Lab V: Logic Coverage

Software Testing and QA (COE891)

Week 10

1 Lab Objectives

- Doing test coverage logically to evaluate boolean variables, predicates, and constraints.
- Satisfying an appropriate combination of clauses in the program.

2 Clauses Determine a Predicate

Finding values for minor clauses c_j is easy for simple predicates. But how to find values for more complicated predicates? As an approach based on definition, $p_{c=true}$ is predicate p with every occurrence of c replaced by true. $p_{c=false}$ is predicate p with every occurrence of c replaced by false. To find values for the minor clauses, connect $p_{c=true}$ and $p_{c=false}$ with exclusive OR:

$$p_c = p_{c=true} \oplus p_{c=false}$$

After solving, p_c describes exactly the values needed for c to determine p.

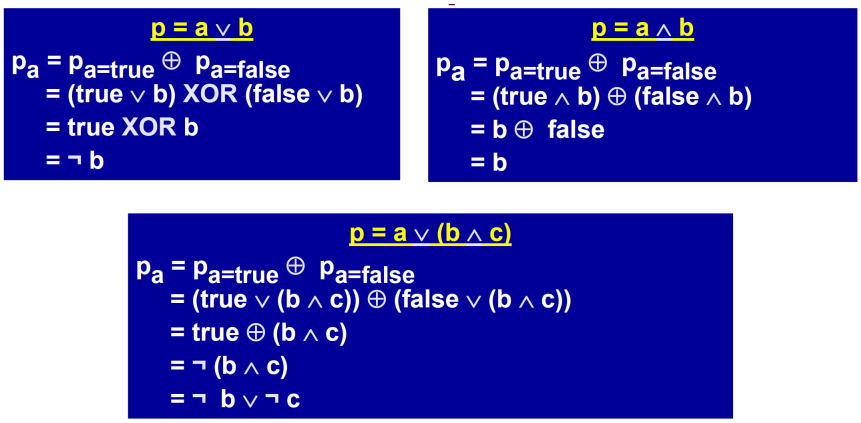

Figure 1

- $\sim b \vee \sim c$ means either b or c can be false.
- RACC requires the same choice for both values of a, CACC does not.

As another example, consider the following predicate and its computation:

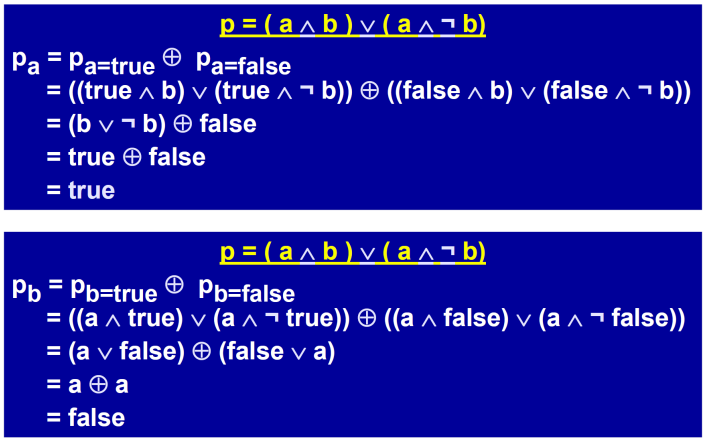

Figure 2

- a always determines the value of this predicate.
- b never determines the value, i.e., b is irrelevant.

2.1 Infeasible Test Requirements

Consider the predicate:

$$p = (a > b \ \wedge \ b > c) \ \vee \ c > a$$

$(a > b) = true$, $(b > c) = true$, $(c > a) = true$ is infeasible. As with graph-based criteria, infeasible test requirements have to be recognized and ignored. Recognizing infeasible test requirements is hard, and in general, undecidable.

3 Assignments

Note that for this lab (Lab XI), you need to exploit a logical coverage tool available at Logic Coverage Web Application. You MUST double check your own answers and/or results with the results achieved using this tool, and submit the pictures/snapshots of the results demonstrated by this tool based on each of the following tasks/questions along with your own results that manually achieved.

Q1. For the following predicate:

$$p = a \wedge (\sim b \vee c)$$

1. Provide the truth table for clauses in p including clause determination predicates.
2. Provide Conditions under which each of the clauses determines p.
3. Provide all row pairs (set of possible tests) for each major clause to satisfy each one of the following:
   - GACC
   - CACC
   - RACC
   - GICC
   - RICC

Q2. Consider the following program code:

```
int x = y;
while (x < 100) {
    if (x < y) {
        x += 1;
        break; }
    for(int z = 1; z < x; z++)
        x = x + z;
    if (x > 5)
        y += 1;
    else
        y += 2; }
System.out.println(x + ',' + y);
```

1. Draw the Control Flow Graph (CFG) of the above program code.
2. For each node n in the CFG in problem 1, please write down def(n) and use(n).
3. For each variable, please write down the DU pairs of each variables.
4. Write down the infeasible test paths based on CFG.
5. Write down a test set that satisfies all-def coverage.
6. Write down a test set that satisfies all-use coverage.
7. Write down a test set that satisfies all-du-paths coverage.

Q3. Provide the answer to the following items listed based on the code below:

```
public class TriangleType {
/** @param s1, s2, s3: sides of the putative triangle
* @return enum describing type of triangle */
public Triangle triangle (int s1, int s2, int s3) {
    // Reject non-positive sides
    if (s1 <= 0 II s2 <= 0 II s3 <= 0)
        return (Triangle.INVALID);
    // Check triangle inequality
    if (s1+s2 <= s3 II s2+s3 <= s1 II s1+s3 <= s2)
        return (Triangle.INVALID);
```

```
11      // Identify equilateral triangles
12      if ((sl == s2) && (s2 == s3))
13          return Triangle.EQUILATERAL;
14      // Identify isosceles triangles
15      if ((sl == s2) II (s2 == s3) II (si == s3))
16          return Triangle.ISOSCELES;
17      return (Triangle.SCALENE); }
18  }
19
20  public enum Triangle {
21      SCALENE, ISOSCELES, EQUILATERAL, INVALID }
```

1. Reachability predicates.
2. TRs and test cases that satisfy PC.
3. TRs and test cases that satisfy CC.
4. Determination predicates (compute and simplify).
5. TRs and test cases that satisfy CACC (or RACC).
6. Infeasible requirements.

4 Submission and Marking Instructions

4.1 Submissions

Once all required tasks are completed, you should submit your lab assignment. Follow the instructions below for a valid submission:

- After checking the accuracy and completeness of your assignment tasks, you should export and submit the FULL Eclipse project/folder as a ZIP file containing packages and source code files (for implementation and coding tasks/questions).

  - Individual files (e.g. java or class files) will NOT be accepted as a valid submission since your submitted package or Java project should be run completely and successfully by itself.

- You MUST submit a single PDF file containing required description or explanation for each of the assignment tasks/questions separately including any required justification, graphs, diagrams, test requirements/cases, calculation, pictures/snapshots from tools or IDE, test results, and so forth.

- The **submission deadline** for this lab (Lab V) is the corresponding lab session in **Week 11**.

- The lab demo and questioning-answering will be held during the lab sessions of the corresponding submission weeks.

4.2 Marking Scheme

- This lab (Lab V) constitutes 4% of your entire grade for this course.

- All assignment tasks/questions in each lab have the same grade.

- The grade for each lab is constituted from 50% for the lab submissions, 10% for the lab attendance, and 40% for demo and questioning-answering during the lab session.

- Note that all the labs constitute 25% of your entire grade for this course.