EE4-57

SOLUTIONS: DISCRETE EVENT SYSTEMS
MASTER IN CONTROL

1. Exercise

a) We flag initially (with a plus) all pair of states that are terminal and non-terminal. This is done according to the following table:

01	+								
02	+								
03	+								
10	+								
11	+								
12	+								
20	+								
21	+								
30	+								
	00	01	02	03	10	11	12	20	21

b) Then, all state pairs that have different sets of enabled events are flagged with an x. This is done according to the following table:

01	+								
02	+								
03	+	x	x						
10	+			x					
11	+			x					
12	+	x	x		x	x			
20	+			x			x		
21	+	x	x		x	x		x	
30	+	x	x		x	x		x	
	00	01	02	03	10	11	12	20	21

c) We flag next with an o those pair of states such that arrival events lead to flagged pair of states. This is done according to the following table:

01	+								
02	+	o							
03	+	x	x						
10	+		o	x					
11	+	o		x	o				
12	+	x	x		x	x			
20	+	o		x	o		x		
21	+	x	x		x	x		x	
30	+	x	x		x	x		x	
	00	01	02	03	10	11	12	20	21

Flagging stops after the first iteration.

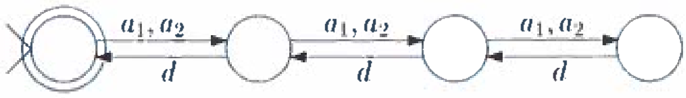

Figure 1.1 Transition diagram of reduced automaton

d) Checking the set of unflagged nodes from the table one can build the following equivalence classes.

$$\{00\}, \{01, 10\}, \{02, 11, 20\}, \{03, 12, 21, 30\}.$$

Notice that d events (which are cause of non-determinism) always map nodes within one equivalence class to nodes within another equivalence class. In this respect, flagging on the ground of d events will not identify any additional non-equivalent states. Therefore all unflagged nodes are equivalent in the sense defined before.

e) A reduced equivalent automaton (which happens to be deterministic) can be found by collapsing each equivalence class into a single state. Its transition diagram is shown in Fig. 1.1.

