



**HAL**  
open science

# An Abstraction Method of Interpreted Petri Nets Preserving the Equivalence of the Controllable Observable Language

Pedro Chavarin-Aguirre, Ernesto López-Mellado, Jean-Jacques Lesage

► **To cite this version:**

Pedro Chavarin-Aguirre, Ernesto López-Mellado, Jean-Jacques Lesage. An Abstraction Method of Interpreted Petri Nets Preserving the Equivalence of the Controllable Observable Language. 14th Workshop on Discrete Event Systems, (WODES'18), May 2018, Sorrento, Italy. pp. 378-384. hal-01757986

**HAL Id: hal-01757986**

**<https://hal.science/hal-01757986>**

Submitted on 4 Apr 2018

**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.

# An Abstraction Method of Interpreted Petri Nets Preserving the Equivalence of the Controllable Observable Language

Pedro Chavarín-Aguirre\* Ernesto López-Mellado\* Jean-Jacques Lesage\*\*

\*CINVESTAV Unidad Guadalajara

Av. Del Bosque 1145 Col El Bajío, 45019 Zapopan, Mexico (email: {pechavarin, elopez}@gdl.cinvestav.mx)

\*\*LURPA ENS de Cachan, Univ. Paris-Sud, Université Paris-Saclay

94235 Cachan, France (e-mail: lesage@lurpa.ens-cachan.fr)

---

**Abstract:** The analysis and control of DES is often supported by interpreted Petri nets (IPN), allowing describing the input-output behavior of the involved components. One of the main challenges of the analysis and synthesis methods is the size of the model when the system is large and performs complex behavior. In order to alleviate this hardship, a model abstraction method for IPN is proposed in which the events and the outputs of the IPN are considered in order to preserve in the reduced model, the controllable observable language of the original IPN. The reductions are considerable in such a manner that reachability set is drastically reduced. The abstraction procedure is polynomial-time on the size of the IPN.

**Keywords:** Interpreted Petri nets; Model abstraction; Controllable observable language.

---

## 1. INTRODUCTION

It is well known that for large Discrete Event Systems (DES) involving complex behaviour, the number of states can be extremely high and consequently, the models are huge despite the power of expression of Petri nets (PN).

PN have been widely used for validation and control of DES in several problems, namely deadlock detection and avoidance, stability analysis and stabilizing control, process identification, supervisory control, and regulation control. One of the challenges in these problems is the efficiency of the methods due to the size of the PN when it is not possible to find structural-based solutions.

In order to alleviate such a hardship, PN reduction methods have been proposed, allowing simplifying the models while preserving key properties for the analysis. The aim was to reduce the size of the PN and thus, reduce the size of the reachability graph. Several approaches can be found in [Berthelot, 1986], [Silva, 1985], [Lee-Kwang, 1985], [Lee-Kwang, 1987] [Murata, 1989], [Desel, 1990], [Esparza, 1994], and [Desel, 1995].

In this paper, we present a model abstraction method for Interpreted PN (IPN), which reduces the size of the model by considering the type of input events associated to transitions and outputs associated to places. In contrast to other proposed methods, the events and the outputs of the IPN associated with transitions and places respectively are taken into account to preserve in the reduced model, the controllable output language of the original IPN.

The method performs structural transformations around a designated subset of places labelled with specified outputs that must remain in the reduced IPN. This allows improving the efficiency of the techniques addressing for example the problems of regulation control [Santoyo, 2008] or stability

analysis in which the reachability of markings involving a subset of places is handled.

The method is based on a set of local transformation operators, which consider the behaviour of the IPN to preserve the properties regarding the controllability of firing sequences and the reachability of a subset of observable places associated to selected outputs. An efficient procedure manages the application of the operators; it obtains a reduced IPN that has the same controllable output language than the original one. The abstraction technique allows a drastic reduction of IPN size, and consequently, of the number of states.

The rest of the paper is organised as follows. Section 2 presents the basic notions of observable behaviour of IPN. Section 3 describes the abstraction technique, based on a kit of transformation operators for ordinary and safe IPN; an example of application to an IPN is presented to illustrate the model abstraction technique.

## 2. THE OBSERVABLE LANGUAGES EQUIVALENCE

### 2.1. Interpreted Petri Nets

**Definition 1.** An ordinary Petri Net structure  $G$  is a bipartite digraph represented by the 4-tuple  $N = (P, T, I, O)$  where:  $P = \{p_1, p_2, \dots, p_{|P|}\}$  and  $T = \{t_1, t_2, \dots, t_{|T|}\}$  are finite sets of vertices named places and transitions respectively;  $I : P \times T \rightarrow \{0,1\}$  ( $O : T \times P \rightarrow \{0,1\}$ ) is a function representing the arcs going from places to transitions (from transitions to places) [Murata, 1989]. A marking function  $M : P \rightarrow \{0,1\}$  represents the number of tokens residing inside each place; it is usually expressed as an  $|P|$ -entry vector.

