ECE 322

SOFTWARE TESTING AND MAINTENANCE

FINAL EXAMINATION (SAMPLE)

OPEN BOOK

2hr exam

Solutions

Name (please print)_____
ID _____

Certification: I certify that when working on the final examination, I have not communicated/worked with any student in the ECE 322 class. Signature:

GOOD LUCK!

1	2	3	4	5	6	7	8	9	Σ
10	16	5	10	10	5	10	10	10	**86**

1. [10 points]
(i) Using the modified condition/branch coverage criterion, develop a suite of test cases for the following composite condition

a||((!b)&& c && d)

Solution

	a	b	c	d	a||((!b)&& c && d)
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	1
10	1	0	0	1	1
11	1	0	1	0	1
12	1	0	1	1	1
13	1	1	0	0	1
14	1	1	0	1	1
15	1	1	1	0	1
16	1	1	1	1	1

a 2-10
b: 4-8
c: 2-4
d: 3-4
the tests are 2-3-4-8-10.

(ii) How many test cases is required when using the weak n x 1 testing strategy for the boundary in the 5-dimensional space (x_1, x_2, x_3, x_4, x_5) described as follows

$$|x_1\text{-}2| + |x_2\text{-}3| + |x_3+1| + |x_4+2| + |x_5+1| \leq 10$$

Could you generalize the result when dealing with an n-dimensional space (x_1, x_2, …, x_n)

Solution
n=2: $2^2(2+1)+1$
n=3: $2^3(3+1)+1$
n=4: $2^4(4+1)+1$
n=5: $2^5(5+1)+1$
...
n: $2^n(n+1)+1$

2. [16 points; each question is worth 2 points]
Circle only one answer for each of the questions posed below. If you find any question that appears to have more than one correct answer, choose the one that is the most specific or appropriate.

1. Suppose that C++ programmer mistakenly writes

```
if (B)
   S1;
   S2;
```

instead of

```
if (B) {
   S1;
   S2;
      }
```

Which of the following statements are true?
(i) it needs only one test to do a statement coverage test for both programs
(ii) it requires at least two test cases to do a branch coverage test for both programs
(iii). The error will never be reflected in the test result if only one test case is used to do the statement coverage test

(a) all
(b) (i) and (ii) only
(c) (i) and (iii) only
(d) (ii) and (iii) only
(e) None of (a) through (d) is a correct answer

2. The presence of data flow anomaly indicates that
(i) the program is definitely in error
(ii) the program is possibly in error
(iii) a datum is defined and then used repeatedly without being redefined

Which of the foregoing statement are true?
a. All
b. All but (i)
c. All but (ii)
d. All but (iii)
e. None of (a) through (d) is a correct answer

3.A test driver (driver) in integration testing is:

(a) a simulator of a calling program (dummy program unit) used in a bottom-up integration testing

(b) a program unit designed to feed the test cases automatically

(c) a special input/output routine used to control the input device

(d) a simulator of a calling program (dummy program unit) used in a top-down integration testing

(e) a program designed to generate test data automatically

4.A stub used in integration testing is:

(a) a driver used in a unit test

(b) a subprogram designed to generate and feed the test cases

(c) a simulator of a called program (dummy program unit) used in a top-down integration testing

(d) a program segment designed to prevent a "runaway" program from damaging other programs during a test

(e) a simulator of a calling program (dummy program unit) used in a bottom-up integration test

5.Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.

a. True

b. False

6.System maintenance is necessary because:

a. Humans never get it right the first time.

b. The deployment platform may change over time.

c. The user's needs may change over time.

d. All of the above.

e. None of the above.

7.In an analysis of some of the life cycle models, we can conclude that the _____ model is the best.

a. Waterfall

b. Spiral

c. XP

d. Formal method

e. Re-use

f. None of the above

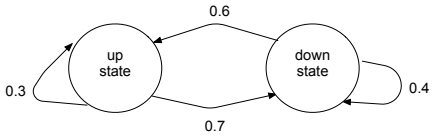
8.The five general phases in the spiral model are:

a. Analysis, Design, Engineering, Testing, and Payment
b. Analysis, Design, Implementation, Testing, and Review
c. Review, Decision, Engineering, Acceptance, and Planning
d. Review, Risk-analysis, Design, Implementation, and Planning
e. Review, Risk-analysis, Prototyping, Engineering, and Planning

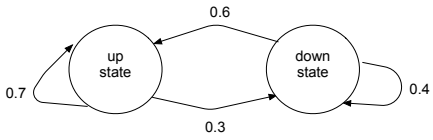
3. [5 points]
(i) Are there any similarities between random testing and operational profiles?

