Department of Computer Science
Purdue University, West Lafayette
CS 49000-ST0 Software Testing
Final Examination: Part A
Closed book/notes.

Q 1 A program that is known to contain errors might have a reliability of 1 when estimated by using an operational profile for testing.

true

Q 2 For a given input domain, equivalence partitioning and boundary value analysis will always lead to the same set of tests.

false

Q 3 Invalid values must be provided for each variable before AETG can generate tests.

false

Q 4 If the number of input variables is 2, then AETG can generate only pairwise tests.

true

Q 5 Program P has been found to be robust. This implies that P

(a) →gracefully handles all unexpected inputs.

(b) is functionally correct.

(c) has been tested adequately with respect to control flow coverage.

(d) does not contain any off-by-1 errors.

Q 6 A program dependence graph captures

(a) Only the flow of control across the basic blocks

(b) →Data and control dependence

(c) Only data dependence

(d) Only control dependence

Q 7 Consider the following code fragment.

```
while (true){
if (a <b) {
     p=p+(x+y)*a; a=a-1;
     }
else {
     p=p+(x-y)*b; b=b+1;
     }
}
```

The number of basic blocks in the above program is

(a) →3 [If a smart static analyzer is used.]

(b) 2

(c) →4

(d) 5

Q 8 Let $C = c_1 \&\& c2$. The number of tests generated by applying BOR method on C is:

(a) 4

(b) →3

(c) 2

(d) 1

Q 9 Which one of the following faults is guaranteed to be detectable by tests generated using the BOR method?

(a) Off-by-1

(b) Off-by ϵ for an arbitrary ϵ

(c) Incorrect relational operator

(d) →Incorrect Boolean operator

Q 10 Consider a mixed-level orthogonal array A=MA$(18, 3^6, 6^1, 2)$. Which one of the following statements is true?

(a) A contains 18 runs, for 9 factors, and has a strength of 2.

(b) A contains 2 runs, for 7 factors, and has a strength of 18.

(c) A contains 2 runs, for 7 factors, and has a strength of 18.

(d) →A contains 18 runs, for 7 factors, and has a strength of 2.

Q 11 Which one of the following statements is true?

(a) For a given set of factors and constraints, a covering array is always smaller than an orthogonal array.

(b) For a given set of factors and constraints, a covering array satisfies the balance requirement and is often smaller than an orthogonal array.

(c) →For a given set of factors and constraints, a covering array is often smaller than an orthogonal array but need not satisfy the balance requirement.

(d) Euler conjectured that MOLS of order 4n+2 do not exist and later researchers proved that indeed this conjecture is true.

Q 12 We are given three factors X, Y, and Z each at, respectively, 3, 4, and 5 levels. The minimum number of tests required to cover all triples is

(a) 12

(b) 20

(c) 15

(d) →60

Q 13 An FSM M is considered *minimal* if

(a) for each state it has transitions corresponding to each member of its input alphabet.

(b) each state in M can be reached from the start state.

(c) →the number of states in M is less than or equal to the number of states in any other machine equivalent to M.

(d) each state in M can be reached from the start state and the start state can be reached from each of the remaining states.

In the next two questions assume that the FSM M is minimal and complete.

Q 14 Let P denote the transition cover set generated from M. Suppose that we use P as the test set (and ignore the W set). The elements of P

(a) cover all the states in M but may not cover all the transitions in M.

(b) →cover all the states and all the transitions in M.

(c) may not cover all the states in M but cover all the transitions in M.

(d) may not cover all the states or transitions in M.

Q 15 Let M be a 1-distinguishable FSM and T a 1-switch cover. Let F denote the set of all possible transfer, operation, missing, and extra state errors in M. Which one of the following statements is correct?

(a) →T will detect all faults of type F.

(b) T will detect all faults of type F except for the missing and extra state errors.

(c) T will detect all faults of type F except for the missing state errors.

(d) T is not guaranteed to detect any fault of type F.

Q 16 Program **P** contains no loops and three `if` statements. Let the triple (**b, c, p**) denote the least number of tests required to cover, respectively, all blocks, all atomic conditions, and all paths in **P**. Which of the following triples contains the correct values of **b**, **c**, and **p**? Assume that there are no infeasible paths.

(a) (1, 1, 3)

(b) (1, 2^3, 1)

(c) (1, 3, 2^3)

(d) →(1, 1 , 4) [This will happen when the if's are nested.]

<End of Part A of Final Exam Solutions CS 49000, Part A. Spring 2011.>

Department of Computer Science
Purdue University, West Lafayette
CS 49000-ST0 Software Testing
Final Examination: Part B
You may consult the textbook and your *hand written* notes.

Q 1 Consider an application that is to be executed in an environment modeled with four factors P, Q, R, and S. Each factor can be set to different levels as follows:

P={p1, p2}; Q={q1, q2}, R={r1, r2, r3}, and S={s1, s2, s3}.