A Petri net (PN) is the pair  $(N, M_0)$ , where  $M_0$  is an initial marking. In a PN, a transition  $t_j$  is enabled at marking  $M_k$  if  $\forall_{p_i \in P}, M_k(p_i) \geq I(p_i, t_j)$ ; an enabled transition  $t_j$  can be fired reaching a new marking  $M_{k+1}$ ; it is computed by the *state*

equation:  $M_{k+1} = M_k - I(p_i, t_j) + O(p_i, t_j)$ . The reachability set of a PN is the set of all possible reachable markings from  $M_0$  firing only enabled transitions; this set is denoted as  $R(G, M_0)$ . For any  $t_j \in T$ ,  $\bullet t_j = \{p_i \mid I(p_i, t_j) = 1\}$ , and  $t_j \bullet = \{p_i \mid O(t_j, p_i) = 1\}$ ; similarly, for any  $p_i \in P$ ,  $\bullet p_i = \{t_j \mid O(t_j, p_i) = 1\}$ , and  $p_i \bullet = \{t_j \mid I(p_i, t_j) = 1\}$ . A PN where  $M_k(p_i) \in \{0, 1\}$  is called *safe* or 1-bounded.

An Interpreted Petri net (IPN) is an extension of a PN that includes events and outputs of a modelled process.

**Definition 2.** An Interpreted Petri Net (IPN) is a 6-tuple  $Q = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$ , also denoted as  $(Q, M_0)$  where:

- $\Sigma = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  is the finite input symbol alphabet.
- $\Phi = \{\phi_1, \phi_2, \dots, \phi_w\}$  is the finite output symbol alphabet.
- $\lambda: T \rightarrow \Sigma \cup \{\varepsilon\}$  is the transition labeling function with the following constraints:  
 $\forall t_j, t_k \in T, j \neq k, \quad \text{if } \forall p_i I(p_i, t_j) = I(p_i, t_k) \neq 0 \quad \text{and} \quad \lambda(t_j), \lambda(t_k) \neq \varepsilon, \text{ then } \lambda(t_j) \neq \lambda(t_k).$
- $\varphi: \mathcal{R}(Q, M_0) \rightarrow (Z^+)^q$  is the output function, which associates each marking in  $\mathcal{R}(Q, M_0)$ , with a  $q$ -entry output vector, where  $q = |\Phi|$ .  $\varphi$  is represented by a  $q \times |P|$  matrix, such that if the output symbol  $\phi_i$  is present every time that  $M(p_j) > 0$ , then  $\varphi(i, j) = 1$ , otherwise  $\varphi(i, j) = 0$ .

The state equation of PN is completed with the marking projection  $Y_k = \varphi M_k$ , where  $Y_k \in (Z^+)^q$  is output vector of the IPN associated with  $M_k$ .

**Definition 3.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an IPN. If  $\lambda(t_i) \neq \varepsilon$ , the transition  $t_i$  is called controllable, otherwise is called uncontrollable.  $T_U$  is the set of uncontrollable transitions and  $T_C$  is the set of controllable transitions in  $(Q, M_0)$ .  $T = T_U \cup T_C$  and  $T_U \cap T_C = \emptyset$ . When an enabled transition  $t_i$  is controllable, then to fire  $t_i$  it is also needed that the input symbol  $\lambda(t_i)$  is present. When an enabled transition  $t_i$  is uncontrollable, it can be fired. A place  $p_i \in P$  is said to be measurable if the  $i$ -th column vector of  $\varphi$  (denoted as  $\varphi(\bullet, i)$ ) is not null, otherwise it is non measurable.  $P = P_\varphi \cup \overline{P_\varphi}$  and  $P_\varphi \cap \overline{P_\varphi} = \emptyset$ ; where  $P_\varphi$  is the set of measurable places and  $\overline{P_\varphi}$  the set of non measurable places.

**Example 1.** In Figure 1 it is shown an IPN  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$ .  $\Sigma = \{q, y, v, e, d, g\}$ .  $\Phi = \{A, B, C\}$ .  $\lambda(t_0) = q, \lambda(t_1) = y, \lambda(t_2) = v, \lambda(t_3) = e, \lambda(t_4) = \varepsilon, \lambda(t_5) = d, \lambda(t_6) = g$ . The matrix representing the output function  $\varphi$  is

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

where every row represents one of the outputs  $A, B, C \in \Phi$ .  $P_\varphi = \{p_1, p_4, p_6\}$ ,  $\overline{P_\varphi} = \{p_2, p_3, p_5\}$ . Pictorially, places in  $P_\varphi$  and transitions in  $T_U$  are dimmed.

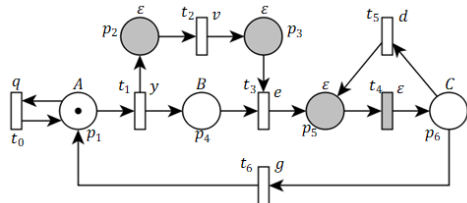


Fig. 1. Interpreted Petri net  $(Q, M_0)$ .

**Definition 4.** Let  $(Q, M_0)$  be an IPN. The set of all firing sequences of  $(Q, M_0)$ , called the firing language of  $(Q, M_0)$ , is  $\mathcal{L}(Q, M_0) = \{\sigma = t_i t_j \dots t_k \mid M_0 \xrightarrow{t_i} M_i \xrightarrow{t_j} \dots \xrightarrow{t_k} M_k\}$ .

