Review session

Problem #1

1. The equivalence classes forming a partition of $\boldsymbol{X}$ =[0,1] x [0,1] are described in the form

$$A_1= \{(x_1,x_2) \mid x_1<a,\ x_2<a\},\ a>0,\ a<1.$$

$$A_2=\boldsymbol{X}-A_1$$

(i)When running random testing, what is the probability that both equivalence classes are covered.

(ii) extend (i) to the input domain being an n-dimensional unit hypercube $\boldsymbol{X}$ = [0,1] x [0,1]x ... x[0,1] with $A_1 =\{(\boldsymbol{x}_1, \boldsymbol{x}_2) \mid x_1<a, x_2<a,\ldots, x_n<a\}$. How does the dimensionality of the input domain (n) impact the probability of coverage of the equivalence classes.

Problem #1

1. The equivalence classes forming a partition of $\boldsymbol{X}$ =[0,1] x [0,1] are described in the form

$$A_1 = \{(x_1, x_2) \mid x_1 < a,\ x_2 < a\},\ a > 0,\ a < 1.$$

$$A_2 = \boldsymbol{X} - A_1$$

(i)When running random testing, what is the probability that both equivalence classes are covered.

(ii) extend (i) to the input domain being an n-dimensional unit hypercube $\boldsymbol{X}$ = [0,1] x [0,1]x ... x[0,1]

with $A_1 = \{(\boldsymbol{x}_1, \boldsymbol{x}_2) \mid x_1 < a, x_2 < a, \ldots, x_n < a\}$. How does the dimensionality of the input domain (n) impact the probability of coverage of the equivalence classes.

(i)prob= $1-a^2$

(ii)prob=$1-a^n$

Problem #2

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 decision table and reduce it.

Problem #2

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 decision table and reduce it.

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

Problem #3

3.Sensors generate measurement results $x_1, x_2, \ldots, x_5$.
A classifier takes a sum of absolute values of these measurements and produces an alarm message when the following relationship is satisfied

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

(i)How many test cases are required when using the weak n x 1 testing.
(ii)Generalize the result when dealing with n-sensors ($x_1, x_2, \ldots, x_n$).

Problem #3

3.Sensors generate measurement results $x_1, x_2, \ldots, x_5$.
A classifier takes a sum of absolute values of these measurements and produces an alarm message when the following relationship is satisfied

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

(i)How many test cases are required when using the weak n x 1 testing.
(ii)Generalize the result when dealing with n-sensors ($x_1, x_2, \ldots, x_n$).

n =2 : 4 line segments ++ +- -+ -- for each line 3 points (2+1 on/off) total $2^2(2+1)$ test cases

n = 3: octahedron centered at the origin; triangular planar face
2^3 planes; for each plane 4 points (3+1) total $2^3(3+1)$ test cases

n =4: 2^4 hyperplanes (cross polytope); for each (4+1) points, total 2^4 (4+1)

n=2: $2^2(2+1)+1$
n=3: $2^3(3+1)+1$
n=4: $2^4(4+1)+1$
n=5: $2^5(5+1)+1$
...
In general, we have $2^n (n+1)+1$ test cases.

Problem #4

4. Given are the following equivalence classes in the two-dimensional space of real numbers.

$$A_1(\boldsymbol{x},\boldsymbol{v}_1,r_1)=\{(x_1, x_2) \mid (x_1-v_{11})^2+(x_2-v_{12})^2 < r_1{}^2)$$
$$A_2(\boldsymbol{x},\boldsymbol{v}_2,r_2)=\{(x_1, x_2) \mid (x_1-v_{21})^2+(x_2-v_{22})^2 < r_2{}^2)$$
$$A_3(\boldsymbol{x},\boldsymbol{v}_3,r_3)=\{(x_1, x_2) \mid (x_1-v_{31})^2+(x_2-v_{32})^2 < r_3{}^2)$$
$$A_4(\boldsymbol{x})= \boldsymbol{X}\text{-}A_1\text{-}A_2\text{-}A_3.$$

where $\boldsymbol{v}_1= [v_{11}\ v_{12}]$, $\boldsymbol{v}_2= [v_{21}\ v_{22}]$, $\boldsymbol{v}_1= [v_{31}\ v_{32}]$.

