NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR
Semester 2 AY2010/2011

CS4271 – CRITICAL SYSTEMS AND THEIR VERIFICATION

Apr/May 2011 **Time Allowed: 2 Hours**

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FOUR (4)** questions and comprises **NINE (9)** printed pages, including this page.
2. Answer **ALL** questions within the space in this booklet
3. This is an Open Book examination.
4. Please write your Matriculation Number below.

MATRICULATION NO: ___________________________________

This portion is for examiner's use only

Question	Marks	Remarks
Q1	15	
Q2	15	
Q3	9	
Q4	11	
Total	50	

Question 1 [6 + 4 + 5 = 15 marks]

A. Consider the following program. Draw its control flow graph. How many paths are there? How many of these paths are feasible and how many are infeasible? Give detailed justifications for your answer.
B. Use the timing schema based Worst-case Execution Time (WCET) calculation method which does not use any infeasible path information. Assume that each basic statement takes 1 time unit.
C. Use the Integer Linear Programming based WCET calculation method which takes into account infeasible path information. Encode as many infeasible path constraints as possible so that the estimated WCET is as tight as possible. How much does the WCET estimate change due to infeasible path information?

```
int P1( int flag,x,y,z ) {
        if( x > 3 ) z = z + 1; else x = flag;
        if( y == 4 ) y = y + 1; else y= y – 1;
        if( x < 2 ) z = z / 2; else z = z – 1;
        return x+y+z;
}
```

Question 2 [6 + 3 + 6 = 15 marks]

Consider an arbitration protocol where multiple hungry processes request for access to one of a pool of resources via an arbiter. Thus, the hungry processes request the arbiter for access to the resource pool. Once they get access they access a specific resource in the pool. Modeling such a system, we should have three classes of objects – *Hungry*, *Resource* and *Arbiter*.

A. Elaborate the design, by giving the UML State Diagrams of each class. Your State Diagram design must satisfy the following criteria --- (a) at most one hungry process must access the resource pool at any time, (b) if there are one or more hungry processes requesting access to the resource pool, the arbiter should have allowed at least one of them, (c) any hungry process requesting access to resource pool should eventually get access.

B. Clearly state what parts of your State Diagram are ensuring each of these three properties. If you make any assumptions for ensuring these properties, you should clearly state all your assumptions.

C. State the three informal properties in part A formally in Linear Time Temporal Logic (LTL). You should clearly state the meaning of any atomic proposition you use in the LTL properties, and the true/false valuations of these atomic propositions in the states of your State Diagram.

Question 3 [3 + (3 + 3) = 9 marks]

A. Suppose several periodic processes are running on a processor (with a cache) which employs a pre-emptive scheduling policy. Thus, the processes share the processor cache. Now, when a process is pre-empted by another process, the pre-empting process pollutes the processor cache which can affect the execution time of the pre-empted process once it resumes execution. In class, when we discussed pre-emptive scheduling policies like RMS, EDF we always assumed a zero context switch overhead. However, in reality, it is not so, and the cache is one reason contributing to a non-zero context switch overhead. Devise an analysis method to estimate the number of cache misses which may be introduced by a single pre-emption. You may want to style your analysis on the cache modeling for WCET analysis we discussed in class.

B. Is the following task set schedulable under the RMS and EDF scheduling policies respectively?

T1 = (1, 4, 4)   T2 = (3, 10, 10)   T3 = (2, 5, 5)

Each task is as a triple (WCET, period, deadline) where deadline = period.

Question 4 [(2 + 4) + 5 = 11 marks]

A. In class we discussed the usage of symbolic execution for test generation. As discussed, we can use symbolic execution to compute a *path condition* – a logical formula which is true for all inputs following a given path. Consider the following program which computes the greatest common divisor of two numbers x and y.

```
scanf("%d", &x); scanf("%d", &y);
while (x != y){ if (x >y) { x = x-y;} else {y = y - x;} }
printf("%d\n", x);
```

What is the path traced by the input (x == 8, y == 6)?
Compute the ***path condition*** of this path.

B. Consider a function *triangle* which takes three integers x,y,z which are the lengths of triangle sides and then calculates whether the triangle is equilateral (that is x == y == z), isosceles (exactly two sides are equal), or scalene (otherwise). Suppose your professor has generated the test-suite for such a function by some means. How would you evaluate whether the professor's test-suite is good enough? You can start by asking some questions to the professor like the following.

- *Do you have a test case for an equilateral triangle?*
- *Do you have a test case for an isosceles triangle?*
- *Do you have at least three test cases for isosceles triangles, where all permutations are considered? (e.g. (x,x,y), (x,y,x), (y,x,x))*
- *Do you have a test case with one side zero?*

Now, I could go on and on – but I need your help here ☺ Suggest some more questions that we should ask to evaluate whether the test-suite generated by the professor is good enough. The more questions you can ask which point to possible corner cases – the more marks you will get. **If you ask ten more questions (over and above the four questions given above), you will get full marks. Good Luck ☺**

END of PAPER