

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2010

MSci Honours Degree in Mathematics and Computer Science Part IV

MEng Honours Degrees in Computing Part IV

MSc in Advanced Computing

MSc in Computing Science (Specialist)

for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the

Associateship of the City and Guilds of London Institute

This paper is also taken for the relevant examinations for the

Associateship of the Royal College of Science

PAPER C481

MODELS OF CONCURRENT COMPUTATION

Thursday 6 May 2010, 10:00

Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

- 1 Modern pedestrian crossings are designed to keep the traffic stopped for as long as it takes the pedestrian to cross, rather than just for a fixed time. Such crossings work by having a sonar sensor which continually ‘pings’ the crossing space to detect objects in it. Consider the CCS process

$$\text{Crossing} = \text{new } L(\text{GoodPedestrian} \mid \text{Control} \mid \text{Light} \mid \text{Sonar}),$$

where

$$\begin{aligned} \text{GoodPedestrian} &= \overline{\text{req}}.\text{AwaitingSignal} + \text{no_ping}.\text{GoodPedestrian} \\ \text{AwaitingSignal} &= \text{green}.\text{StartCrossing} + \text{no_ping}.\text{AwaitingSignal} \\ \text{StartCrossing} &= \text{ping}.\text{InCrossing} + \text{no_ping}.\text{StartCrossing} \\ \text{InCrossing} &= \text{no_ping}.\text{Safe} + \text{ping}.\text{InCrossing} + \text{red}.\text{Litigant} \\ \text{Safe} &= \text{no_ping}.\text{Safe} + \text{red}.0 \\ \text{Litigant} &= \overline{\text{sue}}.0 \end{aligned}$$

where the set L contains all action names used by Crossing except $\overline{\text{sue}}$. The well-behaved pedestrian process uses the following actions: $\overline{\text{req}}$ to send a request to the control to cross; green (red) to acknowledge the green light (red light); ping (no_ping) to indicate that the pedestrian is in the crossing (the sonar ‘ping’ is reflected back to the sonar); and the action $\overline{\text{sue}}$ to sue for mental stress as the red light appeared whilst the pedestrian was crossing.

- Complete the definition of Crossing by defining the CCS processes Control , Light and Sonar , assuming that only one pedestrian is at the crossing.
- Draw the transition graph of Crossing . Explain your answer to part (a), including any assumptions you have made regarding the pedestrian and the sonar.
- If you have correctly implemented Crossing , it should be impossible for a pedestrian to sue the system. Show that $\text{System} \simeq 0$, by giving a weak bisimulation relation containing the pair $(\text{System}, 0)$ and explaining in words why it is a weak bisimulation relation.
[You may appeal to part (b) for this answer.]
- Adapt GoodPedestrian to allow a continual stream of pedestrians to cross. Comment on your answer.

The four parts carry, respectively, 40%, 30%, 20%, and 10% of the marks.

- 2 A simple *broadcaster* $B_n \stackrel{\text{def}}{=} a. (\bar{b}_1 \mid \dots \mid \bar{b}_n)$ waits for an action a , and then in arbitrary order does $\bar{b}_1, \dots, \bar{b}_n$ and evolves to the empty process. For $n > 2$, define a chaining combinator \frown_n by

$$B_{n-1} \frown_n B_2 \stackrel{\text{def}}{=} \text{new } c (B_{n-1}[c/b_{n-1}] \mid B_2[c/a, b_{n-1}/b_1, b_n/b_2]).$$

- a Give the transition graph of $B_2 \frown_3 B_2$.
- b Give a weak bisimulation relation which demonstrates that $B_3 \approx (B_2 \frown_3 B_2)$. Prove that it is a weak bisimulation.
[In fact, $B_n \approx (B_{n-1} \frown_n B_2)$ but you are not requested to show this.]
- c A *repetitive broadcaster* RB_n is similar to B_n except that, after performing the actions $\bar{b}_1, \dots, \bar{b}_n$, it resumes its initial state.
 - i) Amend the definition of B_n to provide a definition of RB_n .
 - ii) Explain in words why $RB_3 \approx (RB_2 \frown_3 RB_2)$ does not hold.
 - iii) Prove that $RB_3 \approx (RB_2 \frown_3 RB_2)$ does not hold.

The three parts carry, respectively, 20%, 40%, and 40% of the marks.

3 a The syntax of the polyadic synchronous π -calculus is defined as

$$P, Q ::= 0 \mid P \mid Q \mid (\nu a)P \mid !P \mid \bar{u}(\tilde{v}).P \mid u(\tilde{x}).P$$

The reduction rule (Comm) is extended to:

$$\bar{a}\langle b_1, \dots, b_n \rangle.P \mid a(x_1, \dots, x_n).Q \rightarrow P \mid Q\{b_1/x_1\} \dots \{b_n/x_n\}$$

- i) Explain the difference between the monadic asynchronous π -calculus and the polyadic synchronous π -calculus.
- ii) Encode the polyadic synchronous π -calculus into the monadic asynchronous π -calculus. Your encoding should be *direct* from the polyadic synchronous π -calculus into the asynchronous π -calculus (i.e. you cannot use the encoding from the polyadic synchronous π -calculus into the monadic synchronous π -calculus from the lecture notes).
- iii) Demonstrate that your encoding is correct by showing that there is no mix-up in the following processes:

$$\llbracket \bar{a}\langle v_1, v_2 \rangle.P_1 \rrbracket \mid \llbracket \bar{a}\langle w_1, w_2 \rangle.P_2 \rrbracket \mid \llbracket a(x_1, x_2).Q \rrbracket$$

I.e. x_1, x_2 is only replaced by v_1, v_2 or w_1, w_2 (Hint: you can use equational law (τ au)) and the reduction congruence (\cong)).

b For this part, use the Session calculus from the lecture notes.