Plot these equivalence classes.
Develop a reduced decision table to test whether these equivalence classes form a partition of $\boldsymbol{X}$.

Problem #4

4. Given are the following equivalence classes in the two-dimensional space of real numbers.

$A_1(\boldsymbol{x},\boldsymbol{v}_1,r_1)=\{(x_1, x_2) \mid (x_1-v_{11})^2+(x_2-v_{12})^2 < r_1^2)$
$A_2(\boldsymbol{x},\boldsymbol{v}_2,r_2)=\{(x_1, x_2) \mid (x_1-v_{21})^2+(x_2-v_{22})^2 < r_2^2)$
$A_3(\boldsymbol{x},\boldsymbol{v}_3,r_3)=\{(x_1, x_2) \mid (x_1-v_{31})^2+(x_2-v_{32})^2 < r_3^2)$
$A_4(\boldsymbol{x})= \boldsymbol{X}-A_1-A_2-A_3.$

where $\boldsymbol{v}_1= [v_{11}\ v_{12}]$, $\boldsymbol{v}_2= [v_{21}\ v_{22}]$, $\boldsymbol{v}_1= [v_{31}\ v_{32}]$.

Plot these equivalence classes.
Develop a reduced decision table to test whether these equivalence classes form a partition of $\boldsymbol{X}$.

A_1-A_4 : Open disks.
F1: $\mathrm{dist}(\boldsymbol{v}_1,\boldsymbol{v}_2) >= r_1+r_2$
F2: $\mathrm{dist}(\boldsymbol{v}_1,\boldsymbol{v}_3) >= r_1+r_3$
F3: $\mathrm{dist}(\boldsymbol{v}_2,\boldsymbol{v}_3) >= r_2+r_3$

F1	F	T	T	T
F2	-	F	T	T
F3	-	-	F	T
partition	N	N	N	Y

Problem #5

5.SHORT QUESTIONS

(1) Given a certain view at software specifications and resulting software system presented below, how would you position test cases produced by black box and white box testing approaches?

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

What are the differences between random testing and fuzzing?

(3) Why would you consider using constraints in the development of cause-effect graphs? In other words, in which sense are they useful?

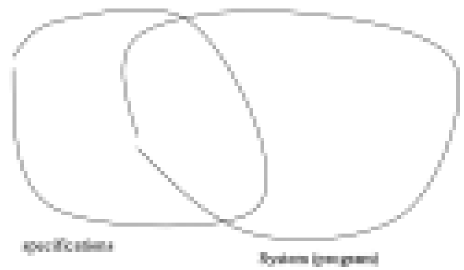
(4) What is the main feature of the McCall software quality model.

(5) What are the main advantages of Petri nets over finite state machines.

Problem #5

5.SHORT QUESTIONS

(1) Given a certain view at software specifications and resulting software system presented below, how would you position test cases produced by black box and white box testing approaches?



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

probabilities

What are the differences between random testing and fuzzing?

level of sophistication, feedback loop

(3) Why would you consider using constraints in the development of cause-effect graphs? In other words, in which sense are they useful?

constraints- reduced number of test cases

(4) What is the main feature of the McCall software quality model.

