Controller synthesis with very simplified linear constraints in PN model
Abbas Dideban, M. Zareiee, Hassane Alla

To cite this version:
Abbas Dideban, M. Zareiee, Hassane Alla. Controller synthesis with very simplified linear constraints in PN model. Dependable Control of discrete Systems, Jun 2009, Bari, Italy. hal-00371113

HAL Id: hal-00371113
https://hal.science/hal-00371113
Submitted on 3 Apr 2009

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.
Controller synthesis with very simplified linear constraints in PN model

Dideban A. * Zareiee M. * Alla H. **

* Semnan University, IRAN (Tel: (+98)231-3354123; e-mail: adideban@Semnan.ac.ir).
* Semnan University, IRAN (Tel: (+98)231-3354123; e-mail: mzsareiee@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 for safe Petri net modeling discrete event systems. We present an efficient method to construct a controller. A set of linear constraints allow forbidding the reachability of specific states. The number of these so-called forbidden states and consequently the number of constraints are large and lead to a large number of control places. A systematic method for constructing very simplified controller is offered. By using a method based on Petri nets partial invariants, maximal permissive controllers are determined.

Keywords: Discrete Event Systems (DES), Petri Nets, Supervisory control, Controller synthesis, Forbidden states

1. INTRODUCTION
A unifying framework for the control of discrete event systems is provided by Ramadge and Wonham (1987, 1989), and so far their supervisory control theory is the most general and comprehensive theory presented for logical DES. In Ramadge & Wonham (1987, 1989), it is shown that we can model the systems by using of formal languages and finite automata. Methods exist for designing controllers based on automata system models, however these methods often involve exhaustive searches or simulations of the system behaviour, making them impractical for systems with large numbers of states and transitions causing event (Moody, Antsaklis 2000). For solving these problems, we can model systems with Petri nets (PNs).

Petri nets are a very appropriate and useful tool for the study of discrete event systems because of their modelling power and their mathematical properties. In the last decade, the research in the field of controller synthesis of discrete event systems has become one of the most active domains (Achour et al. 2004; Giua A & Xie X., 2005).

In Giua et al., (1992) it is shown that it is possible to represent the forbidden states with linear constraints. With the idea presented by Yamalidou et al. (1996) we can design a controller by adding a control place to the PN mode. Each control place corresponds to a constraint. However, when the number of forbidden states is large, then the number of constraints has the same size. In Giua (1992b) it is shown that we can simplify some constraints. But there is no systematic method for simplifying these constraints in general case. This simplification is done by using the PNs structural properties. In Dideban and Alla (2005) a systematic method for reducing the size and the number of constraints was introduced. It was applicable for safe and conservative PNs.

In Dideban and Alla (2008), an efficient method for constructing a controller based on safe PN is presented. A set of linear constraints allow forbidding the reachability of specific states. The number of these so-called forbidden states, and consequently the number of constraints are large and lead to a large number of control places. A systematic method for reducing the size and the number of constraints for safe PN was introduced there.

With applying the idea in Dideban & Alla (2008), it is possible that sometimes the final number of constraints may remain large. In this paper our objective is to develop the idea in Dideban & Alla (2008) and to present a systematic method for safe PN which allows to do more simplification than in the previous approaches.

In this paper, we introduce the concept of Partial Place invariant and then, we present the simplification method. We apply this method to a concrete example to highlights its advantages. We use constraints which are equivalent to forbidden states. These constraints can be calculated in two different ways. They can be given directly as specifications or they can be deduced thanks to the Kumar approach (Kumar & Holloway, 1996).

By applying the idea in Dideban & Alla (2008) and the new approach, it is possible to make a powerful simplification. The advantage of the new idea is to reduce more the number of forbidden states.

This paper is organized as follows: in section 2, some important definitions are introduced. In section 3, the idea of passage from forbidden states to linear constraints will be introduced. The idea in Dideban & Alla (2008) and a practical example are presented in section 4. The basic original idea of simplification and the calculation of control places are presented in section 5. Finally, the conclusion is given in the last section.

