ECE 453/CS 447 Final Exam, W'2004

Electrical and Computer Engineering

ECE 453/CS 447 – Software Testing, Maintenance and Quality Assurance

Instructor: Sagar Naik

April 21, 2004, 2 p.m— 5 p.m

Notes

- You have **three hours** to complete the exam.
- This is a **closed book** exam.
- Continue your answers on the **back** of the facing **page**, if necessary.
- **Answer all questions**. **Hand in all pages**.
- **Questions will not be interpreted in the 3rd hour of the exam.**

Student ID **Student Name** **Student Signature**

Q1	**/ 15**	**Q7**	**/ 10**
Q2	**/ 20**	**Q8**	**/ 20**
Q3	**/ 20**	**Q9**	**/ 10**
Q4	**/ 15**	**Q10**	**/ 10**
Q5	**/ 15**	**Q11**	**/ 20**
Q6	**/ 25**		

Your Total Score = / 180

Q1. [Goodenough and Gerhart's Theory][15 points]

i. [4] Define the following terms: SUCCESSFUL(T), COMPLETE(T, C).

ii. [4] Define the desirable properties of a test selection criterion as proposed by Goodenough and Gerhart.

iii. [7] Clearly explain the difficulties in applying Goodenough and Gerhart's theory of program testing?

Q2. [Domain Testing] [20 points]

i. [5] What are program domains and why do domain errors occur?

ii. [2] State four kinds of domain errors.

iii. [3] Clearly state the test selection criterion in domain based testing.

iv. [10] Show that test cases, generated by using the test selection criterion you have answered in part (iii), can reveal *tilted boundary* errors.

Q3. [CFG/DFG Testing][20 points]

i. [3] Clearly explain the main motivation behind data-flow based testing.

ii. [2] How does data-flow based testing differ from control-flow based testing?

iii. [2] What is a similarity between data-flow based testing and control-flow based testing?

iv. [3] Clearly explain the **all-p-uses** test selection criterion.

v. [10] Suppose that you are going to design a tool to generate test cases using the idea of data-flow based testing. Show the components of such a tool and clearly explain their functions. Finally, explain two limitations of such a tool.

Q4. [Specification-based Testing][15 points]

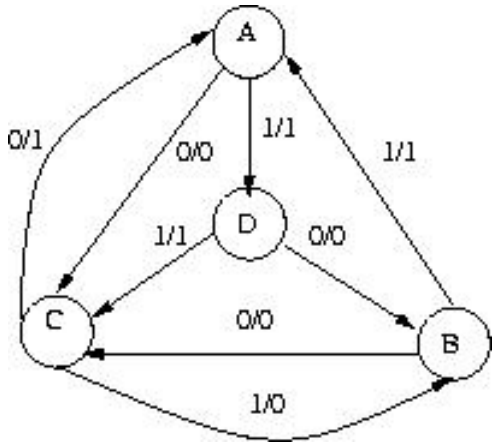
Consider the FSM shown on the following page. This FSM has been completely specified and is deterministic, reduced, and strongly connected. Assume that an implementation has correctly implemented a reset sequence denoted by *ri*. The initial state of the FSM is A.

i. [8] Compute one unique input/output (UIO) sequence for each state, if it exists.

ii. [2] Construct a test sequence to test the state-transition from state D to B. Clearly show the different parts of the test sequence.

iii. [2] What is a potential pitfall of using a UIO sequence as a state-verification sequence in testing a faulty implementation?

iv. [3] What is the main drawback of not implementing a reset sequence?


A
D
C
B
0/1
0/0
1/1
1/1
1/1
0/0
0/0
1/0

Q5. [Test Oracle] [15 points]

i. [3] What is a *test oracle*?

ii. [3] What is the *oracle assumption*?

iii. [3] What is the *partia*l *oracle assumption*?

iv. [3] Clearly explain what will happen if a test engineer does not make at least one of the above two assumptions?

v. [3] Assume that you are working for a company that develops software systems for in-house use in its design of aircraft engines. Being a software engineer you may not have any expertise in aircraft engines. While testing programs for such applications, what are the challenges you are likely to face, and how will you overcome them?

