## ICTAC Summer School, Tunis 2007: CSP Examination

A ticket machine accepts coins and issues tickets. There are two buttons on the front of the machine: a green button—labelled 'press for ticket'—and a red button—labelled 'cancel'. Either of the buttons can be pressed at any time.

If the green button is pressed, and at least one coin has been deposited, the machine should issue a ticket. Whatever coins are currently held in the (temporary) coin buffer are simply consumed by the machine: no change is offered. The machine drops tickets into a ticket tray, from which they can then be collected. If the machine decides to print a ticket, this ticket will eventually be offered to the customer.

If the red button is pressed, the machine should return whatever coins are in the coin buffer: coins inserted but not yet consumed. Once the machine decides to return the coins that it is holding, it starts passing them from the coin buffer to a return mechanism: they will eventually drop into a ticket tray.

1. Using the following abstract events, write a sequential CSP process to describe the behaviour of the machine in interaction with its environment *assuming that the capacity of the coin buffer is a single coin—that is, a second coin cannot be inserted until the first has been consumed or returned, following the press of a button*

   $coin$: a user inserts a single coin

   $green$: a user presses the green button

   $red$: a user presses the red button

   $ticket$: the machine drops a ticket into the ticket tray

   $return$: the machine returns a single coin

   $service$: the machine is serviced, emptying the coin box and re-filling the ticket roll

2. Using the following additional abstract event, show how the behaviour of the machine could be described as a parallel combination of processes representing the coin buffer, the coin store, and the ticket roll:

   $consume$: a single coin is moved from the coin buffer to the coin store

   Your processes should have the following alphabets:

   $\alpha CoinBuffer = \{coin, green, red, consume, return, service\}$

   $\alpha CoinStore = \{consume, service\}$

   $\alpha TicketRoll = \{consume, ticket, service\}$

3. Do your sequential and parallel descriptions of the machine represent the same set of behaviours? Explain how you could check your answer.

4. Write a new version of each of the processes in your parallel combination, using process parameters to represent the number of coins in the buffer, the number of coins in the store, and the number of tickets remaining on the roll. These parameters should return to their original values when the machine is serviced. Explain the resulting behaviour: what can we see now that we have relaxed the assumption in the first question?