Modelling and Validation of Concurrent System

Lab 1: solutions of the exercises on CCS

1. Check, attempting to build derivation trees using the grammar rules, if the following terms are syntactically correct processes.

   (a) $a + b.\mathbf{0}$ is not a syntactically correct process.

   $$\dfrac{\dfrac{}{a \in \text{CCS}}\,(??) \qquad \dfrac{b \in \mathsf{Act} \qquad \dfrac{}{\mathbf{0} \in \text{CCS}}\,(Empty)}{b.\mathbf{0} \in \text{CCS}}\,(Pre)}{a + b.\mathbf{0} \in \text{CCS not derivable}}\,(Sum)$$

   An action by itself is not a process.

   (b) $a.\mathbf{0} \mid b.\mathbf{0}$

   $$\dfrac{\dfrac{a \in \mathsf{Act} \qquad \dfrac{}{\mathbf{0} \in \text{CCS}}\,(Empty)}{a.\mathbf{0} \in \text{CCS}}\,(Pre) \qquad \dfrac{b \in \mathsf{Act} \qquad \dfrac{}{\mathbf{0} \in \text{CCS}}\,(Empty)}{b.\mathbf{0} \in \text{CCS}}\,(Pre)}{a.\mathbf{0} \mid b.\mathbf{0} \in \text{CCS}}\,(Par)$$

   (c) $(a.\mathbf{0} \mid b.\mathbf{0}).\mathbf{0}$ is not a syntactically correct process – only actions can prefix processes (there is no sequencial composition of processes).

   (d) $(\mathbf{new}\,\tau)\tau.\mathbf{0}$ is not a syntactically correct process – only observable action can be hiden and τ is not an observable action.

   (e) $(\tau.\mathbf{0})\{a/\tau\}$ is not a syntactically correct process – only observable action can be substituted and τ is not an observable action.

   (f) $a.\mathbf{0} \mid (b.\mathbf{0} + \mathbf{0})$

2. Consider a subway ticket vending machine. Buyers may choose between three kinds of tickets: single, return, or ten tickets carnets.

   Clients may pay either with cash or card. Encode the values to pay as the actions `singleV`, `returnV`, or `carnetV`.

   The machine allows the client to go back one step, return to the beginning, or terminate the purchase, at any moment.

   Implement in CCS:

   (a) Customers Alice and Bob, who purchase, respectively, a return ticket with cash and 10 tickets with card. Bob starts by asking for a single ticket but changes his mind before paying.

   $$\begin{aligned} Alice &= \overline{return}.\overline{cash}.\overline{returnV}.pick.\mathbf{0} \\ Bob &= \overline{single}.\overline{start}.\overline{ten}.\overline{card}.\overline{carnet}.pick.\mathbf{0} \end{aligned}$$

   (b) A generic client that buys a ticket with card.
   $Cliente(ticket, means, amount) = \overline{ticket}.\overline{means}.\overline{amount}.pick.\mathbf{0}$

(c) Um cliente que compra um bilhete simples com cartão.
$Cliente\langle single, card, singleV\rangle$

(d) The vending machine.

$$\begin{aligned} Start &= single.Pay\langle singleV\rangle + return.Pay\langle returnV\rangle + carnet.Pay\langle carnetV\rangle \\ Pay(value) &= cash.Amount\langle value\rangle + cartao.Amount\langle value\rangle + start.Start \\ Amount(value) &= (value.\overline{pick}.\mathbf{0} + cancel.Pay\langle value\rangle) + start.Start \end{aligned}$$