Q6. [Test Data Adequacy][25 points]

i. [7] Clearly explain the concept of *test data adequacy*.

ii. [8] Clearly explain two ways of evaluating test data adequacy.

iii. [6] Sometimes managers prefer to stop system test when they run out of time or allocated budget with no considerations for “technical details.” Discuss three criteria for stopping system test based on what you have learned in this course.

iv. [4] You know that system test is carried out in the real environment of a software system. Give two examples where system test in the real environment may not be possible. How will you carry out meaningful system test in such cases?

Q7. [Training test personnel][10 points]

Experience shows that it is extremely difficult to find qualified people, such as those with software engineering degrees, for the jobs of test engineers. Sometimes companies have no option but to recruit people with no formal education in software testing. They also recruit people from outside the computer science and engineering disciplines for the task of software testing. Your employer wants you to train such people for the following kinds of projects. Briefly explain how you will devise a training course for those test personnel.

i. [5] Mobile phone system

ii. [5] An information system with a large GUI (graphical user interface) component.

Q8. [Software Reliability][20 points]

i. [5] Define **software reliability** in two ways.

ii. [4] Which of the above definitions will you apply to express reliability of the following systems?

a. A library information system

b. Deep-sea exploration by a robot

c. Control of a hydro-electric power plant

d. Control of a space flight to Mars

iii. [6] Clearly explain three applications of modeling software reliability.

iv. [5] At the beginning of system test assume that the software under development will encounter 100 failures in infinite time. The failure intensity at the start of system test is 5 failures/CPU-hour. As failures are observed, faults are immediately fixed. After several days of testing, you notice that the failure intensity has dropped down to 0.5 failures/CPU-hour. Then, one of your test engineers detects five failures and the development team takes one week to fix the corresponding faults. Calculate the failure intensity at the end of the "one week" mentioned above.

Q9. [Software Quality] [10 points]

i. [5] Explain the *product view* of software quality using two concrete examples.

ii. [5] Explain the *process view* of software quality.

Q10. [Cyclomatic complexity][10 points]

i. [3] Clearly explain the Cyclomatic complexity measure.

ii. [4] Calculate the Cyclomatic complexity of the *Butterfly()* function given on the following page.

iii. [3] Clearly explain two ways to reduce the Cyclomatic complexity of a function.

```
static int Butterfly(int Z, int pos1, int pos2, int control)
    { // if control == 0, return Z. Else return Z after swapping the
      //  bits in positions pos1 and pos2.
      int dest;
      int maskpos1 = (int)Math.pow(2, pos1);
      int maskpos2 = (int)Math.pow(2, pos2);
      int zpos1 = Z & maskpos1;
      int zpos2 = Z & maskpos2;
      int valpos1 = zpos1*maskpos1;
      int valpos2 = zpos2*maskpos2;
      dest = Z;
      if (control != 0){
                  if (zpos1 == 0 && zpos2 != 0)
                        dest = dest + maskpos1 - maskpos2;
                  else if (zpos1 != 0 && zpos2 == 0)
                     dest = dest - maskpos1 + maskpos2;
      }
      return(dest);
}
```

Q11. [Fundamentals][20 points]

ii. [2] Explain the idea of a stub used in program testing?

iii. [2] What are the costs involved in unit testing?

iv. [2] State two factors which determine the effectiveness of unit testing.

v. [2] What is the basis for acceptance test?

vi. [2] Explain the idea of test case dependency?

vii. [2] What is a statement of compliance?

viii. [2] State five different kinds of tests that you need to perform at the system level.

ix. [2] State the three types of maintenance activities.

x. [2] How do you compare two test methods?

xi. [2] What is program slicing?

***** The End *****