Note that for a reachable marking  $M_l \in \mathcal{R}(Q, M_0)$ , the firing language  $\mathcal{L}(Q, M_l) = \{\tau = t_m t_n \dots t_s \mid M_l \xrightarrow{t_m} M_m \xrightarrow{t_n} \dots \xrightarrow{t_s} M_s\}$  denotes the set of all firing sequences enabled from  $M_l$  in  $(Q, M_0)$ .

**Notation:** The Parikh vector  $\vec{\sigma}: T \rightarrow (Z^+)^m$  of  $\sigma \in \mathcal{L}(Q, M_0)$  maps every  $t_i \in T$  to the number of occurrences of  $t_i$  in  $\sigma$ . The input language of  $(Q, M_0)$  is  $\mathcal{L}_{in}(Q, M_0) = \{\lambda(\sigma) \mid \sigma \in \mathcal{L}(Q, M_0)\}$ . Given  $w \in \mathcal{L}$  a word of a language  $\mathcal{L}$  the prefix set of  $w$  is  $\bar{w} = \{w' \mid \exists v \text{ such that } w'v = w\}$ .

## 2.2. IPN Observable language equivalence

In this section we define the observable behaviour to describe when two IPN models can be equivalent, with respect to their output languages.

### 2.2.1. Observable behaviour equivalence

The measurable places in an IPN  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  generate signals that an external agent can observe; thus, given a reachable marking  $M \in \mathcal{R}(Q, M_0)$ , the output  $\varphi(M)$  describes the actual observable state of a process.

The observable equivalence between two IPN determines the behaviour equivalence from an external point of view; i.e., when the observable behaviours of two IPN are indistinguishable regardless the evolution of their markings.

A necessary condition for observable equivalence of an IPN  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$ , with respect to another IPN  $(Q', M'_0) = (N', M'_0, \Sigma', \Phi', \lambda', \varphi')$ , is that they share a set of output symbols  $\tilde{\Phi}$ ;  $(\tilde{\Phi} \subseteq \Phi) \wedge (\tilde{\Phi} \subseteq \Phi')$ .

In order to consider only those outputs of interest, let  $\tilde{\varphi}$  be a new output function of  $(Q, M_0)$  that considers only those measurable places that are related to an output symbol in  $(Q', M'_0)$ ; where  $\tilde{\varphi}$  is composed by only those rows in  $\varphi$  such that the  $i$ -th row of  $\tilde{\varphi}$  represents the activation of the same symbol  $\phi_i$  as the  $i$ -th row of  $\varphi'$ .

The following notion deals with sequences that generate changes in the output.

**Definition 5.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an IPN. Let  $\sigma = t_1 t_2 t_3 \dots t_r \in \mathcal{L}(Q, M_0)$  a sequence of transitions such that  $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \dots \xrightarrow{t_r} M_r$ . It is said that  $\sigma$  is an Output Switching sequence (OSS) in  $(Q, M_0)$  iff there exist  $i, j$  such that:

- $\varphi(M_0) \neq 0, \varphi(M_r) \neq 0$
- $\forall M_x, 0 \leq x \leq i, \varphi(M_x) = \varphi(M_0)$
- if  $j = i + 1$  then  $\varphi(M_0) \neq \varphi(M_r)$ , else  $\forall M_y, i < y < j, \varphi(M_y) = 0$
- $\forall M_z, j \leq z \leq r, \varphi(M_j) = \varphi(M_r)$

The firing of OSSs in  $(Q, M_0)$  determine observable words.

**Definition 6.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an IPN. Let  $\sigma = \sigma_1 \sigma_2 \dots \sigma_l \in \mathcal{L}(Q, M_0)$  such that  $M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_l} M_l$  and,  $\sigma_i (1 \leq i \leq l)$  is an OSS or  $\sigma$  is empty. The observable

language of  $(Q, M_0)$  is  $\mathcal{L}_{obs}(Q, M_0) = \{w \mid \exists \sigma \in \mathcal{L}(Q, M_0)\}$ , where  $w = \varphi(M_0) \varphi(M_1) \dots \varphi(M_l)$  is the *observable word* generated by  $\sigma$  in  $(Q, M_0)$ . The function  $f_{obs}(Q, M_0): \mathcal{L}(Q, M_0) \rightarrow \mathcal{L}_{obs}(Q, M_0)$  maps every firing sequence into its observable word.

**Example 2.** Consider the IPN  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  of Figure 1. When the sequence  $\sigma = t_0 t_1 t_2 t_3 t_4 t_5 t_4 t_6$  is fired in  $(Q, M_0)$  it generates the observable word:

$$w = f_{obs}(Q, M_0)(t_0 t_1 t_2 t_3 t_4 t_5 t_4 t_6) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Now we can define the equivalence between two IPN with respect to their observable behaviour.

**Definition 7.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  and  $(Q', M'_0) = (N', M'_0, \Sigma', \Phi', \lambda', \varphi')$  be two IPN models.  $(Q, M_0)$  is observable equivalent to  $(Q', M'_0)$  iff  $\mathcal{L}_{obs}(Q, M_0) = \mathcal{L}_{obs}(Q', M'_0)$ .

