, 5, 4, 3, 2, 1]*
[2, 1, 5, 4, 3, 2]*
[3, 2, 1, 5, 4, 3]*
[4, 3, 2, 1, 5, 4]*
[5, 4, 3, 2, 1, 5]*

Problem 4: (2 + 3 + 5 + 5 + 5 + 5 = 25 points)

Consider the following data flow graph, which is defined by the sets of nodes, initial nodes, final nodes, edges, and definitions and uses.

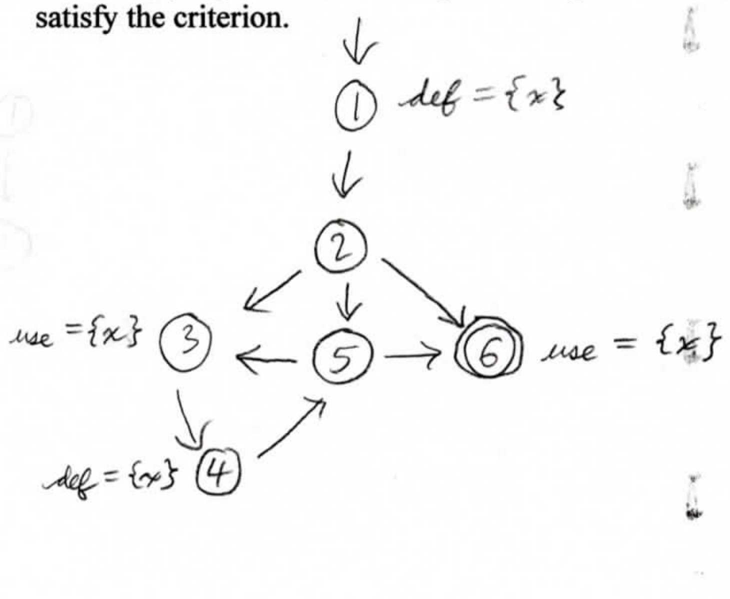
$N = \{1, 2, 3, 4, 5, 6\}$
$N_0 = \{1\}$
$N_f = \{6\}$
$E = \{(1, 2), (2, 3), (2, 5), (2, 6), (3, 4), (4, 5), (5, 3), (5, 6)\}$
def(1) = def(4) = {x}
use(3) = use(6) = {x}

A collection of test paths is also given:

$t1 = [1, 2, 6]$
$t2 = [1, 2, 5, 3, 4, 5, 6]$
$t3 = [1, 2, 3, 4, 5, 3, 4, 5, 6]$

(a) Draw the graph.
(b) List all of the *du-paths* with respect to *x*.
(c) Determine which *du-paths* each test path tours. Write them in a table with test paths in the first column and the *du-paths* they cover in the second column. Consider both direct touring and sidetrips.
(d) List a minimal test set that satisfies *all-defs* coverage with respect to x. (Direct tours only.) If possible, use the given test paths. If not, provide additional test paths to satisfy the criterion.
(e) List a minimal test set that satisfies *all-uses* coverage with respect to x. (Direct tours only.) If possible, use the given test paths. If not, provide additional test paths to satisfy the criterion.
(f) List a minimal test set that satisfies *all-du-paths* coverage with respect to x. (Direct tours only.) If possible, use the given test paths. If not, provide additional test paths to satisfy the criterion.

a)



b) i [1, 2, 3]
ii [1, 2, 6]
iii [1, 2, 5, 3]
iv [1, 2, 5, 6]
v [4, 5, 3]
vi [4, 5, 6]

(This blank page provides additional space for Problem 4.)

c)

test paths	direct	sidetrip
t1 = [1, 2, 6]	ii	
t2 = [1, 2, 5, 3, 4, 5, 6]	iii vi	
t3 = [1, 2, 3, 4, 5, 3, 4, 5, 6]	i v vi	

d) t1 & t2

e) t1 & t3

f) t1 & t3