NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR
Semester 2: AY2011/2012

CS4231 PARALLEL AND DISTRIBUTED ALGORITHMS

Apr/May 2012 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FIVE(5)** questions and comprises **SEVEN(7)** 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 NUMBER: ______________________________

EXAMINER'S USE ONLY		
Question	Marks	Score
Q1	9	
Q2	4	
Q3	5	
Q4	9	
Q5	13	
Total	40	

1. Recall (from Tutorial 1) the swap instruction that is described by

```
swap(a_ptr, b_ptr){        // b_ptr is pointer to b
    tmp ← *a_ptr;          // *a_ptr is a
  *a_ptr ← *b_ptr;
  *b_ptr ← tmp;
}
```

Like TestAndSet, swap is implemented as one atomic hardware instruction.

Show that two processes can use swap to reach consensus (i.e. satisfying Agreement, Validity and Wait-Freedom). [9 marks]

2. Recall the following parallel mutual exclusion algorithm that uses a shared variable turn:

P_0	P_1

RequestCS(0){
  turn $\leftarrow$ 1;
  while(turn==1){};
}
	RequestCS(1){
	  turn $\leftarrow$ 0;
	  while(turn==0){};
	}

One can consider the 2 processes as using the algorithm to reach consensus on who should enter critical section first. However, the consensus number for a register is 1, so a register cannot be used to reach consensus for 2 processes.

Is there a contradiction? Justify your answer. [4 marks]

3. Consider the proof of the Fischer-Lynch-Paterson Theorem. Let S be a global state, P and Q be two processes, e_P an event at P and e_Q an event at Q.

   Explain why $e_P(e_Q(S)) = e_Q(e_P(S))$ if e_P and e_Q are valid message (send or receive) events for S. [5 marks]

4. In **Byzantine Agreement**, each process starts with a binary input value, and they communicate through messages to agree on the output value, so that

   (BA Termination) all nonfaulty processes decide eventually;

   (BA Agreement) all nonfaulty processes decide the same value;

   (BA Validity) if all nonfaulty processes have the same input value v, then all nonfaulty processes decide v.

   Now, change the above to the **Byzantine General** problem: There is a distinguished process that is called the **general**, and the other processes are called **lieutenants**. The general starts with a binary input value v_G. The general and lieutenants communicate through messages to agree on an output value, so that

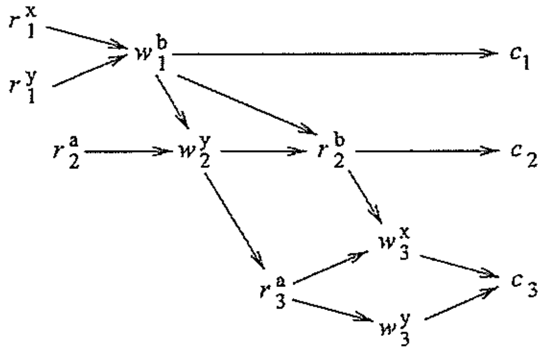
   (BG Termination) all nonfaulty processes decide eventually;

   (BG Agreement) all nonfaulty processes decide the same value;

   (BG Validity) if the general is nonfaulty, then all nonfaulty processes decide v_G.

   Show that any protocol $\mathcal{P}_{\mathrm{BA}}$ for Byzantine Agreement (e.g. $\mathcal{P}_{\mathrm{BA}}$ is the Phase King Protocol) can be used to solve the Byzantine General Problem. [9 marks]

5. Consider the following history H over transactions $\xrightarrow{T_1}$, $\xrightarrow{T_2}$ and $\xrightarrow{T_3}$, where operation o_k belongs to T_k (for clarity, transitive edges are omitted).



(i) Insert lock and unlock operations into H to give a locked history H^L that is two-phase locked. [5 marks]

(ii) Draw the serialization graph $SG(H)$. [3 marks]

(iii) Is H serializable? If so, show a serial history S that is equivalent to H; if not, explain why. [5 marks]

/ End of Paper