2. PRELIMINARY PRESENTATION
In this paper, it is supposed that the reader is familiar with the PNs basis (David & Alla, 2005) and the theory of supervisory control (Ramadge & Wonham, 1987, 1989). We briefly introduce the idea that was presented in (Dideban & Alla, 2008). For more detail, the reader can refer to this reference.
A PN is represented by a quadruplet $R = \{P, T, W, M_0\}$ where $P$ is the set of places, $T$ is the set of transitions, $W$ the incidence matrix and $M_0$ is the initial marking. Here we show the number of tokens in a place $P_i$ with $m_i$. The PN is assumed to be safe; then the marking of each place is Boolean. $\mathcal{M}_R$ denotes the set of PN reachable markings. In $\mathcal{M}_R$, two subsets can be distinguished; the set of authorized states $\mathcal{M}_A$ and the set of forbidden states $\mathcal{M}_F$. The set of forbidden states correspond to two groups:

1. The set of reachable states ($\mathcal{M}_F$) which either don’t respect the specifications or are deadlock states.
2. The set of states for which the occurrence of uncontrollable events leads to states in $\mathcal{M}_F$.

The set of authorized states are the reachable states without the set of forbidden states:

$$\mathcal{M}_A = \mathcal{M}_R \setminus \mathcal{M}_F$$

Among the forbidden states, an important subset is constituted by the border forbidden states denoted as $\mathcal{M}_F$ (Kumar & Holloway, 1996).

**Definition 1.** Let $\mathcal{M}_F$ be the set of border forbidden states:

$$\mathcal{M}_F = \{M_i \in \mathcal{M}_F \mid \exists \sigma \in \Sigma, \text{ and } \exists M_j \in \mathcal{M}_A, M_j \xrightarrow{\sigma} M_i\}$$

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

In this paper we use the concept of support instead of marking as presented in definition 2.

**Definition 2.** The function Support($X$) of a vector $X \in \{0,1\}^n$ is:

$$\text{Support}(X) = \{\text{The set of marked places in } X\}$$

**Definition 3.** Let $\mathcal{R}$ be a PN, $\mathcal{P}$ the set of its places, $\mathcal{M}(\mathcal{P})$ the set of reachable markings from $\mathcal{M}_R$ and $k$ a constant. A place invariant is obtained if there is a sub-set of places $\mathcal{P} = \{P_1, P_2, \ldots, P_r\}$ included in $\mathcal{P}$ and a weighing vector $(q_1, q_2, \ldots, q_r)$ for which all the weights $q_i$ are positive integers such that:

$$q_1m_1 + q_2m_2 + \ldots + q_rm_r = k, \forall m_i \in \mathcal{M}(\mathcal{P})$$

Then we say $\mathcal{P}$ is a place invariant, (David & Alla 2005).

**Remark 1.** With each over-state $b_i$, we associate a constraint $c_i : b_i = (P_{i1}P_{i2}P_{i3} \ldots P_{in}) \Rightarrow c_i = (P_{i1}P_{i2}P_{i3} \ldots P_{i(n-1)})$. That means:

$$m_{i1}+m_{i2}+\ldots+m_{in}=n-1$$

The concept of over-state could be used for the reduction the number of constraints deduced from states or over-states (Dideban & Alla 2008). In the next section we introduce these ideas and the problem to solve.

**3. FORBIDDEN STATES TO LINEAR CONSTRAINTS**

Let $M_i(M_i^r = \{m_1, m_2, \ldots, m_n\})$ be a forbidden state in set $\mathcal{M}_F$ and Support($M$)$=\{P_{i1}P_{i2}P_{i3} \ldots P_{in}\}$ the set of marked places of $M_i$. From a forbidden state, a linear constraint can be constructed (Giua et al., 1992).