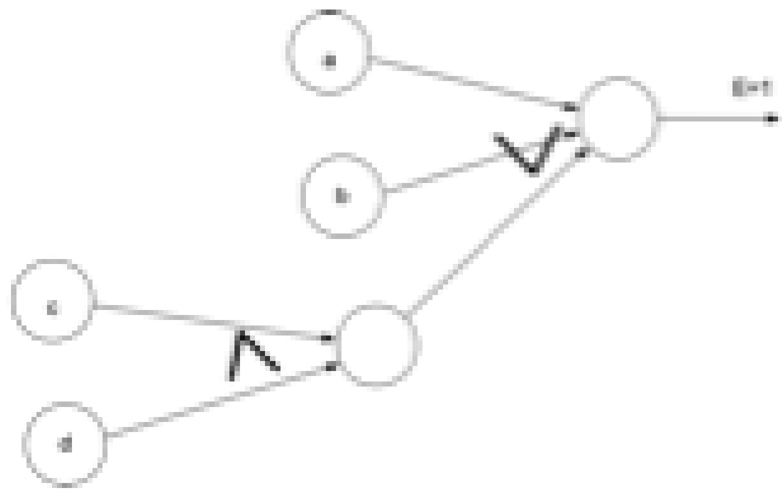
operation, revision, transition

(5) What are the main advantages of Petri nets over finite state machines.

concurrency and parallelism

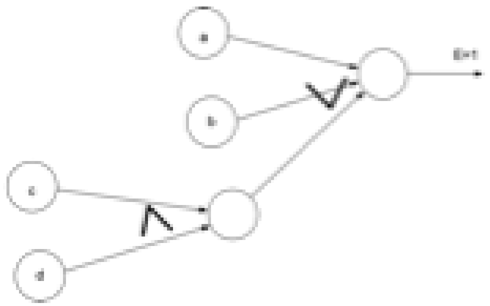
Problem #6

6.For the following cause-effect graphs, develop a decision table for $E = 1$.



Problem #6

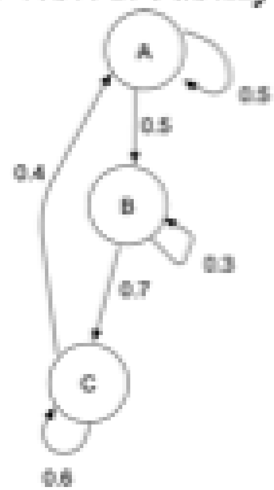
6.For the following cause-effect graphs, develop a decision table for $E = 1$.



a	0	1	0
b	0	0	1
c	1	0	0
d	1	0	0
E	1	1	1

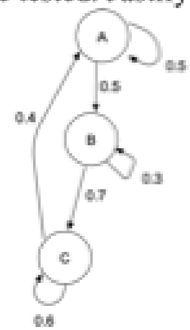
Problem #7

7. The software system is composed of three modules and the interaction between them is modeled by a Markov model with the transition values shown below. If you were to test the modules, what would be an order in which they are to be tested. Justify your choice.



Problem #7

7.The software system is composed of three modules and the interaction between them is modeled by a Markov model with the transition values shown below. If you were to test the modules, what would be an order in which they are to be tested. Justify your choice.



$$p_A = 0.5p_A + 0.4p_C$$
$$p_B = 0.5p_A + 0.3p_B$$
$$p_C = 0.7p_B + 0.6p_C$$

The probabilities are p_A, p_B, and p_C organized in a vector $\boldsymbol{p}$

$$\boldsymbol{p} = [0.3373, 0.2410, 0.4217]$$

Problem #7

```
# Define the coefficient matrix A
A = np.array([[0.5,0,-0.4], [0.5,-0.7,0],
              [1,1,1]])

# Define the constant vector b
b = np.array([0,0,1])

# Solve the system
x = np.linalg.solve(A, b)

print("Solution (x, y):", x)
```

Problem #8

8. Given is a context free grammar describing computing with trigonometric functions.

<expr >::=<expr> + <term>

< expr>::= <expr> - <term>

<expr>::= <term>

< term>::= <term> * <factor>

<term>::= <term>/ <factor>

<term>::= <factor>

< factor>::= <trigFunc>

<factor >::=(<Expr >)

