

The University of New South Wales

# **Final Exam**

**Session 2, 2017**

**COMP3151/COMP9151**

## **Foundations of Concurrency**

Time allowed: **2 hours + 10 minutes reading time**

Total number of questions: **5**

Total number of marks: **50**

Textbooks, lecture notes, etc. are not permitted, except for 2 double-sided A4 sheets of hand-written notes.

Calculators may not be used.

Not all questions are worth equal marks.

Answer all questions.

Answers must be written in ink.

You can answer the questions in any order.

You may take this question paper out of the exam.

Write your answers into the answer booklet provided. Use a pencil or the back of the booklet for rough work. Your rough work will not be marked.

Number of pages in this exam paper: 3

## Shared-Variable Concurrency (20 Marks)

### Question 1 (12 marks)

Consider a C program in which  $x$  is a shared variable declared and initialised by `volatile int x = 0;` and  $s$  points to a shared binary semaphore initialised to 1. The program consists of three threads. The bodies of these threads are given as follows.

**Thread  $p$ :** `sem_wait(s); x = x + 1; sem_post(s);`

**Thread  $q$ :** `sem_wait(s); x = x + 2; sem_post(s);`

**Thread  $r$ :** `sem_wait(s); x = x + 1; sem_post(s);`

- (2 marks:) What are the possible final values of  $x$  and the semaphore?
- (4 marks:) Faithfully model this situation as a parallel composition of three transition diagrams, one for each of the threads.
- (6 marks:) Prove your answer to (a) using the Owicki/Gries method on your answer to (b).

### Question 2 (8 marks)

*The One-Lane Bridge.* Cars coming from the North and the South arrive at a one-lane bridge. Cars heading in the same direction can cross the bridge concurrently, but cars heading in opposite directions cannot. We model cars as processes

Car ( $N$ processes)	
	bit dir
loop forever	
p1:	B.enter(dir)
p2:	B.exit(dir)
p3:	dir $\leftarrow$ 1 - dir

which synchronise via a monitor  $B$  offering the two procedures `enter` and `exit`. Each time a car has crossed the bridge it turns around and attempts to cross in the other direction. Write the bridge monitor  $B$  that satisfies the specification. Do not worry about fairness but ensure that the bridge does not remain empty when there are cars waiting to cross. Explain how your monitor models the bridge faithfully.

## Message-Passing Concurrency (30 Marks)

Answers to questions in this section that require programming can be formulated using either Ben-Ari's pseudo-code notation or Promela. Shared mutable state, semaphores, monitors, locks etc. are not allowed.

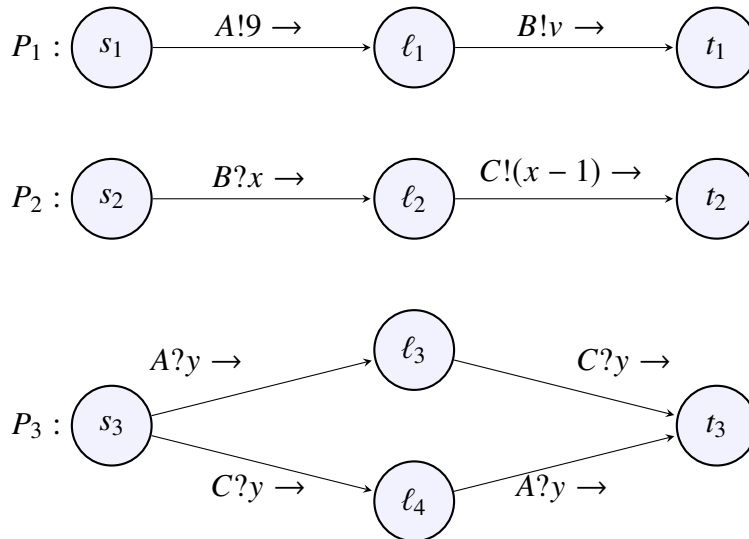
**Question 3 (10 marks)**

Consider  $n$  bees and a single bear. They share an initially empty pot that can hold up to  $k > 0$  units of honey. The bear naps until the pot is full, then consumes all honey in the pot and returns to napping under a tree. Each bee repeatedly gathers one unit of honey and puts it in the pot; the bee who fills the pot awakens the bear. The pot is inaccessible for bees if it is full or if the bear is in the process of eating honey.

Write a program to simulate the above with  $n + 2$  processes: one for the bear, one for the pot, and  $n$  for the bees.

**Question 4 (10 marks)**

Prove that  $\{true\} P_1 \parallel P_2 \parallel P_3 \{y = v - 1\}$  holds for the synchronous transition diagram:



**Question 5 (10 marks)**

Develop a program whose output is the sequence of all multiples of 2, 3, and 5 in ascending order. The first elements of the sequence are 0, 2, 3, 4, 5, 6, 8, 9, 10, 12, 14. Your program consists of four concurrent processes: one each to calculate the multiples of the numbers 2, 3, and 5, respectively, and a fourth process to merge the results and send them on output channel  $O$ .

— END OF EXAM PAPER —