Determination of Minimal Sets of Control Places for Safe Petri Nets
Abbas Dideban, Hassane Alla

To cite this version:

HAL Id: hal-00356523
https://hal.science/hal-00356523
Submitted on 27 Jan 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.
Determination of Minimal Sets of Control Places for Safe Petri Nets

A. DIDEBAN, H. ALLA

Abstract—Our objective is to design a controlled system with a simple method for discrete event systems based on Petri nets. It is possible to construct the Petri net model of a system and the specification separately. By synchronous composition of both models, the desired functioning closed loop model is deduced. Often uncontrollable transitions lead to forbidden states. The problem of forbidden states is solved using linear constraints. A set of linear constraints allows forbidding the reachability of these states. Generally, the number of these so-called forbidden states and consequently the number of constraints are large and lead to a great number of control places. A systematic method to reduce the size and the number of constraints for safe Petri Nets is given. By using a method based on the Petri nets invariants, maximal permissive controllers are determined. The size of the controller is close to the size of the specified model, and it can be implemented on a PLC in a structural way.

Index Terms — Controller, Discrete Event Systems (DES), Forbidden states, marking invariant, Petri Net.

I. INTRODUCTION

Supervisory control theory is essentially a theory for restricting the behavior of the plant to satisfy given specifications that describe which evolutions of the plant should not be allowed. The theory of Ramadge and Wonham [1][2] is based on the modeling of the systems using formal languages and finite automata. However, the great number of states representing the behavior of system, and the lack of structure in the models, limit the possibility of developing an effective algorithm for the analysis and the synthesis of real systems. To solve these problems, several methods of controller synthesis based on Petri Nets (PNs) were proposed. PNs are a suitable tool to study Discrete Event Systems (DES) due to its capability in modeling and its mathematical properties.

In [3] and [4], the authors use the marking invariants to determine algebraically the incidence matrix of the supervisor PNs model. This method is very simple to be used. However, if some transitions are uncontrollable, it does not give the maximal permissive solution. This technique presents two other disadvantages: 1) it is not always possible to describe the specifications by constraints and, 2) the number of constraints can be very large.

A general objective of the control synthesis is to prevent from forbidden states. These states may be deduced from specifications, they can also be deadlock states. A method to minimize the addition of PN places is proposed in [5], it is based on elementary siphons. There are some drawbacks in this paper. Firstly, one can see that it is based on the computation of minimal siphons and secondly the proposed method is not generally optimal. Another problem is that uncontrollable transitions cannot be considered. In [6] and [7], the authors proposed a method for solving the problems of forbidden states by the theory of regions. The advantage of this method is its generality for non safe PNs. However there are some drawbacks for this method:

- Generally the number of control places is large.
- The computation time for solving the set of integer equations can be very large.

In [8], it is shown that it is possible using the linear constraints to specify forbidden states for safe and conservative PNs. The proposed approach is based on the equivalence between the set of forbidden states and the set of linear constraints, which are deducted from it. Using invariants technique presented in [3], allows building a set of control places, which constitute the optimal controller. However, the number of control places and consequently the number of constraints are large and lead to a large number of control places. In [8], it is also shown that some constraints can be replaced by a single constraint; however there is no systematic method to calculate the simplified constraints in a general case. The problem comes from the linear constraints, which can be simplified taking the PN structural properties into account. In [9], a systematic method has been presented to reduce the number of constraints for safe and conservative PNs. The equations deduced from P-invariants property in conservative PNs are used for simplifications.

In this paper, a method is proposed to reduce the number of linear constraints for safe PNs. The expected advantage of this method is its applicability for all kind of safe PN models, and not only for the conservative PNs. The time and memory space for simplification is less than those presented in [9].

In our approach and also in [9], we use the Reachability Graph (RG) as an intermediate step for calculating the controller. Although the complexity of the computation of RG is exponential, this calculation is performed off-line. Moreover, the implemented final controller is a PN model,
whose size is very close to the starting model. Generally, few control places are added.

In this paper, an important concept of over-state will be defined; it corresponds to a set of markings which have the same property. This idea will help us to build the simplest constraints which forbid a greater number of states. A property for the existence of the maximal permissive controller will be analytically proved. In some very particular cases of non conservative PNs, the optimal solution does not exist. We show that this approach allows highlighting this problem in a simple way.

This paper is organized as follows: In Section 2, the motivation and the fundamental definitions will be presented and illustrated via an example. In Section 3, the idea of passage from forbidden states to the linear constraints will be introduced. The concept of over-state will be defined in Section 4. Section 5 constitutes the part of this paper where the fundamental properties for the simplification methods are presented. The calculation of maximal permissive controller will be described in Sections 6 and 7.

