XX (117182.1)

CINVESTAV
Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Observabilidad y síntesis modular de modelos en redes de Petri de Sistemas de Eventos Discretos

Tesis que presenta:
José Israel Rivera Rangel

para obtener el grado de:
Doctor en Ciencias

en la especialidad de:
Ingeniería eléctrica

Directores de Tesis
Dr. Antonio Ramírez Treviño
Dr. Luis Ernesto López Mellado

Guadalajara, Jalisco, Octubre del 2004.

CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Observability and modular synthesis of Petri net models of Discrete Event Systems

Tesis que presenta:
José Israel Rivera Rangel

para obtener el grado de:
Doctor en Ciencias

en la especialidad de:
Ingeniería eléctrica

Directores de Tesis
Dr. Antonio Ramírez Treviño
Dr. Luis Ernesto López Mellado

Guadalajara, Jalisco, Octubre del 2004.

Observabilidad y síntesis modular de modelos en redes de Petri de Sistemas de Eventos Discretos

Tesis de Doctorado en Ciencias
Ingeniería eléctrica

Por:

José Israel Rivera Rangel

Maestro en Ciencias

CINVESTAV Unidad Guadalajara 1998-2000

Directores de Tesis

Dr. Antonio Ramírez Treviño

Dr. Luis Ernesto López Mellado

CINVESTAV del IPN Unidad Guadalajara, Octubre del 2004.

Observability and modular synthesis of Petri net models of Discrete Event Systems

Tesis de Doctorado en Ciencias
Ingeniería eléctrica

Por:
José Israel Rivera Rangel
Maestro en Ciencias
CINVESTAV Unidad Guadalajara 1998-2000

Directores de Tesis
Dr. Antonio Ramírez Treviño
Dr. Luis Ernesto López Mellado

CINVESTAV del IPN Unidad Guadalajara, Octubre del 2004.

Resumen

Esta tesis presenta tres contribuciones importantes en el area del análisis y diseño de Sistemas de Eventos Discretos (DES). La primera es una nueva metodología de modelado que es bottom-up y modular. Se distingue de otras técnicas de síntesis en que se usan dos operadores de composición y no solo el producto síncrono introducido por Ramadge-Wonham en la teoría de control supervisor. Usando el nuevo operador de composición, llamado permisivo,se garantiza que el modelo final obtenido contiene solo comportamientos que son realizables en el sistema.

La segunda contribución de este trabajo es una definición formal de observabilidad que es adecuada para cualquier tipo de sistema (determinista o no determinista) que este descrito en términos de variables de estado.

Finalmente, se presenta una caracterización de los modelos que presentan observabilidad. Esta caracterización provee algoritmos polinomiales que pueden usarse tanto en el analisis como en el diseño de sistemas.

Abstract

This dissertation presents three major contribution on the field of analysis and design of Discrete Event Systems (DES). The first one is a novel modeling methodology which is bottom-up and modular. It differs from other synthesis technique in that two composition operators are used and not only the synchronic product used in the Ramadge-Wonham supervisory control theory but also a novel permissive composition operator. By using these operators it is guarantied that the obtained model will include only the behavior that is feasible within the system. The second contribution of this work is a formal definition of observability which is suitable for any kind of dynamic system (deterministic or nondeterministic) described in terms of state variables. Finally, models exhibiting observability are characterized. This characterizations provide polynomial algorithms that can be used both in the analysis of an exisiting system or in the system design process.

Agradecimientos

Al CONACYT, al CINVESTAV y a la Unidad Guadalajara, por proveer los medios necesarios para la realización de esta investigación.

En particular a mis asesores Dr. Antonio Ramírez Treviño y Dr. Luis Ernesto López Mellado, por transmitirme sus conocimientos y guiar este trabajo.

A mi madre, por haberme brindado todo su apoyo siempre que la he necesitado.

A mi esposa, porque sin su amor no hubiese sido posible sacar adelante este proyecto.

A mi hija, a quien dedico este trabajo.

Y al resto de mi familia, en la que incluyo a tod@s aquell@s que sin llevar mi apellido son mis herma@s.

Observability and modular synthesis of Petri net models of Discrete Event Systems

Presented by

José Israel Rivera Rangel

as partial requirement to obtain the Ph. D. degree.