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

4. [10 points]
Which of the two software systems (A or B) has higher availability? By availability we mean the probability being in the up state.



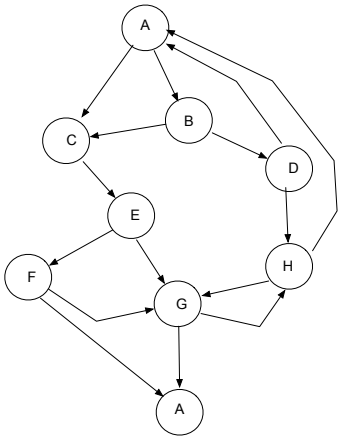
System A



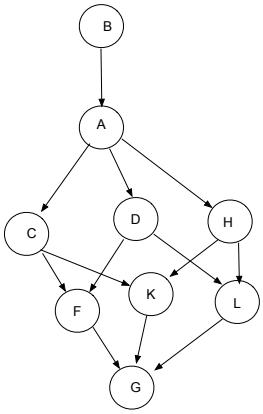
System B

Solution
System B has higher availability.

5. [10 points]
(i) Determine cyclomatic complexity of the following control flow graphs



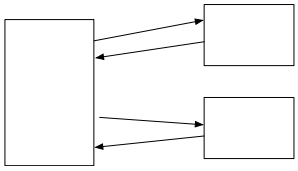
(a)



(b)

Solution
a: Planar regions: 8
b: e =13; n=9 $v(G)$=13-9+2 = 6

(ii) Determine the cyclomatic complexity of the overall system $v(G)$ composed of three modules as shown below and knowing that each of them has the cyclomatic complexity $v(G_1)$, $v(G_2)$, and $v(G_3)$.



Solution $v(G_1)= e_1-n_1+2$, $v(G_2)= e_2-n_2+2$, $v(G_3)= e_3-n_3+2$.
$v(G)= e_1+e_2+e_3+4 -(n_1+n_2+n_3)+2 = v(G_1) + v(G_2) + v(G_3)$.

6. **[5 points]**
A software module realizes a sorting procedure of an n-dimensional array $\boldsymbol{a}$ of real numbers. Propose a metamorphic relation to carry out metamorphic testing.

Solution
Permutation of original array, perm($\boldsymbol{a}$)

Idempotence: sort(sort($\boldsymbol{a}$))=sort($\boldsymbol{a}$)

7.[10 points]
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 decision table and reduce it.

Solution

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

8. [10 points]
(i) Complete the missing entries of the table to make it 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

(ii) Identify equivalence classes and then develop test cases for the problem in which there are two input variables:
Name: string of letters of length >1 and lower than 60
Product price <20$ or >100$
How many test cases are required?

Solution
Equivalence classes

Name: string of letters of length >1 and < 60
Valid equivalence class string of length (1, 60)
Invalid equivalence class: string of length less or equal to 1,
string of length equal or greater than 60

valid equivalence class <20 $
valid equivalence class >100$
Invalid equivalence classes
Price in range [20, 100]

Weak normal equivalence class testing [*single* fault assumption]
Design a test case so that it covers one and only one invalid equivalence class and as many as possible valid equivalence classes
3 test cases

Strong normal equivalence class testing [*multiple* fault assumption]
Select test cases from the Cartesian product of equivalence classes.
Covers all equivalence classes; one of each possible combinations of inputs
9 test cases

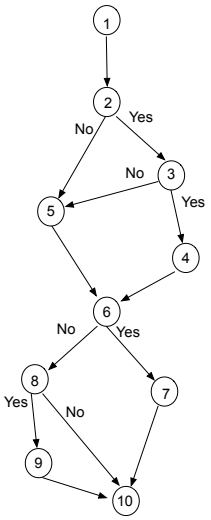
9. [10 points]
Given is the following code

```
#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);
}
```

(i)Draw a control flow graph and determine its cyclomatic complexity.
(ii) suggest test cases to complete branch/decision coverage criterion. What is the smallest number of test cases required to satisfy this requirement. Show the corresponding parts of the graph being traversed by the developed test cases. How does the number of test cases compare with the cyclomatic complexity?

Solution




```
#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);
}
```

Cyclomatic complexity: 5
Test cases:

p1.x=2, p2.x=2 p1.y =1, p2.y =1 traverses the path 1-2-3-4-6-8-10

p1.x=2, p2.x=1 p1.y =1, p2.y =1 traverses the path 1-2-5-6-7-10

p1.x=2, p2.x=4 p1.y =3, p2.y =1 traverses the path 1-2-3-5-6-8-9-10