UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MEng Honours Degrees in Computing 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 City and Guilds of London Institute

PAPER 4.81

MODELS OF CONCURRENT COMPUTATION

Tuesday, April 29th 1997, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1a State the Expansion Theorem for CCS processes in Standard Concurrent Form (involving parallel composition, restriction and relabelling).

b Here is a buffer of capacity one in CCS:

$B = in(x).B(x)$
$B(x) = \overline{out}\,(x).B$

i) Specify a buffer B_2 of capacity two.

ii) Construct a buffer of capacity two by chaining two copies of B, explaining your construction briefly.

iii) Use equational reasoning to show that the process you constructed in (ii) is equal to B_2 as defined in (i).

c i) Explain how the following process acts as a counter storing a natural number:

$C = up.(C \mid D)$
$D = down.\mathbf{0}$

ii) It is desired to modify C so that it can indicate when it is storing 0 (action *zero*). Explain why the following does not work:

$Z = up.(Z \mid D) + \overline{zero}\,.Z$
$D = down.\mathbf{0}$

iii) Use chaining to modify Z (and D if necessary) so that Z acts correctly as a counter with zero indicator. Explain your construction briefly; you need not prove that it works.

The three parts carry, respectively, 15%, 40%, 45% of the marks.

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

Use your definition from (i) to show

ii) $a.\tau.b + a.(\tau+c+\tau.b) \approx a.(\tau+c+\tau.b)$ (the **0**'s are omitted)

iii) $P \mid Q \approx Q \mid P$ (any CCS processes P, Q)

b Let P, Q, R, S range over CCS processes, and define

$P \approx^{+} Q$ iff for all R, $P+R \approx Q+R$

i) Show that if $P \approx^{+} Q$ then $P+S \approx^{+} Q+S$ (any P, Q, S). State any properties of $\approx$ you use.

ii) Suppose that $P \approx^{+} Q$.
Show that if $P \rightarrow^{\tau} P'$ then there is Q´ such that $Q \;{}^{\tau}\!\Rightarrow Q'$ and $P' \approx Q'$.
[Here $Q \;{}^{\tau}\!\Rightarrow Q'$ means $Q\,(\rightarrow^{\tau})^{n}\,Q'$ for some $n \geq 1$.]

iii) Use (ii) to show that it is not in general the case that $P \approx Q$ implies $P \approx^{+} Q$.

The two parts carry, respectively, 60%, 40% of the marks.

Turn over

3a By using structural congruence find the redex in

$$!((\nu c)(y(a).\overline{a}\ c\ .\mathbf{0})) \mid \overline{y}\ b\ .Q$$

and give the result of the reduction.

b i) Explain the difference between structural congruence ≡ and strong equivalence = in the pi-calculus

ii) State the laws for + for structural congruence ≡ and strong equivalence = respectively.

c Explain why the following equation

$$x \mid \overline{y} = x.\overline{y} + \overline{y}.x \qquad (*)$$

is problematic in the pi-calculus. How can the problem be resolved?

d A telephone chatline service works as follows: Customers ring the number of the service (action *req*) and leave their own number. After the first two customers have rung, the service rings them back and sets up two channels *a,b* for them to talk one to one (without revealing their numbers to each other). Channel *a* allows customer 1 to talk to 2, and *b* allows 2 to talk to 1. The chat is terminated by the service (action *end*). Subsequent customers are similarly paired off by the service and any number of conversations can take place at once.

Model the service *S* and a customer *C(no)* with telephone number *no* in the pi-calculus.

The four parts carry, respectively, 15%, 25%, 25%, 35% of the marks.

4a A scheduler S controls when n processes start up and close down. Each process P_i must be constrained to alternately start up (event a_i) and close down (event b_i). Processes should start up in cyclic order starting with P_1 ($a_1, a_2, \ldots, a_n, a_1, \ldots$) and close down in any order. Additionally, no more than three processes are active at any time. Initially no process is active.

i) Model S as a CSP process.

ii) Give a failures-style specification (with traces and refusals) of S.

b i) Explain briefly the distinction between the operators □ and ⊓ of CSP.

ii) Give the interpretation of □ and ⊓ in the failures model of CSP, paying attention to alphabets.

In each of the following cases state whether the law is valid in the failures model of CSP, giving a proof or counterexample as appropriate:

iii) $a \rightarrow (P \,\Box\, Q) = (a \rightarrow P) \,\Box\, (a \rightarrow Q)$

iv) $a \rightarrow (P \sqcap Q) = (a \rightarrow P) \sqcap (a \rightarrow Q)$

End of paper