Assume that combination (p1, r3) is infeasible. Use the IPO algorithm to construct a covering array that covers all feasible pairs among the four factors. Underline entries that have been added in the *horizontal growth* mode and, using an arrow (→), point to rows that have been added in the *vertical growth* mode. [Hint: The IPO algorithm as given in the text book does not account for constraints. Thus, it may be best for you to apply IPO first and then adjust the entries generated so that the constraint above is satisfied.]

5 points

	Run	**P**	**Q**	**R**	**S**
	1	p1	q1	r1	s1
	2	p2	q2	r1	s2
	3	p1	q2	r1	s3
	4	p2	q1	r2	s1
→	5	p1	q1	r2	s2
→	6	p2	q2	r2	s3
→	7	p2	q2	r3	s1
→	8	*	q1	r3	s2
→	9	*	q1	r3	s3

p1 cannot replace * in runs 8 and 9. All entries under columns labelled R and S in rows 1 through 4 are the consequence of horizontal growth.

Q 2 Method `main` given below is intended to compute and display the factorial of a positive non-zero integer input by the user. The method terminates when the user types "quit" or an end-of-file character ("control D" or "∧D") . However, the placement of the initialization of `fact` is erroneous and causes the program to fail on certain inputs. Answer the questions below with respect to `main`.

(a) On page 2 draw a data flow graph for `main`. Label the nodes as 1, 2, 3, and so on. In the table on page 3 fill the line numbers, definitions, c-uses, and p-uses that correspond to each node.

6 points

(b) Develop an MC/DC adequate test set for `main` that *does not reveal* the error.

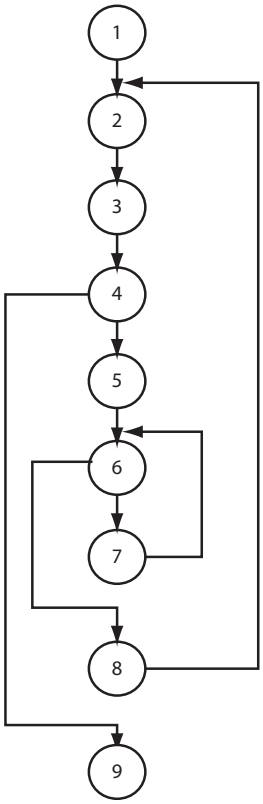
2 points

(c) In `main`, does there exist a def-use pair (c-use or p-use) that when covered is *guaranteed* to reveal the error? If yes, then construct a test case that covers such a def-use and show how it reveals the error. If not, then construct a test set that covers all def-uses but does not reveal the error (i.e., does not cause the program to fail).

3 points

```
1     public static void main(String [] args) throws Exception{
2         BufferedReader in = new
3                     BufferedReader(new InputStreamReader(System.in));
4         int fact=1;
5         for( ; ;) {
6             System.out.print("Factorial finder ");
7             String line = in.readLine();
8             if ((line == null) || line.equals("quit"))
9                break;
10             int x = Integer.parseInt(line);
11             int  count=1;
12             while (count<=x){
13                    fact=fact*count; count++;
14              }
15             System.out.println(x + "! = " + fact);
16            }
17        }
```

[(a) Control-flow graph portion of the data flow graph.]



Node	Lines	def	c-use	p-use
1	1, 2, 3, 4	in, fact, args		
2	5	--	--	--
3	6, 7, 8	line	in	line, line
4	9	--	--	--
5	10, 11	x, count	line	--
6	12	--	--	count, x
7	13, 14	fact, count	fact, count	--
8	15, 16	--	x, fact	--
9	17	--	--	--

[Fill the following table with the def-use information. Add additional rows in the table if you have more than 9 nodes in your flow graph, or leave the rows blank if you have fewer than 9 nodes. The column labeled "Lines" must list line numbers that correspond to `main`.]

See the table next to the graph.

(b) MC/DC adequate test set for the condition in the **if** statement:

Let C_1: line==null, C_2: line=="quit", and $C : C_1||C_2$. Thus the MC/DC adequate test set must demonstrate the the impact of C_1 and C_2 on C. We assume that short-circuit evaluation of compound conditions has been turned off.

t1: <line=null>

t2: <line="quit">

t3: <line=1, line="quit">

Test	C_1	C_2	C	**Comment**
t1	true	false	true	t1 and t3 demonstrate the effect of C_1 on C.
t2	false	true	true	t2 and t3 demonstrate the effect of C_2 on C.
t3	false	false	false	The second part of this test, line="quit", is redundant for MC/DC adequacy. This test also covers the decision controlling the `while` loop.

(c) def-use adequate test set or a test case that causes `main` to fail:

There is no def-use pair coverage of which *guarantees* the detection of the error, i.e., causes `main` to fail. Hence, we provide a def-use adequate test set T that does not cause `main` to fail.

T={t1, t2};

t1:<line=0, line=null> This covers the def of `fact` at line 4 and its use at line 15, and the definition of `x` at line 10 and its use at line 15.

t2:<line=2, line="quit"> This covers the remaining def-use pairs.

<End of Final Exam Solutions CS 49000, Part B Spring 2011.>