Feedback control logic synthesis for non safe Petri nets
Abbas Dideban, Hassane Alla

To cite this version:
Abbas Dideban, Hassane Alla. Feedback control logic synthesis for non safe Petri nets. 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM’09), Jun 2009, Moscou, Russia. pp.946-951, 2009.

HAL Id: hal-00463404
https://hal.archives-ouvertes.fr/hal-00463404
Submitted on 12 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Feedback control logic synthesis for non safe Petri nets

Dideban A*. Alla H.**

* Semnan University, IRAN (Tel: (+98)231-3354123; e-mail: adideban@Semnan.ac.ir).
** GIPSA lab, 38402 St Martin d’Heres Cedex FRANCE, Hassane.alla@inpg.fr)

Abstract – This paper addresses the problem of forbidden states of non safe Petri Net (PN) modelling discrete events systems. To prevent the forbidden states, it is possible to use conditions or predicates associated with transitions. Generally, there are many forbidden states, thus many complex conditions are associated with the transitions. A new idea for computing predicates in non safe Petri nets will be presented. Using this method, we can construct a maximally permissive controller if it exists.

Keywords: Controller Synthesis, Discrete event system, Forbidden states, Petri Nets

1. INTRODUCTION

Real – life discrete-event systems (DES) are becoming more and more complex and highly automated which makes it tricky the realization of an efficient and realistic control system. Given a discrete-event model of the plant and the specification of the desired behaviour, the objective is to synthesize appropriate supervisor that will act in closed-loop with the plant according to the desired behaviour. Finite-state machines and formal languages are the modelling framework considered in the approach of Ramadge and Wonham (1989). The main limitation in such an approach is the lack of structure in controlled automata.

Petri nets have been proposed as an alternative modelling formalism for DES control. There have been many attempts to solve the control problem for DES with PN modelling. Li and Wonham (1994) have presented an algorithm, which calculates the optimal solution for nets whose uncontrollable subnets are loop-free. The theory of regions (Ghaffari et al. 2003a), allows the design of a maximally permissive PN controller. However, the number of control places is equal to the number of forbidden states and sometimes this leads to complex solutions. Holloway and Krogh have presented a method for controller calculating in real time for a safe and cyclic marked graph (Holloway and Krogh,1990). An effective method for controller synthesis was presented in (Dideban and Alla, 2006), however this method is applicable only on safe PNs. Moreover the final model may be complex.

In this paper, a method is presented to solve the problem of forbidden states for controlled Petri Nets. We develop the method presented in (Dideban and Alla, 2006) for non safe PN. Moreover, in comparison with (Ghaffari et al. 2003b) and (Holloway et al. 1996), the final condition will be very simple. The disadvantage of this approach is the calculation of the reachability graph that is fortunately performed off-line. In this paper we use the “over-state” concept that was presented in (Dideban and Alla, 2008).

This paper is organized as follows: In the second section, the fundamental definitions will be presented. The motivations for this approach will be presented in Section 3 and in Section 4, a method for calculating the condition of forbidding transitions will be presented. Then, in Section 5, the method for simplification of the conditions in safe PNs is called. In Section 6, a compact algorithm will formalize this method and solving the problem of forbidden states will be illustrated via an example. In Section 7, this method will be extended for non safe PN. Finally, the conclusion is given in the last section.

2. FUNDAMENTAL DEFINITIONS

2.1. Petri Nets

A PN is presented by a 4-uple $N = \{P, T, W, C\}$ where: 1) $P$ is the set of places, 2) $T$ is the set of transitions, 3) $W: (P \times T) \cup (P \times T)$, is the incidence matrix, and 4) $C$ is the firing conditions associated with each controllable transition.

The reachability graph consists of nodes, which correspond to the accessible markings $M_a$, and arcs to the firing of the transitions. In the reachability graph, there are two types of states: the authorized state $M_d$ and the forbidden state $M_f$. Among the forbidden states, a particular and important subset is constituted by the border forbidden states, which are denoted by the set $M_b$. These states are such that all the input transitions are controllable.

In this paper, we use the word state instead of marking.

Definition 1: The set $\{0,1\}^N$ represents all the Boolean vectors of dimension $N$.

The set of the marked places of a marking $M$ is given by a Support function that is defined in the following.

