UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1999

MEng Honours Degrees in Computing Part IV
MSci Honours Degree in Mathematics and Computer Science Part IV
MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the Royal College of Science
Associateship of the City and Guilds of London Institute

PAPER 4.81

MODELS OF CONCURRENT COMPUTATION

Friday, April 30th 1999, 10.00 – 12.00

Answer THREE questions

For admin. only:
paper contains 4 questions

1a Let a CCS process P be defined by

$$P = (a.b.c.\mathbf{0} \mid (\bar{a}.b.\mathbf{0} + d.\mathbf{0}))\backslash a$$

Use the Expansion Theorem to find a process Q such that Q = P and Q is defined using only prefixing, plus and nil.

b i) State the τ-laws of CCS.

ii) State without proof sufficient conditions for a recursive equation in CCS to have a unique solution. Explain any terms you use.

c A system S of CCS processes is defined as follows:

$P = a_3.\bar{b}.\bar{c}.P'$ $\quad P' = d.e.P$

$Q_1 = b.Q_1'$ $\quad Q_1' = a_1.Q_1''$ $\quad Q_1'' = \bar{d}.Q_1$

$Q_2 = c.Q_2'$ $\quad Q_2' = a_2.Q_2''$ $\quad Q_2'' = \bar{e}.Q_2$

$S = (P|Q_1|Q_2)\backslash\{b,c,d,e\}$

i) Draw a static (configuration) diagram for S.

ii) Obtain a process T such that T = S and T uses only prefixing, plus and recursion. Mention any laws you use in showing T = S.

You can abbreviate S to $P||Q_1||Q_2$, etc.

(Hint: You may find it useful to let $S_1 = P'||Q_1''||Q_2'$ and $S_2 = P'||Q_1'||Q_2''$, and to use the Expansion Theorem to find S in terms of S_1 and S_2).

The three parts carry, respectively, 20%, 25%, 55% of the marks.

2a i) Define *weak bisimulation* and *weak equivalence* ($\approx$) for CCS processes.

ii) Use your definition in (i) to show that $a.(b.\mathbf{0}+\tau.c.\mathbf{0}) \approx a.(b.\mathbf{0}+\tau.c.\mathbf{0}) + a.c.\mathbf{0}$

Is it the case that $a.(b.\mathbf{0}+\tau.c.\mathbf{0}) = a.(b.\mathbf{0}+\tau.c.\mathbf{0}) + a.c.\mathbf{0}$, and why?

iii) Use your definition in (i) to show that $a.Q + P \approx a.Q + \tau.(a.Q + P)$ (any P,Q)

Is it the case that $a.Q + P = a.Q + \tau.(a.Q + P)$ (any P,Q), and why?

b i) Give the rules for the transition relation in Temporal CCS (TCCS) which corresponds to idling for one time unit, distinguishing in particular between the two forms of choice.

ii) A piece of equipment requires checking every 10-15 time units. To assist with this a safety device has three lights – green, amber and red. Exactly one light is on at a time. If the equipment does not yet need checking then green is displayed. When 10 time units have elapsed since the last check then amber comes on. When 15 units have elapsed since the last check then red comes on. The equipment can be checked at any time, and this turns the device to green.

Model the device as a process D in TCCS, using actions 'green', 'amber', 'red' and 'check'. Assume that initially the equipment has just been checked.

The two parts carry, respectively, 55%, 45% of the marks.

3a Give the sequence of reductions of the following pi calculus process, continuing as far as possible:

$$(\nu x)\ \bar{a}\langle x\rangle.\mathbf{0} \mid a(y).\ \bar{b}\langle y\rangle.\bar{y}\langle c\rangle.\mathbf{0} \mid b(z).z(w).P$$

State any laws of structural congruence which you use.

b A software company has an enquiries telephone line. Enquiries can deal with one customer at a time. When customers contact Enquiries, they give their name (which is initially private to them) and say whether they wish to be put in touch with accounts or user support. They are then transferred to either Accounts or Support, which then gives them information directly on a new private channel.

Model the scenario in the pi calculus. You can use recursion and matching (equality test). Define a process 'Company' to model the company, and define an example customer 'Cust' who wants information from accounts. Accompany your definitions with a diagram to show the initial configuration of processes and connections.

c Two parties A and B communicate according to the following protocol: A sends B a message M encrypted with the shared key K_{AB}. B decrypts the message it receives and sends it back encrypted with the shared key K_{BA}. A decrypts the message it receives from B, and checks whether it is equal to M. If so, A sends B the message 'success' in plain text.

Model the protocol in the spi calculus. The channel from A to B is c_{AB}, and the channel from B to A is c_{BA}.

The three parts carry, respectively, 30%, 45%, 25% of the marks.

4a Give the interpretations of the parallel composition (||) and action prefixing ($\rightarrow$) operators of CSP in the failures model, paying attention to alphabets, but ignoring divergence.

b State whether the following "laws" of CSP are true or false, giving a proof in the failures model or a counterexample as appropriate:

i) $a \rightarrow P \parallel a \rightarrow Q = a \rightarrow (P \parallel Q)$ where $a \in \alpha P \cap \alpha Q$

ii) $(a \rightarrow P) \parallel Q = a \rightarrow (P \parallel Q)$ provided $a \notin \alpha Q$

c Give event structures corresponding to the following CSP processes:

i) $a \rightarrow b \rightarrow c \rightarrow a \rightarrow STOP_{\{a,b,c\}} \parallel b \rightarrow d \rightarrow STOP_{\{b,d\}}$

ii) $[(a \rightarrow b \rightarrow STOP_{\{a,b,c\}}) \Box (c \rightarrow b \rightarrow STOP_{\{a,b,c\}})] \parallel b \rightarrow d \rightarrow STOP_{\{b,d\}}$

iii) $[(a \rightarrow b \rightarrow STOP_{\{a,b,c\}}) \sqcap (c \rightarrow a \rightarrow STOP_{\{a,b,c\}})] \parallel d \rightarrow a \rightarrow STOP_{\{a,d\}}$

d A pure interleaving parallel composition operator ||| for CSP is informally defined as follows: The process P|||Q consists of P and Q operating in parallel, without communicating or synchronising on common events. Each event of P|||Q is either an event of P or an event of Q.

i) Suggest an interpretation of ||| in the failures model, in the same style as your answer to (a) above.

ii) How should ||| be defined as an operator over event structures?

End of paper