ECE 322

SOFTWARE TESTING AND MAINTENANCE

Fall 2022

EXAMPLE PROBLEMS

1. Answer the following questions, and please be concise:

(i) Are there any similarities between random testing and operational profiles?

Solution
Both use a concept of randomness (probability) to choose test cases; however, these probabilities come from different sources.

(ii) Is the black box testing aimed at software validation or software verification? Justify your answer.

Solution
Validation, driven by specifications formulated by the user.

(iii) The module supposed to compute the value of the expression *sin*(x) + *cos*(x). The test case was completed for $x = \pi/4$, and the returned result is $\sqrt{2}$. Is the module free of faults?

Solution
We cannot say this. One might have encountered an effect of coincidental correctness.

(iv) Identify valid and invalid equivalence classes for the system having two inputs, namely, a name and a phone area code of a customer located in North America. A name is a string of letters longer than one and shorter than 60 characters. Answer to the following question.

1) What are valid equivalence classes.
2) Develop a suite of test cases for weak and strong equivalency testing.

Solution
Valid equivalence class for the name:
- Only letters, length of string of letters in the range of [2, 60)

Invalid equivalence classes for the name:

- Length of string of letters equal to 0 or 1.
- Length of string of letters greater or equal to 60

One can also consider strings composed of other symbols than letters; there could be other equivalence classes, say a mixture of letters/other symbols.

Equivalence classes for phone area. see e.g., https://en.wikipedia.org/wiki/List_of_North_American_Numbering_Plan_area_codes

Valid equivalence class for the phone area:

- 3-digit integer numbers in [200, 999]

Invalid equivalence classes for the phone area:

- Integers in the range of $(-\infty, 200)$
- Integers in the range of $(999, \infty)$

Note that even in the range [200, 999] you have some special cases (currently not used). There could be invalid equivalence classes formed by strings including letters, special symbols, etc.

Considering the previously mentioned equivalence classes, we have (invalid classes are shown in red):

- Name (Length of string of letters):
  - [0, 2) 1, [2, 60) 2, [61, *N*) 3 (*N*-the length of the longest string that can be accommodated in the system)

- Phone area code:
  - (0, 200) 4, [200, 999] 5, (999, M) 6 (assuming that more than 3 digits are accepted by the system)

Using the weak normal equivalence class strategy, the tests are:

(1, 5), (3, 5), (2, 4), (2, 6)

There could be some other options depending which combinations of valid and invalid equivalence classes have been selected.

Using the strong normal equivalence class strategy, we end up with 3 * 3 = 9 tests. The tests are:

(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)

(v) Complete the array below (fill in missing entries), so that it becomes an orthogonal array.

a	*b*	*c*	*d*
1	1	1	1
1		2	2
1	3	3	3
2	1	2	3
2	2		1
2	3	1	2
3	1	3	2
3		1	3
3	3	2	1

Solution

a	*b*	*c*	*d*
1	1	1	1
1	2	2	2
1	3	3	3
2	1	2	3
2	2	3	1
2	3	1	2
3	1	3	2
3	2	1	3
3	3	2	1

---

2. The university computer system allows students an allocation of disc space depending on their projects. If they have used allotted space, they have only allowed limited access, i.e., to delete files, not to create them. This is assuming they have logged in with a valid username and password.

Construct a reduced decision table and list a collection of tests.

Solution
Reduced decision table:

Input conditions				
Valid username	F	T	T	T
Valid password	-	F	T	T
Account in credit	-	-	F	T
Output condition				
Login accepted	F	F	T	T
Restricted access	-	-	T	F

---

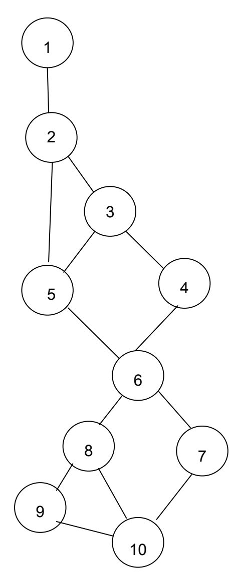
3. Given is the following code:

```c
#include <math.h>
#define PI 3.14159

/* function to compute a pseudo-angle */
double theta(Point2D p1, Point2D p2)
{
  double dx,dy,ax,ay,t;
  dx = p2.x - p1.x;
  ax = fabs(dx);
  dy = p2.y - p1.y;
  ay = fabs(dy);
/* check if line is vertical */
  if(dx==0.0 && dy==0.0)
    t=0;
  else
    t = dy/(ax+ay);
/* correct for quadrant */
 if(dx < 0.0)
    t = 2-t;
  else if (dy < 0.0)
    t = 4+t;
  return(t*PI/2);
}
```

Draw a control flow graph and determine its cyclomatic complexity.