The linear constraint deduced from the forbidden state $M_i$ is given below. The state $M_i$ does not verify this relation. Therefore, by applying this relation, $M_i$ will be forbidden.

$$\sum_{i=1}^{n} m_{ik} \leq n - 1$$

Where $n = \text{Card} [\text{support (M_i)}]$, is the number of marked places of $M_i$, and $m_{ik}$ is the marking of place $P_{ik}$ of state $M_i$. Let $M_i(M_i^r = \{m_1, m_2, \ldots, m_n\})$ be a general marking and $M_i$ be a forbidden state. The constraint (forbidden state $M_i$) is denoted by $c_i$ and can be rewritten as follows:

$$M_i \subseteq M_i^r.M \leq \text{Card}[\text{support}(M_i)] - 1$$

For example if:

$$M_1^r = [0,1,1,0,0,0,1,0] \Rightarrow \text{Card}[\text{support}(M_1)] = 3$$

$$\Rightarrow m_1 + m_2 + m_3 \leq 2$$

Verifying the relation in above is equivalent to forbidding state $M_i$ when the PN model is conservative. However, in a safe PN not necessarily conservative, this equivalence is not always true.

In follow we recall the concept of over-state that was introduced in Dideban & Alla (2008).

**Definition 5.** Let $M_2 = P_{21}P_{22} \ldots P_{2n}$ be an accessible state, $M_1 = P_{11}P_{12} \ldots P_{1n}$ will be an over-state of $M_2$ if:

$$M_1 \subseteq M_2$$

That means all of the places in $M_1$ are in $M_2$.

For example $M_1 = P_1P_3$ is an over-state of $M_2 = P_1P_3P_4P_5$. The name over-state is used because the constraint corresponding to an over-state holds constraint. For example, the constraint $m_1 + m_2 \leq 1$ that is correspond to the over-state $M_1 = P_1 P_2 P_3$ holds both following constraints:

$$m_1 + m_3 + m_4 \leq 3, \quad m_1 + m_4 + m_5 \leq 3$$

These two constraints forbid states $M_0 = P_1P_2P_3$ and $M_2 = P_2P_3P_4P_5$, $P_5P_6$ is an over-state of both states $P_1P_2P_3P_4P_5$ and $P_2P_3P_4P_5$ which could be verified by $M_1 \subseteq M_6$ and $M_2 \subseteq M_6$. Thus by using only the constraint $m_1 + m_2 \leq 1$, both states $M_6$ and $M_2$ will be forbidden.

**4. RECUCTION THE NUMBER OF CONSTRAINTS**

Now we explain the idea in Dideban & Alla (2008). Suppose that in a system we have the set of authorized states $\mathcal{M}_A$ and the set of border forbidden states $\mathcal{M}_F$. In the first step of reduction, we must construct the sets $\mathcal{B}_1$ (All the over-states of border forbidden states) and $\mathcal{A}_1$ (All the over-states of authorized states). Then we remove from $\mathcal{B}_1$ the states or over-states that exist in $\mathcal{A}_1$ and call this set $\mathcal{B}_2$. In the next step we remove the states which their over-states are in $\mathcal{B}_2$ from it and we call it $\mathcal{B}_3$.

Sometimes it is possible, to simplify the reduced set $\mathcal{B}_n$, by application of the idea that is presented in Dideban (2007). But sometimes it is not the simplest set. The rules to choose the final over-states are similar to the rules of the Quine-McCluskey method to simplify the logical expressions (Morris Mano 2001).

**4.1. Introduction of a practical example**
Consider a system with three machines and two robots. The closed loop PN model of this system is presented in fig. 1. The start command of each machine is accomplished by controllable transitions $c_1$, $c_2$, $c_3$ and the stop command is accomplished by uncontrollable transitions $f_1$, $f_2$, $f_3$. Two robots are used to free three machines. All of machines can be turned on or off independently. The end of process for each machine when at least one of the robots are not free, is forbidden.

