DD2459 Software Reliability

Lab - 1

Md Sakibul Islam
msis3@kth.se
Md Ahsanul Karim
makari@kth.se

White-box Testing
Feb 16, 2026

Question 1: Draw a condensation graph for the Triangle Test algorithm.

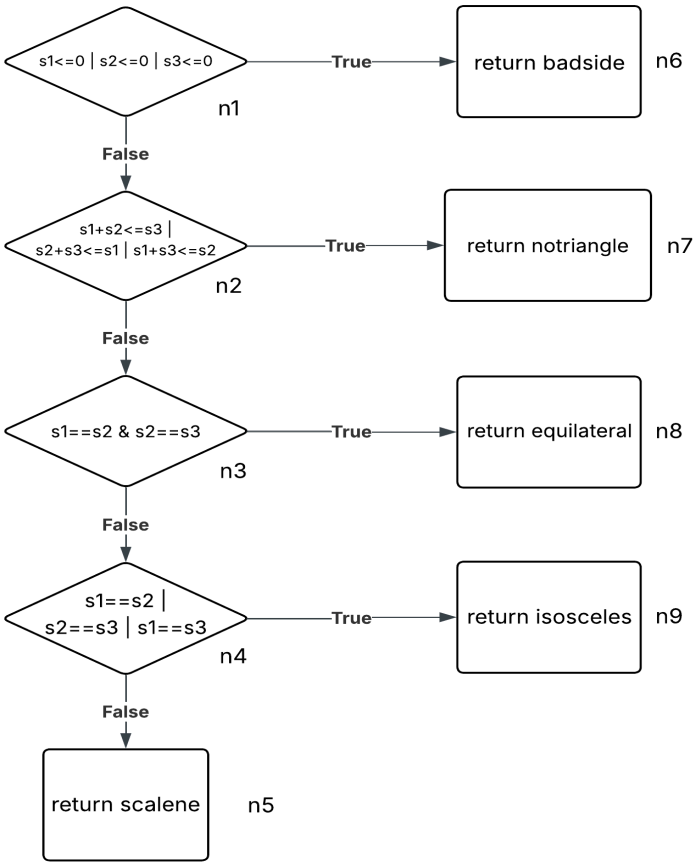

Figure - 1 : Condensation graph

Answers

1.1 (a). Test Requirements for 100% Node Coverage

NC TR1: n_1, n_6

NC TR2: n_1, n_2, n_7

NC TR3: n_1, n_2, n_3, n_8

NC TR4: n_1, n_2, n_3, n_4, n_9

NC TR5: n_1, n_2, n_3, n_4, n_5

1.1 (b). Minimized Test Cases:

TR1 : triangleTest(-1,1,1)
TR2 : triangleTest(2,3,5)
TR3 : triangleTest(2,3,5)
TR4 : triangleTest(2,2,2)
TR5 : triangleTest(4,5,6)

1.2 (a). Test Requirements that achieve full edge coverage (EC):

EC TR1: n_1, n_6

EC TR2: n_1, n_2, n_7

EC TR3: n_1, n_2, n_3, n_8

EC TR4: n_1, n_2, n_3, n_4, n_9

EC TR5: n_1, n_2, n_3, n_4, n_5

1.2 (b) Minimized corresponding set of test cases:

TR1 : triangleTest(-1,1,1)
TR2 : triangleTest(2,3,5)
TR3 : triangleTest(2,3,5)
TR4 : triangleTest(2,2,2)
TR5 : triangleTest(4,5,6)

1.2 (c) Because in this example, every edge connects unique nodes, so visiting all edges automatically visits all nodes and vice versa. Therefore node coverage and edge coverage require the same set of test cases.

