The University of New South Wales

Final Exam

2010/11/02

COMP3151/COMP9151

Foundations of Concurrency

Time allowed: **2 hours (within 17:45–20:00)**
Total number of questions: **4**
Total number of marks: **45**

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

Calculators may not be used. (Not that they would be of any help.)

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.

Shared-Variable Concurrency (15 Marks)

Question 1 (8 marks)

Let A and B be two algorithms which were designed to solve the mutual exclusion problem, and let C be the algorithm obtained by replacing the critical section of A with the algorithm B:

```
                 Algorithm: $C$ ($n$ processes)
                      shared vars of $A$
                      shared vars of $B$
    loop forever
p1:     non-critical section
p2:     entry protocol of $A$
p3:     entry protocol of $B$
p4:     critical section
p5:     exit protocol of $B$
p6:     exit protocol of $A$
```

Assume that the shared variables of A are disjoint from those of B. Are the following statements correct? Justify each answer briefly (i.e., with a sentence or two).

(a) If either A or B satisfies mutual exclusion then C satisfies mutual exclusion.

(b) If A has no unnecessary delay and B satisfies mutual exclusion then C has no unnecessary delay.

(c) If A satisfies mutual exclusion and B has no unnecessary delay then C has no unnecessary delay.

(d) If A guarantees eventual entry and B is deadlock-free then C guarantees eventual entry.

Question 2 (7 marks)

Suppose president Alice and bodyguard Bob both consume milk. They are both capable of buying milk at the corner store. To ensure a regular supply they are looking for a protocol to ensure that:

(a) Only one person buys milk, when there is no milk.

(b) Someone buys milk, when there is no milk.

They look into protocols involving post-it notes stuck to the fridge door. There's an implicit loop forever around each process's code.

```
              Algorithm: Alice's suggestion
              bool noteA ← 0, noteB ← 0
          Alice                          Bob
p1: noteA ← 1                  q1: noteB ← 1
p2: if ¬noteB                  q2: await ¬noteA
p3:     if ¬milk               q3:     if ¬milk
p4:         buy milk           q4:         buy milk
p5: noteA ← 0                  q5: noteB ← 0
```

```
                         Algorithm: Bob's suggestion
                              bool note ← 0
                Alice                          |                Bob
p1: if ¬note                                   | q1: if ¬note
p2:     if ¬milk                               | q2:     if ¬milk
p3:         note ← 1                           | q3:         note ← 1
p4:         buy milk                           | q4:         buy milk
p5:         note ← 0                           | q5:         note ← 0
```

Are these suggestions correct solutions to the problem? Prove your answers.

Message-Passing Concurrency (30 Marks)

Question 3 (15 marks)

President Alice and bodyguard Bob offer different services at a fundraiser for the CPM[1]. Alice can mix mean cocktails and play Checkers whereas Bob can fix neck pain and play Checkers. Concerned citizens appear at the fund raiser event and non-deterministically decide to request either one of the three services offered in return for their generous donations to the CPM. Note that those who desire a mean cocktail must be served by Alice, those who have neck pain can only be helped by Bob, but those who would like to play Checkers are ok with either one. A pair of trustworthy party secretaries is in charge of dealing with the requests from the citizens. Their job is to ensure that every citizen gets serviced while never letting two citizens be serviced by the same person at the same time. It must be possible that two games of Checkers take place concurrently–one involving Alice and one involving Bob.

- Secretary G is in charge of receiving requests from citizens along a single *request* channel. Those messages arrive in the form (i, r) where i is a citizen ID and r is a request type, that is, one of D, N, C for a drink, neck treatment, and Checkers, respectively.
- Secretary R deals with those citizens who have received their desired service from Alice or Bob. Such a citizen would send a message along a second channel, called *release*, in the format (i, p), where i is again the citizen's ID and p is either A for Alice or B for Bob, i.e., the person who provided the service to the citizen.

A typical (partial) scenario could look like this:

```
Citizen 10 requests a neck massage.
Secretary G provides access to Bob for citizen 10.
Citizen 3 requests a game of Checkers.
Citizen 7 requests a mean cocktail.
Bob fixes citizen 10's neck.
Secretary G provides access to Alice for citizen 3.
Alice plays Checkers with citizen 3.
Citizen 3 tells secretary R that the game vs Alice has finished.
Secretary G provides access to Alice for citizen 7.
Alice mixes a mean cocktail for citizen 7.
Citizen 10 tells secretary R that the neck massage by Bob has finished.
```

[1] **C**orruption **P**arty of **M**acadamia

```
Citizen 3 requests a mean cocktail.
...
```

3 marks: Develop a citizen process description in Promela.[2]

2 × 4 marks: Again using Promela notation, develop processes to model the secretaries. These two processes may share variables.

4 marks: Formulate relevant correctness criteria for your processes in LTL.

Question 4 (15 marks)

Suppose a sender process S and a receiver process R share a secret key k that can be used to encrypt and decrypt the elements of a sequence $(a_i)_{i<L}$ of $L \in \mathbb{N}$ data items (e.g. by XORing each element with the key). The sender S has a public channel to R that is capable of transmitting one encrypted sequence item at a time. The channel is FIFO and reliable. The receiver R needs to learn the sequence without exposing its content on the public channel.

3 marks: Construct synchronous transition diagrams for S and R.

3 marks: Devise a precondition ϕ and a postcondition ψ for $P = S \parallel R$ to capture correct transmission of the sequence.

4 marks: Prove $\{\phi\}\, P\, \{\psi\}$.

5 marks: The previous proof ignored the secrecy requirement. How would you prove that no spy process P who sees the channel contents can learn the sequence of data items. Formulate all assumptions that are needed for your proof to go through. Sketch the proof based on those assumptions.

[2] Promela syntax is not really essential. If in doubt, make up notation yourself and explain what you mean.