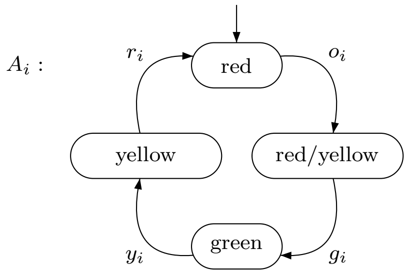
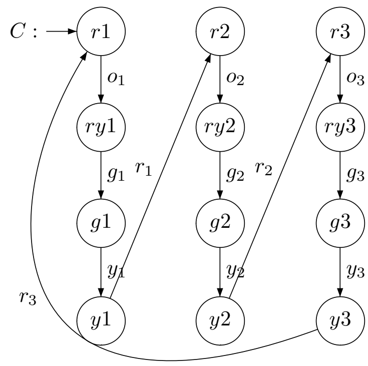
Exercises of Chapter 2

Answer to Exercise 2.3

(a) Let $Act = \bigcup_{0<i\leqslant 5}\{\, r_i, o_i, g_i, y_i \,\}$. Labeling the transition system A_i yields:



(b) The controller has to synchronize with the traffic lights. Note that the actions defined in part (a) uniquely identify the i-th transition system. This is exploited by the controller in the following way. The controller synchronizes with the traffic lights using pairwise handshaking.



(c) Let $TS_1 \| \cdots \| TS_n$ denote the parallel composition of TS_1 through TS_n where TS_i and TS_j $(0 < i < j \leqslant n)$ synchronize over the set of actions $H_{i,j} = Act_i \cap Act_j$ such that $H_{i,j} \cap Act_k = \varnothing$ for $k \notin \{\, i, j \,\}$. The inference rules for the transition relation are:

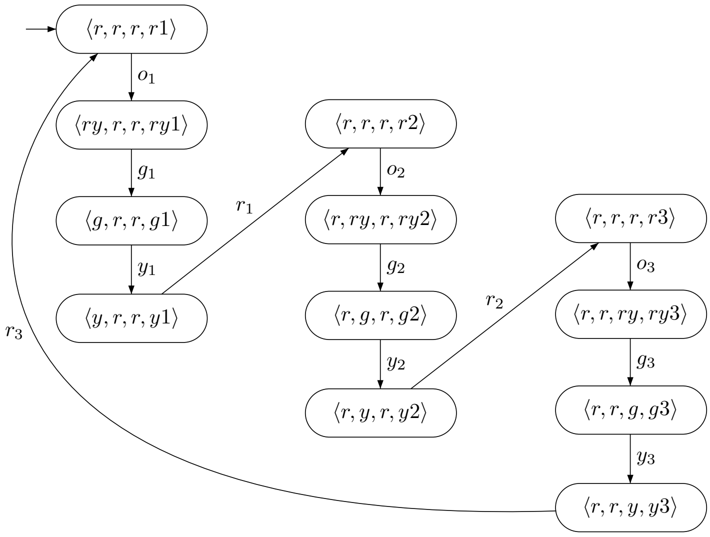
– if $\alpha \in Act_i \setminus \bigcup_{0<j\leqslant n, i\neq j} H_{i,j}$ and $0 < i \leqslant n$:

$$\frac{s_i \xrightarrow{\alpha}_i s_i'}{\langle s_1, \ldots, s_i, \ldots, s_n \rangle \xrightarrow{\alpha} \langle s_1, \ldots, s_i', \ldots, s_n \rangle}$$

– if $\alpha \in H_{i,j}$ and $0 < i < j \leqslant n$:

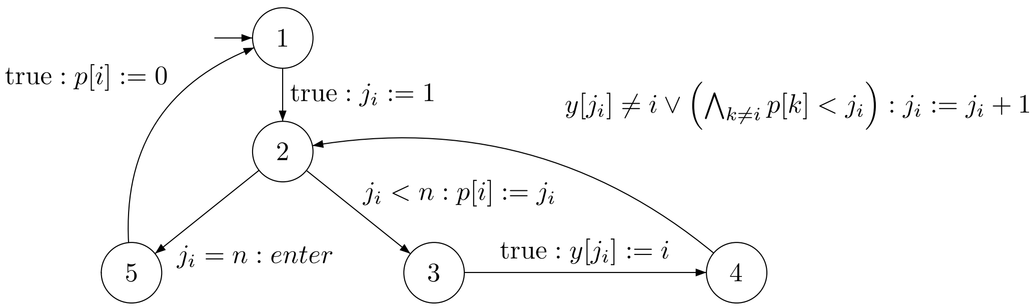
$$\frac{s_i \xrightarrow{\alpha}_i s_i' \qquad s_j \xrightarrow{\alpha}_j s_j'}{\langle s_1, \ldots, s_i, \ldots, s_j, \ldots, s_n \rangle \xrightarrow{\alpha} \langle s_1, \ldots, s_i', \ldots, s_j', \ldots, s_n \rangle}$$

