Concurrent Programming - Exercises 6

Value passing CCS

- 1. For each process P compute the LTS $[\![P]\!]$
 - (a) $(a!21.0|a?x.b!(x*2).0) \setminus \{a\}$
 - (b) $(a!42.0|a?x.Sender[x]) \setminus \{a\}$

2. For the following program compute its LTS.

Sender	:=	put?x.send!x.Sending[x]
Sending[x]	:=	receiveAck?.Sender + receiveNAck?.send!x.Sending[x]
Receiver	:=	receive?x.get!x.sendAck!.Receiver +
		gargled?.sendNAck!.Receiver
Medium	:=	send?x. (receive!x. Medium + i. garbled!. Medium)
AckMedium	:=	sendAck?.receiveAck!.AckMedium +
		send NAck?. received NAck!. Ack Medium
DupMedium	:=	Medium AckMedium
Protocol	:=	$(Sender \mid Receiver \mid DupMedium) \setminus$
		$\{send, receive, sendAck, receiveAck,$
		$receiveNAck, sendNAck, garbled\}$

Compute also $[(Protocol|put!1.put!2.put!3.put!4.0) \setminus \{put\}]$. Restrict the values to rangeR := 0..9.

3. Given

$$\begin{aligned} Fac[n,j] &:= when(j>0)i.Fac[n*j,j-1] \\ &+ when(j==0)println!n.0 \end{aligned}$$

Compute [[Fac[1, 5]]].

4. Consider the following vending machine that accepts coins of 1, 2 and 5 euros. If the price of a coffee has not been paid yet, more coins are required before a coffee is dispensed. Once enough money has been paid, no more coins are accepted, and a coffee is dispensed. If the last inserted coin caused the coffee to be overpaid, some change is given. Each coffee cost 5 euros.

$$\begin{split} Machine[b] &:= when (b < 5) coin?c.Machine[b+c] + when (b \geq 5) coff ee!.ReturnMachine[b-1] \\ ReturnMachine[b] &:= when (b > 0) change!.ReturnMachine[b-1] + Machine[0] \\ User &:= coin!2.coin!2.coin!2.coff ee?.change?.0 \end{split}$$

Construct $[(Machine[0]|User) \setminus \{coin, change, coffee\}]].$

5. Given

 $\label{eq:compute} \begin{tabular}{ll} Compute \begin{tabular}{ll} (Cells|Serve|Use) \label{eq:compute} \end{tabular} \end{tab$

 $6.\ {\rm Consider}$ the exercise 9 from Practical 5 with value passing CCS.