Directors: Dr. Antonio Ramírez Treviño

and

Dr. Luis Ernesto López Mellado.

September 17, 2004

Contents

Chapter 1

Introduction

Summary: In this chapter it is established that the scope of this dissertation is the observability problem. This problem is loosely defined as computing the state of a systems model using the information provided by the inputs, outputs and structure of the model. Observability is relevant in control and fault detection of dynamic systems as well as to provide a back up of the systems information and to reduce network loads. The literature related to this topic in the field of discrete event systems is reviewed and also, the necessity of a formal modeling tool, as well as a simple modeling methodology is established.

1.1 Overview

This dissertation addresses the observability problem for Discrete Event Systems (*DES*). These systems are characterized because the set of values that their state can take is numerable and their dynamics are even-driven, i.e., the state changes abruptly in response to event occurrences. Therefore, *DES* are commonly described using other modeling formalisms that those used for continuous systems.

The observability problem can be roughly described as the possibility of computing the state of a system from the knowledge of its inputs, and outputs, given that the system model is known.

The relevance of this problem arises from the fact that sensing, processing and transmitting all the information generated by a system can result unpractical, overwhelming or even impossible in many applications. Indeed, assuming that the system output does not contain all the information of the state of system is common in control theory [7][23].

By addressing the observability problem, the minimal amount of information needed for an accurate development of other tasks can be determined, as well as output feedback control laws [34][42]. The device devoted to compute the system state is commonly named observer [29] in control theory and, besides of providing an estimation of the system state, it can be useful for failure detection [14] and as a back up of the system information. Moreover, this devices are commonly used when decentralized control [21] is implemented since each of the controlling devices is assumed to have access to only some portion of the system information and then an observer is created for each controller, so that it can estimate the complete system status.

Being observability a property that depends on the system model, it results important to use a formal modeling tool that allows to properly describe those system characteristics that are relevant for the problem in a framework that allows a compact representation and provides powerful mathematical analysis tools.

Noteworthy among *DES* modeling formalisms are Finite Automata (*FA*) [18], process algebras [2] and Petri Nets (*PN*) [11], because they are widely used for *DES* researchers. *FA* provide a simple graphic representation for systems whose behavior is restricted to an small number of states and that do not present much of parallelism. However, since all the trajectories that the system state can take are explicitly described in this formalism, when the size of the system grows and its behavior is complex, the size of the model presents and exponential explosion. Several attempts have been made in order to cope with this problem [27][6][44]. In most of the cases, these formalisms are equivalent to *PN*, a modeling tool that allows to describe highly concurrent system in a compact graphic representation and provides a simple and sound mathematical background to analyze models. Unlikely, even when process algebras are suitable for building *DES* models from a functional description, there exist few analysis tools for this formalism.

One of the earlier reported modeling techniques for *DES* is [36]. A modular methodology, based on the synchronic product is proposed in that work, using *FA* as formalism. Besides of the state explosion problem mentioned above, the major drawback that this approach exhibits is that a refinement stage, at the end of the modeling process, is required in order to guarantee that the model describes only behaviors that are feasible in the system. Trying to avoid the state explosion problem, this modeling methodology has been translated to the *PN* formalism [14] [20]. However, this approach inherits the need of a refinement stage from the *FA* modeling methodology.

In this dissertation, a novel modeling methodology, which is modular and does not require the final refinement stage is proposed. It uses an extension to *PN* named Interpreted Petri Nets (*IPN*) as modeling formalism. This extension on *PN*, allows to include in the model the input signals associated to the event occurrences and the output signals that the states of the system generated, without modifying the analysis power of *PN*, which occurs when they essentially changed as proposed in works like [22], [24] and [3].

The observability problem in *DES* was first addressed using *FA* as a description formalism [35]. In [26] observability using the formal languages point of view is defined. That concept focuses on determining the conditions necessary for the existence of a device capable of restricting the behavior of the *DES* to a desired subset of event sequences. However, even when the conditions established in that work are held, this does not guarantee that the system state can be computed. Similar observability concepts have been presented in later works also addressing the controllability of *DES* under partial event observation [8] [25][42]. In the *FA* approach, observers are defined either in terms of an algorithm that estimates the initial and current state of the system, assuming that transition events are completely known [4], or [5] as a non-deterministic *FA* which is a copy of the system model where those events that are not observable are ϵ−labeled and that is later reduced to a deterministic *FA*. In [9] resilent observers, that are equivalent to those obstained after reducing the non-deterministic *FA* used by [5], are introduced.