Definition 2: The function $Support(X)$ of a vector $X \in \{0,1\}^N$ is: $Support(X) = \{0,1\}^N$.
The support of vector $M_0 = [1, 0, 1, 0, 0, 0, 0]$ is:

**Support** ($M_0$) = $\{P_1, P_3, P_6\}$

**Definition 3:** Let $M_1$ and $M_2$ be 2 states of the system, and $P = \{P_1, P_2, ..., P_N\}$ the PN set of places, $M_2$ is an over-state of $M_1$ if: $\forall P_i \in P / m_i(P_2) \geq m_i(P_1)$ and $\exists P_i \in P / m_i(P_2) > m_i(P_1)$

This relation is represented as shown below:

$M_1 > M_2$

**Definition 4:** Informally, the forbidden states are:
- The states reachable in the process but not authorized by the specification.
- Deadlock states.

### 2.2. Critical and sound transitions

In the PN modelling, when a controllable event is associated with a transition, a controller can be calculated for this transition. Then we use the controllable transition instead of the uncontrollable event. Firing of some controllable transitions can lead to forbidden states. This set is named set of critical transitions. The rest of the controllable transition is named sound transitions.

#### 2.3. Critical and sound states

The border forbidden states are reached from the admissible states by the occurrence of controllable events. Preventing the occurrence of the controllable events can forbid entering to a forbidden state. By constructing the reachability graph, we can divide the admissible states for each controllable event. Firing of some controllable events can forbid entering to a state. The border forbidden states are reached from the admissible states by the occurrence of controllable events. This set is named critical states. The first and second groups are non critical states. The first and second group are non critical states. The first and second group are non critical states.

**Definition 5:** Let $M_{C_i}$ be the set of critical states and $M_{A_i}$ the set of admissible states. The sets of $t_i$, critical states $M_{C_i}$, and $t_i$ sound states $M_{S_i}$, are defined as follows:

$\forall t_i \in \sum \ M_{C_i} = \{M_i \in M_A | M_i \xrightarrow{t_i} M_j, M_j \in M_A\}$

Where $\Sigma$ is the set of controllable transitions.

**Definition 6:** Let $M_{C_i}$ be the set of critical states for the critical transition $t_i$. Control $U_{t_i}(M_{C_i}, M_i) \rightarrow \{0,1\}$ is defined as follow:

$U_{t_i}(M_{C_i}, M_i) = \begin{cases} 0 & M_j \in M_{C_i} \\ 1 & \text{if not} \end{cases}$

The control relation is modeled in Figure 1.

Fig. 1. Adding the condition as a control

This is similar with the approach presented in (Holloway et al. 1996). The difference between both approaches is the method of calculation of the control $U_{t_i}$. As it will be shown the advantage of our approach is to provide a method to determine simple forbidding conditions. To achieve this goal, we need to build the reachability graph as an intermediate step. Our approach is applicable to ordinary PNs. Firstly we present it on safe PN and then for non safe PN. It is supposed that all of the events are independent.

### 3. MOTIVATION

We first present our ideas via a simple example. Consider the classical system composed of two machines $M_1$ and $M_2$ and one buffer $S_1$. The specification constraint is the capacity of the buffer (Figure 2). Firstly we suppose that the capacity of $S_1$ is one part, it will be changed later in order to have a non safe model.

Fig. 2. A simple system

We suppose here that only the starts of the tasks (event $c_1$, $c_2$, i.e. transitions $t_1$, $t_2$) are controllable and the ends of task (event $f_1$, $f_2$ i.e. transitions $t_3$, $t_4$) are uncontrollable. The desired functioning in closed loop for this system is given in Figure 3.

The goal is to find a control such that the border forbidden states are never reached. For this, we must construct the reachability graph of the closed loop model. The reachability graph for this example is given in Figure 4.
In this state the firing of \( t_1 \) is not possible. Now, the generalization of this idea is given in the following section and a method will be presented for the simplification of these conditions thanks to the concepts of over-state which corresponds to a significant contribution.

4. CALCULATION OF FORBIDDING CONDITIONS

From the sets \( M^F \) and \( M^C \) for each controllable event or transition, there are two ways to construct the controller: the calculation of the conditions of forbidding or the calculation of the conditions for enabling each controllable event. In this paper, the first method is called but in general case it is better to examine both methods and to select the simpler solution. Now we explain how these states can be forbidden (Dideban and Alla, 2006).

