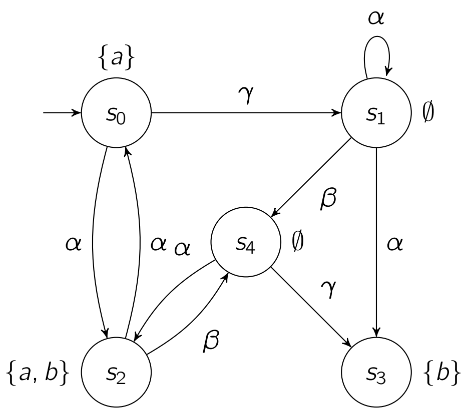
Exercise 1 (Transition Systems): (2 points)

For this exercise we give the following definition:

Definition 1. *Deterministic Transition System Let* $TS = (S, Act, \rightarrow, I, AP, L)$ *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.



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 (Reachability in Parallel Composition): (1 point)

We are given three (primitive) processes P_1, P_2, and P_3 with shared integer variable x and local registers r_1, r_2 and r_3. The program of process P_i is as follows:

```
for k := 1 to 10 do {
   r_i := x
   r_i := r_i + 1
   x := r_i
}
```

That is, P_i executes ten times the assignment $x := x + 1$. Consider now the parallel program $P = P_1 \parallel P_2 \parallel P_3$ with x initially being 0. Does P have an execution that halts with the terminal value $x = 2$? Justify your answer.

Exercise 3 (Mutual Exclusion): (5 points)

The following program is a mutual exclusion protocol for two processes due to Pnueli. There is a single shared variable s which is either 0 or 1, and initially 1. Besides, each process has a local Boolean variable y that initially is 0. The program text for process P_i ($i = 0, 1$) is as follows:

```
while (true) do {
  // non-critical section
  (y_i, s) := (1, i);
  wait until ((y_{1-i} = 0) ∨ (s ≠ i));
  // critical section
  y_i := 0
}
```

Here, the statement $(y_i, s) := (1, i)$; is a *multi-assignment*, i.e. the assignments $y_i := 1$ and $s := i$ are carried out in a single (atomic) step. Also, checking the boolean condition is done atomically.

a) Define the program graph of a process in Pnueli's algorithm. Make sure that both the critical and the non-critical section correspond to distinguished locations.

b) Construct the transition system of their parallel composition $\mathrm{TS}(P_0 \mid\mid\mid P_1)$ over the state space (l_i, l_j, y_0, y_1, s). *Hint:* If you wish, you may treat the states $(l_i, l_j, y_0, y_1, 0)$ and $(l_j, l_i, y_1, y_0, 1)$ to be equivalent. In other words, whenever you encounter a transition to a state with $s = 0$, you may swap the locations of the processes P_0 and P_1, swap the values of their private variables and set s to 1 and make the transition target this "new" state instead. This operation considerably reduces the size of the resulting state space.

c) Check whether the algorithm ensures mutual exclusion, i.e. both processes are never in their critical section at the same time. Justify your answer.

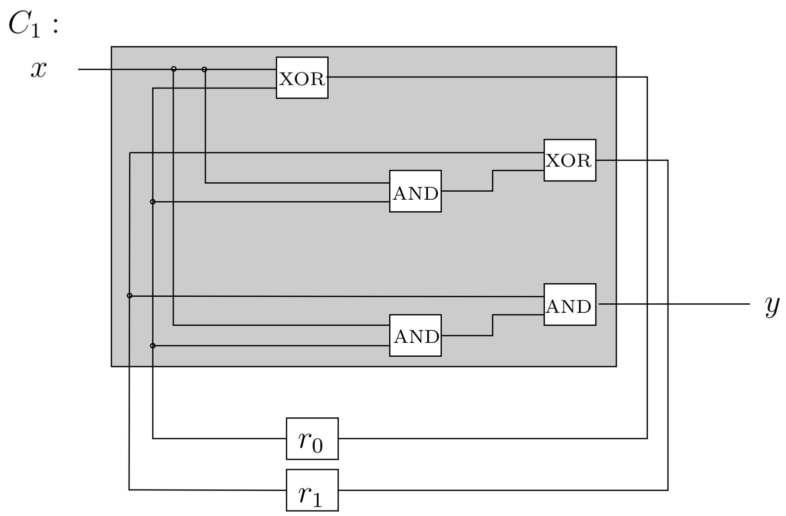
d) Does the algorithm guarantee that every time a process wants to enter its critical section it can eventually do so (starvation freedom)? Justify your answer.

e) What happens, if we modify the condition for the waiting phase to $y_{1-i} = 0$ (that is, we drop the second disjunct)? In particular, does the algorithm then satisfy mutual exclusion and/or starvation freedom? Justify your answer.

Hint: If possible, you may use observations on the transition system as a justification.

Exercise 4 (Circuits): (2 points)

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.