

MSc and EEE/ISE PART IV: MEng and ACGI

© Imperial College London

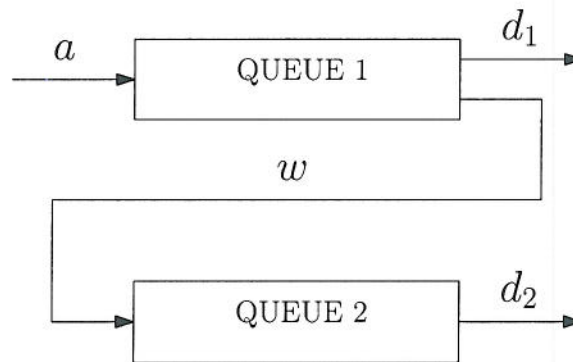


Figure 1.1 Office structure

1. An office is organized with two queues. All customers arriving need to wait at the first queue (event  $a$ ); some of them are served and leave the office (event  $d_1$ ), others need to wait also at the second queue (event  $w$ ). Then, after being served, they are free to leave (event  $d_2$ ). This is schematically summarized in the Figure above.
  - a) Model Queue 1 and Queue 2 (separately) by means of finite deterministic automata, assuming a maximum of three people waiting for each queue. [ 3 ]
  - b) Model the office by means of a single finite deterministic automaton. [ 4 ]
  - c) A sensor is placed at the entrance to detect arrivals (event  $a$ ) and departures (event  $d$ ); the sensor is not able to discriminate between  $d_1$  and  $d_2$  events. Design a deterministic automaton  $G_o$ , which acts as an observer to estimate the current number of customers in the queues by processing only  $d$  and  $a$  events. [ 8 ]
  - d) Let  $p \prec w$  denote that “ $p$  is a prefix of  $w$ ” and  $|w|_e$  be the number of occurrences of  $e$  in the word  $w$ . Write, in set-theoretic terms  $\mathcal{L}(G_o)$ , the language generated by the observer automaton. [ 5 ]

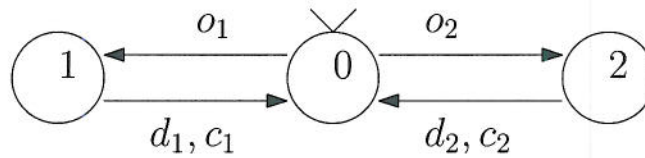


Figure 2.1 Specification automaton  $G_a$

2. Two users  $U_1$  and  $U_2$  can access records in a database. Each user may issue a request to ‘open a record’ (event  $o_i, i \in \{1, 2\}$ ) after which it simply waits for a ‘denied access’ event  $d_i$  or a ‘granted access’ event  $g_i$ . When editing of the record is complete the user may issue a ‘close record’ notification, event  $c_i$ . Assume for the sake of simplicity a single record in the database.

- Model the behaviour of  $U_i$  by means of a finite deterministic automaton. [ 3 ]
- Compute the finite deterministic automaton  $G$ , which models the two users acting simultaneously on the database. [ 3 ]
- Consider the automaton  $G_a = \{X_a, E_a, f_A, 0, \Gamma_A\}$ , with

$$X_a = \{0, 1, 2\}, \quad E_a = \{o_1, c_1, d_1, o_2, c_2, d_2\},$$

and transition diagram shown in Fig. 2.1. A supervisor  $S$  is defined as follows:

$$S(s) = \{g_1, g_2\} \cup \Gamma_A(f_A(0, s)).$$

Compute the automaton that generates the closed-loop language  $\mathcal{L}(S/G)$ .

[ 5 ]

- Assuming the set of uncontrollable events to be  $E_{uc} = \{o_1, o_2, c_1, c_2\}$  is  $S(\cdot)$  an admissible supervisor? Is the language  $\mathcal{L}(G_a)$  controllable? [ 3 ]
- Build an automaton for the following specification: access to the database can only be granted to a single user at a time. Let  $K$  denote its generated language. [ 3 ]
- Is  $K$  controllable with respect to  $\mathcal{L}(G)$  and  $E_{uc}$ ? [ 3 ]

3. In a small store three types of boxes are collected. Assume a maximum capacity of five units (for instance five square feet) and assume that box  $i$ ,  $i = 1, 2, 3$  occupies  $i$  units.
- Model the store with a marked Petri Net  $(N, M_0)$  with transitions  $t_{a1}, t_{a2}, t_{a3}$  representing arrivals of boxes of type 1, 2 and 3 respectively and transitions  $t_{d1}, t_{d2}$  and  $t_{d3}$  representing departures of boxes of type 1, 2 and 3. What is the physical meaning of the places you used to model the system? [ 5 ]
  - Compute the  $P$ -semiflows of the network and provide an interpretation of the result obtained. [ 3 ]
  - Compute the reachable set  $\mathcal{R}(M_0)$  and the associated transition diagram. [ 2 ]
  - Discuss the following properties (justify your answers):
    - Reversibility
    - Liveness
    - Boundedness [ 4 ]
  - Modify the network in order to model the fact that boxes of type 1 and 3 are always dispatched together as they are complementary components of the same product. [ 2 ]
  - How does this affect the Liveness and Reversibility properties of the net for the same initial marking  $M_0$ ? [ 4 ]

