Concurrency and Process Algebra

Date: Wednesday 16th May 2012
Time: 14:00 - 16:00

Please answer any THREE questions from the FIVE questions provided

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text.
1. Modelling concurrent systems in FSP.

A university is updating the management of its two campus car parks. The east and west car parks are barrier controlled and currently operate independently of each other. A valid permit is needed to gain entry to the car parks. The current behaviour of the car park controls is modelled by the following FSP definitions.

```
Spaces(N=2) = Spaces[N],
Spaces[i:0..N] =
    ( when (i > 0) arrival -> Spaces[i-1]
      | when (i < N) departure -> Spaces[i+1] )

Barrier =
    ( permit -> raise -> past -> lower -> Barrier ).

||CarParkControl(M=2) =
    ( Spaces(M)/{entry.raise/arrival, exit.permit/departure}
      || {entry, exit}:Barrier
    )\{entry.permit,exit.raise,{entry,exit}.lower,{entry,exit}.past}.

||CampusCarParks =
    ( east:CarParkControl(2) || west:CarParkControl(3) ).
```

a) Explain clearly what the processes CampusCarParks, CarParkControl, Barrier and Spaces do and how the number of cars in the car parks is controlled. To do so, explain the action labelling and the parallel compositions in the definitions above.

(5 marks)

b) The car park director has noticed that a number of car park users have been sharing a permit and therefore cheating the university. Explain how such cheating is possible in the above model.

(4 marks)

c) By assuming that each permit is uniquely identified by a member of a set ID, informally describe how a shared resource storing permit identifiers can be used to solve the director's problem and explain how it may be modelled in FSP (you need not write FSP definitions but should describe what modifications are needed to the above description).

(4 marks)
d) Fixit Control Systems supply a component with the following FSP description.

```
range B = 0..1

Var = Var[0],
Var[v:B] = ( when (v==0) in -> Var[v]
             | when (v==1) out -> Var[v]
             | when (v==0) toggle -> Var[1]
             | when (v==1) toggle -> Var[0]).
```

Explain how such a process can record whether a permit has been used to enter a car park and then, using Var, define an FSP process that can model an array indexed by a permit identifier (i.e. an element of ID) to record whether any particular permit is in use. (2 marks)

e) Fixit Control Systems also supply a modified Barrier process with the following FSP description.

```
EntryBarrier =
   ( permit[id:ID] -> CheckID[id] ),
CheckID[id:ID] = (store[id].acquire ->
                ( store[id].in -> RaiseEntryBarrier[id]
                | store[id].out -> blocked[id]
                -> store[id].release -> EntryBarrier ))),
RaiseEntryBarrier[id:ID] =
   ( raise -> store[id].toggle -> past -> lower
```

This interfaces with the array store of permit identifiers via the in, out and toggle actions. EntryBarrier models, of course, the entry barrier to a car park. Explain the purpose of the store[id].acquire and store[id].release actions, and hence define an associated LOCK process for the store array. (3 marks)

f) Finally, by modifying the definition in part 1(e), define an appropriate ExitBarrier process. (2 marks)
2. On FSP semantics.

Consider the following FSP process definitions of a simple resource shared between two users with mutual exclusion:

\[ \text{RESOURCE} = (\text{acquire} \rightarrow \text{release} \rightarrow \text{RESOURCE}). \]
\[ \text{USER} = (\text{acquire} \rightarrow \text{use} \rightarrow \text{release} \rightarrow \text{USER}). \]
\[ \text{RESOURCE\_SHARE} = \{a: \text{USER} \mid b: \text{USER} \mid \{a,b\}_::\text{RESOURCE} \}. \]

a) What is meant by the alphabet of an FSP process? Using the recursive definition of the alphabet of a process, calculate the alphabet of \text{RESOURCE\_SHARE}. You should show the main steps of the calculation. (3 marks)

b) What is meant by a finite trace of an FSP process? Give an example of a trace of the process \text{RESOURCE\_SHARE}, and also an example of a sequence of actions that the process cannot make. (3 marks)

c) State the FSP rules for the parallel composition of processes. (3 marks)

d) Apply these rules, and others, to give an explicit derivation for one of the initial actions that the process \text{RESOURCE\_SHARE} may undertake. You should give full details of the derivations, including the rules you use at each step. (5 marks)

e) Describe precisely how a labelled transition system may be constructed from an FSP process using FSP rules. (3 marks)

f) Apply your construction to produce a labelled transition system for the process \text{RESOURCE\_SHARE}. (3 marks)
3. The equivalence of FSP processes.