II. PRELIMINARY PRESENTATION

In this paper, it is supposed that the reader is familiar with the PNs basis [10] and the theory of supervisory control [1], [2]. Here, we present only the notations and definitions that will be used later.

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, \( W \) is the incidence matrix and \( M_0 \) is the initial marking. This PN is assumed to be safe; the marking of each place is a Boolean.

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

A marking of a safe PN containing \( N \) places is a vector of the set \( \{0,1\}^N \).

The set of the marked places of a marking \( M \) is given by a function support defined as below:

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

\[ \text{Support}(X) = \text{the set of marked places in } X. \]

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

Support\((M_0) = \{P_1, P_3P_6\} \) or more simply:

Support\((M_0) = P_1P_3P_6\)

To simplify the writing of the formal expressions, we will use the support of a marking instead of its corresponding vector.

\( \mathcal{M}_b \) denotes the set of PN reachable markings. In \( \mathcal{M}_b \), two sub sets 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 do not respect the specifications or are deadlock states and, 2) the set of states such that 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:

\[ \mathcal{M}_a = \mathcal{M}_b \setminus \mathcal{M}_f \]

Among the forbidden states, an important subset is constituted by the border forbidden state denoted as \( \mathcal{M}_b \).

**Definition 3:** Let \( \mathcal{M}_b \) be the set of border forbidden states:

\[ \mathcal{M}_b = \{M_j \in \mathcal{M}_f | \exists \sigma \in \Sigma_c \text{ and } \exists M_f \in \mathcal{M}_f, M_f \xrightarrow{\sigma} M_j \} \]

Where \( \Sigma_c \) is the set of controllable transitions.

We will use the following example in order to illustrate the definitions and the results developed in this paper.

Consider a system composed of two machines \( M_{a1} \) and \( M_{a2} \) which can work independently. The starting and the end of jobs on these machines are respectively realized by controllable events \( c_1 \) and \( c_2 \), and uncontrollable events \( f_1 \) and \( f_2 \). When the machine \( M_{a1} \) ends its job on a part, it stays available for a new job while machine \( M_{a2} \) has to transfer its produced part in a buffer before beginning a new job (event \( t_2 \)). The specification imposes a sequence of the events \( f_1 \) and \( t_2 \). An elementary process corresponds to the treatment of a part by \( M_{a1} \) followed by the treatment of a part by \( M_{a2} \). This production is repeated in a cyclic way. The synchronous composition of the system’s model and the specification’s model is given in Figure 1.

![PN model of the system](image)

The existence of uncontrollable events leads to the existence of forbidden states. The set of forbidden states can be determined by the algorithm established by Kumar [11]. Figure 2 gives the reachability Graph of the PN presented in Figure 1. The forbidden states are indicated in dark gray and the authorized states in white. In this example, there is no deadlock state.

From the set of forbidden states \( \mathcal{M}_f = \{M_5, M_6, M_7, M_8, M_9, M_{10}, M_{12}\} \), we can construct the set of border forbidden states.
be a forbidden state. The constraint (forbidden state denoted as places of
the important concept of "over-state"
the number and the size of the linear constraints for a given set
constraint \[4\]. In this paper, we propose a method to reduce
controller model increases when the number of forbidden
we will need
places of
by applying this relation,
constructed \[8\].
Where
Hypothesis 1:
1) All of the events are independent.
2) The forbidden states are non marked states and all the
authorized states are marked.

III. FROM FORBIDDEN STATES TO LINEAR CONSTRAINTS
Let \( M_i, \) \( M_i = \{ m_{i1}, m_{i2}, \ldots, m_{in}\} \) be a forbidden state in set
\( M^f \) and Support(\( M_i \)) = \{ \( P_1, P_2, P_3, \ldots, P_n\) \} be
the set of marked places of \( M_i \). From a forbidden state, a linear constraint can be
constructed \[8\].