4. Consider a 3-bits shift register. Each time a new bit comes in, the content of the 3-bits is shifted towards the left and the new bit is stored in the rightmost flip-flop.
- a) Model the register as a finite deterministic automaton  $G_1$ , with event set  $E = \{0, 1\}$  and 8 states corresponding to all possible configurations of flip-flops. [ 4 ]
  - b) Modify the previous model by adding faulty transitions  $f$  between each pair of states that differ by a single bit (automaton  $G_2$ ). [ 3 ]
  - c) Modify the previous model assuming that in the string of transmitted bits a parity check event  $p$  may occur (leaving the flip-flop states unchanged) at any state for which the last three bits transmitted contain an even number of 1s (automaton  $G_3$ ). [ 3 ]
  - d) Assume next that  $f$  events are unobservable. Design a diagnoser  $G_D$  to detect occurrence of  $f$  events, (Hint: what is the  $\varepsilon$ -reach from state  $x$  once  $f$  events have been replaced by  $\varepsilon$ ?). [ 10 ]

5. Mark has two options to go to work. One is to take a single bus, line A, or two buses one after the other, namely line B and C. When changing from bus B to C, bus A can no longer be taken as it follows a different route.
- a) Model the above scenario by means of a suitable timed discrete event model, assuming:
    - i) The trip time is negligible;
    - ii) Arrival times of bus A are exponentially distributed with a rate of 1;
    - iii) Arrival times of bus B and C are exponentially distributed with a rate of  $\lambda$ ;
    - iv) Mark always takes the first bus that gets to his bus stop. [ 4 ]
  - b) Compute the average time it takes for Mark to reach his workplace (as a function of  $\lambda$ ). [ 4 ]
  - c) What is the probability that Mark will take bus A to go to work? What is the probability that he will take bus B and C instead? [ 4 ]
  - d) Use the above results to compute the average time to get to the workplace assuming trip time is constant and no longer negligible: bus A 3 time units, bus B 1 time unit and bus C 1 time unit. [ 3 ]
  - e) Since the parameter  $\lambda$  is unknown to Mark (say traffic dependent), he has decided to adopt the following policy: if bus B will come first and within 1 time unit he will take it; if not he will take bus A no matter which one comes first. Can you model the above scenario by means of a Markov chain? Explain why. (Hint: can you use non-homogeneous Markov chains?) [ 2 ]
  - f) Explain how to compute the average travel time assuming Mark adopts the policy described above (and again zero trip time). [ 3 ]



6. A gambling machine works as follows: a player joins the game with 2 tokens. Each time he wins, which happens with probability  $p$ , he doubles the number of tokens, each time he loses he sees the number of tokens divided by 2. If he only has one token and loses again, then he ends up with no tokens. If he has 4 tokens and wins then he leaves the game with 8 tokens and can no longer continue to play (unless he is willing to put 2 tokens in and start all over). Winning and losing events are statistically independent random variables.
- a) Model the machine by means of a discrete time Markov chain; [ 4 ]
  - b) What is the probability of winning two consecutive times ? What is the probability of losing two consecutive times ? [ 2 ]
  - c) Sampling the amount of tokens every two gambles a new stochastic process is obtained. Model this process by a reduced order Markov chain. [ 4 ]
  - d) Use the reduced model to compute the probability of a player winning 8 tokens; [ 4 ]
  - e) What is a fair value of  $p$  so that in average the player has zero losses (and zero profit); [ 2 ]
  - f) Find the average number of times a player gambles before it either wins or loses. [ 4 ]

# SOLUTIONS: DISCRETE EVENT SYSTEMS

## MASTER IN CONTROL

292

### 1. Exercise

- The first queue can be modeled with a finite deterministic automaton  $G_1$  of alphabet  $E_1 = \{a, d_1, w\}$ . The second queue can be modeled with a finite deterministic automaton  $G_2$  of alphabet  $E_2 = \{w, d_2\}$ . See Fig. 1.1 for their transition diagram.
- The office can be modeled by taking the parallel composition  $G = G_1 || G_2$ , as shown in Fig. 1.2
- In order to design the observer we first define a non-deterministic automaton  $G_{ND}$  with alphabet  $E = \{a, d, \varepsilon\}$  obtained from the automaton  $G$  by replacing  $w$  events with the empty string  $\varepsilon$  and by collapsing events  $d_1$  and  $d_2$  into a single  $d$  event. See Fig. 1.3. The observer automaton  $G_o$  can be designed by finding an equivalent deterministic automaton. See Fig. 1.4.
- The observer automaton is simply a queue with maximum length equal to 6. Hence the language generated by the automaton can be expressed as:

$$\mathcal{L}(G_o) = \{w \in \{a, d\}^* : \forall v \prec w, |v|_d \leq |v|_a \leq |v|_d + 6\}$$