Is easy to see that for  $(Q, M_0)$  and  $(Q', M'_0)$  in Figures 1 and 2.a respectively,  $(Q, M_0)$  is observable equivalent to  $(Q', M'_0)$ .

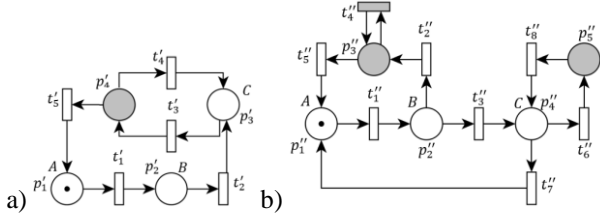


Fig. 2. Interpreted Petri nets  $(Q, M_0)$  and  $(Q', M'_0)$ .

### 2.2.2. Controllability

Let  $(Q, M_0)$  be an IPN and  $K \subseteq \mathcal{L}(Q, M_0)$  be the specification language. The language  $K$  is controllable with respect to  $\mathcal{L}(Q, M_0)$ , iff  $\forall t_i \in T_U$  it holds that  $\bar{K}t_i \cap \mathcal{L}(Q, M_0) \subseteq \bar{K}$ .

Now it is possible to define all those observable words that can be generated by an IPN by the firing of controllable firing sequences only.

**Definition 8.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an IPN model. The controllable observable language of  $(Q, M_0)$  is  $\mathcal{L}_{obs}^c(Q, M_0) = \{w \in \mathcal{L}_{obs}(Q, M_0) \mid \exists \sigma \text{ controllable in } (Q, M_0) \text{ s.t. } f_{obs}(Q, M_0)(\sigma) = w\}$ ; it is the set of all the controllable observable words in  $\mathcal{L}_{obs}(Q, M_0)$ .

Now we can define when two IPN models can be equivalent, with respect to their controllable observable languages.

**Definition 9.** Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  and  $(Q', M'_0) = (N', M'_0, \Sigma', \Phi', \lambda', \varphi')$  be two IPN models.  $(Q, M_0)$  is controllable observable equivalent to  $(Q', M'_0)$  iff  $\mathcal{L}_{obs}^c(Q, M_0) = \mathcal{L}_{obs}^c(Q', M'_0)$ .

Given  $(Q, M_0)$  and  $(Q', M'_0)$ , when the controllable observable languages of  $(Q, M_0)$  and  $(Q', M'_0)$  are equal, it means that for every controllable firing sequence in  $(Q', M'_0)$  that generates an observable word  $w$ , there exists a controllable firing sequence in  $(Q, M_0)$  that generates  $w$  and vice versa.

It is easy to see that for  $(Q, M_0)$  and  $(Q', M'_0)$  in Figures 1 and 2.b respectively,  $\mathcal{L}_{obs}^c(Q, M_0) = \mathcal{L}_{obs}^c(Q', M'_0)$ ; thus,  $(Q, M_0)$  is controllable observable equivalent to  $(Q', M'_0)$ .

## 3. IPN ABSTRACTION

Several transformation methods have been proposed in [Berthelot, 1986], [Murata, 1989] and [Lee-Kwang, 1987], which are applicable to general PN. The rules from [Berthelot, 1986] preserve the paths in the original net that always fire consecutively the transitions to be merged by the transformation. Whereas the interest in some of the reduction rules from [Murata, 1989] and [Lee-Kwang, 1987] is to remove unnecessary paths and redundant nodes.

The above-mentioned transformations do not consider properties regarding the controllability of events and reachability of observable states. Then, an abstraction technique based on a kit of graph rewriting operators  $\mathcal{O}$  for safe IPN is proposed in this paper; the aim is to eliminate structural redundancies and unnecessary parallel evolutions while preserving the observable language and controllable observable language with respect to a set of outputs  $\tilde{\Phi}$ .

### 3.1 Transformation operators

A transformation operator  $\mathcal{T}_x$  changes a part of a source IPN  $(Q, M_0)$  into another one preserving desired properties while reducing the number of nodes. The resulting IPN after the application of an operator  $\mathcal{T}_x$  is denoted as  $(Q, M_0)|_{\mathcal{T}_x}$ .

Six transformation operators  $\mathcal{O} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4, \mathcal{T}_5, \mathcal{T}_6\}$  are presented; every  $\mathcal{T}_x$  is illustrated through an example.

#### 3.1.1 Operator $\mathcal{T}_1$ : post-fusion of transitions

This operator eliminates a non-measurable place  $p$  and merges the pre-set  $B$  of  $p$  with its post-set  $F$ ; that is, the transitions in  $B$  and  $F$  are substituted by a set of transitions that represent the possible firing sequences from a transition in  $B$  to a transition in  $F$ . Then, the original paths are recorded as transition sequences in the new transitions of the abstracted net.