From constructing the reachability graph, we can calculate the set of border forbidden states.

Fig. 1. closed loop PN model

Fig. 2. A part of reachability graph of the PN model presented in fig. 1. The construction of the reachability graph is stopped when a forbidden state is reached. The state after these forbidden states is presented by $\otimes$.

As it is obvious in fig. 2, the existence of uncontrollable events leads to the forbidden states. For example when the system is in state $(P_1P_2P_3P_{10})$, it is possible to fire the uncontrollable event $f_1$ while it is not authorized by the specifications. This state is called as a forbidden state. And these forbidden states are determined from the specifications. This state is called as a forbidden state. And these forbidden states can be determined from these forbidden states.

In this example the set of border forbidden states is:

$$M_B = \{ P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}, P_1P_2P_3P_{10}\}$$

And The set of authorized states is:

$$M_A = \{ P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}\}$$

Now, the sets of over-states of $M_A$ and $M_B$ are constructed. The set of over-states of $M_B$ is shown with $B_1$ and $M_A$ with $A_1$.

Hence the set of over-states of $M_B$ is:

$$A_1 = \{ P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}, P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}\}$$

After this step we remove from $B_1$, the states that are common in $B_1$ and $A_1$, and the residual set is named $B_2$.

Remark 2. we don’t write the set of over-states of authorized states $A_1$, since with having $M_A$ we can remove the common states from $B_1$.

$$B_2 = \{ P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}, P_2P_4P_5P_6P_7P_8P_9P_{10}\}$$

By the method presented in Dideban & Alla (2008), the controller can be synthesized by final selection of simplified over-states.

4.2. Controller synthesis

Final selection is similar to the McCluskey method for simplifying logical terms. In this method we construct a table (table 1) where the first row represents the set of border forbidden states $M_B$, and the first column is the set of simplified over-states $B_2$. The relation between border forbidden states and the simplified over-states can be expressed by the following definition:

**Definition 6.** let $B_2=\{b_1,b_2,...,b_n\}$ be the set of simplified over-states and $M_B=\{M_1,M_2,...,M_M\}$ be the set of border forbidden states. The relation $R : M_B \times B_2 \rightarrow \{0,1\}$ is as:

$$R(M_i,b_j) = \begin{cases} 1 & \text{if } M_i(b_j) \in \text{over state of } M_i \\ 0 & \text{if not.} \end{cases}$$

The covering of a marking is an integer number:

$$Cv(M_i) = \sum R(M_i,b_j)$$

$Cv(M_i) \geq 1$ means that forbidden state $M_i$ is covered by at least one over-state. 

| Table 1 | Relation between over-states and border forbidden states |
|---------|--------------------------------------------------|
| $M_B$   | $B_2$                                           |
| $M_1$   | $b_1$                                           |
| $M_2$   | $b_2$                                           |
| $M_M$   | $b_n$                                           |


The final constraints must cover all the border forbidden states. This means that with applying the constraints equivalent to the final sets, all the border forbidden states are forbidden. These constraints are as follow:

\[ B_k = \{ P_1 P_2 P_3, P_1 P_2 P_4, P_1 P_2 P_5, P_1 P_2 P_6, P_1 P_2 P_7, P_1 P_2 P_8, P_1 P_2 P_9, P_1 P_2 P_{10} \} \]

Before using the method in Dideban & Alla (2008), the number of constraints and equivalently the number of border forbidden states for our example was 13, but with using this method, the number of constraints is 9. It is clear that the number of simplified constraints is large, and for designing a controller by the method presented by Yamalidou et al. (1996), the number of control places is also large. However, is it possible to simplify these constraints more? The answer to this question is given by a method based on the concept of partial place invariant that was defined in section 2.