By applying these inference rules, the transition system $A_1 \| A_2 \| A_3 \| C$ becomes:



Answer to Exercise 2.7

(a) The program graph PG_i for process i is given as:



Note that we consider i as a constant here and that the variables j_i are private to process i.

(b) The cardinality of the set of states of $TS(PG_1 ||| \cdots ||| PG_n)$ can be deduced as follows. Let $PG_i = (Loc_i, Act_i, Effect_i, \hookrightarrow_i, Loc_0, g_0)$ be the formal representation of the program graph from part (a) where:

- $Loc_i = \{ 1, 2, 3, 4, 5 \}$
- $Act_i = \{ j_i := 1, p[i] := j_i, y[j_i] := i, j_i := j_i{+}1, enter, p[i] := 0 \mid i \in \{1, \ldots, n\} \}$

According to the algorithm, we have:

$$
\begin{array}{lcll}
dom(y[k]) & = & \{ 1, \ldots, n \} & \text{for all } k \in \{0, \ldots, n-1\} \\
dom(j_i) = dom(p[k]) & = & \{ 0, \ldots, n-1 \} & \text{for all } k, i \in \{1, \ldots, n\}
\end{array}
$$

Therefore it follows $|dom(y[k])| = |dom(j_i)| = |dom(p[k])| = n$. The arrays y and p have capacity n. The state space of the transition system is:

$$S = Loc_1 \times \cdots \times Loc_n \times Eval\left(\{p[k], y[l], j_i \mid i, k \in \{1, \ldots, n\} \text{ and } l \in \{0, \ldots, n-1\}\}\right).$$

Therefore we obtain $|S| = 5^n \cdot n^{3n}$.

(c) We prove a stronger statement that implies mutual exclusion:

At level $j \in \{0, \ldots, n-1\}$, at most $n-j$ processes are at level $\geqslant j$

By definition, process P_i is at level j iff $p[i] = j$. We proceed by induction over j:

- **basis** ($j = 0$): The statement trivially holds, as $n-j = n-0 = n$ and there are at most n processes in the system.
- **induction step** ($j \rightsquigarrow j+1$): The induction hypothesis implies that there are at most $n-j$ processes at level $\geqslant j$. We show that there is at least one process that cannot move from level j to level $j+1$. By contradiction, assume there also were $n-j$ processes at levels $\geqslant j+1$ (i.e., no process is stuck at level j). Let i be the last process that writes to $y[j]$. Therefore, the old value of $y[j]$ that corresponds to the previous process k at level j is overwritten and we have $y[j] = i$. Hence the condition $y[j] \neq i$ cannot be true. According to the algorithm,
  - ∗ process k writes $p[k]$ before it writes $y[j]$ and
  - ∗ process i reads $p[k]$ only after it wrote to $y[j]$.

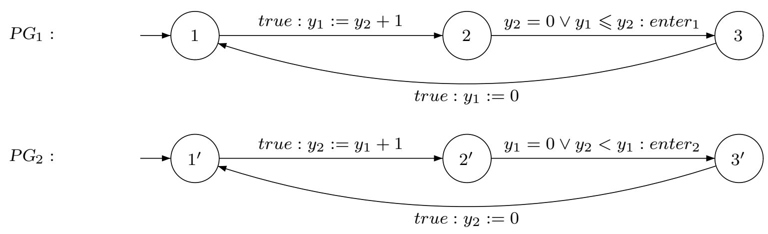
  Therefore every time process i reads $p[k]$, process k already set $p[k] = j$ and for process i, the second condition $p[k] < j$ is not fulfilled either.

  We assumed that process i enters level $j+1$. This yields a contradiction since it cannot leave the wait–loop.