Question 2: Logic Coverage

Answers

2.1 (a) Test Requirements for PC:

TR1:S1≤0||S2 ≤0||S3 ≤0;
TR2:!(S1≤0||S2 ≤0||S3 ≤0);
TR3:S1+S2≤S3||S1 + S3 ≤S2||S2 + S3 ≤S1;
TR4:!(S1+S2≤S3||S1 + S3 ≤S2||S2 + S3 ≤S1);
TR5:S1 == S2&&S2 == S3;
TR6:!(S1 == S2&&S2 == S3);
TR7:S1 == S2||S2 == S3||S1 == S3;
TR8:!(S1 == S2||S2 == S3||S1 == S3);

Test cases:
PC TC1:triangleTest(−1,1,3);
PC TC2:triangleTest(1,1,3);
PC TR3:triangleTest(3,1,3);
PC TR4:triangleTest(3,1,3);
PC TR5:triangleTest(4,5,3);

2.1 (b) No. The node coverage and the predicate coverage are not always the same. NC focuses on visiting nodes (structural) while PC requires testing predicates to both TRUE and FALSE (logical). In Triangle Test they're equal due to tree structure, but they differ when paths merge or when nodes can be visited without testing all predicate outcomes.

2.1 (c) Modified Condensation Graph

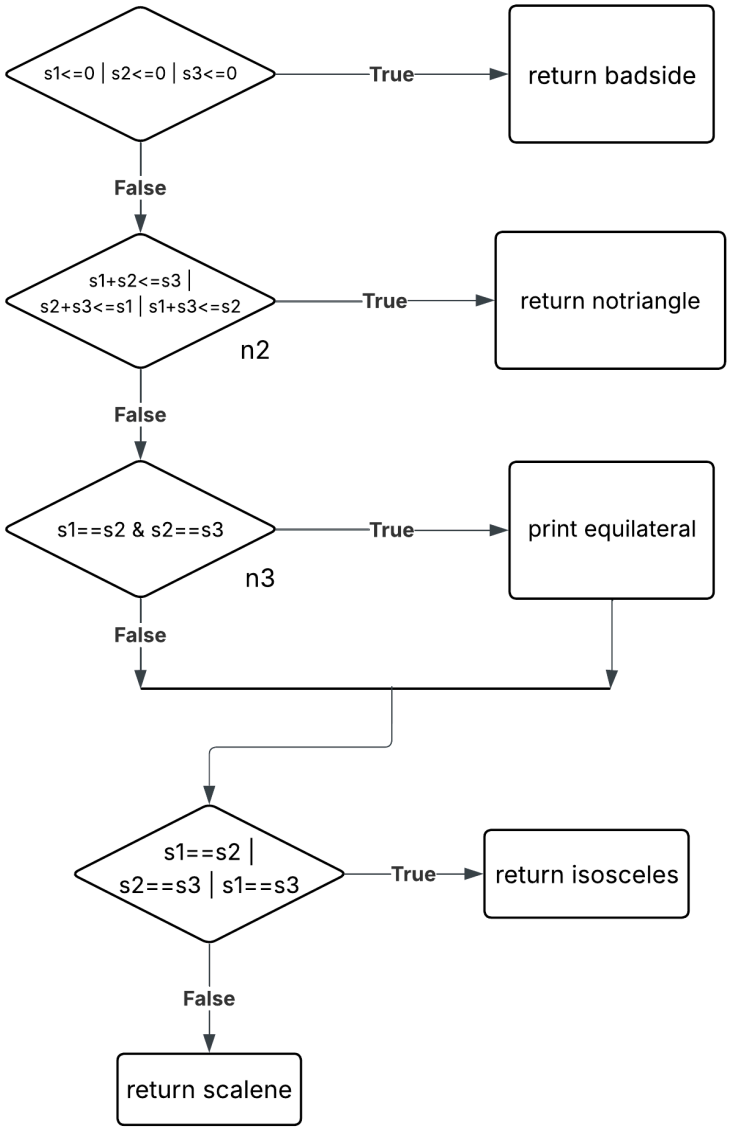

Figure - 2 : Modified Condensation graph (Q2)