**Property 1:** Let \( M_1 \) be a critical state for transition \( t_i \) in state \( M_1 \), \( U_i(M_1) = \prod_j c_j \) if \( M_1(M_2) = 0 \) and \( M_2 \) is an over-state of \( M_1 \). For constructing a maximal permissive controller, the modification of the conditions will be considered in the next property.

**Property 2:** Let \( M^C \) be a critical state for transition \( t_i \) critical states, the condition for forbidding the firing towards the forbidden states will be calculated as follows:

\[
U_i(M^C) = \left( \bigcup_{k=1}^{\text{card}(M^C)} \left( \prod_{j=1}^{\text{card}(\sup\text{port}(M^C))} m(P_j) \right) \right)
\]

Now, the condition of firing can be calculated for each transition. However, sometimes the conditions are complex and makes very difficult to understand the dynamic behaviour of the system. Moreover, the calculation time for the conditions in real time can be very large. It is then necessary to use similar simplification methods presented in (Dideban and Alla, 2005, 2008).

5. SIMPLIFICATION OF THE CONDITIONS

The simplification method that is presented in this paper is not exactly the same that presented in (Dideban and Alla, 2008). It is based on the concepts of critical and sound states which are reachable states, while the previous approach uses mainly the concepts of forbidden states.

**Property 3:** Let \( M_2 \) be an over-state of a \( t_i \) critical state \( M_1 \). Forbidding the firing of transition \( t_i \) by control \( U_i(M_2) \) leads to forbidding the firing of transition \( t_i \) in state \( M_1 \).

\[
U_i(M_2) = 0 \text{ and } M_2(M_1) \Rightarrow
\]

Transition \( t_i \) is non fireable from \( M_1 \).
This property guarantees that transition \( t_i \) is not fireable in state \( M_1 \) by using of any over-state of \( M_1 \) in \( t_i \) control. But the condition deduced from an over-state may forbid the firing of transition \( t_i \) in some of the \( t_i \) sound states. This problem is formalized via the following property:

**Property 4:** Let \( M_t^C \) be the set of \( t_i \) critical states and \( M_t^S \) the set of \( t_i \) sound states such that \( M_1 \in M_t^C \) and \( M_2 < M_1 \).

The control \( Ut(M_1) \) can be replaced by the control \( Ut(M_2) \) if there is no state \( M_3 \in M_t^S \) such that: \( M_2 < M_3 \)

An over-state can often cover several critical states, and then the condition computed for this over-state gives the control for all these critical states. Then the forbidding condition will be simpler. Our objective is to find the over-states that cover the maximum number of critical states. To reach this goal, a set of all over-states for the \( t_i \) critical state is constructed and the \( t_i \) sound states should be deleted from this set. This construction is similar to the one presented in (Dideban and Alla, 2008).

### 6. CONTROLLER SYNTHESIS

Firstly the different steps for the controller synthesis are presented by an algorithm and then illustrated via an example.

**Algorithm 1:**

1. Calculate the set of border forbidden states \( M_b \).
2. Calculate the set of the critical-states and the transitions that can be fired (C).
3. Reorganize set C for each transition (\( M_t^C \)).
4. Calculate set of sound-states (\( M_t^S \)) for all transitions in C.
5. Calculate the forbidding conditions for each controllable transition.
6. Simplify the forbidding conditions (Property 4)
7. Select the necessary and sufficient set of over-states.

**Safe PN example:** For the example presented in Section 3, the following results are obtained:

\[
Ut(M_t^C) = ((m(P_1)\Lambda m(P_3)\Lambda m(P_5)) \lor (m(P_1)\Lambda m(P_3)\Lambda m(P_6)))
\]

In the following, the simplification method is described.