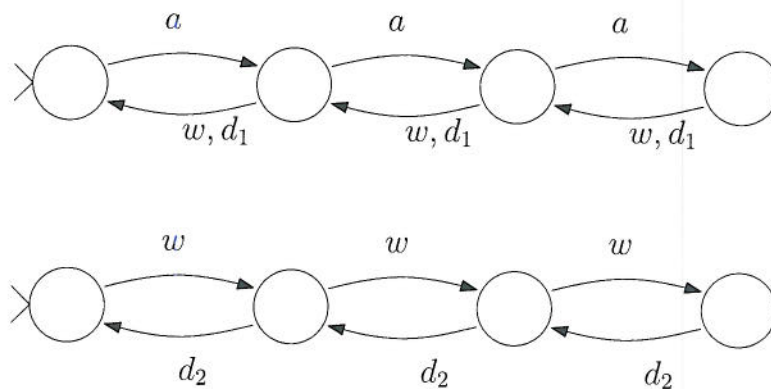


Figure 1.1 Finite Deterministic Automata  $G_1$  and  $G_2$



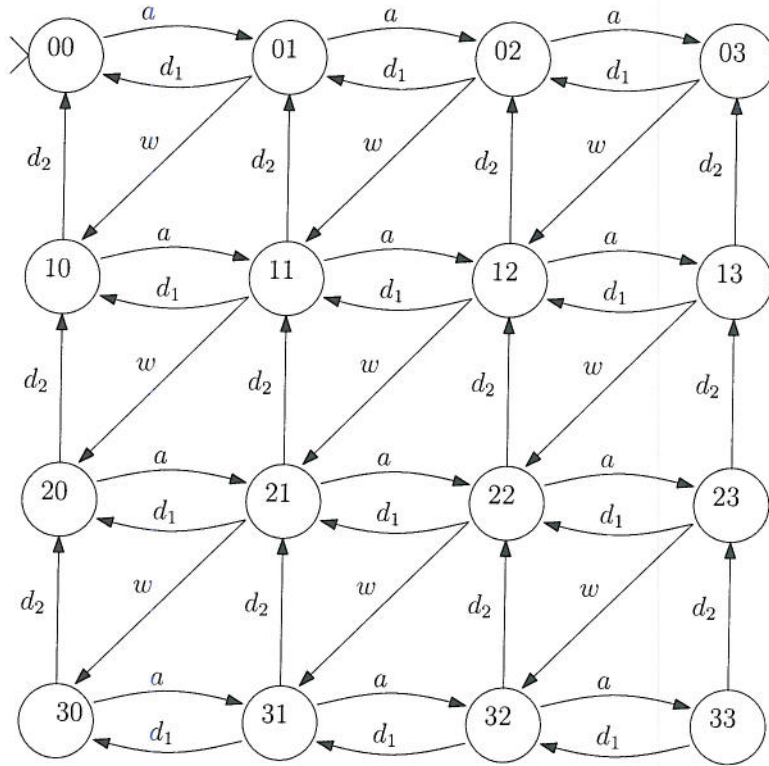


Figure 1.2 Parallel composition  $G_1 || G_2$

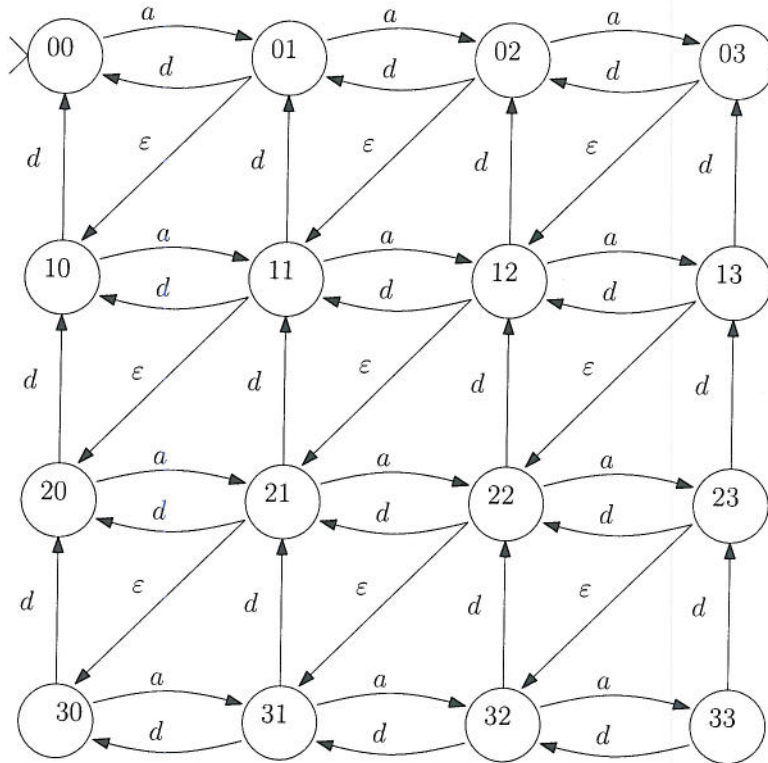


Figure 1.3 Finite non-deterministic automaton  $G_{ND}$

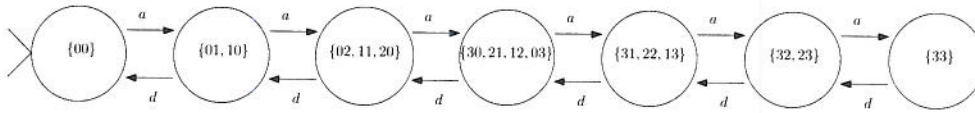


Figure 1.4 Observer automaton

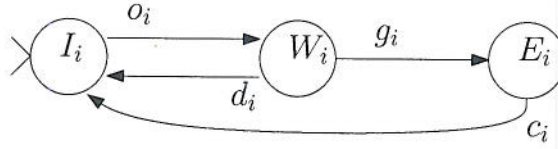


Figure 2.1 Transition diagram of  $G_i$

## 2. Exercise

- Each user can be modeled by an automaton  $G_i = \{X_i, E_i, f_i, I_i\}$  where  $X_i = \{I_i, W_i, E_i\}$ , where  $I$  stands for Idle,  $W$  for Waiting and  $E$  for Editing. The transition diagram of  $G_i$ ,  $i = \{1, 2\}$  is shown in Fig. 2.1
- The two users acting in parallel can be modeled by computing  $G = G_1 || G_2$ . The transition diagram of the resulting automaton is shown in Fig. 2.2.
- Clearly  $\mathcal{L}(S/G) = \mathcal{L}(G || G_a)$ . The transition diagram of automaton  $G || G_a$  is shown in Fig. 2.3.
- As emphasized in Fig. 2.3, the supervisor  $S$  is disabling event  $o_1$  in states IW2 and IE2, and event  $o_2$  in states W11 and EI1. As such events are uncontrollable the supervisor  $S(\cdot)$  is not admissible. In fact, the specification  $\mathcal{L}(G_a)$  is uncontrollable. Notice that  $o_1 o_2 \in \mathcal{L}(G_a) E_{uc} \cap \mathcal{L}(G)$  but it is not a legal string thus violating controllability.
- The specification is a 'forbidden state' type of specification. Its associated automaton can simply be obtained by removing state EE from the automaton  $G$ , (see Fig. 2.4).
- As emphasized in Fig. 2.4, the specification automaton disables events  $g_1$  in state WE and  $g_2$  in state EW, both of which are controllable; hence the specification is controllable.