- i) Write a small variable agent $\mathbf{Var}(x, v_1, v_2, \dots, v_n)$ (storing value v_1, \dots, v_n at x) which satisfies the following specification:
 - * if read_i is chosen, then it returns the stored value v_i and recurs to the same variable;
 - * if write_i is chosen, then it receives a value w and returns to the variable with the new state $\mathbf{Var}(x, v_1, \dots, v_{i-1}, w, v_{i+1}, \dots, v_n)$; and
 - * if quit is chosen, it ends the loop (i.e. it terminates with 0).
- ii) Give a process $\llbracket x.i ::= 7 \rrbracket$ which has the following property:

$$\mathbf{Var}\langle x, v_1, \dots, v_i, \dots, v_n \rangle \mid \llbracket x.i ::= 7 \rrbracket \rightarrow^* \mathbf{Var}\langle x, v_1, \dots, 7, \dots, v_n \rangle.$$

Explain why the following equation does not hold with a counterexample:

$$\mathbf{Var}\langle x, v_1, \dots, v_i, \dots, v_n \rangle \mid \llbracket x.i ::= 7 \rrbracket \cong \mathbf{Var}\langle x, v_1, \dots, 7, \dots, v_n \rangle.$$

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

- 4a Assume the monadic asynchronous π -calculus. Recall the forwarder process $\mathbf{FW}\langle b, x \rangle$ and the equator process $\mathbf{EQ}\langle b, x \rangle$ from the lecture notes:

$$\mathbf{FW}(a, b) \stackrel{\text{df}}{=} a(z).\bar{b}\langle z \rangle \quad \text{and} \quad \mathbf{EQ}(a, b) \stackrel{\text{df}}{=} !\mathbf{FW}\langle a, b \rangle \mid !\mathbf{FW}\langle b, a \rangle$$

In this part, assume the laws for the reduction congruence \cong from the lecture notes and state the names of the laws if used.

- i) Prove $(\nu d, e)(d(y).\bar{e}\langle v \rangle \mid \bar{d}\langle w \rangle) \cong 0$.
- ii) Prove $\mathbf{EQ}\langle b, x \rangle \mid \mathbf{EQ}\langle c, x \rangle \cong !(\mathbf{FW}\langle c, x \rangle \mid \mathbf{FW}\langle x, c \rangle \mid \mathbf{FW}\langle c, b \rangle \mid \mathbf{FW}\langle b, c \rangle)$ using the law (Multi!!) $!(R_1 \mid R_2) \cong !R_1 \mid !R_2$.

- iii) Let P and Q be

$$\begin{aligned} P &= a(x).(\mathbf{EQ}\langle b, x \rangle \mid \mathbf{EQ}\langle c, x \rangle) \\ Q &= (\nu c)(\mathbf{FW}\langle a, c \rangle \mid \\ &\quad !c(x).(\nu d, e)(\bar{c}\langle x \rangle \mid \mathbf{FW}\langle c, x \rangle \mid \mathbf{FW}\langle x, c \rangle \mid \mathbf{FW}\langle c, b \rangle \mid \mathbf{FW}\langle b, c \rangle \\ &\quad \mid d(y).\bar{e}\langle v \rangle \mid \bar{d}\langle w \rangle)). \end{aligned}$$

If $P \cong Q$, prove $P \cong Q$. If $P \not\cong Q$, then give a context able to tell the two processes apart.

- b Consider the following protocol:

$$\begin{array}{c} A \xleftarrow{n} B \\ \xrightarrow{\{\{n, m\}_{k_A^-}\}_{k_B^+}} \end{array}$$

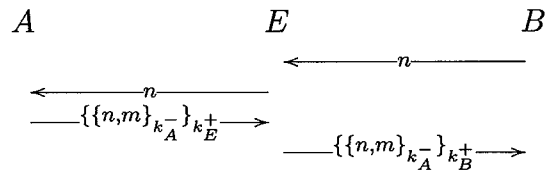
where $\{-\}_{k_I^-}$ and $\{-\}_{k_I^+}$ denote respectively a signature with a private key and an encryption with a public key. The name n is fresh. This protocol is supposed to establish the secrecy and authenticity of m , i.e. only B learns m and there is the guarantee that m originates from A .

- i) Define the equational theories of signatures and asymmetric encryption.
- ii) Explain informally why this protocol claims to provide secrecy and authenticity properties. Explain also the role of the nonce (name) n .
- iii) We model this protocol in the applied π -calculus as follows.

$$\begin{aligned} A &\stackrel{\text{df}}{=} (\nu m)c(x).\bar{c}\langle \text{encr}(\text{sign}((x, m), \text{sk}(a)), \text{pk}(b)) \rangle.Q \\ B &\stackrel{\text{df}}{=} (\nu n)\bar{c}\langle n \rangle.c(y).C \\ C &\stackrel{\text{df}}{=} (\nu z, w)(\{\text{decr}(y, \text{sk}(b))/z\}\{\text{check}(z, \text{pk}(a))/w\} \mid [\text{fst}(w) = n] P) \\ \text{Sys} &\stackrel{\text{df}}{=} (\nu a)(A \mid \{\text{pk}(a)/\text{pk}_A\}) \mid (\nu b)(B \mid \{\text{pk}(b)/\text{pk}_B\}) \end{aligned}$$

Show the reductions of a successful run.

iv) There is a man-in-the-middle attack breaking the secrecy of message m :



Write a process E which implements the attack in the following way: E sits in parallel with A and B and emits a barb on a channel ω when the message m is equal to m_0 .

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