In the first step in Section 2, the set of critical states for each controllable transition were calculated:

\[
M_t^C = \{P_1, P_3, P_5, P_4, P_6\}
\]

Now we calculate the set of sound states for each controllable transition. The occurring of event \( c_1 \) (transition \( t_1 \)) is only authorized in state \( M_1 \) (Figure 5).

\[
M_t^S = \{P_1, P_3, P_5\}
\]

Using the same method presented in (Dideban and Alla, 2008) for each controllable transition, the sets of all over-states for critical states and sound states must be constructed. For transition \( t_1 \), the set of its over-states for \( M_t^C \), \( (C_1^t) \) and \( M_t^S \), \( (S_1^t) \) are computed as follows:

\[
C_1^t = \{P_1, P_3, P_5, P_4, P_6\}
\]

\[
S_1^t = \{P_1, P_3, P_5, P_4, P_6\}
\]

Now, all over-states that exist in set \( S_1^t \) should be deleted from set \( C_1^t \).

\[
C_2^t = C_1^t \setminus S_1^t = \{P_4, P_5, P_6\}
\]

In addition, the over-states which are covered by another over-state can be deleted.

\[
C_3^t = \{P_4, P_5, P_6\}
\]

Now there are two over-states for transition \( t_1 \). The simpler condition for each transition must be selected. These conditions must forbid the firing of these transitions in all critical states (Table 1).

**Table 1. Final choice for \( t_1 \) in safe PN**

| Critical states | \( P_1, P_2, P_3 \) | \( P_2, P_4, P_5 \) | Choice (\( C_4^t \)) |
|----------------|----------------|----------------|----------------|
| \( P_4 \)     | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| \( P_5 \)     | \( \checkmark \)     |               |               |

For selecting the simpler condition, we write all of the critical states in first line and all of the simplified over-states in first column. For each over-state, we mark all of the states covered by it. Now for choosing the final result as for the Cluskey method (Morris Mano 2001), firstly we select the over-state covering only one state. Then for the state that is covered by two or more over-states, we select the over-state that covers the more non selected states. The final condition for our example is: \( C_4^t = \{P_4\} \Rightarrow Ut(M_t^C) = (m(P_4))^i \)

Controlable transition \( t_1 \) has not to be controlled since no forbidden state is reached.

There is a dual method for the controller synthesis. We can calculate the firing conditions instead of forbidding conditions. For this we must change the critical and sound sets. In this case for our example we have:

\[
S_1^t = S_1^t \setminus C_1^t = \{P_1, P_3, P_4, P_5\}
\]

\[
S_2^t = \{P_3\} \quad S_3^t = \{P_3\} \quad Ut(M_t^C) = (m(P_3))
\]

The simpler solution must be kept. Here they are equivalent.

### 7. EXTENSION OF THE SIMPLIFICATION METHOD TO NON SAFE PETRI NETS
In non safe Petri Nets, the number of marks in a place is not a Boolean variable. In this section, the controller synthesis method will be developed for non safe Petri Nets.

7.1 New Definition

Definition 7: The function \( \text{Support} (X) \) of a vector \( X \in \{0,1,\ldots,\}^n \) is presented as below:

\[
\text{Support}(X) = \{ m \in \text{marks} \in X \text{ with the number of marks presented as the power for each place} \}
\]

The support of vector \( M_0^3 = [1, 0, 2, 0, 0, 1, 0] \) is:

\[
\text{Support} (M_0) = \{ P_1P_2P_3P_6 \} = \{ P_1P_2P_3 \}
\]

There is no change in the definition of an over-state. If \( M_2 \) is an over-state of \( M_1 \), then \( M_2 < M_1 \).

Let \( M_i^c \) be the set of critical states for a controllable transition \( t_i \). The control \( U_{t_i} (M_i^c, M_i) \rightarrow \{0,1\} \) is defined exactly in the same way:

\[
U_{t_i} (M_i^c, M_i) = \begin{cases} 
0 & \text{if } \forall m \in M_i^c \\
1 & \text{if not}
\end{cases}
\]

We have seen that the prevention of firing in the critical state in a safe PN is simple to compute. For example, for the critical state \( P_1P_2P_3 \) we can use the Boolean condition \( m(P_1), m(P_2), m(P_3) \). But how is that for non safe PN? A definition is presented as bellow:

Definition 8: Let \( M_1 = (P_1^{k_1}P_2^{k_2} \ldots P_{1m}^{k_{1m}} \ldots P_{1n}^{k_{1n}}) \) be a critical state where \( P_{1m} \) represents the marked place, \( km \) the number of marks and \( Ut_i(M_1) \), the condition for preventing of firing from \( M_1 \), \( Ut_i(M_1) \) can be calculated as this:

\[
U_{t_i} (M_1) = (C_{11}C_{12} \ldots C_{1m} \ldots C_{1n})
\]

So that
\[
C_{1m} = \begin{cases} 
1 & m_j(P_{1m}) \geq k_{1m} \\
0 & m_j(P_{1m}) < k_{1m}
\end{cases}
\]

The simplification method by using the concept of over-state is different of the one used for safe PN. But the definition of an over-state is not changed for a non safe PN.

