1. Consider a vending machine with two buttons, B1 and B2, accepting coins of 1 Euro and 2 Euros. If button B1 is pressed the machine outputs product X, whereas if button B2 is pressed, the machine outputs product Y. Product X costs 3 Euros and product Y costs only 1 Euro. The machine can give back change (assume that it always has enough coins to give back as change).

   a) Model the behaviour of this vending machine using a labelled transition system with the following labels:

   - B1 – button B1 is pressed
   - B2 – button B2 is pressed
   - I1 – a coin of 1 Euro is inserted
   - I2 – a coin of 2 Euro is inserted
   - OX – product X is output
   - OY – product Y is output
   - C1 – a coin of 1 Euro is given back as change
   - C2 – a coin of 2 Euros is given back as change

   Assume that the user first chooses the product and then inserts the coins.  

   b) Write a CTL formula specifying that “the vending machine cannot output product X without first (earlier) a user has pressed button B1”.

   c) Let P be the atomic proposition “the user has pressed button B1” and Q be the atomic proposition “the machine has output product X”. Label with P each state with an incoming transition labelled by B1, and label with Q each state with an incoming transition labelled by OX. Does the state where the user chooses the product satisfies your CTL specification above.

2. Let p and q be two atomic propositions.

   a) Show from the CTL definition of AU that A[T U q] holds if q holds somewhere along every path.

   b) Using the fixed-point method described in the course, give the set of all states of the following transition system satisfying the CTL formula E[p U EG q].
3. Prove the partial correctness of following Hoare triple: \[10\text{ points}\]

\[(a[i] = x \land a[j] = y) ; a[i] := a[j] ; a[j] := z ; (a[i] = y \land a[j] = x)\]

4. Explain why one needs to introduce logical variables when reasoning about partial and total correctness of a command. \[10\text{ points}\]

5. Let \(a[0..n]\) be an array of integer and consider the following program MIN:

\[
t := 1;
min := a[0]
while \(t \leq n\) {
\quad if \ a[t] < min then {
\quad \quad min := a[t]
\quad } else {
\quad \quad \text{skip}
\quad }
\quad t := t + 1
}
\]

a) Write a postcondition \(\phi\) expressing the fact that the above program calculates the minimum number in the array \(a[0..n]\). \[10\text{ points}\]

b) Give an invariant for the while command establishing the postcondition \(\phi\). \[10\text{ points}\]

c) Give the variant function for proving the termination of the while command. \[10\text{ points}\]

d) Give a proof outline for the total correctness of \((\text{true}) \; \text{MIN} \; (\text{\phi})\). \[10\text{ points}\]

The final score is given by the sum of the points obtained divided by 10 (with a maximum of 10).

---

**Proof system for partial correctness:**

1. \((l \; \phi \; l) \; \text{skip} \; (l \; \phi \; l)\)
2. \((l \; \phi[l/x] \; l) \; x := e \; (l \; \phi \; l)\)
3. \((l \; \phi \; l) \; c1 \; (l \; \phi \; l) \; (l \; \phi \; l) \; c2 \; (l \; \psi \; l)\)
4. \((l \; \phi \land b \; l) \; c1 \; (l \; \psi \; l) \; (l \; \phi \land \lnot b \; l) \; c2 \; (l \; \psi \; l)\)
5. \((l \; \phi \; l) \; \text{while} \; b \; \text{do} \{ c \} \; (l \; \phi \land \lnot b \; l)\)
6. \((l \; \phi \; l) \; \text{if} \; b \; \text{then} \{ c1 \} \; \text{else} \{ c2 \} \; (l \; \psi \; l)\)

\(\phi \Rightarrow \phi l \; (l \; \phi l) \; c \; (l \; \psi l) \; \psi l \Rightarrow \psi\)

\((l \; \phi \; l) \; c \; (l \; \psi \; l)\)