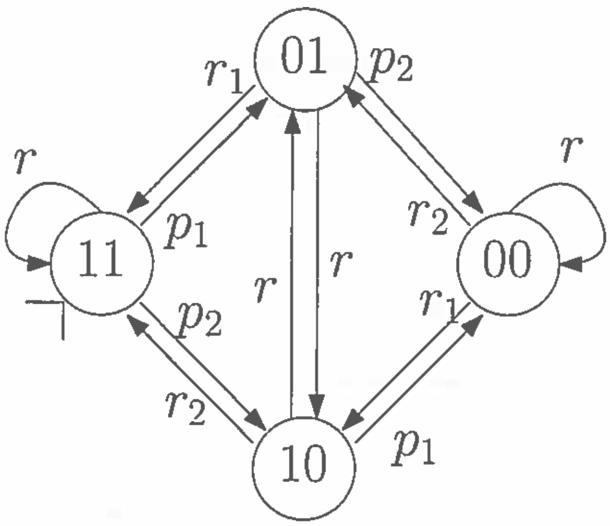

Figure 2.1 Transition diagram of automaton G

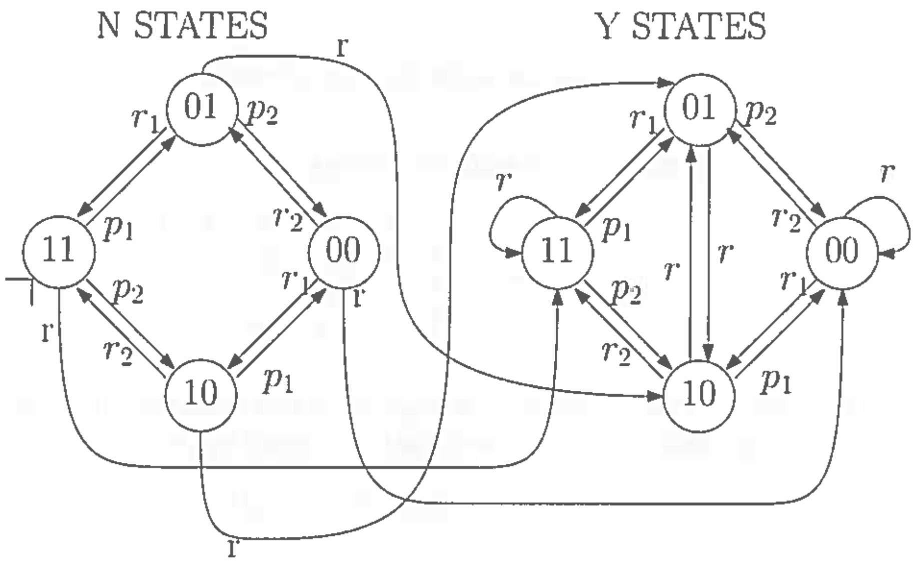

Figure 2.2 Parallel composition of labelling automaton and G

2. Exercise

a) The automaton operates with events $E = \{p_1, p_2, p_3, r_1, r_2, r_3, r\}$. We adopt a state space $X = \{000, 001, 010, 011, 100, 101, 110, 111\}$ where a 0 in position i denotes an empty slot and a 1 in position i denotes a full slot. In particular, 111 is the initial state. The automaton G, modeling the rotating platform can be provided as in Fig. 2.1.

b) The labelling automaton as usual has two states N and Y, a single event r, and two transitions, $f_L(N,r) = Y$ and $f_L(Y,r) = Y$.Initial state is N. Accordingly parallel composition of G and G_L yields the automaton in Fig. 2.2. The next step is to replace r events with ε events, and then apply the algorithm for the computation of the Observer automaton to the resulting nondeterministic automaton. In particular, the initial state will be the ε-Reach of $11N$, and is therefore the set, $\{11N, 11Y\}$. From there, exploration of all accessible states leads to the Observer automaton shown in Fig. 2.3

3. Exercise

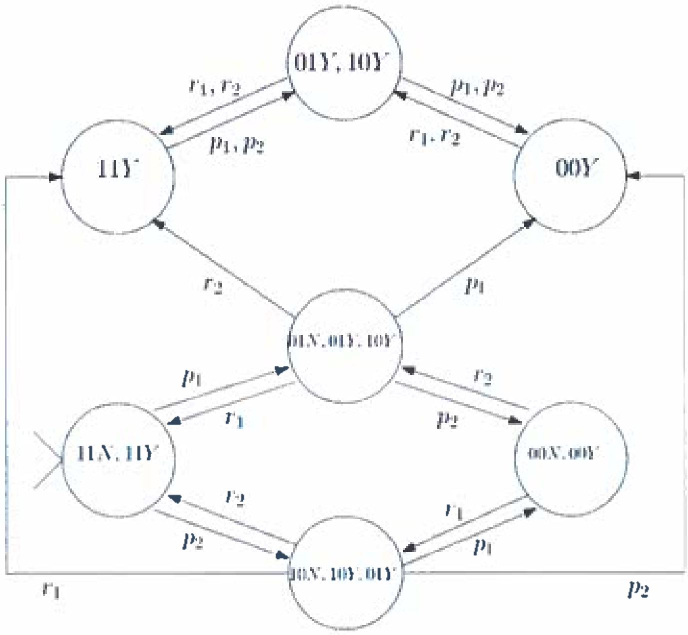

Figure 2.3 Diagnoser for detection of r events

a) The incidence matrix C is computed as follows:

$$C = Post - Pre = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

b) The minimal P-semiflows are non-negative vectors that annihilate C from the left. In particular there are 3 P-semiflows of minimal support:

$$[1,0,0,0], [0,1,0,0], [0,0,1,1].$$

c) The network is structurally bounded as $[1,1,1,1]$ is a P-semiflow of support equal to the set of all places.

d) From the initial marking $[1,1,1,1]'$ two additional markings can be reached: $[1,1,0,2]'$ and $[1,1,2,0]'$. These are reached with transitions t_1 (or t_4) and t_2 (or t_3) respectively. Conversely, from the marking $[1,1,0,2]'$, transitions t_2 and t_3 bring back to the initial marking $[1,1,1,1]'$. Similarly, from the marking $[1,1,2,0]'$, transitions t_1 and t_4 bring back to the initial marking $[1,1,1,1]'$.

e) Notice that, from each reachable state it is possible to get back to the initial state, hence the Petri Net is reversible.