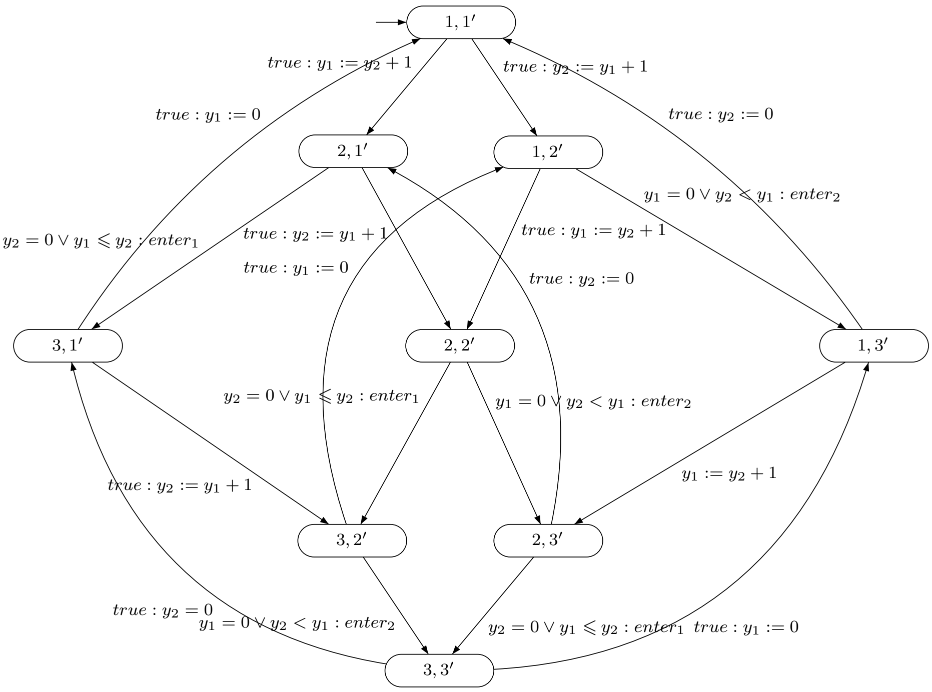
According to the idea of the algorithm, a process enters the critical section when it leaves the wait–loop at level $n-1$. As we proved, at level $n-1$, there may only be $n-(n-1) = 1$ processes at level $\geqslant (n-1)$. Therefore, the mutual exclusion property holds.

Answer to Exercise 2.9

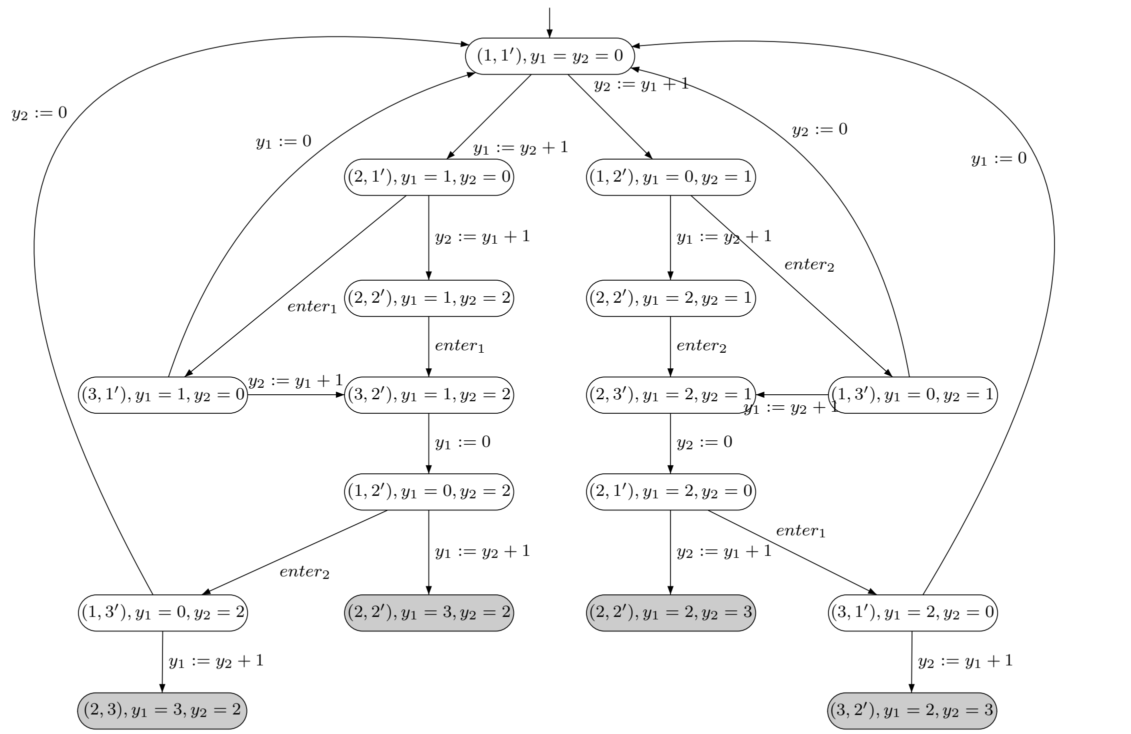
(a) The program graphs of the two processes can be depicted as follows:



(b) First, we provide the program graph $PG_1 \| PG_2$:



The transition system $TS(PG_1|||PG_2)$ for $y_1 \leqslant 2$ and $y_2 \leqslant 2$ becomes:



(c) To show that the complete transition system is infinite, we consider the infinite execution:

$$
\begin{aligned}
\langle 1, 1', y_1 = y_2 = 0\rangle \quad & \xrightarrow{y_1:=y_2+1} \langle 2, 1', y_1 = 1, y_2 = 0\rangle \\
& \xrightarrow{y_2:=y_1+1} \langle 2, 2', y_1 = 1, y_2 = 2\rangle \\
& \xrightarrow{enter_1} \langle 3, 2', y_1 = 1, y_2 = 2\rangle \\
& \xrightarrow{y_1:=0} \langle 1, 2', y_1 = 0, y_2 = 2\rangle \\
& \xrightarrow{y_1:=y_2+1} \langle 2, 2', y_1 = 3, y_2 = 2\rangle \\
& \xrightarrow{enter_2} \langle 2, 3', y_1 = 3, y_2 = 2\rangle \\
& \xrightarrow{y_2:=0} \langle 2, 1', y_1 = 3, y_2 = 0\rangle \\
& \xrightarrow{y_2:=y_1+1} \langle 2, 2', y_1 = 3, y_2 = 4\rangle \\
& \xrightarrow{enter_1} \langle 3, 2', y_1 = 3, y_2 = 4\rangle \\
& \ldots
\end{aligned}
$$

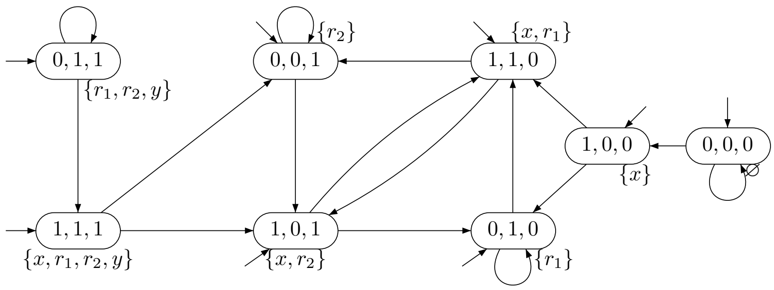
Answer to Exercise 2.11

(a) The output and the circuit control functions for C_1 are as follows:

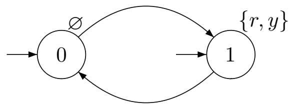
$$
\begin{aligned}
\lambda_y &= r_1 \wedge r_2 \\
\delta_{r_1} &= (x \wedge \neg r_1) \vee (\neg x \wedge r_1) = x \oplus r_1 \\
\delta_{r_2} &= (\neg x \wedge r_2) \vee (x \wedge r_1)
\end{aligned}
$$

The transition system is given by $TS(C_1) = TS_1 = (Eval(\{x, r_1, r_2\}), Act, \rightarrow, I, AP, L)$ where

– $I = \{ (x, r_1, r_2) \mid x, r_1, r_2 \in \mathbb{B} \} = Eval(\{ x, r_1, r_2 \})$
– $AP = \{ x, r_1, r_2, y \}$
– $L : S \to 2^{AP}$ as given in the figure below where evaluations are represented as triples (x, r_1, r_2).