The linear constraint deduced from the forbidden state \( M_i \) is given
below. The state \( M_i \) does not verify this relation. Thus, by
applying this relation, \( M_i \) will be forbidden.
\[
\sum_{b=1}^{n} m_{kb} \leq n - 1
\]
Where \( n = \text{Card} \{ \text{Support}(M_i) \} \) is the number of marked
places of \( M_i \), and \( m_{kb} \) the marking of place \( P_k \) of state \( M_i \).
Let \( M (M^f = [m_1, m_2, \ldots, m_n]) \) be a general marking and \( M_i \)
be a forbidden state. The constraint (forbidding state \( M_i \)) is
denoted as \( c_i \) and can be rewritten in the following form:
\[
M_i^T \cdot M \leq \text{Card} \{ \text{Support}(M_i) \} - 1
\]
For example if:
\[
M_i^T = [0, 1, 1, 0, 0, 0, 1] \Rightarrow \text{Card} \{ \text{Support}(M_i) \} = 3
\]
\[
\Rightarrow m_2 + m_3 + m_7 \leq 2
\]
Verifying relation 1 is equivalent to forbid state \( M_i \) when

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{reachability_graph.png}
\caption{Reachability graph}
\end{figure}

IV. OVER–STATE

A. Definition of an over–state
The concept of over-state is very important in this paper.
An over-state can represent a complete state or a part of this
one. In the example of two machines, \( P_1P_2P_3P_4 \) is a complete
state that represents the situation of both machines and the
specification. \( P_2P_3 \) is an over-state of this state that represents
a partial state of the system. We have seen that a state can be
forbidden by a linear constraint. In the same way, it is possible
to forbid an over-state by its corresponding constraint.

Definition 4: Let \( M_1 = P_1, P_2, \ldots, P_{2m} \) an accessible state,
\( M_2 = P_1, P_2, \ldots, P_{2n} \) will be an over-state of \( M_1 \) if:
\[
M_1 \leq M_2
\]

For example \( M_1 = P_1P_3 \) is an over-state of \( M_2 = P_1P_2P_4 \).
The name “over-state” is used because the constraint corresponding
to an over-state holds the state’s constraint. For example
the constraint \( m_4 + m_6 \leq 1 \) corresponding to the over-state
\( M_1 = P_4P_6 \) holds both following constraints:
\[
m_1 + m_4 + m_6 \leq 2
\]
\[
m_2 + m_4 + m_6 \leq 2
\]
These two constraints forbid the states \( M_6 = P_1P_2P_4P_6 \) and
\( M_7 = P_1P_2P_4 \). \( P_2P_6 \) is an over-state of both states \( P_1P_2P_4P_6 \) and
\( P_2P_3P_6 \) which could be verified by \( M_1 \leq M_6 \) and \( M_1 \leq M_7 \).
Thus only using the constraint \( m_4 + m_6 \leq 1 \) both states \( M_6 \) and
\( M_7 \) will be forbidden. However, this reduction is not always so
simple; sometimes it is possible that the simplified constraint
forbids also some authorized states. We present below a method
of simplification, which guarantees that the constraints forbid
only the forbidden states.

Remark 1: With each over-state \( b_i \), we associate a
constraint \( c_i \) in the following way:
\[
b_i = (P_1P_2P_3 \ldots P_{2m}) \Rightarrow c_i = (P_1P_2P_3 \ldots P_{2m}, n-1)
\]
That means:
\[
m_1 + m_2 + \ldots + m_n \leq n-1
\]

Remark 2: There are two relations of inclusion, which
work in opposite direction: a set inclusion and a marking
inclusion. Let \( M_i \leq M_j \):
1) The set of the marked places in the over-state \( M_i \) is
included in the set of the marked places in the state \( M_j \).
2) The set of the markings covered by \( M_i \) contains those
covered by marking \( M_j \).

Property 1: Let \( M_1 \) and \( M_2 \) be two vectors of \( \{0, 1\}^n \) and \( c_1 \)
and \( c_2 \) be two corresponding constraints. If \( M_1 \leq M_2 \) (\( M_1 \) is an
over-state of \( M_2 \)) and \( c_1 \) is true, then \( c_2 \) is also true:
$M_1 \leq M_2$ and $c_1$: \[ M_1^T, M \leq \text{Card}(\text{Support}(M_1))-1 \]

\[ c_2: M_2^T, M \leq \text{Card}(\text{Support}(M_2))-1 \]

All the demonstrations of the properties used in this paper can be found in [12].

B. Set of over-states

We have seen that for forbidding a state it is enough to forbid its over-state but which over-state? This question will be answered in this paper. To achieve this goal, we need to construct the set of over-states for the forbidden states.

Firstly, we calculate the set of over-states for each state and then the union of all the over-states giving the final set.

Definition 5: Let $M_i = \{P_1, P_2, P_3, ... P_m\}$ be a state of the system. The set of the over-states of $M_i$, denoted as $M_i^{over}$, is equal to the set of the subsets of $M_i$ without the empty set.