**Definition 10.** Operator  $\mathcal{T}_1$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The operator  $\mathcal{T}_1$  is applicable to  $(Q, M_0)$  if there exist a non-measurable place  $p \in P$  and sets  $B = {}^*p, F = p^* \subseteq T$  such that:

- i.  $B \cap F = \emptyset$ ,  $p \notin {}^*B$  and  $p \notin F^*$
- ii.  ${}^*F = \{p\}$ , The only input of  $F$  is  $p$
- iii.  $F^* \neq \emptyset$ ,  $F$  has at least one output place
- iv.  $(|B| = 1) \vee (|F| = 1)$   $B$  or  $F$  has only one transition
- v.  $M_0(p) = 0$ ,  $p$  is initially unmarked
- vi.  $p \notin P_\varphi$ ,  $p$  is not a measurable place
- vii. One of the following conditions holds:
  - vii.1.  ${}^*B \cap P_\varphi = \emptyset$ ,  $B^* = \{p\}$  and either vii.1.1 or vii.1.2 is fulfilled
    - vii.1.1.  $(B \cup F) \cap T_U = \emptyset$  and  $\forall t \in B, \forall t' \in (({}^*t) \setminus \{t\}) \cap T_U; {}^*t' \subseteq {}^*t$
    - vii.1.2.  $F \subseteq T_U$  and  $|F| = 1$
  - vii.2.  $F^* \cap P_\varphi = \emptyset, (B^* \setminus \{p_i\})^* \cap T_U = \emptyset, B^* \cap F^* = \emptyset$  and (vii.2.1. or vii.2.2. )
    - vii.2.1.  $(B \cup F \cup F^*) \cap T_U = \emptyset$
    - vii.2.2.  $F \subseteq T_U, |F| = 1$

therefore,  $(Q, M_0)$  can be transformed into  $(Q, M_0)|_{\mathcal{T}_1}$  after the application of the following steps:

1.  $\forall t_i \in \bullet p, \forall t_j \in p^\bullet$ 
  - 1.1.  $T = T \cup \{t_k\}$
  - 1.2.  $\forall p_q \in \bullet t_i, I(p_q, t_k) \leftarrow 1$
  - 1.3.  $\forall p_s \in (t_i^\bullet \setminus p \cup t_j^\bullet), O(p_s, t_k) \leftarrow 1$
  - 1.4.  $\lambda(t_k) \leftarrow \lambda(t_i) \cdot \lambda(t_j)$
2.  $T \leftarrow T \setminus (\bullet p \cup p^\bullet)$
3.  $P \leftarrow P \setminus \{p\}$

Notice that  $\lambda(t_k)$  created in step 1.4 is the concatenation of the labels of the merged transitions.

The conditions from *i* to *iii* are sufficient to preserve liveness and safeness; they are equivalent to those in [Berthelot, 1986] when the conditions are restricted to safe Petri nets. The condition *iv* is needed to assure that every application of  $\mathcal{T}_1$  always reduces the number of nodes of the original IPN. The application conditions *v* and *vi* are needed to preserve the initial marking and the measurable places of interest. Conditions in *vii* are necessary to preserve the observable and controllable observable languages from the original IPN.

**Example 3.** Fig. 3.a shows an IPN that satisfies  $\bullet B \cap P_\varphi = \emptyset, B^\bullet = \{p_1\}$ , it also satisfies conditions *vii.1.1* and *vii.1.2* respectively. The transformed IPN is shown in figure 3.b. The dashed places and arrows represent structural conditions that are not allowed for applying the operator  $\mathcal{T}_1$ .

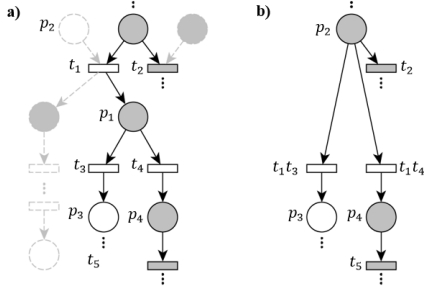


Fig. 3. a) IPN  $(Q, M_0)$  satisfying *vii.1.1*. b)  $(Q, M_0)|_{\mathcal{T}_1}$ .

### 3.1.2. Operator $\mathcal{T}_2$ : pre-fusion of transitions

Alike to  $\mathcal{T}_1$ , this operator merges the single transition  $b$  in the pre-set of a non measurable place  $p$ , with the set of transitions  $F$  of the post-set of  $p$ , when the enabling of any transition in  $F$  cannot occur before the firing of  $b$ .

**Definition 11.** Operator  $\mathcal{T}_2$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The transformation operator  $\mathcal{T}_2$  is applicable to  $(Q, M_0)$  if there exist a place  $p \in P$ , a transition  $b \in T$  and a set  $F = p^\bullet \subseteq T$  such that:

- i.  $\bullet p = \{b\}$ , The only input of  $p$  is  $b$
- ii.  $b^\bullet = \{p\}$ , The only output of  $b$  is  $p$
- iii.  $\bullet b \neq \emptyset$ ,  $b$  has at least an input place
- iv.  $b \notin F$ ,  $b$  is not an output of  $p$
- v.  $M_0(p) = 0$ ,  $p$  is initially unmarked
- vi.  $(\bullet b)^\bullet = \{b\}$ ,  $b$  does not share its input places
- vii.  $p \notin P_\varphi$ ,  $p$  is not a measurable place
- viii.  $\bullet B \cap P_\varphi = \emptyset$  and one of the following conditions:

- viii.1.  $(\{b\} \cup F) \cap T_U = \emptyset$
  - viii.2.  $F \cap T_U \neq \emptyset, b \in T_U$  and  $\bullet(F \cap T_U) = \{p\}$
- then,  $(Q, M_0)$  is transformed into  $(Q, M_0)|_{\mathcal{T}_2}$  after the application of the following steps:

1.  $\forall t_i \in p^\bullet$ 
  - 1.1.  $T = T \cup \{t_k\}$
  - 1.2.  $\lambda(t_k) \leftarrow \lambda(b)\lambda(t_i)$
  - 1.3.  $\forall p_r \in (\bullet t_i \setminus \{p\} \cup \bullet b), I(p_r, t_k) \leftarrow 1$
  - 1.4.  $\forall p_q \in \bullet t_i, O(p_q, t_k) \leftarrow 1$
2.  $T \leftarrow T \setminus (\{b\} \cup p^\bullet)$
3.  $P \leftarrow P \setminus \{p\}$

Similarly to  $\mathcal{T}_1$ , the application conditions from *i* to *vi* are sufficient to preserve liveness and safeness, since they are equivalent to those in [Berthelot, 1986] when the conditions are restricted to ordinary and safe Petri nets. Conditions *vii* and *viii* are necessary to preserve the observable language and controllable observable language from the original IPN.

**Example 4.** In the IPN shown in Figure 4, the place  $p_1$  is removed after the application of the operator  $\mathcal{T}_2$ , since the conditions  $\bullet B \cap P_\varphi = \emptyset$  and *viii.1* are satisfied. The dashed places and arrows represent some structures not allowed.

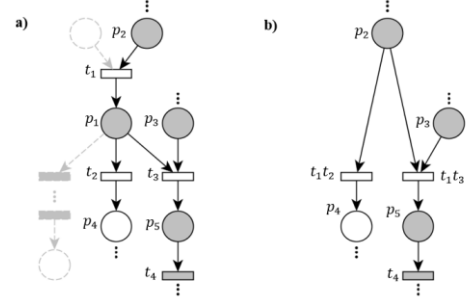


Fig. 4. a) IPN  $(Q, M_0)$  satisfying *viii.1*. b)  $(Q, M_0)|_{\mathcal{T}_2}$ .

### 3.1.3. Operator $\mathcal{T}_3$ : parallel place removal

This transformation operator removes a non measurable place  $p$  in the Petri net, when there exist another place that has the same pre-set and post-set than  $p$ , such that the initial marking assigns the same number of tokens to them.

**Definition 12.** Operator  $\mathcal{T}_3$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The transformation operator  $\mathcal{T}_3$  is applicable to  $(Q, M_0)$  if there exist non-measurable places  $p, r \in P$  where  $p \neq r$ , such that:

- i.  $\bullet p = \bullet r$ ,
- ii.  $p^\bullet = r^\bullet$ ,
- iii.  $M_0(p) = M_0(r)$ ,
- iv.  $p \notin P_\varphi$  or  $r \notin P_\varphi$ .

then,  $(Q, M_0)$  is transformed into  $(Q, M_0)|_{\mathcal{T}_3}$  after applying the update: if  $p \in P_\varphi$  then  $P \leftarrow P \setminus \{r\}$  else  $P \leftarrow P \setminus \{p\}$ .

It is easy to see that this operator preserves liveness and safeness of the PN. Furthermore, this transformation does not alter the firing sequences, since no transition is ever removed and for every reachable marking, both  $r$  and  $p$  have the same number of tokens.

**Example 5.** Figure 5.a shows an IPN  $(Q, M_0)$ , where the places  $p_5$  and  $p_6$  are parallel to  $p_2$  and  $p_3$  respectively. Figure 5.b shows  $(Q, M_0)|_{\mathcal{T}_3}$ , the resulting IPN after the application of the operator  $\mathcal{T}_3$ , where the non measurable parallel places  $p_5$  and  $p_6$  are eliminated.

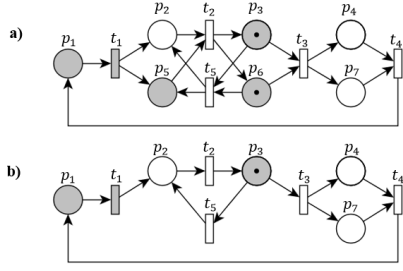


Fig. 5. a) IPN  $(Q, M_0)$ . b)  $(Q, M_0)|_{T_3}$ .

### 3.1.4. Operator $T_4$ : identical transition removal

The identical transition removal operator deletes a controllable transition when there exists another transition with the same pre-set and post-set, because the firing of either one will lead to the same marking. If one transition is not controllable, then the controllable one is eliminated. This transformation operator preserves liveness and safeness of the IPN.

**Definition 13.** Operator  $T_4$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The transformation operator  $T_4$  is applicable to  $(Q, M_0)$  if there exist transitions  $t, y \in T$ ,  $t \neq y$  such that:

- i.  $\bullet t = \bullet y$ ,
- ii.  $t \bullet = y \bullet$
- iii.  $t \notin T_U$  or  $y \notin T_U$

$(Q, M_0)$  is transformed into  $(Q, M_0)|_{T_4}$  after the application of the following step:

If  $t \notin T_U$  then  $T \leftarrow T \setminus \{t\}$  else  $T \leftarrow T \setminus \{y\}$

**Example 6.** Figure 6.a shows an IPN  $(Q, M_0)$ , where the transition  $t_4$  is redundant with respect to  $t_1$ . Fig. b shows the resulting IPN after the application of the operator  $T_4$ , where the identical transition  $t_4$  is removed.