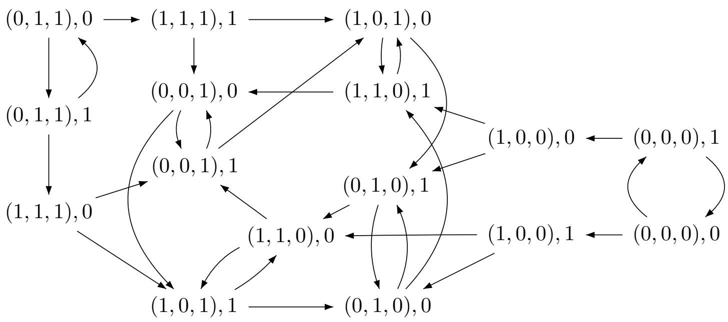
(b) The transition system representation TS_2 of the circuit C_2 is given by:



The synchronous composition of TS_1 and TS_2 is defined as the transition system

$$TS_1 \otimes TS_2 = (S_1 \times S_2, Act, \to, I_1 \times I_2, AP_1 \uplus AP_2, L)$$

It is given by the diagram below. Note that the set of initial states in this case equals the set of states $S_1 \times S_2$, i.e., every state can serve as an initial state. Therefore we do not indicate the initial states. We also omit the atomic propositions. These can be defined analogously to part (a) by renaming the variables of C_2 to r_1' and y', respectively.



Answer to Exercise 2.12

(a) The program graph for process P_i is given by $PG_i = (Loc_i, Act_i, Effect_i, \hookrightarrow_i, Loc_0^i, g_0^i)$ over the set of variables $Var_i = \{ id_i, m_i \}$. To interconnect the processes, we define the channels as

$$Chan := \{c_{ij} \mid i \in \{1, \ldots, n\}, j = (i+1) \bmod n\}$$

where each channel has capacity n, i.e., $cap(c) := n \quad \forall c \in Chan$. The domain of the channel is defined by $dom(c) := \{1, \ldots, n\} \quad \forall c \in Chan$.

The program graph for process P_i is given by:

$$\begin{array}{lcl} Loc_i & := & \{start_i, recv_i, test_i, stop_i\} \\ Act_i & := & \{noop_i\} \\ Effect_i(noop_i, \eta) & := & \eta \\ Loc_0^i & := & \{ start_i \} \\ g_0^i & := & id_i = id \wedge m = 0 \end{array}$$

The transition relation is given as follows:

$$\begin{array}{l} start_i \xrightarrow{c_{i,i+1}!id_i} recv_i \\ recv_i \xrightarrow{c_{i-1,i}?m_i} test_i \\ test_i \xrightarrow{m_i=id_i:noop_i} stop_i \\ test_i \xrightarrow{m_i>id_i:c_{i,i+1}!m_i} recv_i \\ test_i \xrightarrow{m_i<id_i:noop_i} recv_i \end{array}$$

(b) The transition system for $CS = (P_1|P_2|\cdots|P_n)$ is given by:

$$TS(CS) = (S, Act, \rightarrow, I, AP, L) \quad \text{where}$$

- $S := Loc_1 \times \ldots \times Loc_n \times Eval\left(\bigcup_{i=1}^n Var_i\right) \times Eval(Chan)$
- $Act := \{noop_i \mid i \in \{1, \ldots, n\}\} \cup \{\tau\}$
- $\rightarrow \subseteq S \times Act \times S$
- $I := \{(start_1, \ldots, start_n, \eta, \xi_0) \mid \bigwedge_{i=1}^n \eta \models g_{0,i}\}$ where $\xi_0 : Chan \rightarrow dom(c)^*$ denotes the initial channel evaluation
- $AP := \{1, \ldots, n\} \cup \bigcup_{i=1}^n Loc_i$
- $L : S \rightarrow 2^{AP} : (l_1, \ldots, l_n, \eta, \xi) \mapsto \{l_1, \ldots, l_n\} \cup \{id_i = n \mid n \in \mathbb{N}\}$

An initial execution fragment is:

$$\begin{array}{ll} & (st, st, st, \overbrace{(id_1/1, id_2/2, id_3/3, m_1/0, m_2/0, m_3/0)}^{\eta}, c_{1,2} = c_{2,3} = c_{3,1} = \varepsilon) \\ \xrightarrow{\tau} & (rc, st, st, \eta, c_{1,2} = 1, c_{2,3} = c_{3,1} = \varepsilon) \\ \xrightarrow{\tau} & (rc, st, rc, \eta, c_{1,2} = 1, c_{2,3} = \varepsilon, c_{3,1} = 3) \\ \xrightarrow{\tau} & (test, st, rc, \eta\,(m_1/3)\,, c_{1,2} = 1, c_{2,3} = c_{3,1} = \varepsilon) \\ \xrightarrow{\tau} & (rc, st, rc, \eta\,(m_1/3)\,, c_{1,2} = 13, c_{2,3} = c_{3,1} = \varepsilon) \\ \ldots & \ldots \end{array}$$