Software Testing, Quality Assurance & Maintenance (ECE453/CS447/SE465): Assignment 3 v1

Patrick Lam

Due: March 3, 2010

You may discuss the assignment with others, but I expect each of you to do the assignment independently. I will follow UW's Policy 71 if I discover any cases of plagiarism.

Question 1 (10 points)

(based on Section 3.2, Q1) For the following predicates:

1. $p = a \vee (b \rightarrow c)$
2. $p = a \leftrightarrow (b \vee c)$

(a, 0 points) Identify the clauses that go with each p.

(b, 2 points) Compute (and simplify) the conditions under which each of the clauses determines predicate p.

(c, 2 points) Write the complete truth table for all clauses. Label your rows. Use the format from the table underneath combinatorial coverage from Section 3.2. Include columns for the truth value of the predicate and for the conditions under which each clause determines the predicate.

(d, 2 points) Identify all pairs of rows from your table that satisfy general active clause coverage (GACC) with respect to each clause.

(e, 2 points) Identify all pairs of rows from your table that satisfy correlated active clause coverage (CACC) with respect to each clause.

(f, 2 points) Identify all pairs of rows from your table that satisfy restricted active clause coverage (RACC) with respect to each clause.

Question 2 (10 points)

Common programming languages (like C and Java) include short-circuit operators; that is, in the expression `a && b`, we only evaluate clause `b` if `a` is true, and in the expression `a || b`, we only evaluate clause `b` if `a` is false.

In this question, you will explore the impact of short-circuit operators on logic coverage criteria. (a, 2 points) What is the effect of short-circuit evaluation on active clause coverage? (b, 6 points) For each logic coverage criterion we've seen, discuss whether its test requirements make sense in the presence of short-circuit criteria—that is, do any test requirements become infeasible or illogical? (c, 2 points) Show an example of a test case for Clause Coverage which makes no sense in the context of short-circuit evaluation.

Question 3 (10 points)

Consider predicates of the form,

$$p = c_1 \wedge p';$$

where c_1 is a clause and p' is another predicate. Prove that any test suite T which satisfies GACC with respect to c_1 also satisfies CACC w.r.t. c_1.

Question 4 (20 points)

As I mentioned in class, one of my goals is to develop the ability to read and modify realistically-sized software systems. I realize that these systems require quite a bit of fiddling around to get working, but that's always the case with large software systems, and the amount of fiddling decreases with experience (which is the point of the exercise!)

This time, the software system will be Ehcache, whose homepage is at `http://ehcache.org`. This 4.5-year old webpage may help, but it may be out of date:

`http://twasink.net/blog/2005/10/ehcache-dissected/`

Your main task will be to write a simple test suite for Ehcache which satisfies certain graph coverage criteria.

Bonus (10 points). Read Ehcache's features document at

`http://ehcache.org/features.html`

and suggest a feasible strategy for improving Ehcache's test suite. ("Include more tests" and "achieve 100% coverage" are not the answer here; actually, I can't think of a good way to do this.)

Main Questions. (2 points) What is the most likely source of problems that you may encounter when using Ehcache? (Your answer should be a couple of lines long). (8 points) Create a Finite State Machine which summarizes the lifecycle of a cache and one key x in the cache; document your assumptions about the environment. (That is, you do not need to consider the behaviour of the cache with keys other than x.) (10 points) Pick a suitable graph coverage criterion for testing the FSM and create JUnit tests that exercise this criterion on Ehcache.