Definition 9: In non safe PN, a state \( M_2 = (P_1^{k_1}P_2^{k_2} \ldots P_{2n}^{k_{2n}} \ldots P_{2n}^{k_{2n}} \ldots P_{2n}^{k_{2n}}) \) is an over-state of a state \( M_1 = (P_1^{k_1}P_2^{k_2} \ldots P_{1n}^{k_{1n}} \ldots P_{1n}^{k_{1n}}) \) if \( \forall P_{1k}, P_{2k} \in \text{support}(M_2) \exists P_{1l} \in \text{support}(M_1) / P_{1l} = P_{2k} \) and \( k_{2k} \leq k_{1l} \) then:

\[
M_2 < M_1
\]

However, there is a need of a new definition for the Boolean control given by an over-state. Since this control must cover its original state, it is equal to one (or satisfied) if the number of marks in each place is greater or equal to the number of corresponding marks in the over-state. This new definition for the control calculation is presented as bellow:

\[
\text{Definition 10: Let } M_t = (P_{11}^{k_1}P_{12}^{k_2} \ldots P_{1m}^{k_{1m}} \ldots P_{1n}^{k_{1n}}) \text{ be a critical state or an over-state deduced from it, that } P_{1m} \text{ present the marked places and } km \text{ present the number of marks and } Ut_i(M_t), \text{ the condition for preventing of firing in } M_t, \text{ } Ut_i(M_t) \text{ can be calculated as this:}
\]

\[
U_{t_i} (M_t) = (C_{11}C_{12} \ldots C_{1m} \ldots C_{1n})
\]

So that
\[
C_{1m} = \begin{cases} 
1 & m_j(P_{1m}) \geq k_{1m} \\
0 & m_j(P_{1m}) < k_{1m}
\end{cases}
\]

As for safe PN, this property guaranties that transition \( t_i \) is not fireable from any over-state of \( M_1 \). And by testing the sound states, we are sure that no authorized state is forbidden.

Remark 3: By this definition, it is possible to arrive to an empty set after the reduction process. In this case we can use Definition 8 for the control calculation which cannot be simplified.

7.2 Example of a non safe PN

Consider the example presented in Figure 3 modified such that the capacity of buffer is 2. The closed loop PN model for this the system is given in Figure 6.

Fig. 6. Desired functioning in closed loop in a non safe PN

The marking graph for this example is given in Figure 7.

Fig. 7. Marking graph for a non safe PN

As previously, the goal is to find a control such that the border forbidden states are never reached. So we must control transition \( t_1 \) (event \( c_1 \)) in states \( M_6 \) and \( M_{10} \), therefore:

\[
M_i^c = [M_6, M_{10}] \quad M_6 = P_3P_4P_6 \quad M_{10} = P_1P_2P_3P_6
\]

\[
Ut_i (M_i^c) = (C_{81} \lor C_{82} \lor C_{83}) \lor (C_{101} \lor C_{102} \lor C_{103})
\]

\[
C_{81} := m(P_3) \geq 1 \quad C_{82} := m(P_2) \geq 2 \quad C_{83} := m(P_3) \geq 1
\]

\[
\text{By convention, if } m(P_{2a}) = 0, \text{ we say that } km = 0.\]
The set of sound states for each controllable transition we must calculate the sets of critical states $M_c^k$ and sound states $M_s^k$.

$$M_c^k = \{P_2P_3P_5, P_1P_4P_6\}$$

The set of sound states for $t_1$ is:

$$M_s^k = \{P_1P_3P_5, P_1P_4P_5, P_1P_3P_6, P_1P_3P_6, P_1P_4P_6\}$$

The set of over-states for critical states of $t_1$ is:

$$C^k_1 = \{P_1, P_4, P_5, P_1P_4, P_2P_3, P_2P_5, P_2P_3, P_2P_6, P_1P_3P_5, P_2P_5, P_2P_6, P_2P_3, P_1P_3P_6, P_1P_4P_6\}$$