For Figure No: 2, we modified the condensation graph. Now we have these test cases below :

NC TC1: s1 = -1, s2 = 1, s3 = 1
NC TC2: s1 = 1, s2 = 1, s3 = 5
NC TC3: s1 = 3, s2 = 3, s3 = 3
NC TC4: s1 = 3, s2 = 3, s3 = 4
NC TC5: s1 = 3, s2 = 4, s3 = 5

Now the modified graph demonstrates NC ≠ PC because Node Coverage requires 4 test cases {TC1, TC2, TC3, TC5} to visit all nodes while Predicate Coverage requires 5 test cases {TC1, TC2, TC3, TC4, TC5} to test all predicates in both TRUE and FALSE states, with TC4 being necessary only for PC to test p_4=TRUE via the direct path from n3(FALSE).

2.2 (a) Test Requirements for Clause Coverage (CC)

Clauses :
CC TR1: s1<=0
CC TR2: s2<=0
CC TR3: s3<=0
CC TR4: s1> 0
CC TR5: s2> 0
CC TR6: s3> 0

CC TR7: s1+s2<=s3
CC TR8: s2+s3<=s1
CC TR9: s1+s3<=s2
CC TR10: s1+s2>s3
CC TR11: s2+s3>s1
CC TR12: s1+s3>s2
CC TR13: s1==s2
CC TR14: s2==s3
CC TR15: s1!=s2
CC TR16: s2!=s3
CC TR17: s1==s3
CC TR18: s1!=s3

2.2 (b) Test Cases for CC

CC TC1: S1=0, S2=0, S3=0; (Covers TR1, TR2, TR3)

CC TC2: S1=1, S2=2, S3=4; (Covers TR4, TR5, TR6, TR7)

CC TC3: S1=4, S2=1, S3=2; (Covers TR8)

CC TC4: S1=1, S2=4, S3=2; (Covers TR9)

CC TC5: S1=3, S2=3, S3=3; (Covers TR10, TR11, TR12, TR13, TR14, TR15)

CC TC6: S1=3, S2=4, S3=5; (Covers TR16, TR17, TR18)

2.3 (a)Test Requirements for Restricted Active Clause Coverage

RACC TC1: S1=0, S2=1, S3=1 = T

RACC TC2: S1=1, S2=0, S3=1 = T

RACC TC3: S1=1, S2=1, S3=0 = T

RACC TC4: S1=1, S2=1, S3=1 = F

RACC TR5: S1+S2>S3 & S2+S3>S1 & S1+S3>S2; = F

RACC TR6: S1+S2<=S3 & S2+S3>S1 & S1+S3>S2; = T

RACC TR7: S1+S2>S3 & S2+S3<=S1 & S1+S3>S2; = T

RACC TR8: S1+S2>S3 & S2+S3>S1 & S1+S3<=S2 = T

RACC TR9: S1==S2 & S2==S3; = T

RACC TR10: S1!=S2 & S2==S3; = F

RACC TR11: S1==S2 & S2!=S3; = F

RACC TR12: S1!=S2 & S2!=S3 & S1!=S3; = F

RACC TR13: S1==S2 & S2!=S3 & S1!=S3; = T

RACC TR14: S1!=S2 & S2==S3 & S1!=S3; = T

RACC TR15: S1!=S2 & S2!=S3 & S1==S3; =T

2.3 (b) Test Cases for RACC

RACC TC1: S1=2, S2=2, S3=2; (Covers TR1, TR5, TR9)

RACC TC2: S1=-1, S2=2, S3=2; (Covers TR2)

RACC TC3: S1=2, S2=-1, S3=2; (Covers TR3)

RACC TC4: S1=2, S2=2, S3=-1; (Covers TR4)

RACC TC5: S1=2, S2=2, S3=5; (Covers TR6)

RACC TC6: S1=5, S2=2, S3=2; (Covers TR7)

RACC TC7: S1=2, S2=5, S3=2; (Covers TR8)