## 3. Exercise

- We may model the store by having a place  $p_4$  representing the number of free space units; then tokens in places  $p_1$ ,  $p_2$  and  $p_3$  could represent the number of boxes of types 1, 2 and 3 respectively. Accordingly the associated Petri Net is given as in Fig. 3.1. Notice the initial marking  $M_0 = [0, 0, 0, 6]$ .
- The incidence matrix  $C$  of the network is given by:

$$C = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & -2 & -3 & 1 & 2 & 3 \end{bmatrix}$$

Computing  $P$ -semiflows yields the following table:

Step 1:

$$\left[ \begin{array}{cccccc|cccc} 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & -2 & -3 & 1 & 2 & 3 & 0 & 0 & 0 & 1 \end{array} \right]$$

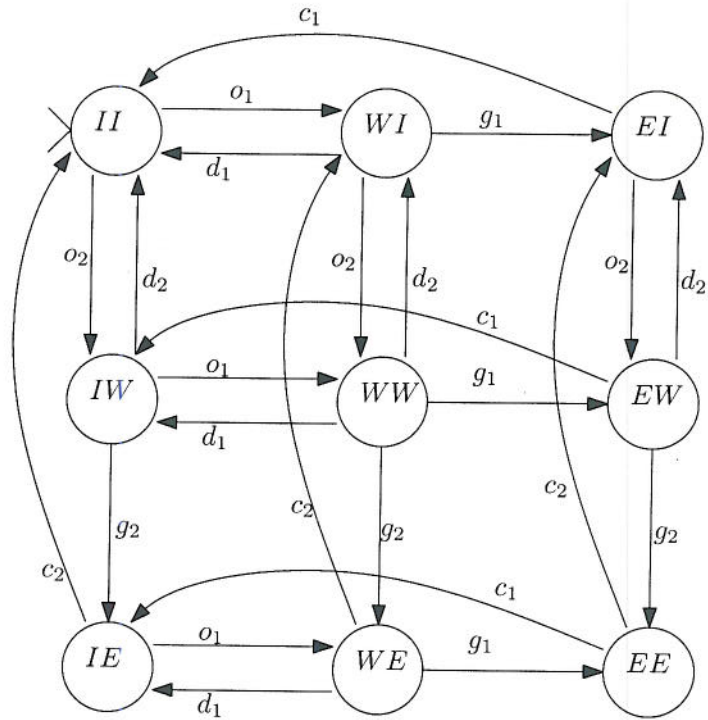


Figure 2.2 Transition diagram of  $G$

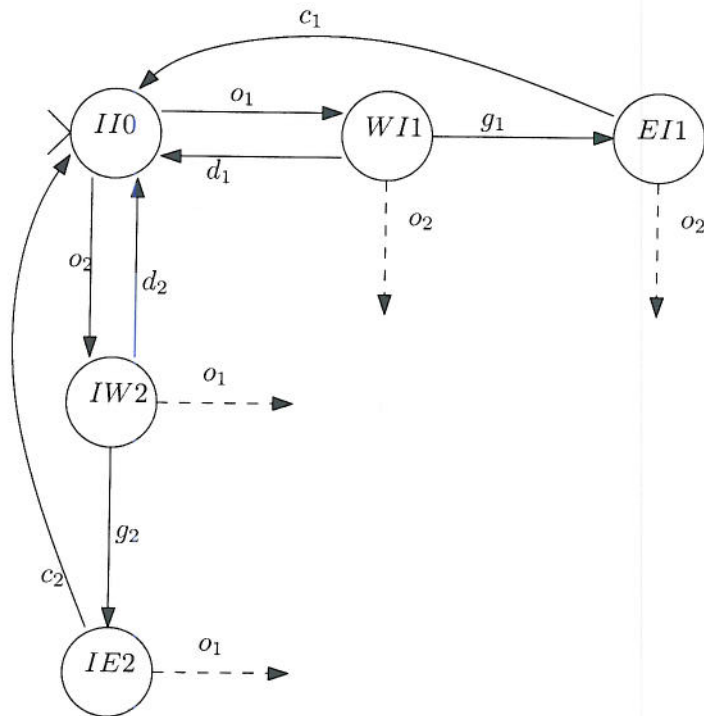


Figure 2.3 Automaton  $G \parallel G_a$

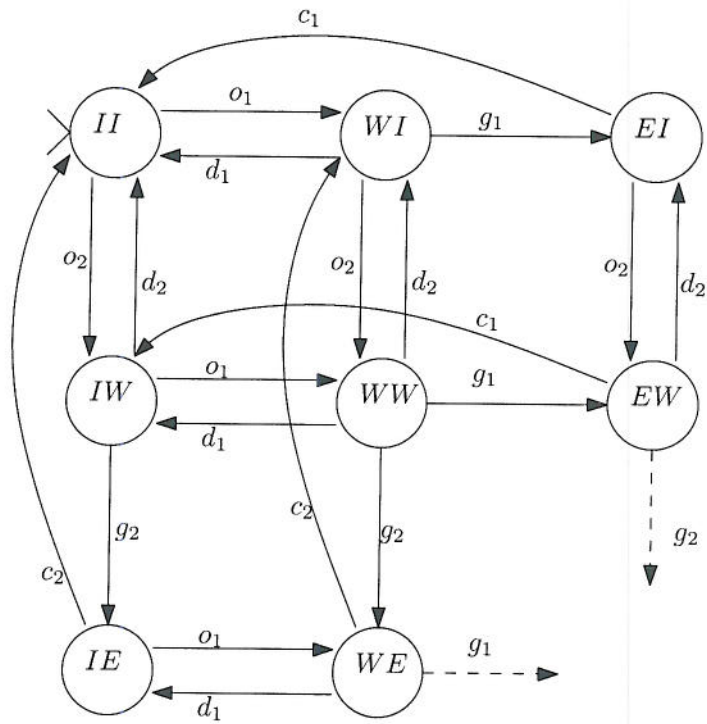


Figure 2.4 Forbidden state specification automaton

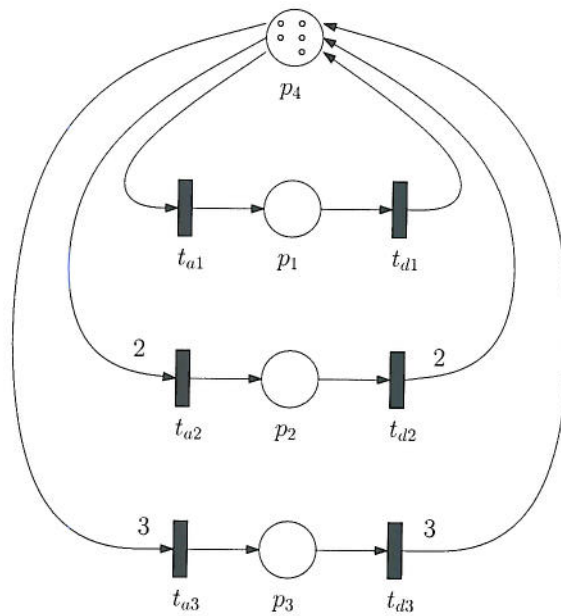


Figure 3.1 Petri Net modeling the store

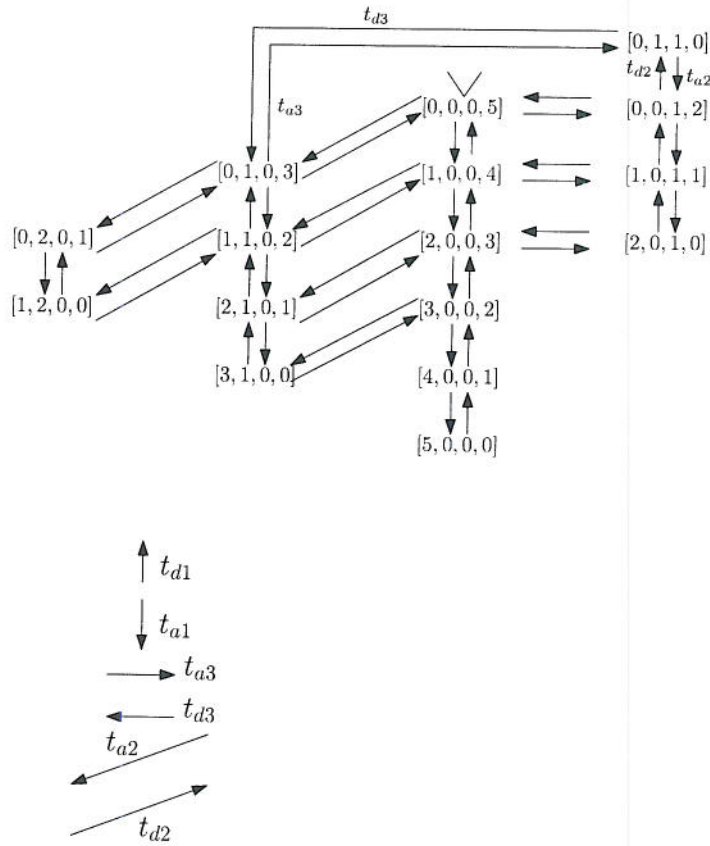


Figure 3.2 Reachable set and transition graph

Step 2:

$$\left[ \begin{array}{cccccc|cccc} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 0 & 2 & 3 & 1 & 0 & 0 & 1 \end{array} \right]$$

Step 3:

$$\left[ \begin{array}{cccccc|cccc} 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 0 & 0 & 3 & 1 & 2 & 0 & 1 \end{array} \right]$$

Step 4:

$$\left[ \begin{array}{cccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 \end{array} \right]$$

Hence  $[1, 2, 3, 1]$  is the unique  $P$ -semiflow of minimum support. This is consistent with intuition as  $M(p_1) + 2M(p_2) + 3M(p_3)$  represents the space occupied by the boxes in the store. This added up with  $M(p_4)$ , that is the currently available space in the store gives the total store capacity (which is clearly a constant).

- c) The reachable set can be computed exploring it starting from the initial marking  $M_0 = [0, 0, 0, 5]'$ . The resulting transition graph is represented in Fig 3.2. Notice that equally oriented arrows represent the same transitions, apart for a few anomalies which have been explicitly labeled.
- d) Notice that firing of transitions  $t_{a1}$ ,  $t_{a2}$  and  $t_{a3}$  leads to states where the 'inverse' transitions  $t_{d1}$ ,  $t_{d2}$  and  $t_{d3}$  are enabled (and viceversa). This clearly implies that the transition graph is strongly connected. It is also easily seen from its graphical representation. As a consequence, the Petri Net is *reversible*. Moreover, each transition  $t_{a1}$ ,  $t_{a2}$ ,  $t_{a3}$ ,  $t_{d1}$ ,  $t_{d2}$  and  $t_{d3}$  appears at least once in the transition



