Software Validation and Verification
First Exercise Sheet

Exercise 1

For this exercise we give the following definition:

Definition 1. *Deterministic Transition System Let $TS = (S, Act, \to, I, AP, L)$ be a transition system.*

1. *TS is called* action-deterministic *if* $|\, I \,| \;\leq\; 1$ *and* $|\, Post(s, \alpha) \,| \;\leq\; 1$ *for all states* s *and actions* α.
2. *TS is called* AP-deterministic *if* $|\, I \,| \;\leq\; 1$ *and* $|Post(s) \cap \{s' \in S \mid L(s') = A\}| \;\leq\; 1$ *for all states* s *and* $A \in 2^{AP}$.

Consider the following the transition system TS_1.

$$\begin{tikzcd}
{} \arrow[r] & s_0 \ \{a\} \arrow[rr, "\gamma"] \arrow[dd, bend right, "\alpha"'] & & s_1 \ \emptyset \arrow[loop above, "\alpha"] \arrow[dl, "\beta"] \arrow[dd, "\alpha"] \\
& & s_4 \ \emptyset \arrow[dl, bend right, "\alpha"'] \arrow[dr, "\gamma"] & \\
& \{a, b\} \ s_2 \arrow[uu, bend right, "\alpha"'] \arrow[ur, bend right, "\beta"'] & & s_3 \ \{b\}
\end{tikzcd}$$

a) Give the formal definition of TS_1.

b) Specify a finite and an infinite execution of TS_1.

c) Decide whether TS_1 is an *AP*-deterministic or an action-deterministic transition system. Justify your answer.

Exercise 2

Consider the following mutual exclusion algorithm that uses the shared variables y_1 and y_2 (which are initially both 0):

Process P_1:

```
while true do
  ... non-critical section ...
  $y_1 := y_2 + 1$
  wait until $(y_2 = 0) \vee (y_1 \leq y_2)$
  ... critical section ...
  $y_1 := 0$
  ... non-critical section ...
od
```

Process P_2:

```
while true do
  ... non-critical section ...
  $y_2 := y_1 + 1$
  wait until $(y_1 = 0) \vee (y_2 < y_1)$
  ... critical section ...
  $y_2 := 0$
  ... non-critical section ...
od
```

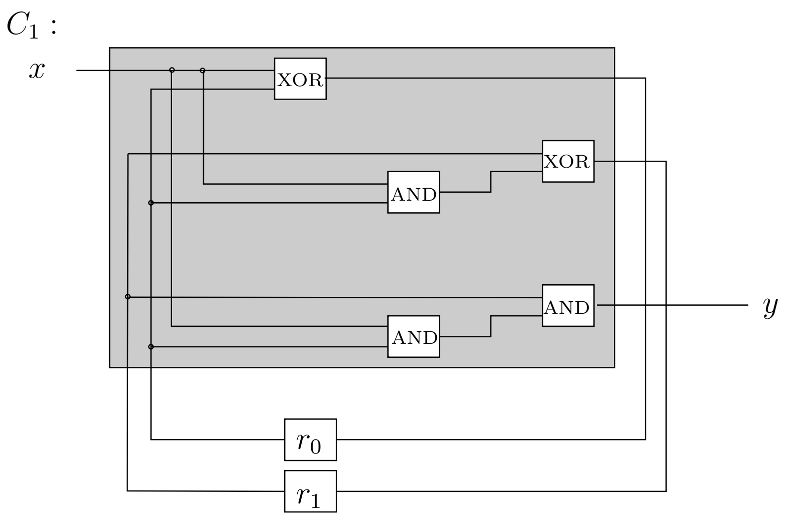
Questions:

a) Give the program graph representation of both processes.
   (A pictorial representation suffices)

b) Give the reachable part of the transition system of $P_1 ||| P_2$ where $y_1 \leq 2$ and $y_2 \leq 2$.

Exercise 3

The circuit C_1 describes the layout of a hardware adder that stores a 2-bit binary number represented by the registers r_0 and r_1. In each cycle, the value of x is added to the currently stored value; y is used as the carry bit:



Give the transition system representation TS_1 of the circuit C_1.