In order to cope with the state explosion problem, research groups throughout the world are increasingly adopting the *PN* formalism for the analysis of observability.

One of the earliest reported works on observability using *PN* is [19], in that work Ichikawa and Hiraishi present an algorithm to compute a set of possible initial markings; it is based on the analysis of the incidence matrix. They first compute the fired transition sequence, then they determine the set of possible initial markings.

Giua [15] addressed the problem of computing the initial marking when the firing transition sequence is fully known. Based on this assumption, an algorithm to determine the initial marking of a *PN* when a macro marking law is known, as well as an observability definition were proposed. This definition, however, does not take advantage of the system's output information, thus many observable *PN* are not characterized.

In this context, Aguirre et al. [1] proposed a *PN* based observer to determine the initial marking of a PN as well as a new observability definition. They argue that since both, the system and the observer are represented by PN, then all structural analysis techniques of PN can be used to determine liveness, boundedness, dead lock-freeness, etc. of the system-observer pair. The observability definition, however, does not exploit the system's output information and the observer is restricted to the binary case.

An extension to the previous work was presented in [37]. In that paper Ramírez et al. address the problem of computing the initial marking of a binary PN when only partial information of the fired transition sequence is available. In that case, both the observability definition and the PN observer take into account the system's output information.

More recently, in [16] different observability definitions were introduced. Most of them are related with the use of the output system signals, leading to an incomplete observability concept; these definitions establish that for any sequence of system outputs, there exists another finite sequence of system outputs such that the system state can be computed. The errors in these definitions appear since the first sequence could become infinite or since they are searching for a special sequence. Thus, even if these sequences exist, they could never be executed, and then the system state could never be computed.

The observability concept given in [40] is closely related to the concept of strong uniform marking observability given by Giua and Seatzu. It is assumed that the initial marking belongs to a given set and it is said that the Petri net is observable if all event sequences provide enough information to uniquely determine the initial state or can be completed to sequences that provide that information. Also in this case the system is allowed to infinitely iterate before its state can be computed.

The definition herein presented overcomes all these difficulties since it includes both input and output system signals, and it is based in some properties of all finite sequences. Using this new definition, the IPN models exhibiting this property are structurally characterized.

The characterization of IPN exhibiting the observability property is based on two important properties: a) event detectability and b) marking detectability. Event detectability of live and bounded IPN is characterized from the net structure, more precisely, the rows of the incidence matrix corresponding to measurable places. Marking detectability of live and bounded IPN is also characterized from the structure, using a basis of T-semiflows and their relationship with a subset of P-semiflows. The characterization of these properties is performed using polynomial time algorithms and are useful in the analysis of existing DES. Also results that can be applied in the design of DES are herein presented. This results are necessary conditions for the models obtained by using the modeling methodology proposed in this dissertation to be observable.

1.2 Objectives of this work

As mentioned earlier, this dissertation addresses the observability problem for DES modeled in terms of IPN. This problem can be, at this point established as follows.

Problem 1 *Computing the initial state of an IPN model given the knowledge of a finite sequence of input signals, using the output information that it generates and the model structure.*

However, the first objective of this work is providing a formal definition of observability that, besides of capturing the meaning of this property as it is understood in dynamic systems theory, can be applied to any dynamic system model given in terms of state variable, even if it is deterministic, non-deterministic, time or event-driven.

Providing necessary and sufficient conditions (or at least sufficient) for observability, that can be verified from the model structure, is the second objective of this thesis. The usefulness of these conditions relays in the fact that dynamic properties definitions commonly refer to the systems evolution, and therefore, verifying these properties directly form the definition, requires describing all the trajectories that the state of the system may take Fortunately, using conditions that only depend on the model structure, polynomial algorithms to verify observability can be proposed.