5. INTRODUCTION OF THE NEW IDEA FOR SIMPLIFICATION

We consider the partial invariant for safe and conservative PNs. From the invariant relations we can construct partial invariants. For example:

\[ m_1 + m_2 + \ldots + m_n = k \Rightarrow m_1 + m_2 + \ldots + m_n \leq k \]

In safe PN in general case, it is not possible to construct this equation directly. In the following, we present some properties based on partial invariants that in many cases allow us to simplify the set of constraints representing the forbidden states.

Property 1: suppose that \( m_{i_1}, m_{i_2} \) are the number of tokens in the places \( P_{i_1}, P_{i_2} \) respectively. If \( P_{i_1}P_{i_2} \) is not in the set of over-states of authorized states we have:

\[ m_{i_1} + m_{i_2} \leq 1 \]

\[ \Box \]

Proof: Suppose that the constraint is not true. Then we have:

\[ m_{i_1} + m_{i_2} > 1 \]

So, for safe PNs we have:

\[ m_{i_1} + m_{i_2} = 2 \Rightarrow m_{i_1} = m_{i_2} = 1 \]

This means \( P_{i_1}P_{i_2} \) is in the set of over-states of authorized states that is not true. Then:

\[ m_{i_1} + m_{i_2} \leq 1 \]

\[ \Box \]

Property 2 (Extension of Property 1), suppose that the constraint \( m_{i_1} + \ldots + m_{i_n} = k \) is true. If all of the over-states \( \{ P_{i_1}P_{i_2} \ldots P_{i_4}P_{i_5} \ldots P_{i_{(n+1)}} \} \) are not in the set of over-states of authorized states, we have:

\[ m_{i_1} + \ldots + m_{i_n} + m_{i_{(n+1)}} \leq 1 \]

\[ \Box \]

**Proof:** The proof of this property is similar to the proof of property 1. Suppose that this relation is not true, so we can write:

\[ m_{i_1} + \ldots + m_{i_n} + m_{i_{(n+1)}} > 1 \]

hence we have:

\[ m_{i_1} + \ldots + m_{i_n} + m_{i_{(n+1)}} = 2 \]

\[ m_{i_1} + \ldots + m_{i_n} \leq 1 \]

And also from the constraint, we have:

\[ (m_1 + m_2 + \ldots + m_r) + m_1 + \ldots + m_j \leq k \]

By adding two constraints, we have:

\[ (m_1 + m_2 + \ldots + m_r) + m_k + \ldots + m_j \leq k + 1 \]

We show that the value \( k + 1 \) is not any time accessible. Suppose that \( k + 1 \) is accessible. We have:

\[ m_1 + m_2 + \ldots + m_r = 1 \Rightarrow m_1 + \ldots + m_j = 1 \]

From the \( r \) constraints and equation \( m_1 + \ldots + m_k = k \), we have:

\[ m_1 = m_2 = \ldots = m_k = 0 \]

Then:

\[ (m_1 + m_2 + \ldots + m_r) = 0 \]

That is not true.

**Sufficient condition:**

\[ (m_1 + \ldots + m_k) + m_1 + \ldots + m_j \leq k \]

\[ \forall q \in \{1, \ldots, r\} \quad m_q \geq 0 \Rightarrow m_q + m_1 + \ldots + m_j \leq k \]
Generally, by application of these properties, it is possible to simplify the constraints. However, for our example, it is not possible to do more simplification using these properties. Suppose that the sets $A_1$ and $B_1$ be the sets of authorized and forbidden states, respectively:

$$A_1 = \{P_1P_3P_5, P_1P_3P_7, P_1P_4P_5P_8, P_2P_1P_5P_8\}$$

$$B_1 = \{P_1P_4P_5, P_2P_4P_5, P_2P_5P_7, P_4P_5P_7\}$$

By application of property 1 and 3 for states $\{P_1P_3P_5, P_2P_3P_5\}$, we have: $m_1 + m_2 + m_4 + m_5 \leq 2$. For states $\{P_1P_3P_7, P_2P_3P_7\}$, the constraint $m_2 + m_3 \leq 1$ is not verified, but is it necessary? We show that it is not necessary. When the borne of constraints is more than 1, for having $m_2 + m_4 + m_5 \leq 2$, it is not necessary $m_2 + m_4 \leq 1$, but also it is necessary that $m_2 + m_4 + m_5 \leq 2$ and $m_2 + m_4 + m_7 \leq 2$ be true.