For example, the state $M_1 = P_1P_2P_3$ gives:

\[ M_1^{over} = \{P_1, P_2, P_3, P_1P_2, P_1P_3, P_2P_3, P_1P_2P_3\} \]

Among, the set of forbidden states in $\mathcal{M}_E$, only the borders states have to be considered in the controller synthesis. Let $\mathcal{M}_B$ be this set and $B_1$ be the set of over-states of $\mathcal{M}_B$.

\[ B_1 = \bigcup_{i=1}^{\text{Card}(M_1)} M_i^{over} \]

V. BUILDING THE REDUCED SET OF OVER-STATES

It is needed to build two sets of over-states; a set of the authorized over-states $A_1$, and that of the forbidden states $B_1$. It is obvious that no over-state of $A_1$ must be forbidden. Thus it is necessary to remove from the set $B_1$, all the over-states which are in $A_1$.

Property 2: Let $B_1$ be the set of over-states of $\mathcal{M}_B$ and $A_1$ be the set of over-state of $\mathcal{M}_A$ and:

\[ B_2 = B_1 \setminus A_1 \]

Verifying the set of constraints $C_2$ (equivalent to $B_2$) do not forbid any authorized state.

The proof of this property is obvious.

Another point is that in set $B_2$, it may occur that two over-states $M_1$ and $M_2$ are such as $M_1 \leq M_2$. In that case, $M_2$ is removed giving the set $B_3$ defined formally as follows:

\[ B_3 = B_2 \setminus \{M_2 \in B_2 \mid \exists M_2 \leq M_2 \geq M_2\} \]

$B_3$ is the set of the over-states to be forbidden.

For the example of Figure 1, the sets of borders forbidden states and authorized states are:

$\mathcal{M}_B = \{P_1P_2P_6, P_2P_3P_6, P_3P_6, P_3P_7, P_4P_9, P_2P_7\}$

$\mathcal{M}_A = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

$M_i = \{P_1P_2P_6, P_2P_3P_6, P_3P_6, P_3P_7, P_4P_9, P_2P_7\}$

The different sets $A_1$, $B_1$, $B_2$ and $B_3$ are then calculated below:

\[ B_1 = \bigcup_{i=1}^{\text{Card}(M_1)} M_i^{over} \]

\[ B_2 = \bigcup_{i=1}^{\text{Card}(M_1)} M_i^{over} \]

\[ B_3 = B_2 \setminus \{M_2 \in B_2 \mid \exists M_2 \leq M_2 \geq M_2\} \]

VI. MAXIMAL PERMISSIVE CONTROLLER

In the previous section, we have determined the set $B_3$, which is the greatest set of over-states that must be forbidden. With the concept of marking covering, we are going to present in the two following sections, the necessary and sufficient conditions to obtain a maximal permissive controller.

With each over-state of $B_3$, we associated a constraint in the following way:

\[ b_i = (P_1P_2P_3 ... P_m) \Leftrightarrow c_i = (P_1P_2P_3 ... P_m, n+1) \]

Let $C_i$ be the set of these constraints in our example:

\[ C_i = \{(P_1P_2P_3), (P_2P_3, 1), (P_2P_5, 1)\} \]

This set $C_i$ defines the set of non-forbidden states, denoted as $\mathcal{M}_E$. Now our objective is to compare the set of authorized states $\mathcal{M}_A$ and $\mathcal{M}_E$.

Remark 3: Constraint $c_i$ and over-state $b_i$ are equivalent as shown above.

Definition 7: Let $B_3 = \{b_1, b_2, ..., b_n\}$ be the set of simplified over-states and $\mathcal{M}_B = \{M_1, M_2, ..., M_m\}$ be the set of border forbidden states. The relation $R$: $\mathcal{M}_B \times B_3 \rightarrow \{0, 1\}$ is such as:

\[ R(M_i, b_j) = \begin{cases} 1 & b_j \leq M_i \text{ (b_j is over-state of } M_i) \\ 0 & \text{if not} \end{cases} \]

The covering of a marking is an integer number:

\[ Cv(M_i') = \sum_{j=1}^{n} R(M_i', b_j) \]

\[ Cv(M_i') \geq 1 \text{ mean that forbidden state } M_i \text{ is covered by at least one over-state.} \]
Property 3: The set of non forbidden state $\mathcal{M}_E$ is equal to the set of authorized state $\mathcal{M}_I$ if and only if:

$$\forall M_i \in \mathcal{M}_B \quad CV(M_i) \geq 1$$

We are going to illustrate the results established above on the example of Figure 1. Property 3 should initially be checked. For this, we construct a table (Figure 3) where the first line represents the set of forbidden states $\mathcal{M}_B$ and the first column, the set of the simplified over-states $\mathcal{B}_s$.