4. Exercise

a) The transition diagrams of Automata G_A and G_B are shown in Fig. 4.1.

b) The parallel composition $G = G_A || G_B$ is shown in Fig. 4.2.

c) The specification can be modelled with the automaton G_{spec} in Fig. 4.3. This allows to model the situation in which only a single tool 1 is available for the 2 robots. Notice that we may afford to allow apparently meaningless sequences such as p_{1a}, r_{1b} or $p_{1b}r_{1a}$, as these are not allowed by the open loop language, anyway. The specification avoids that tool 1 be picked twice without it being released first.

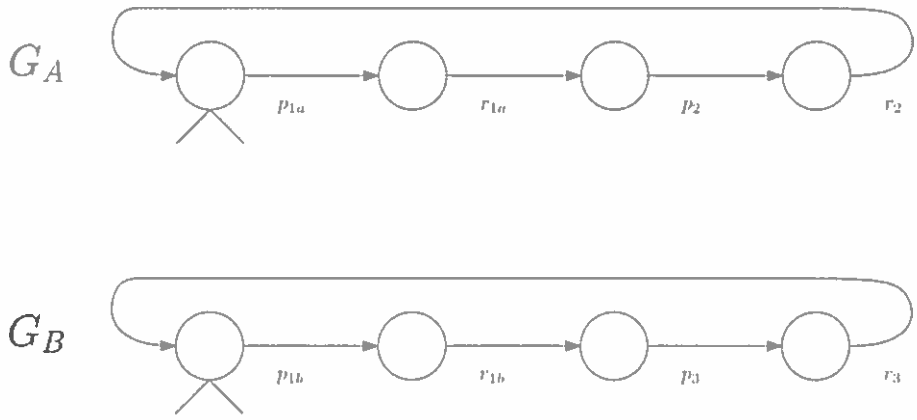

Figure 4.1 Automata: G_A and G_B

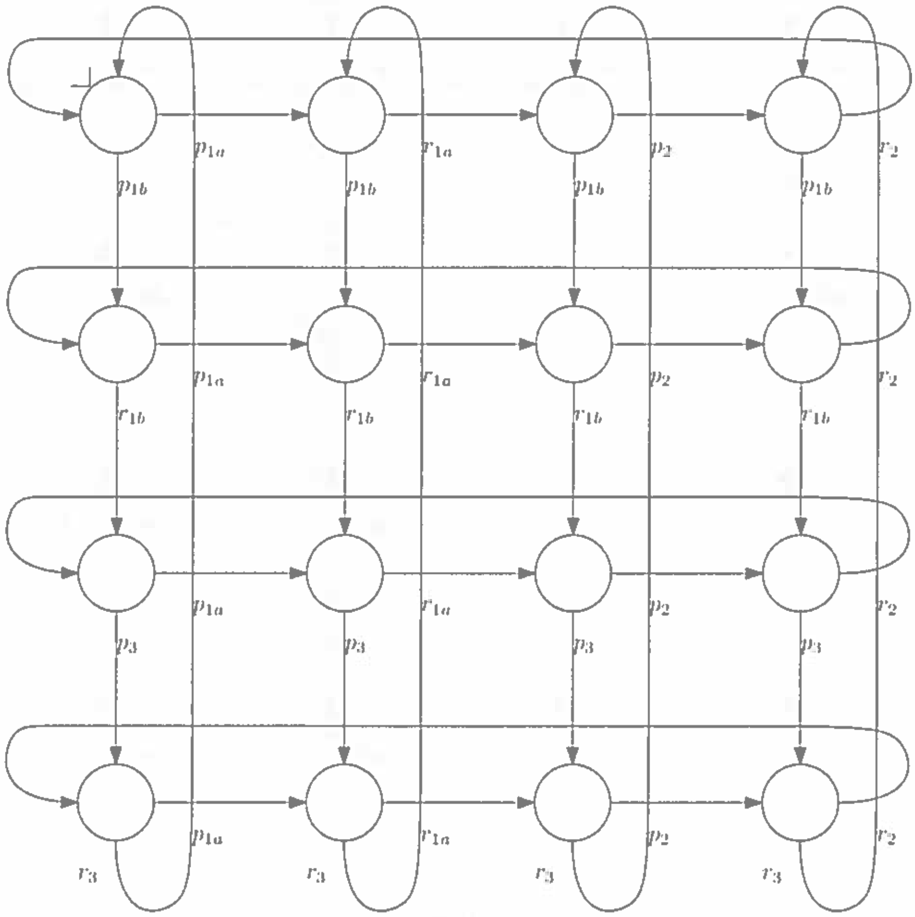

Figure 4.2 Automaton $G = G_A || G_B$

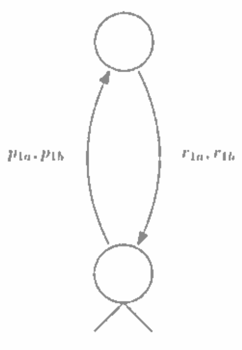

Figure 4.3 Specification automaton G_{spec}

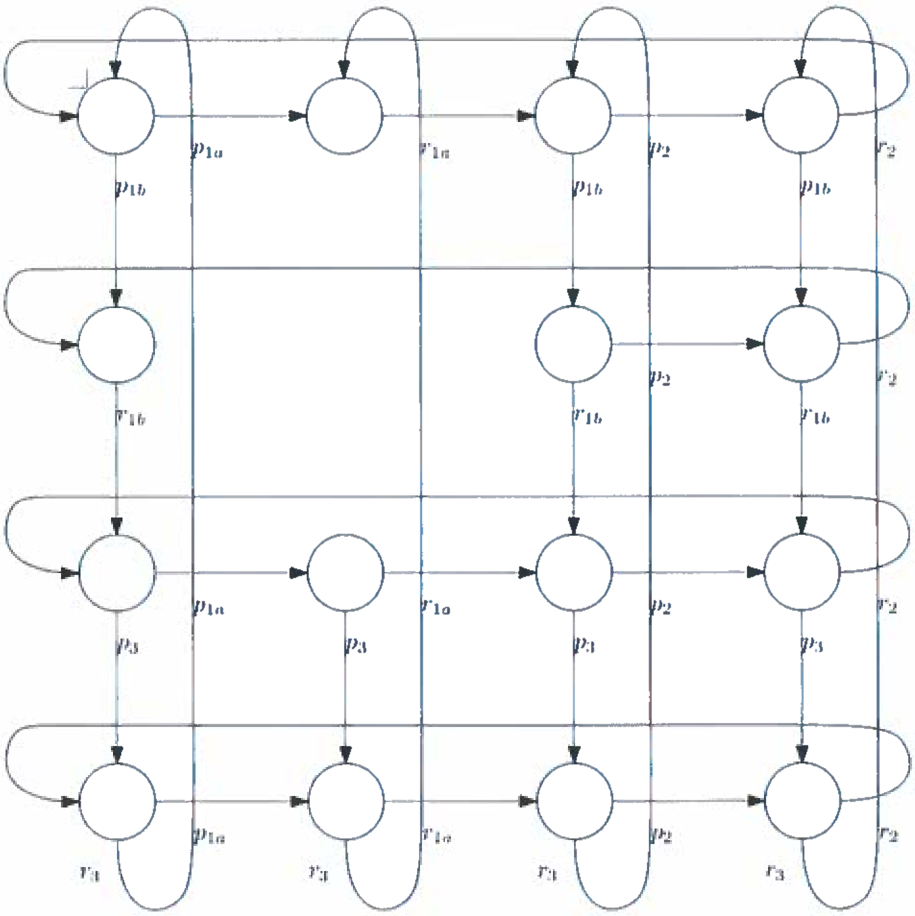

Figure 4.4 Parallel composition $G||G_{spec}$

d) The parallel composition of the automaton G with G_{spec} is shown in Fig. 4.4

e) Concerning controllability it can be seen that the only disabled events are p_{1a} and p_{1b}, which are controllable. Hence, the specification is controllable.