Solution

```c
#include <math.h>
#define PI 3.14159

/* function to compute a pseudo-angle */
double theta(Point2D p1, Point2D p2)
{
  double dx,dy,ax,ay,t;
  dx = p2.x - p1.x;
1 ax = fabs(dx);
  dy = p2.y - p1.y;
  ay = fabs(dy);
/* check if line is vertical */
  if(dx==0.0 && dy==0.0)
4   t=0; 2          3
  else
  5 t = dy/(ax+ay);
/* correct for quadrant */
6 if(dx < 0.0)
  7 t = 2-t;
  else if (dy < 0.0) 8
  9 t = 4+t;
  return(t*PI/2); 10
}
```



The number of binary decision boxes is 4, so the cyclomatic complexity is 5. One can also count the number of regions (the graph is planar) and here we obtain 5.

If (2 and 3) are treated as a single decision box, then the cyclomatic complexity is 4 (one binary decision box less).

---

4. Consider a program that solves 2nd order differential equation:

$$y'' + ay' + cy = 0$$

(i) Identify valid equivalence classes.

Solution
We consider a characteristic polynomial

$$k^2 + ak + c = 0$$

and determine

$$D = a^2 - 4c$$

There are three cases implying a certain type of solution to this equation:

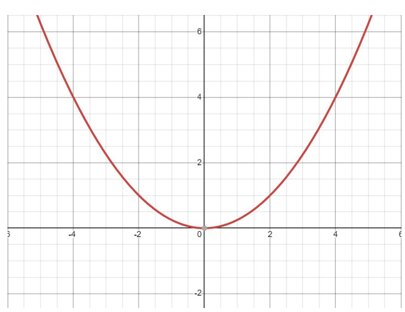
- $D > 0$: two real roots.
- $D = 0$: roots are real and equal.
- $D < 0$: there are two complex conjugate roots.

Thus, we have three equivalence classes W_1, W_2, W_3 formed in the (a, c) input domain:

- $W_1 = \{(a, c) \mid D > 0\}$
- $W_2 = \{(a, c) \mid D = 0\}$
- $W_3 = \{(a, c) \mid D < 0\}$

(ii) Are the boundaries linear? Sketch them.

Solution
The boundaries are nonlinear; a parabolic function in the (a, c) coordinates.



Consider x-axis is representing a, and y-axis is representing c. W_1 is the region below the curve. W_2 are the points on the curve. W_3 is the region above the curve.

(iii) Develop test cases.

Solution
We have to select one test case for each equivalence class; hence, one the test cases in the form of (a, c) would be:

$$(2, 0) \text{ in the } W_{1,} \text{ and } (4, 4) \text{ in the } W_{2,} \text{ and } (0, 2) \text{ in the } W_3$$

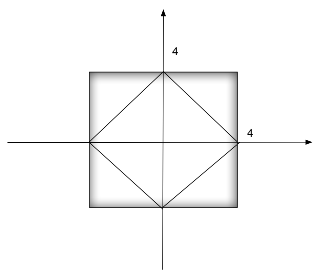
---

5. The subdomain is described by the following relationships:

- $|x_1| + |x_2| > 4$
- $\text{Max}(|x_1|, |x_2|) \leq 4$

(i) Plot this subdomain. Is it open or closed?

Solution
The subdomain is illustrated in the figure below, which is neither open nor closed:

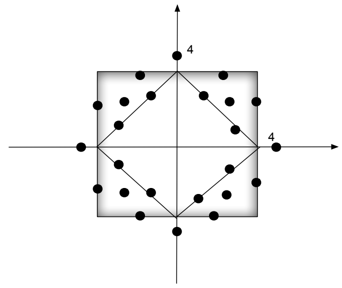


(ii) Use the EPC strategy and list a set of test cases.

Solution
the EPC strategy positions the test cases at the corners of the square.

(iii) Show test cases produced by the weak n x1 strategy.

Solution
The weak n x1 strategy is realized by test cases indicated by black dots.


4
4