PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS MAY/JUNE 2003

SWANSEA

Computer Science

CS 116 Modelling Computing Systems

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_116

MODELLING COMPUTING SYSTEMS

(*Attempt 2 questions out of 3*)

Question 1

In the game of **NIM**, an arbitrary number of piles of coins are formed, each with an arbitrary number of coins in them, and two players alternate in removing one or more coins from any one pile. The player who takes the last coin is declared to be the winner.

(a) Explain carefully how you can determine who has the winning strategy in NIM, and which moves are the winning moves. Use examples, and make it clear why the procedure works.

[8 marks]

(b) Without referring to the above general theory of NIM, argue that in 2-pile NIM:

- the 2nd player has the winning strategy if the piles contain an *equal* number of coins;
- the 1st player has the winning strategy if the piles contain an *unequal* number of coins.

[4 marks]

(c) Who has the winning strategy in NIM when you start with n piles each containing the same number of coins? Explain your answer.

[3 marks]

The game of **MISÈRE NIM** is played like NIM, but the player who takes the last coin is declared to be the *loser*.

(d) Who has the winning strategy in 1-pile MISÈRE NIM?

[3 marks]

(e) Who has the winning strategy in 2-pile MISÈRE NIM when one pile has only 1 coin?

[3 marks]

(f) Argue that in 2-pile MISÈRE NIM, when one of the two piles contains at least 2 coins:

- the 2nd player has the winning strategy if the piles contain an *equal* number of coins;
- the 1st player has the winning strategy if the piles contain an *unequal* number of coins.

[4 marks]

Question 2

For any integer $\mathrm{k}>0$, a k-***counter*** is a system which stores an integer value between 0 and k (inclusively). The k-counter can be:

- *incremented*, as long as its value is less than k;
- *decremented*, as long as its value is greater than 0; and
- *tested* if its value is zero.

We can give the following definition for a 1-counter:

$$\boxed{\mathrm{C} \stackrel{\text{def}}{=} \mathit{iszero}.\mathrm{C} + \mathit{inc}.\mathit{dec}.\mathrm{C}}$$

(a) List the *actions*, *states* and *transitions* of the system C.

[2 marks]

(b) Draw the labelled transition system for C.

[2 marks]

(c) For each state of this system, give a property of the modal logic M which distinguishes that state from the other states (or explain why this is impossible).

[2 marks]

We can give the following definition for a 2-counter:

$$\begin{aligned}
\mathrm{C}_2 &\stackrel{\text{def}}{=} \mathit{iszero}.\mathrm{C}_2 + \mathit{inc}.\mathrm{C}_2^1\\
\mathrm{C}_2^1 &\stackrel{\text{def}}{=} \mathit{inc}.\mathrm{C}_2^2 + \mathit{dec}.\mathrm{C}_2\\
\mathrm{C}_2^2 &\stackrel{\text{def}}{=} \mathit{dec}.\mathrm{C}_2^1
\end{aligned}$$

(d) List the *actions*, *states* and the *transitions* of the system C_2.

[2 marks]

(e) Draw the labelled transition system for C_2.

[2 marks]

(f) For each state of this system, give a property of the modal logic M which distinguishes that state from the other states (or explain why this is impossible).

[2 marks]

(g) Give a definition for a 3-counter C_3, and draw its labelled transition system.

[4 marks]

(Please turn over.)

Recall the definition of the ***synchronization merge*** operator:

- Each process E has a ***synchronization sort*** $\mathcal{L}(\mathsf{E})$. (Every state of a given process has the same sort.)
- Concurrent processes must synchronize on actions which are common to their sorts.

The synchronization merge $\mathsf{E} \mid\mid \mathsf{F}$ is then defined by the following three rules:

- If $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ and $a \notin \mathcal{L}(\mathsf{F})$ then $\mathsf{E} \| \mathsf{F} \xrightarrow{a} \mathsf{E}' \| \mathsf{F}$.
- If $\mathsf{F} \xrightarrow{a} \mathsf{F}'$ and $a \notin \mathcal{L}(\mathsf{E})$ then $\mathsf{E} \| \mathsf{F} \xrightarrow{a} \mathsf{E} \| \mathsf{F}'$.
- If $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ and $\mathsf{F} \xrightarrow{a} \mathsf{F}'$ and $a \in \mathcal{L}(\mathsf{E}) \cap \mathcal{L}(\mathsf{F})$ then $\mathsf{E} \| \mathsf{F} \xrightarrow{a} \mathsf{E}' \| \mathsf{F}'$.

We can "implement" a k-counter system by merging together k copies of the 1-counter system C, defining $\mathcal{L}(\mathrm{C}) = \{\mathit{iszero}\}$. For example, a 3-counter system can be implemented as $\mathrm{C} \| \mathrm{C} \| \mathrm{C}$.

(h) Draw the labelled transition system for $\mathrm{C} \| \mathrm{C} \| \mathrm{C}$.

[4 marks]

(i) Prove that $\mathrm{C}_3 \sim \mathrm{C} \| \mathrm{C} \| \mathrm{C}$.

[5 marks]

Question 3

Consider the following process definition.

$$
\begin{aligned}
X_1 &\overset{\text{def}}{=} a.X_1 + b.X_3 & \qquad X_4 &\overset{\text{def}}{=} a.X_4 + b.X_3 \\
X_2 &\overset{\text{def}}{=} a.X_3 + a.X_6 + b.X_1 & \qquad X_5 &\overset{\text{def}}{=} a.X_3 + a.X_6 + b.X_1 \\
X_3 &\overset{\text{def}}{=} a.X_5 & \qquad X_6 &\overset{\text{def}}{=} a.X_3 + a.X_5 + b.X_4
\end{aligned}
$$

(a) Draw the labelled transition system for the above process.

How many states, actions, and transitions does it have?

[4 marks]

(b) What is a bisimulation colouring?

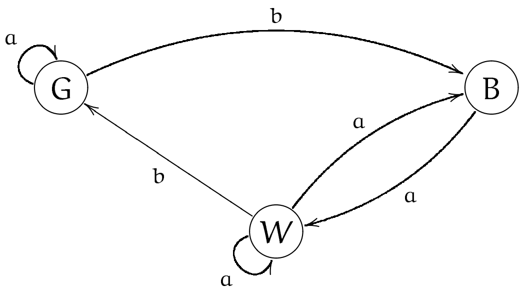
[3 marks]

(c) Carry out the bisimulation colouring algorithm step-by-step on the labelled transition system from part (a).

Explain the steps of the algorithm as you do.

[5 marks]

(d) The following labelled transition system represents the minimal process equivalent to the above process.



Explain in what sense this is true, and how this minimized transition system can be derived from the above bisimulation colouring algorithm

[4 marks]

(e) Give a process definition for the labelled transition system from part (d).

[3 marks]

(f) For each of the three state of the labelled transition system from part (d), give a modal formula which is true of it but not of the other two states.

[6 marks]