Another objective of this work, which arises from the fact that having a suitable model of the system is necessary in order to simplify the analysis of dynamic properties, is providing a modular modeling methodology for DES in terms of IPN. The advantages of using this modeling methodology for the analysis of observability is also one of the goals of this dissertation.

1.3 Contains of this work

Chapter 2 presents the basic notions of both PN and IPN, the modeling formalism that will be used in this work, because of its modeling power and its capability to include in the system model both the input and output signals associated to the system.

Chapter 3 introduces a novel modeling methodology for DES in terms of IPN. This methodology is modular bottom-up and allows to clearly distinguish the state variables used to describe the system components. Also a transformation technique that allows to obtain equivalent models with less places is introduced in that chapter.

Chapter 4 presents the formal definition of observability and a geometric characterization of this property is provided. Later, this characterization is traslated to structural properties of a broad class of IPN that allow to derive polynomial algorithms to determine if a given IPN is observable.

Chapter 5 takes advantage of the structure of the models obtained using the proposed modeling methodology to characterize observability and shows that this property is invariant under the transformation technique proposed.

Finally, some concluding remarks are presented in chapter 6.

Chapter 2

Basic Notions

Summary: This chapter presents basic concepts and notation of *PN*, a formal modeling tool for *DES* that allows to obtain compact graphical representation of systems and also provide the possibility of analyzing systems from simple mathematical expressions.

Later *IPN*, the modeling formalism that will be used in this work, are presented. *IPN* is an extension to *PN* that besides of inherit all *PN* features, allow to capture in the model the input and output signals associated to the *DES*.

2.1 Introduction

The most commonly used modeling formalisms in *DES* literature are Finite Automata (*FA*) and Petri Nets (*PN*), even when there are many others that result useful for certain purposes. *FA* provide a simple graphic representation of the behavior of *DES* when they have a small number of states and do not exhibit concurrent behavior, because this causes a prohibitive state explosion. However, when modeling large *DES* exhibiting concurrence and parallelism, *PN* are usually preferred. This is because, while *FA* explicitly describe all the states that the system may reach, *PN* achieve to provide a compact representation of the states by associating the *DES* evolution with a token game. Moreover, *PN* provide a simple and sound mathematical background to work with.

Interpreted Petri Nets (*IPN*) [30] is an extension to *PN* that allows to include in the model the relations among the *DES* input signals and the corresponding *DES* event occurrences, as well as the *DES* output signals with the *DES* states that generate them. This is achieved by associating input and output alphabets to *PN* transitions and *PN* marking, respectively. In this way *IPN*, besides of providing the advantages of *PN*, allow to capture the physical meaning of the model within it.

In this work *IPN* will be used as modeling formalism. This chapter introduces *IPN* formal definition, along with basic concepts and properties that will be useful for the remaining of the work.

2.2 Petri nets

As mentioned earlier, *PN* is a formal *DES* modeling tool that allows to associate the *DES* evolution with a token game. Its structure is described by a graph containing two sets of vertices called places and transitions, which are represented as circles and rectangles, respectively. These vertices are joined by directed and weighted arcs that can go from places to transitions or from transitions to places. Arc weights are always positive integers and, when different from 1, the weight is pictorially represented by a number beside the corresponding arrow.

The formal definition of a *PN* structure is given as follows.

Definition 2 *A Petri Net structure G is a bipartite digraph represented by the 4-tuple $G = (P, T, I, O)$ where:*

- $P = \{p_1, p_2, ..., p_n\}$ *is a finite set of vertices called places,*
- $T = \{t_1, t_2, ..., t_m\}$ *is a finite set of vertices called transitions,*

- $I : P \times T \longrightarrow \mathbb{Z}^+$ *(where* $\mathbb{Z}^+$ *is the set of non-negative integers) is a function representing the weighted arcs going from places to transitions, and*

- $O : P \times T \longrightarrow \mathbb{Z}^+$ *is a function representing the weighted arcs going from transitions to places.*

In PN literature it is common to denote with ${}^\bullet t_j$ the set of all places p_i such that $I(p_i, t_j) \neq 0$ and with $t_j{}^\bullet$ the set of all places p_i such that $O(p_i, t_j) \neq 0$. Analogously, ${}^\bullet p_i$ denotes the set of all transitions t_j such that $O(p_i, t_j) \neq 0$ and $p_i{}^\bullet$ the set of all transitions t_j such that $I(p_i, t_j) \neq 0$.