RACC TC8: S1=4, S2=6, S3=6; (Covers TR10, TR14)

RACC TC9: S1=5, S2=5, S3=7; (Covers TR11, TR13)

RACC TC10: S1=4, S2=6, S3=7; (Covers TR12)

RACC TC11: S1=6, S2=4, S3=6; (Covers TR15)

Question 3: Loop Program Analysis

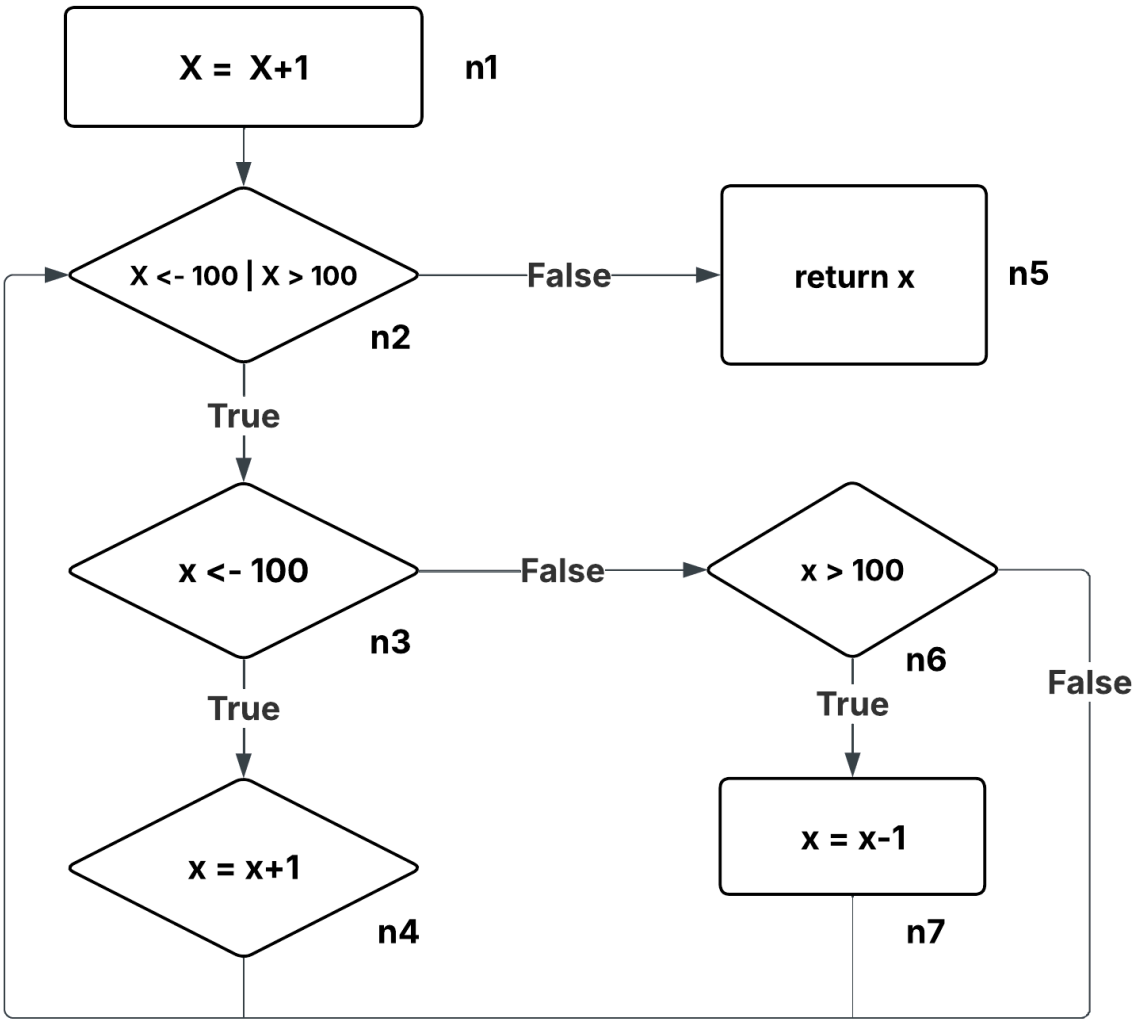

Figure - 3 : Condensation graph of (Q3)

3(a) Drawn the figure in no. 3 as per given piece of code.