To choose the final results, the table of Figure 3 is used and McCluskey method to simplify the logical expressions [13].

Corollary 1: The set of the non forbidden states $\mathcal{M}_E$ defined by the set of the constraints deduced from $\mathcal{B}_s$ is equal to $\mathcal{M}_E$ if and only if: $\forall M_i \in \mathcal{M}_B \quad CF(M_i) = 1$

This corollary means that it is necessary for each forbidden states to be covered at least by one over-state. When this is verified, the maximal permissive controller is obtained.

B. Controller synthesis

The set of the constraints equivalent to $\mathcal{B}_s$ is denoted as $\mathcal{C}_s$. To calculate the control places corresponding to each linear constraint, we will use the method developed in [4]. This technique will be called invariant approach recalled briefly below. Let $WR$ be the incidence matrix of the system (process and specifications). Each place of the controller will add a line to this matrix. Let $WRC$ be the incidence matrix of the PN model corresponding to the controlled system. It is made up of two matrices, the original matrix of system $WR$ and the incidence matrix of the controller $WC$. From the set of constraints $\mathcal{C}_s$, the matrix $L$ and the constant vector $C_{\text{bound}}$ can be constructed. It is possible to calculate in an algebraic way the incidence matrix of the controller as it is presented below. $M_{B_i}$ is the initial marking of system $R$ and $M_{C_i}$ is the control places initial marking. The very simple way to calculate $WC$ makes this approach very popular.

$$WC = -L.WR \quad , \quad M_{C_i} = C_{\text{bound}} - L.M_{B_i}$$

Let us take again the example of Figure 1, the set of final constraints ($C_4$) is:

$$m_4 + m_6 \leq 1 \quad , \quad m_2 + m_7 \leq 1$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow WC = \begin{bmatrix} 0 & 1 & -1 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{C_1} = C_{\text{bound}} - L.M_{B_1} \Rightarrow M_{C_1} = 0; \quad M_{C_2} = 1$$

Yamalidou [4] showed that if all the events are controllable, the controller is maximal permissive. However, if there are uncontrollable events, the extended method presented in [3] does not give the optimal solution in the
general case. The problem exists when a control place is synchronized with a place of the process by an uncontrollable event. In this case, the process cannot respect the PN firing rules. It means that the set of the reachable states will be greater than that the set given by the PN model. We will indicate this set as $\mathcal{A}_{RC}$.

We are going to show that if Corollary 1 is true, the controller obtained is maximal permissive even if uncontrollable transitions exist.

**Remark 4:** A marking of the set $\mathcal{A}_{RC}$ differs from a marking of $\mathcal{M}_P$ because of the added control places. This is only a coding of these sets. To be able to compare the various sets of states, we will omit the control places for the elements of the set $\mathcal{A}_{RC}$.

**Property 4:** Let $\mathcal{M}_P$ be the set of authorized states by the constraints deduced from $B_4$ and let $\mathcal{A}_{RC}$ be the automaton corresponding to the set of accessible state in the controlled system. If $\mathcal{M}_P = \mathcal{M}_d$ then $\mathcal{A}_{RC}$ is isomorphic to $\mathcal{M}_P$ and the controller obtained by the invariant approach is maximal permissive.

In the case of our example the function $C(M)$ (final covering) is equal to 1 for each $M_i \in \mathcal{M}_p$, therefore $\mathcal{M}_P = \mathcal{M}_d$ (Corollary 1) then the controller is maximal permissive (Property 4). The PN model of the final controller is represented in Figure 8.

It should be noticed that there are some control places with output uncontrollable transitions. However, that never leads to a bad behavior, i.e. when a control place is not marked; there is at least one non marked input place of this uncontrollable transition, which belongs to the process. Moreover, controllable events $e_1$ and $e_2$ have been removed since the control is now performed by the control places. The different sets computations are in polynomial complexity except for the $\mathcal{M}_I$ over-states computation which is exponential. Fortunately the number of border states is often small.

![Fig. 8. PN Model in closed loop with control places](image)

VIII. 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 the non reachable states and by building the constraints with a systematic method. The important concept of over-state has been defined; it corresponds to a set of markings having the same property (forbidden or authorized). From the forbidden states the set of over-states is calculated. The utilization of over-state concept allows simplifying the constraints. Properties giving necessary and sufficient conditions for the existence of a maximal permissive controller were established. After the simplifications, the existence of the controller is proved formally. When this one exists, the invariant approach allows computing the controller.

REFERENCES