a) Provide and explain a formal definition of \textit{(not an algorithm for)} strong bisimilarity between two labelled transition systems, and hence of two FSP processes. \hspace{1cm} (4 marks)

b) Define what is meant by two labelled transition systems being \textit{trace equivalent}. Give an example of two labelled transition systems that are trace equivalent but not strongly bisimilar. Justify your answer. \hspace{1cm} (4 marks)

c) Outline an algorithm for computing whether two given FSP processes are strongly bisimilar. Your description should explain clearly the steps in the algorithm. \hspace{1cm} (6 marks)

d) Consider the following two definitions of FSP processes, \texttt{DAY1} and \texttt{DAY2},

\[
\text{DAY1} = (\text{up} \rightarrow (\text{tea} \rightarrow W \mid \text{coffee} \rightarrow W)) ,
\]
\[
W = (\text{work} \rightarrow \text{sleep} \rightarrow \text{DAY1}).
\]

\[
\text{DAY2} = (\text{up} \rightarrow B) ,
\]
\[
B = (\text{work} \rightarrow W1 \mid \text{coffee} \rightarrow \text{work} \rightarrow W2) ,
\]
\[
W1 = (\text{work} \rightarrow \text{sleep} \rightarrow \text{DAY2}) ,
\]
\[
W2 = (\text{sleep} \rightarrow \text{up} \rightarrow \\
(\text{tea} \rightarrow \text{work} \rightarrow W2 \mid \text{coffee} \rightarrow W1)).
\]

Use the algorithm you gave in answer to part 3c to determine whether the process \texttt{DAY1} is strongly bisimilar to \texttt{DAY2}. Explain fully your working and how you arrive at your result. \hspace{1cm} (6 marks)
4. Concurrency concepts and their implementation in Java

a) Briefly explain the following concepts:

i. An interleaving model of concurrency
ii. Semaphores
iii. Mutual exclusion

(6 marks)

b) Consider the following FSP definitions that model a stack with operations push, top and pop.

```
range Item = 1 .. 3
set ItemList = { [Item], [Item][Item], [Item][Item][Item] }

STACK = ( push[i:Item] -> STACK[i] ),
STACK[i:Item] = ( push[j:Item] -> STACK[j][i]
| top[i] -> STACK[i]
| pop -> STACK ),
STACK[t:Item][r:ItemList] =
| push[j:Item] -> STACK[j][t][r]
| top[t] -> STACK[t][r]
| pop -> STACK[r] ).
```

The stack is to be shared between several concurrent users.

A Java class, Stack, is required to implement the above model of a stack so that it operates safely with multiple concurrent users. There should be methods corresponding to the operations push, top and pop. The stack object may be assumed to hold just a finite number of items.

Give a Java class definition for the FSP process STACK with a constructor and three accessor methods. You may assume the constructor is passed the maximum number of items to be stored in the stack.

(8 marks)

c) Explain clearly how multiple concurrent users can access your stack safely. You should explain the mechanisms in Java which allow multiple concurrent threads to be handled and what safe access to the stack means and how it is ensured by your Java definition and the thread mechanisms.

(6 marks)
5. On properties and property checking.

a) For concurrent systems, explain, briefly but clearly, what is meant by (i) a safety property, (ii) a liveness property, and (iii) a progress property. (4 marks)

b) Explain clearly the difference between the LIGHTS_PROPERTY property and the LIGHTS process below.

```
property
LIGHTS_PROPERTY =
    ( red \rightarrow amber \rightarrow green \rightarrow LIGHTS_PROPERTY ).

LIGHTS = ( red \rightarrow amber \rightarrow green \rightarrow LIGHTS ).
```

Draw the labelled transition system for LIGHTS_PROPERTY. (4 marks)

c) Consider the following FSP model of a state machine for a traffic signal.

```
range S = 0..3

SIGNAL = SIGNAL[0],
SIGNAL[s:S] =
    ( when (s == 0) red \rightarrow SIGNAL[1]
    | when (s == 1) amber \rightarrow SIGNAL[2]
    | when (s == 2) green \rightarrow SIGNAL[3]
    | when (s == 3) red \rightarrow SIGNAL[0] ).
```

Construct the labelled transition system corresponding to the process

```
( SIGNAL \parallel LIGHTS_PROPERTY )
```

Using this labelled transition system, determine whether the SIGNAL process satisfies the LIGHTS_PROPERTY? Provide a justification for your answer, for which, if 'no', you should include a counterexample trace. (10 marks)

d) Fix the SIGNAL FSP process definition so that it satisfies the desired property. (2 marks)