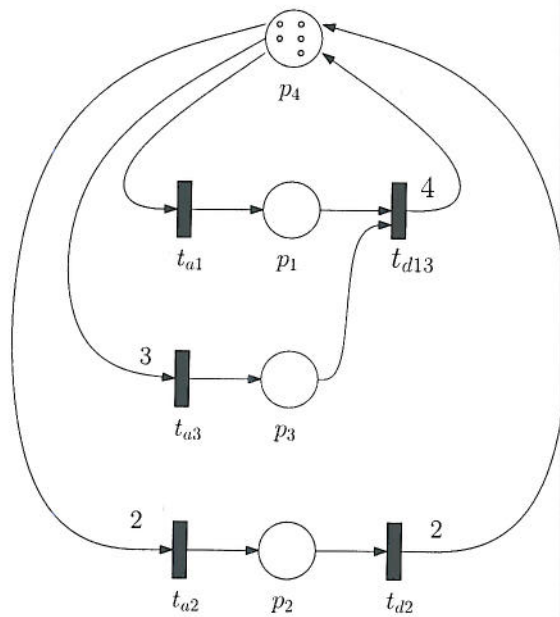


Figure 3.3 Petri Net modeling the store with synchronization

graph. This means that all transitions are live. Boundedness is easily seen as the reachable set is finite. Alternatively, existence of a P-semiflow of support equal to  $P$ , implies structural boundedness and therefore also boundedness for the initial marking  $M_0$ .

- e) The store with a synchronization constraint between events  $t_{d1}$  and  $t_{d3}$  can be modeled by the Petri Net in Fig. 3.3
- f) After the modification the reachable set is unchanged. However, its transition diagram is modified as in Fig. 3.4. Notice that there exists 2 states:  $[4, 0, 0, 1]$  and  $[5, 0, 0, 0]$  from which  $M_0$  cannot be reached. Hence the network is not reversible. In particular, because of state  $[5, 0, 0, 0]$  in which no transition is enabled, it follows that all transitions are  $L3$ -live but none of them is  $L4$ -live.

#### 4. Exercise

- a) The automaton  $G_1$  is shown in Fig. 4.1.
- b) Adding  $f$  events the automaton is modified as in Fig. 4.2.
- c) The automaton  $G_3$  is obtained adding self-loops with events  $p$  in nodes 000, 011, 101 and 110.
- d) To design a diagnoser we first do the concurrent composition between  $G_3$  and the automaton  $N \rightarrow Y$ , with transition occurring on event  $f$  and a self-loop in  $Y$  with event  $f$ . The resulting automaton is sketched in Fig. 4.3. Notice that the  $\varepsilon$ -reach of state  $x$  is equal to  $\{x, Y \text{ states}\}$ . Therefore the Diagnoser automaton is very similar to the actual  $G_3$  automaton, except for an additional state. See Fig. 4.4.

#### 5. Exercise

- a) We can model the scenario described by means of a continuous time homogeneous Markov chain with 3 states, as shown in Fig. 5.1. The equations