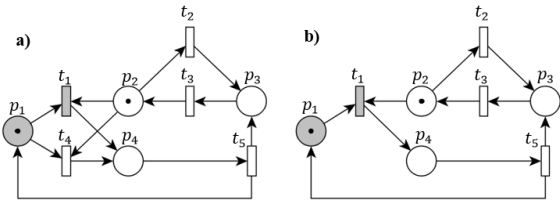


Fig. 6. a) IPN  $(Q, M_0)$ . b)  $(Q, M_0)|_{T_4}$ .

### 3.1.5. Operator $T_5$ : self-loop places removal

The transformation operator  $T_5$  removes an initially marked self-loop place when it is non-measurable.

**Definition 14.** Transformation operator  $T_5$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The transformation operator  $T_5$  is applicable to  $(Q, M_0)$  if there exist a non-measurable place  $p \in P$  and transition  $t \in T$  such that:

- i.  $\bullet p = p \bullet$ ,
- ii.  $M_0(p) = 1$ ,
- iii.  $p \notin P_\varphi$
- iv.  $|P| > 1$

$(Q, M_0)$  is transformed into  $(Q, M_0)|_{T_5}$  after the application of the following step:  $P \leftarrow P \setminus \{p\}$

Is easy to see that this transformation operator preserves liveness and keeps the PN safe. Since no transition is removed, this transformation does not alter the firing sequences.

**Example 7.** Figure 7 shows two IPN representing the original IPN (a) and the resulting IPN after the application of the transformation operator  $T_5$ (b); the initially marked non measurable self-loop place  $p_3$  is eliminated.

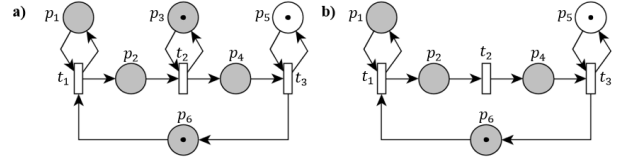


Fig. 7. a) IPN  $(Q, M_0)$  b)  $(Q, M_0)|_{T_5}$ .

### 3.1.6. Operator $T_6$ : self-loop transitions removal

The self-loop transition removal transformation operator eliminates a self-loop transition when it is controllable.

**Definition 15.** Transformation operator  $T_6$ . Let  $(Q, M_0) = (N, M_0, \Sigma, \Phi, \lambda, \varphi)$  be an ordinary and safe IPN. The transformation operator  $T_6$  is applicable to  $(Q, M_0)$  if there exist a transition  $t \in T$ , and a non-measurable place  $p \in P$  such that:

- i.  $\bullet t = t \bullet = \{p\}$
- ii.  $t \notin T_U$

$(Q, M_0)$  is transformed into  $(Q, M_0)|_{T_6}$  after the application of the following step:  $T \leftarrow T \setminus \{t\}$

A self-loop transition can be removed because its firing leads to the same marking; then it is not useful for the reachability analysis. The constraint regarding the controllable transitions is held to preserve the controllability of the IPN firing sequences. This transformation also preserves liveness and safeness of the transformed IPN.

**Example 8.** In Figure 8 it is shown two IPN representing the original IPN (a) and the resulting IPN after the application of the transformation operator  $T_6$  (b), where the controllable self-loop transitions  $t_3$  and  $t_5$  are removed.

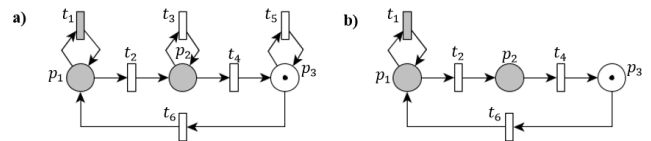


Fig. 8. a) IPN  $(Q, M_0)$ . b)  $(Q, M_0)|_{T_6}$ .

After the application of the operators, the incidence matrix changes. After removing a place  $p_i$  the  $i$ -th column of the matrix  $\varphi$  is removed. Similarly, for a transition  $t_j$  the  $j$ -th row is removed.

## 3.2. Properties of the model transformed by the operators

As mentioned in subsection 3.1, all the operators preserve liveness and safeness. Besides safeness, in order to apply a transformation operator to an already transformed IPN it is necessary that the resulting IPN remains ordinary.





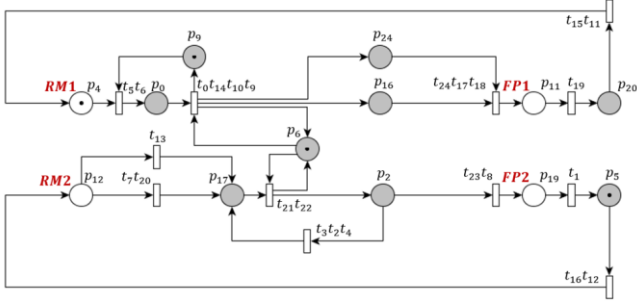


Fig. 10: Resulting IPN after  $\mathcal{T}_1^*$ .