The following property generalizes this idea.

**Property 4** suppose that we have the set of constraints:

$$C = \{(P_1P_3P_5...P_wn), (P_1P_3P_5...P_wn)\}$$

Consider $n$ over-states as follow:

$$P_1P_3P_5P_7...P_{j+1}...P_n \text{ for } 2 \leq j \leq n-1$$

$$P_1P_3P_5P_7...P_{n-1}P_{n+1}P_{n+2}...P_n$$

If none of these states are in the set of over-states of authorized states, then we can reduce these constraints to one constraint as follow:

$$P_1P_3P_5P_7...P_n$$

that means:

$$m_1 + m_2 + m_3 + m_4 + ... + m_n \leq n$$

**Proof:** the proof of this property is similar. Suppose that the inequality $m_1 + m_2 + m_3 + m_4 + ... + m_n \leq n$ is not true then $m_1 + m_2 + m_3 + m_4 + ... + m_n = n + 1$. That means $(n+1)$ places is marked in the set of $\mathcal{P} = (P_1, P_2, P_3, ..., P_n)$ then at least one of the condition in the property is violated. So our supposition is not true.

**Property 5** Suppose that we have the constraints $C_1 = \{(P_1P_2P_5...P_{w+1}P_{w}P_5...P_n), (P_1P_2P_5...P_{w+1}P_{w}P_5...P_n)\}$, that verify the authorized states. If all of the over-states content $n+1$ places from the sets $\mathcal{P} = (P_{n+1}, P_{n+2}, ..., P_m)$ are not exist in the set of over-states of authorized states, the two constraints $C_1$ and $C_2$ can be replaced by one constraint as follow:

$$P_1P_2P_3P_5...P_{w+1}P_{w}P_5...P_n$$

**Proof:** the proof of this property is similar to the proof of property 4. Suppose that the inequality $m_1 + m_2 + m_3 + m_4 + ... + m_n \leq n$ is not true then $m_1 + m_2 + m_3 + m_4 + ... + m_n = n + 1$, that means $(n+1)$ places is marked in the set of $\mathcal{P} = (P_1, P_2, ..., P_m, P_{n+1}, P_1, P_2, ..., P_n)$ then at least one of the condition in the property is violated. So our supposition is not true.

**5.1. More simplification in our example by new idea**

From section 4.1 we have:

$$B_3 = \{(P_2P_4P_5, P_2P_4P_5, P_2P_4P_{10}, P_2P_9P_{10}, P_2P_9P_{10} + P_2P_9P_{10}, P_2P_9P_{10} + P_2P_9P_{10}, P_2P_9P_{10})\}$$

We can select the subset $C_1$ that contains two common places and one different place. $C_1 = (P_2P_9P_{10}, 2)\). $P_2P_9P_{10}$, in the set of over-states of authorized states. So it is impossible to use the property 2, but the states $P_2P_9P_{10}$, $P_2P_9P_{10}$, $P_2P_9P_{10}$, are not in the set of over-states of authorized states, then it is possible to use the property 4.

$$C_1 = (P_2P_9P_{10}, 2) \Rightarrow C_1 = (P_2P_9P_{10}, 2)$$

For applying the property 5 on the sets of $C_1$ and $C_2 = (P_2P_9P_{10}, 2), (P_2P_9P_{10}, 2) \Rightarrow C_2 = (P_2P_9P_{10}, 2)$

The simplified sets $C_2$ and $C_3$ and $C_4$ are the same. Then the final set is: $C_2 = (P_2P_9P_{10}, 2)$

So it is obvious that with using these properties the 9 constraints have reduced to one constraint that is good reason for utilization of these properties.