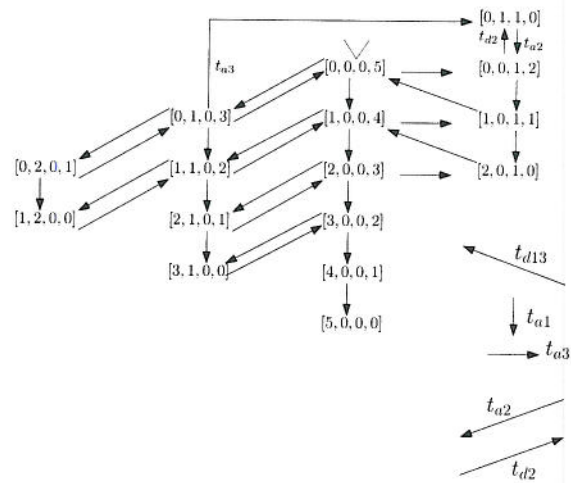


Figure 3.4 Transition diagram of the modified Petri Net

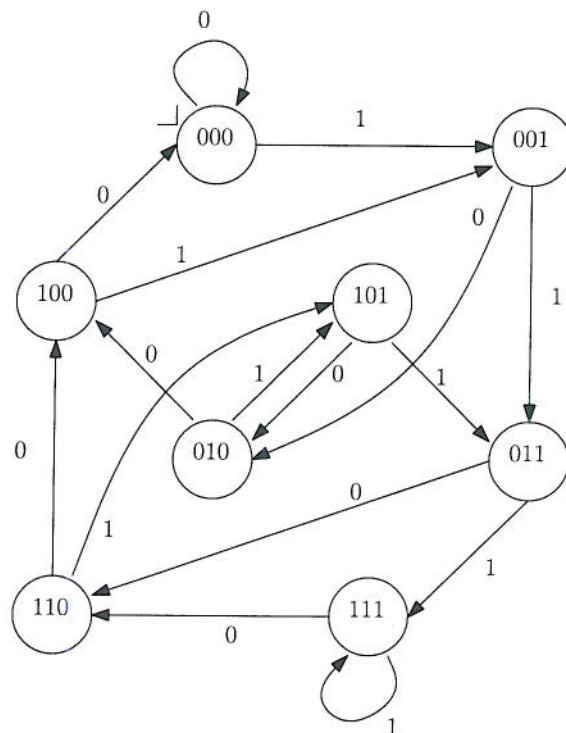


Figure 4.1 Automaton modeling a shift register

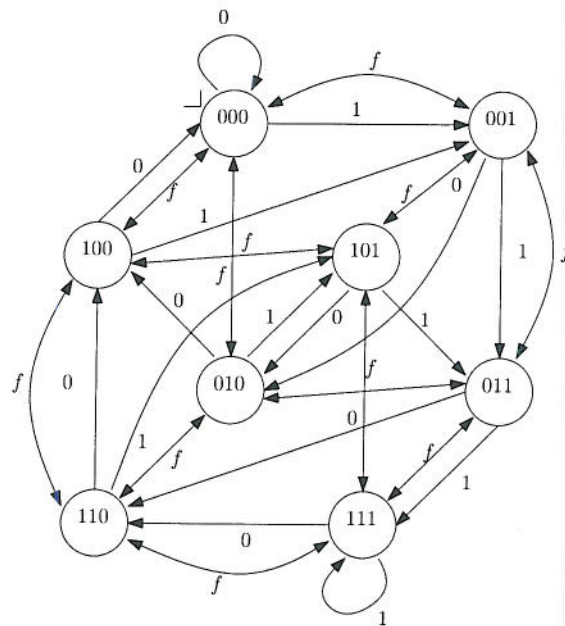


Figure 4.2 Automaton modeling a shift register with faulty transitions

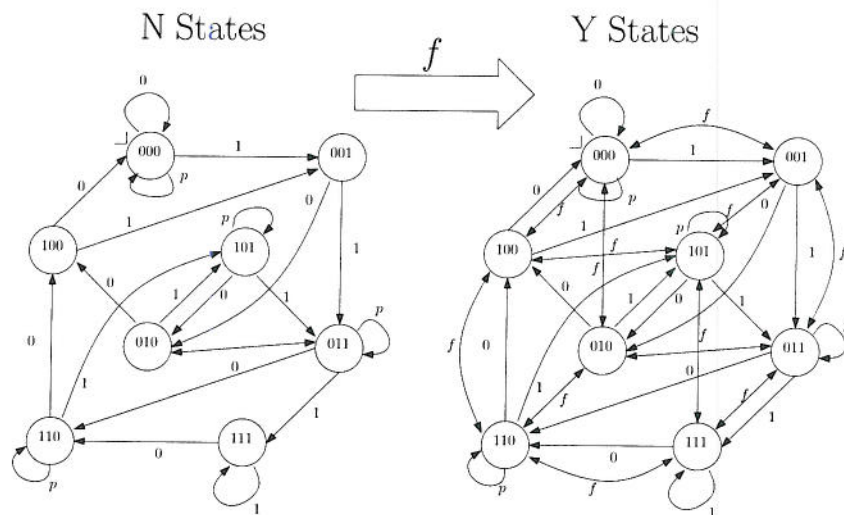


Figure 4.3 Concurrent composition of  $G_3$  and  $N \rightarrow Y$  automaton

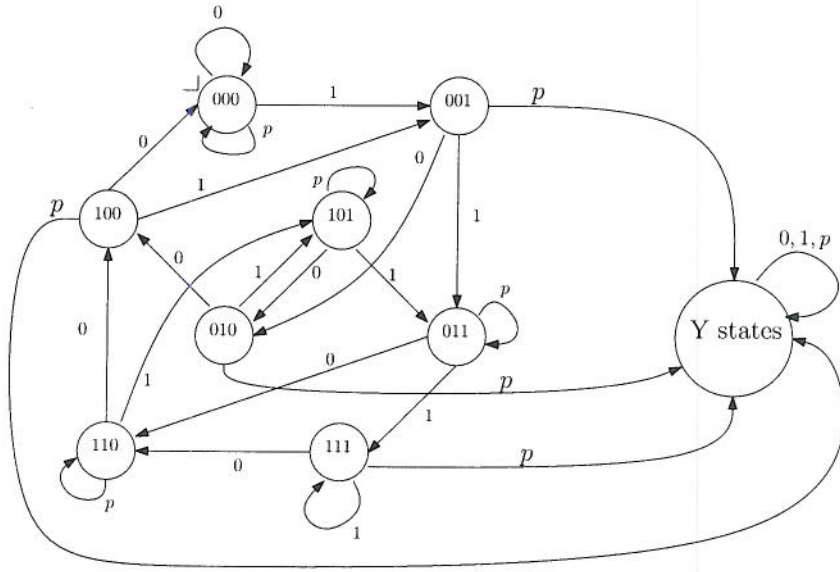


Figure 4.4 Diagnoser automaton  $G_D$

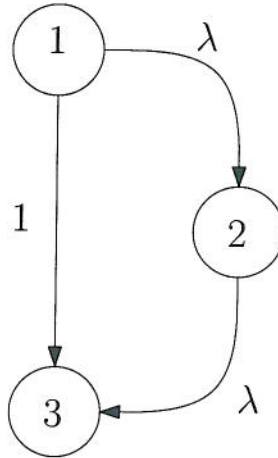


Figure 5.1 Markov chain modeling bus waiting times

associated to it are given by:

$$\dot{\pi} = \pi \begin{bmatrix} -\lambda - 1 & \lambda & 1 \\ 0 & -\lambda & \lambda \\ 0 & 0 & 0 \end{bmatrix}.$$

- b) In order to compute the average time to work we need to set  $\pi(0) = [1, 0, 0]$  and compute the average absorption time of state 3. Notice that in the Laplace domain:

$$\mathcal{L}[\dot{\pi}_3] = [1, 0, 0] \begin{bmatrix} s + \lambda + 1 & -\lambda \\ 0 & s + \lambda \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \lambda \end{bmatrix}.$$

In order to compute the average absorption time we may exploit the final value theorem:

$$\lim_{s \rightarrow 0} -\frac{d}{ds} \mathcal{L}[\dot{\pi}_3] = \lim_{s \rightarrow 0} -\frac{d}{ds} \frac{s + \lambda + \lambda^2}{(s + \lambda)(s + \lambda + 1)} = \frac{2}{\lambda + 1}.$$