Exercise 4

In the following, whenever transition systems are compared via $=$ or $\neq$, this means (in)equality **up to renaming of states** (i.e. isomorphism).

a) Show that, the handshaking $\|_H$ operator **is not** associative, i.e. that in general

$$(\mathrm{TS}_1 \|_H TS_2) \|_{H'} \mathrm{TS}_3 \neq \mathrm{TS}_1 \|_H (\mathrm{TS}_2 \|_{H'} \mathrm{TS}_3)$$

b) The handshaking operator $\|$ that forces transition systems to synchronize over their common actions **is** associative. Show that

$$\underbrace{(\mathrm{TS}_1 \| \mathrm{TS}_2) \| \mathrm{TS}_3}_{L} = \underbrace{\mathrm{TS}_1 \| (\mathrm{TS}_2 \| \mathrm{TS}_3)}_{R}$$

where $\mathrm{TS}_1, \mathrm{TS}_2, \mathrm{TS}_3$ are arbitrary (finite) transition systems. To this end, show that the bijective function $f_\approx \colon (S_1 \times S_2) \times S_3 \to S_1 \times (S_2 \times S_3)$ given by $f_\approx(\langle\langle s_1, s_2\rangle, s_3\rangle) = \langle s_1, \langle s_2, s_3\rangle\rangle$ preserves the transition relation in the sense that

$$l \xrightarrow{\alpha}_L l' \iff f_\approx(l) \xrightarrow{\alpha}_R f_\approx(l') \tag{1}$$

where $l, l' \in S_L$, S_L is the state space of transition system L and $\longrightarrow_L$, $\longrightarrow_R$ are the transition relations of L and R, respectively.

Hint: When considering an action α, you need only distinguish the cases

(i) $\alpha \in \mathrm{Act}_1 \backslash (Act_2 \cup Act_3)$

(ii) $\alpha \in (\mathrm{Act}_1 \cap \mathrm{Act}_2) \backslash Act_3$

(iii) $\alpha \in \mathrm{Act}_1 \cap Act_2 \cap Act_3$

(Act_i is the action set of TS_i) as all other cases are symmetric. Also, for simplicity, it suffices to show the direction "$\Longrightarrow$" of condition (1). However, keep in mind that L and R are not necessarily action-deterministic (see exercise sheet 1).

Exercise 5

Consider the following leader election algorithm: For $n \in \mathbb{N}$, n processes $\mathcal{P}_1, \ldots, \mathcal{P}_n$ are located in a ring topology where each process is connected by an unidirectional, asynchronous channel to its neighbour as outlined below.

$$\begin{tikzcd}
 & \mathcal{P}_1 \arrow[dr, bend left] & \\
\mathcal{P}_n \arrow[ur, bend left] & & \mathcal{P}_2 \arrow[d, bend left] \\
\mathcal{P}_{n-1} \arrow[u, bend left] & & \mathcal{P}_3 \arrow[d, bend left] \\
\mathcal{P}_{n-2} \arrow[u, bend left] & & \mathcal{P}_4 \arrow[dl, bend left] \\
 & \ldots \arrow[ul, bend left] &
\end{tikzcd}$$

To distinguish the processes, each process i is assigned a unique identifier $id(\mathcal{P}_i) \in \{1, \ldots, n\}$ that is written to private variable id_i. The aim of the algorithm is to elect the process with the highest identifier as the (unique) leader within the ring. Therefore each process executes the following algorithm using another private variable m_i (which is initially 0):

```
send($id_i$); // send own id to next process.
while (true) do {
  receive ($m_i$);
  if ($m_i$ == $id_i$) then stop; // process i is the leader
  if ($m_i$ > $id_i$) then send($m_i$); // forward other identifier
}
```

a) Model the leader election protocol for n processes as a channel system.

b) Give an initial execution fragment of $TS([\mathcal{P}_1 \mid P_2 \mid \mathcal{P}_3])$ such that at least one process has executed its send-statement within the body of the while-loop. Assume for $1 \leq i \leq 3$, that process P_i has identifier $id_i = i$.