In this example, there is not the final selection, but generally by the method as the same was presented in last section, we can select the final result.

**5.2. Calculation of control places**

To calculate the control places corresponding for each linear constraint, we will use the method developed in Yamalidou et al. (2006). Now we briefly present this method that is based on concept of $P$-invariant. Suppose that we have the set of constraints as $L,m_{\ell} \leq b$ that $m_{\ell}$ is the marking vector of system, $L$ is a $n \times n$ matrix, $b$ is a $n \times 1$ vector, $n$, is the number of constraints and $n$ is the number of places. In this method for each constraint, we add a place to the PN model of the system. In Yamalidou et al. (2006), it is proved that this places guarantees that these constraints are respected. Suppose $W_p$ be the incidence matrix of the system. For each constraint we add a row to $W_p$ and we show this as $W_p$, corresponds to the incidence matrix of the controller. and $W_c$ is calculated as follows:

$$W_c = L,W_p$$

$W_c$ is added to $W_p$ as follow:

$$W = \begin{bmatrix} W_p \end{bmatrix} \begin{bmatrix} W_c \end{bmatrix}$$

If the initial marking of system is $m_0$, the initial marking for the added places is calculated as follow:
Now we calculate the control places for our example. The simplified constraint was \((P_2 P_4 P_6 P_8 P_{10}, 2)\), Then we have:
\[ L = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \Rightarrow W_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \]
\[ M_{s0} = 2 \]

The PN model of the final controller after adding the control place is indicated in fig 3. The control place and corresponding arcs are shown with gray color.

Fig. 3. Controlled PN model

As it is obvious from fig. 3, with adding \(P_{c1}\), forbidden states can not be reached.

6. CONCLUSION

In this paper, we have presented a systematic method to reduce the number of linear constraints corresponding to the forbidden states for a safe PN. This is realized by using non-reachable states and by building the constraints using a systematic method. The important concept of over-state has been used; it corresponds to a set of markings which keep the same property (forbidden or authorized). From the forbidden states, the set of over-states is calculated. The utilization of non-reachable markings allows great simplification of the constraints.

In this paper, we have extended the idea presented in Dideban & Alla (2008). We have used the partial invariant for more simplifications. Finally, we have shown that sometimes it is possible to simplify without invariant or partial invariant. By this method we can arrive to very simplified constraints deduced from the forbidden states.

REFERENCES

Achour Z., & N.Rezg, X. Xie . (2004). Supervisory controller of Petri nets under partial Observation. Proc. IFAC WODES04: 7th Workshop on DISCRETE EVENT SYSTEMS(Reims, france), September.

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., 2007, Synthesis of discrete controllers by simplification of constraints and conditions, Ph.D. Thesis, UJF, France.

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

Giua, A., DiCesare, F.M., & Silva. (1992). Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions. In Proc. IEEE int. conf. on systems, man, and cybernetics (pp. 974–799).

Giua, A., & Xie, X. (2005). Control of safe ordinary Petri Nets using unfolding. Discrete Event Dynamic Systems: Theory and Applications, 15, 349–373.

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

Moody, J.O., & Antsaklis, P. (2000). Petri net supervisor for DES with uncontrollable and unobservable transition. IEEE Trans. Automatic Control, 45(3): 462-476, mars.

Morris Mano, M. (2001). Digital design. Prentice Hall.

Ramadge, P.J., & Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. SIAM Journal of Control and Optimization, 25 (1):206-230.

Ramadge, P. J., & Wonham, W. (1989). The control of discrete event systems. Dynamics of discrete event systems [Special issue]. Proceedings of theIEEE, 77(1), 81–98.

Rene, D., Alla, H. (2005). Discrete, Continuous, and Hybrid Petri Net. Springer.

Yamalidou, K., Moody, J., Lemmon, M., & Antsaklis, P. (1996). Feedback control of petri nets based on place invariants. Automatica, 32(1), 15–28.