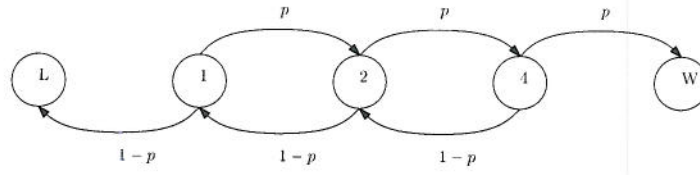


Figure 6.1 Transition diagram of the gambling machine

- c) The probability that Mark will take bus A or bus B & C are proportional to the transition rates out of state 1, namely 1 for bus A and  $\lambda$  for bus B. This gives the following probabilities:

$$p_A = \frac{1}{1+\lambda} \quad p_{BC} = \frac{\lambda}{1+\lambda};$$

- d) The average travel time if trip times are not negligible is simply given by the average absorption time plus the average trip time computed by taking into account  $p_A$  and  $p_{BC}$ , namely:

$$\frac{2}{1+\lambda} + \frac{3}{1+\lambda} + \frac{2\lambda}{1+\lambda} = \frac{5+2\lambda}{1+\lambda};$$

- e) We may define the function  $\lambda(t)$  as follows:

$$\lambda(t) = \begin{cases} \lambda & \text{if } t \in [0, 1] \\ 0 & \text{if } t \geq 1 \end{cases}$$

The described policy can be modeled by a non-homogeneous Markov chain of equations:

$$\dot{\pi} = \pi \begin{bmatrix} -\lambda(t) - 1 & \lambda(t) & 1 \\ 0 & -\lambda & \lambda \\ 0 & 0 & 0 \end{bmatrix}.$$

- f) Given the cascaded structure, the above time-varying systems of linear differential equations can be explicitly solved. Hence, the average time to reach the workplace  $T_a$  can be computing by integrating:

$$T_a = \int_0^{+\infty} t \dot{\pi}_3(t) dt.$$

## 6. Exercise

- a) The gambling machine can be described by means of a Markov chain with 5 states:  $\{L, 1, 2, 4, W\}$ , which stand for Loose, 1 token, 2 tokens, 4 tokens, Win. Its transition diagram is shown in Fig. 6.1. The matrix of one-step ahead transition probabilities is given by:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1-p & 0 & p & 0 & 0 \\ 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The equation describing the evolution of  $\pi$  is therefore  $\pi(t+1) = \pi(t)Q$ .

- b) The probability of winning two times is  $p^2$  as winning events are independent. Similarly the probability of loosing twice is  $(1-p)^2$ ;

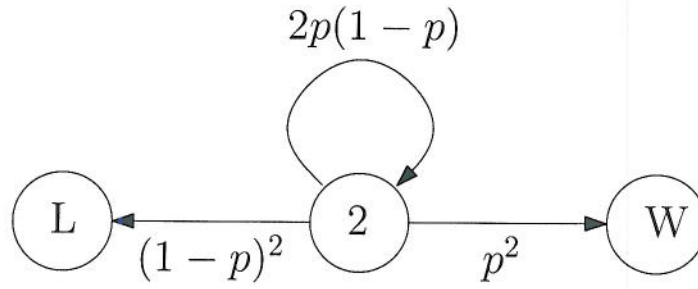


Figure 6.2 Transition diagram of the reduced model

- c) The reduced order model can be built considering the probabilities of winning twice or loosing twice. In this model only the states  $\{L, 2, W\}$  will be involved. The corresponding transition diagram is shown in Fig. 6.2. The matrix  $Q_{red}$  of one-step ahead transition probabilities for the reduced model is given by:

$$Q_{red} = \begin{bmatrix} 1 & 0 & 0 \\ (1-p)^2 & 2p(1-p) & p^2 \\ 0 & 0 & 1 \end{bmatrix}.$$

- d) From the reduced model we see that the probability  $P_W$  of winning 8 tokens is proportional to  $p^2$ , while that of loosing all tokens is proportional to  $(1-p)^2$ , hence:

$$P_W = \frac{p^2}{(1-p)^2 + p^2} \quad P_L = \frac{(1-p)^2}{(1-p)^2 + p^2}.$$

Alternatively one may compute the  $z$  transform of  $\pi_W(t)$ :

$$\pi_W(z) = z[0, 1, 0] [zI - Q_{red}]^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

From the final value theorem:

$$P_W = \lim_{t \rightarrow +\infty} \pi_W(t) = \lim_{z \rightarrow 1} (z-1) \pi_W(z).$$

- e) Notice that a player loosing by definition loses 2 tokens, while a player winning gains  $8 - 2 = 6$  tokens. Hence, for a probability of winning  $P_W = 1/4$  we have a probability of loosing equal to  $P_L = 3/4$  and an average profit:

$$-2P_L + 6P_W = 0$$

which corresponds to a fair choice of winning chances. Solving the equation  $P_W = 1/4$  with respect to  $p$  yields  $p = \frac{-1+\sqrt{3}}{2}$ .

- f) We need to compute the average absorption time for the states  $\{L, W\}$ . We may use again the reduced order model as exit times are always even numbers. Notice that the probability of being in state 2 after  $t$  pairs of gambles is:

$$\pi_2(t) = [2p(1-p)]^t.$$

The probability of exiting at time  $t+1$  is therefore  $[p^2 + (1-p)^2] \pi_2(t)$ . The average absorption time can be computed as:

$$t_a = \sum_{t=0}^{+\infty} (t+1) \pi_2(t) [p^2 + (1-p)^2].$$

Hence the average number of gambles is twice  $t_a$ .