< factor>::=< number>

< factor >::=<variable>

<trigFunc>::= < funcName (funcArg)>

<funcArg >::=<expr>

<funcName>::= sin | cos | tan | cot | sec | csc

< number>::= < digit >| <digit><number>

<digit>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

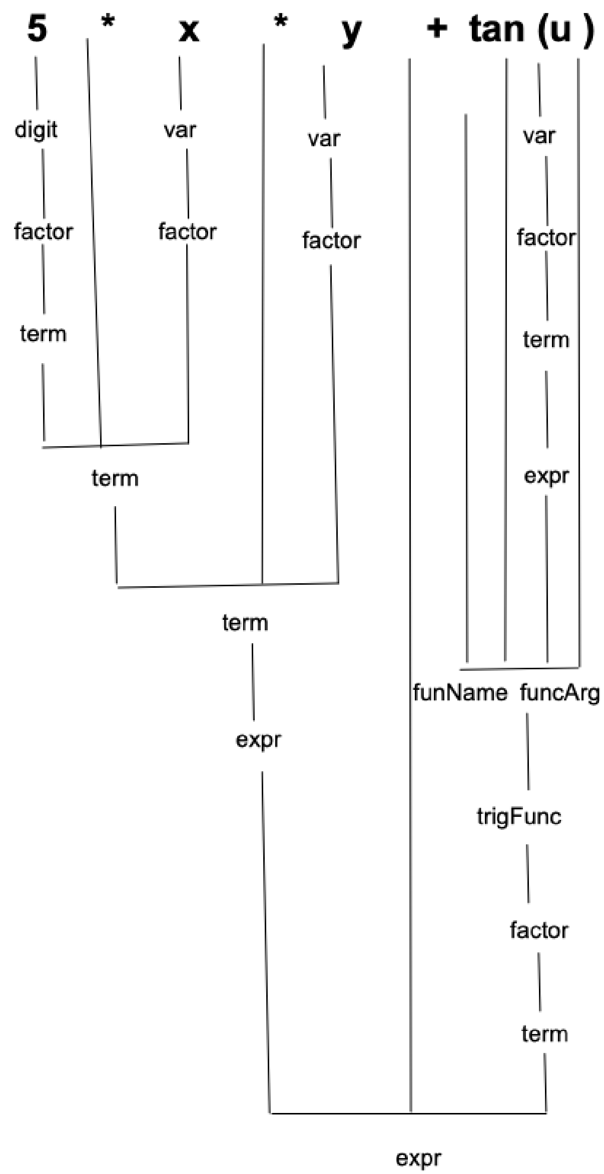
<variable>::= x | y | z | a | b | c | θ | α | β | ... | π

(i)Discuss the number of test cases to satisfy the criterion of production rules coverage.

(ii)Show production rules that are covered by the following expression

5*x*y + tan(u).

Problem #8



<expr >::=<expr> + <term>

< expr>::= <expr> - <term>

<expr>::= <term>

< term>::= <term> * <factor>

<term>::= <term>/ <factor>

<term>::= <factor>

< factor>::= <trigFunc>

<factor >::=(<Expr >)

< factor>::=< number>

< factor >::=<variable>

<trigFunc>::= < funcName (funcArg)>

<funcArg >::=<expr>

<funcName>::= sin | cos | tan | cot | sec | csc

< number>::= < digit >| <digit><number>

<digit>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<variable>::= x | y | z | a | b | c | θ | α | β | ... | π

Problem #9

9. Determine a cyclomatic complexity of the following code

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        # Last i elements are already sorted
        for j in range(0, n - i - 1):
            # Compare adjacent elements
            if arr[j] > arr[j + 1]:
                # Swap if elements are in wrong order
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr
```

Problem #9

```
def bubble_sort(arr):
 1 n = len(arr)
 2 for i in range(n):
      # Last i elements are already sorted
   3 for j in range(0, n - i - 1):
        # Compare adjacent elements
    4   if arr[j] > arr[j + 1]:
          # Swap if elements are in wrong order
      5   arr[j], arr[j + 1] = arr[j + 1], arr[j]
  return arr
```