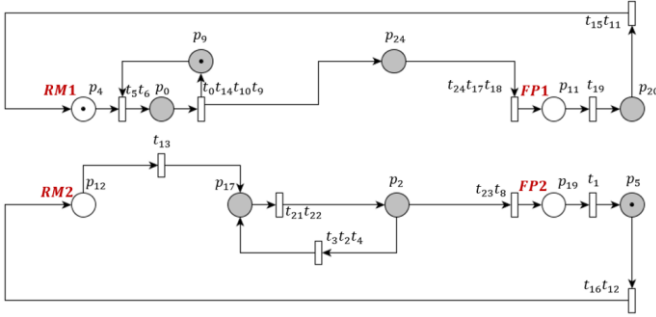


Fig. 11: IPN after the removal of nodes  $p_6$ ,  $t_7$ ,  $t_{20}$ , and  $p_{16}$ .

The removal of the redundant nodes create new application cases; the nodes  $p_2$  and  $p_0$  are removed by the operators  $\mathcal{T}_2$  and  $\mathcal{T}_1$  respectively. After that, the self-loop transition  $t_{21}t_{22}t_3t_2t_4$  and the self-loop place  $p_9$  are removed by the operators  $\mathcal{T}_6$  and  $\mathcal{T}_5$  respectively. Those were the final possible reductions by  $\pi$ , the resulting IPN representing the abstract model  $(Q_A, M_0^A)$  of the process  $(Q_P, M_0)$  with respect to the output symbols of interest, is shown in Figure 12.

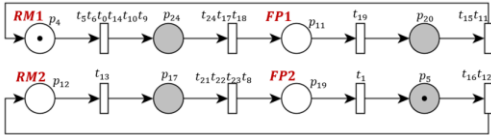


Fig. 12: Resulting IPN from  $(Q_P, M_0)$  of Example 9.

It is easy to verify that both the observable language and the controllable observable language are preserved. Furthermore, the reachability set is highly decreased: the reachability graph of the original IPN has 315 states with 890 arcs, while that of the abstract model has 16 states with 32 arcs.

The abstract IPN of this example (and many others) has been obtained with the help a software developed using the proposed technique. This is reported in [Chavarin, 2017], where the abstract models are used in regulation control scheme.

#### 4. CONCLUSIONS

A model abstraction technique for safe Interpreted Petri Nets (IPN) has been proposed. The technique is oriented to the analysis and control of Discrete Event Systems (DES); it can be used to simplify the controllability analysis and the controller synthesis in a regulation control framework. The abstraction method includes a kit of local transformation operators that simplifies stepwise the structure of an initial IPN

by reducing its size progressively. The abstraction is done with respect to a subset of observable places of interest, by preserving the controllability feature of the paths in the transformed structures. Thus, the abstracted IPN preserves the observable language and the controllable observable language with respect to the original IPN.

#### REFERENCES

- Berthelot, G. (1986) “Checking Properties of Nets Using Transformations”, in Rozenberg, G.(ed.) APN 1985, LNCS, vol 222, pp. 19–40.
- Chavarin-Aguirre, P. (2017) “Interpreted Petri Nets Abstraction for Regulation Control of Discrete Event Systems”. Thesis report, Cinvestav Unidad Guadalajara, December 2017.
- Desel, J. (1990) “Reduction and design of well-behaved concurrent systems,” Lecture Notes Comp. Sci., vol. 458, pp. 166–181.
- Desel, J., J. Esparza (1995) “Free choice Petri nets”, Cambridge tracts in theoretical comp. sc, Vol. 40.
- Esparza, J. (1994) “Reduction and synthesis of live and bounded free choice Petri nets,” Inf. Comput., vol. 114, no. 1, pp. 50–87.
- Lee-Kwang, H. and J. Favrel (1985), “Hierarchical reduction method for analysis and decomposition of Petri nets”, IEEE Trans. Syst. Man Cybern., Vol. 15, no 3, 272–280.
- Lee-Kwang,H., J. Favrel, and P. Baptiste (1987), “Generalized Petri net reduction methods”, IEEE Trans. Syst., Man, Cybern., vol.17, no. 2, pp. 297–303, Apr. 1987.
- Lutz-Ley, A. and E. López-Mellado (2013), “State-Stability Analysis of Discrete Event Systems using Petri-net Branching Processes”.IFAC Workshop on Dependable Control of Discrete Event Systems, York, U. K.
- Murata, T. (1989) “Petri nets: Properties, analysis, and application”, in Proc. IEEE, Vol. 77, No.4, pp. 541-580.
- Santoyo-Sanchez, A., A. Ramírez-Treviño, C. De Jesús Velásquez, L.I. Aguirre-Salas (2008), “Step State-feedback Supervisory Control of Discrete Event Systems using Interpreted Petri Nets”, in Proc.13th IEEE Int. Conf. on Emerging Technologies and Factory Automation, pp. 926 – 933, Hamburg, Germany.
- Silva, M. (1985). “Las Redes de Petri en la Automática y la Informática”, Editorial AC, Madrid.