UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE FOUR

COMMUNICATION AND CONCURRENCY

Tuesday 23 April 2002

09:30 to 11:00

Examiners:

Prof. K. Turner (External)

Prof. C. Stirling (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. The ISO standard CCR service ensures coordination between two (or more) servers in the sense that all servers carry out a request or none do. The service involves communication between a controller and servers with a 4-stage mechanism

   - Controller sends request to servers.
   - A server sends a reply to the controller saying whether or not it can carry out the request.
   - If a server is able and the controller wishes to proceed, then the controller tells the server to prepare to carry out the request. If not, the controller tells the server to ignore the request.
   - If the controller tells a server to act, the server carries out the request and then sends an acknowledgement to the controller.

   Below is a description of a server S_i

$$
\begin{array}{lcl}
S_i & \stackrel{\text{def}}{=} & \mathtt{req}_i.(\overline{\mathtt{yes}}_i.Y_i + \overline{\mathtt{no}}_i.S_i) \\
Y_i & \stackrel{\text{def}}{=} & \mathtt{prepare}_i.\overline{\mathtt{act}}_i.\overline{\mathtt{ok}}_i.S_i + \mathtt{ignore}_i.S_i
\end{array}
$$

   The actions $\mathtt{req}_i$ are requests, $\mathtt{yes}_i$ and $\mathtt{no}_i$ are whether the server can or cannot carry out the request. Actions $\mathtt{prepare}_i$, $\mathtt{ignore}_i$ are preparing for or ignoring the request. The action $\mathtt{act}_i$ is the carrying out of the request and $\mathtt{ok}_i$ is the acknowledgement. Assume two servers S_1 and S_2.

   (a) Write a CCS definition of *Controller* process to achieve the informal specification above when there are two servers. [*8 marks*]

   (b) Consider the processes

$$
\begin{array}{lcl}
\mathit{Service} & \stackrel{\text{def}}{=} & (\mathit{Controller}|S_1|S_2)\backslash L \\
D & \stackrel{\text{def}}{=} & \overline{\mathtt{act}}_1.\overline{\mathtt{act}}_2.D + \overline{\mathtt{act}}_2.\overline{\mathtt{act}}_1.D
\end{array}
$$

   where L is all actions except for $\mathtt{act}_1$ and $\mathtt{act}_2$. If you have correctly implemented the controller then

$$\mathit{Service} \approx D$$

   Prove this. [*10 marks*]

   (c) Consider the following two temporal properties

      i. Eventually $\overline{\mathtt{act}}_1$ is possible

      ii. Always, whenever $\overline{\mathtt{act}}_2$ happens eventually $\overline{\mathtt{act}}_1$ will be possible

   Write these properties as formulas of CTL^{-}. Which of the properties does D have and which does *Service* have? Explain your answers. [*7 marks*]

2. The syntax of Hennessy-Milner logic (HML) is given by the following grammar:

$$\Phi ::= \mathtt{tt} \mid \mathtt{ff} \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \langle K \rangle \Phi_1 \mid [K]\Phi_1$$

where K denotes a set of actions.

(a) Define the satisfaction relation $E \models \Phi$, where E is a process and Φ is a formula of HML. *[7 marks]*

(b) Consider the two CCS processes *[6 marks]*

$$P \stackrel{\text{def}}{=} a.(b.\mathsf{0} + c.\mathsf{0}) \qquad Q \stackrel{\text{def}}{=} a.b.\mathsf{0} + a.c.\mathsf{0}$$

i. Give formulas Φ_1, Φ_2 of HML such that $P \models \Phi_1$, $Q \not\models \Phi_1$, $Q \models \Phi_2$, and $P \not\models \Phi_2$.

ii. Give a formula Φ_3 of HML such that $P \models \Phi_3$ and $P + Q \not\models \Phi_3$.

(c) Given two formulas Φ and Ψ of HML, we say that Φ is *stronger* than Ψ if the set of processes satisfying Φ is included in the set of processes satisfying Ψ. For example, $\langle a \rangle \langle b \rangle \mathtt{tt}$ and $\langle a \rangle \mathtt{tt} \wedge \langle b \rangle \mathtt{tt}$ are both stronger than $\langle a \rangle \mathtt{tt}$. Let $Q \stackrel{\text{def}}{=} a.b.\mathsf{0} + a.c.\mathsf{0}$. *[12 marks]*

i. Give a formula $\Phi_{b.0}$ of HML such that $b.\mathsf{0} \models \Phi_{b.0}$ and $\Phi_{b.0}$ is as strong as possible.

ii. Give a formula Φ_Q of HML such that $Q \models \Phi_Q$ and Φ_Q is as strong as possible.

iii. Show that there is no formula Φ of HML satisfying the following two properties

- $Q \models \Phi$.
- If $E \models \Phi$ then the transition systems of Q and E are isomorphic. (That is, one can be transformed into the other by renaming the states.)

3. (a) State what it means for a relation to be a strong bisimulation, and define strong bisimulation equivalence, $\sim$. *[4 marks]*

(b) Consider the following processes

$$
\begin{aligned}
P &\stackrel{\text{def}}{=} a.(\tau.b.c.\mathsf{0} + \tau.c.b.\mathsf{0}) \\
Q &\stackrel{\text{def}}{=} a.\tau.(b.c.\mathsf{0} + c.b.\mathsf{0}) \\
R &\stackrel{\text{def}}{=} (a.d.b.\mathsf{0} | \overline{d}.c.\mathsf{0}) \backslash d
\end{aligned}
$$

For each of the following claims prove or disprove it *[6 marks]*

i. $P \sim R$

ii. $Q \sim R$

(c) Prove the following two statements, for any CCS processes E, F, G. *[6 marks]*

i. If $E \sim F$, then $E + G \sim F + G$.

ii. If $E \sim F$, then $E|G \sim F|G$.

(d) A process E is *prime* provided that for all F and G: if $E \sim F|G$, then $F \sim \mathsf{0}$ or $G \sim \mathsf{0}$. For example, $a.b.\mathsf{0}$ is prime whereas $a.b.\mathsf{0} + b.a.\mathsf{0}$ is not prime. A process E has a prime decomposition if $E \sim F_1| \ldots |F_n$ and each process F_i, $1 \leq i \leq n$ is prime. *[9 marks]*

i. Is $a.a.(b.c.\mathsf{0} + c.b.\mathsf{0})$ prime ? Justify your answer.

ii. A CCS process is finite if it is built from the combinators $\mathsf{0}$, $+$, and prefixing a. Prove that any finite process has a prime decomposition.

iii. Give an example of a CCS process that does not have a prime decomposition. (Hint, use the $\stackrel{\text{def}}{=}$ operator.)