And for sound states:

$$S^k = \{P_1, P_3, P_4, P_5, P_2P_3, P_2P_6, P_1P_3P_5, P_2P_5, P_2P_6, P_1P_3P_6, P_1P_4P_6\}$$

Now we calculate the set of over-states of critical states that are not sound states.

$$C^k_2 = C^k_1 \setminus S^k = \{P_4^2, P_2P_3^2, P_2P_5^2, P_2P_6^2, P_1P_3P_5^2, P_2P_5^2, P_2P_6^2, P_1P_3P_6^2, P_1P_4P_6^2\}$$

The final choice is the same that used for safe Petri Nets model. $C_1^{\text{f}} = \{P_2^2\} \Rightarrow U(t_1(M_c^k)) := (m(P_2) \geq 2)$

**Table 2. Final choice for $t_1$ in non safe PN**

| Critical states | $P_1P_4P_3$ | $P_1P_5P_6$ | Choice $(C_1^{\text{f}})$ |
|-----------------|-------------|-------------|---------------------|
| $P_2^2$        | $\checkmark$ | $\checkmark$ | $\checkmark$        |
|                 | $\checkmark$ | $\checkmark$ |                   |

That means if the number of marks in place $P_2$ equal to 2, this condition equal to zero and transition $t_1$ is not fireable. The controlled model is presented in Figure 8.

![Fig. 8. Final controlled model](image)

8. CONCLUSION

This paper presents an efficient approach for solving forbidden states control problems. Petri Nets are used for modelling discrete event systems and the controls correspond to conditions associated with transitions. The basic idea is to use simpler conditions for preventing the forbidden states. A condition is a Boolean expression deduced from the marking of special places for every controllable event. The concept of over-state covering several markings allows significant simplification of these conditions, which are determined in a formal way. This concept has been generalized to non safe Petri nets taking into account the number of marks in each place.

This method needs the determination of the reachability graph, hopefully it is done offline and the size of the final model is close to the specifications. It is maximal permissive, since all of the used properties meet this condition. By using a dual method sometimes it is possible to calculate a simpler controller. Each one gives a maximal permissive controller and the simplest model must be chosen.

REFERENCES

Dideban, A., Alla, H. (2005). “From forbidden state to linear constraints for the optimal supervisory control”, *Control Engineering and applied Informatics (CEAI)*, 7(3), 48-55.

Dideban, A., Alla, H. (2006). “Solving the problem of forbidden states by feedback control logical synthesis”, *The 32nd Annual Conference of the IEEE Industrial Electronics Society*, 7-10 Nov, Paris, FRANCE.

Dideban, A., & Alla, H. (2008). Reduction of Constraints for Controller Synthesis based on Safe Petri Nets. *Automatica*, 44(7): 1697-1706.

Ghaffari A., Rezg N.and Xie X.-L., (2003a). ”Design of Live and Maximally Permissive Petri Net Controller Using Theory of Regions”, *IEEE Trans. On Robotics and Automation*, vol. 19, no 1.

Ghaffari A., Rezg N. and Xie X. (2003b). “Feedback control logic for forbidden-state problems of marked graphs: Aplication to a real manufacturing system”, *IEEE Trans. Automatic Control*, Vol.48(1):18-29.

Holloway L. E. and Krogh B. H. (1990). “Synthesis of feedback logic for a class of controlled Petri nets”. *IEEE Trans. Autom. Control*, AC-35, 5, 514-523.

Holloway L. E., Guan X. and Zhang L. (1996). “A Generalisation of state avoidance Policies for Controlled Petri Nets”. *IEEE Trans. Autom. Control*, AC-41, 6, 804-816.

Kumar R., Holloway L.E. (1996). “Supervisory control of deterministic Petri nets with regular specification languages”, *IEEE Trans. Automatic Control*, 41(2):245-249.

Li Y., Wonham W. M., (1994). “Control of discrete-event systems-part ii: controller synthesis”. *IEEE Trans. Automatic Control*, 39(3):512-531.

Morris Mano M. (2001). *Digital Design*, Prentice Hall , ch 3.

Ramadge P. J., Wonham W.(1989). “The Control of Discrete Event Systems”, *Proceedings of the IEEE, Special issue...*
on Dynamics of Discrete Event Systems, Vol. 77, No. 1:81-98.