3(b) The algorithm constrains the input to the range [-100, 100]. To achieve full node coverage, a minimum of **two** test cases are needed:

TR1: n1, n2, n3, n6, n7, n2, n5 (for x > 100)

TR2: n1, n2, n3, n4, n2, n5 (for x < -100)

If the input is greater than 100, the code goes through n1, n2, n3, n6, n7. At n7 it decreases x by 1 and loops back to n2. This repeats until x is no longer greater than 100, at which point the code exits through n5 and returns x. Only n4 is not covered since the input is not less than -100. Therefore we need a second test case with input less than -100 to reach n4, which goes through n1, n2, n3, n4, loops back to n2, and exits through n5.

With these two test cases we achieve 100% node coverage of all nodes n1, n2, n3, n4, n5, n6, n7.

3(c) Test Cases for Node Coverage

TC1: x=105

TC2: x=-102

3(d) Predicate coverage means each predicate should be tested true and false . Predicates for the code are :

P1: x<-100 | x>100

P2: x<-100

P3: x>100

With TC1: x=105 we will get P1(true & false), P2(false) and p3(true)

With TC2: x=-102 we will get P1(true & false), P2(true). However P3 never evaluates to false while the loop is still running — once x reaches 100 the loop exits through P1=false before P3 can be evaluated as false inside the loop.

Therefore predicate coverage cannot be fully achieved with these two test cases, meaning the test suite for predicate coverage is not better than node coverage in this case.

Question 4: Self-Assessment

1. Do you have a test case that represents a valid scalene triangle?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.
Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

2. Do you have a test case that represents a valid equilateral triangle?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.
Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

3. Do you have a test case that represents a valid isosceles triangle?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.
Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

4. Do you have at least three test cases that represent valid isosceles triangles such that you have tried all three permutations of two equal sides?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

5. Do you have a test case in which one side has a zero value?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

6. Do you have a test case in which one side has a negative value?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.

Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

7. Do you have a test case with three integers such that the sum of two is equal to the third?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

8. Do you have at least three test cases in category 7 such that you have tried all three permutations where the length of one side is equal to the sum of the lengths of the other two sides?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

9. Do you have a test case with three integers greater than zero such that the sum of two numbers is less than the third?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

10. Do you have at least three test cases in category 9 such that you have tried all three permutations ?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

11. Do you have a test case in which all sides are zero?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.

Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

12. Do you have at least one test case specifying non-integer values or does this not make sense?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

13. Do you have at least one test case specifying the wrong number of values (2 or less, four or more) or does this not make sense?
Answer : Node Coverage (NC) - No.
Edge Coverage (EC) - No.
Predicate Coverage (PC) - No.
Clause Coverage (CC) - No.
Restricted Active Clause Coverage (RACC) - No.

14. For each test case, did you specify the expected output from the program in addition to the input values?
Answer : Node Coverage (NC) - Yes.
Edge Coverage (EC) - Yes.
Predicate Coverage (PC) - Yes.
Clause Coverage (CC) - Yes.
Restricted Active Clause Coverage (RACC) - Yes.

Comparison Table:

Coverage	NC	EC	PC	CC	RACC
Points	6	**6**	**6**	**9**	**8**

Clause Coverage (CC) achieves the highest score (9/12 points) and provides the best balance between test thoroughness and efficiency. For the questions evaluating functional requirements, CC reaches the most circumstances with only 10 test cases, making it the optimal choice for this scenario. Additionally, RACC is also a good choice with 8/12 points, offering the most rigorous logic testing despite requiring more test cases.