Using functions I and O, the incidence matrix of a PN can be defined as $C = [c_{ij}]$, where $c_{ij} = O(p_i, t_j) - I(p_i, t_j)$. However, in many cases it is useful to separately consider the arcs coming into places as $C^+ = [c_{ij}^+]$ with $c_{ij}^+ = O(p_i, t_j)$; and the arcs coming into transitions as $C^- = [c_{ij}^-]$ with $c_{ij}^- = I(p_i, t_j)$.

Once that the PN structure is defined, the state of the DES is described in terms of a marking function or vector, which is formally defined as follows.

Definition 3 *The marking function* $M : P \longrightarrow \mathbb{Z}^+$ *is a mapping from each place to the non-negative integers representing the number of tokens (depicted as dots) residing in each place. The marking of a PN is usually expressed as an* $n-$*entry vector, where* n *is the number of places.*

Thus, a state of a DES can be associated to a PN marking and therefore, a DES model in terms of PN is obtained when both, a PN structure and an appropriate marking are established.

Definition 4 *A Petri Net system or simply Petri Net (PN) is the pair* $N = (G, M_0)$*, where* G *is a PN structure and* M_0 *is an initial token distribution, or initial marking.*

Now, the evolution of the DES modeled is described by the marking evolution of the IPN. Marking changes in IPN are based on the following transition enabling and firing rules: in a PN, a transition t_j is enabled at marking M_k if $\forall p_i \in P, M_k(p_i) \geqslant I(p_i, t_j)$; an enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed as $M_{k+1} = M_k + Cv_k$, where $v_k(i) = 0$, $i \neq j$, $v_k(j) = 1$. This equation is called the PN state equation.

The complete behavior of a PN can be computed if the set of all markings reachable from the initial one M_0 is available.

Definition 5 *The reachability set of a PN is the set of all possible reachable marking from* M_0 *firing only enabled transitions; this set is denoted by* $R(G, M_0)$*.*

Moreover, a directed graph, whose nodes are those markings reachable from M_0, can be obtained. This graph is called the reachability graph and is equivalent to a FA model for the DES. Most of PN properties can be directly characterized using the reachability graph. However, in the general case, computing the reachability set and the reachability graph is a hard problem due to the high combinatorial contained in the PN evolution. Therefore, it is desirable to derive characterizations of PN properties based on the PN structure. Basic for this structural characterization of PN properties are the conservative and repetitive components, they are formally established as follows.

Definition 6 *Let N be a PN. Vectors X_i (Y_i) such that $CX_i = 0$, $X_i \geqslant 0$ ($Y_i^T C = 0, Y_i \geqslant 0$) are called T-semiflows (P-semiflows). A T-semiflow X_i (P-semiflow Y_i) is called canonical if the greatest common divisor of all $X(k)$ (all $Y(k)$) is 1. A T-semiflow X_i (P-semiflow Y_i) is called minimal if there exists no other T-semiflow X_j (P-semiflow Y_j) such that $X_j(k) \leq X(k), \forall k$ ($Y_j(k) \leq Y(k), \forall k$). A T-semiflow X_i (P-semiflow Y_i) is called elemental if it is canonical and minimal.*

The set of places (transitions) associated to a P-semiflow Y_i (T-semiflow X_i) is called the P-semiflow (T-semiflow) support.

Definition 7 *The support of a T-semiflow X_i (P-semiflow Y_i) denoted $\langle X_i \rangle$ ($\langle Y_i \rangle$) is a set of transitions $T_i = \{t_j | X_i(j) > 0\}$ (places $P_i = \{p_j | Y_i(j) > 0\}$).*

Intuitively, the support of a P-semiflow Y_i is a set of places such that a weighted addition of the tokens residing inside them is constant along any PN evolution. Analogously, the support of a T-semiflow X_i is a set of transitions that when fired (if this is possible) reach the initial marking once again.

The subnet $\mathbf{X}_i$ generated by a T-semiflow X_i is called a T-component and analogously, the subnet $\mathbf{Y}_i$ generated by a P-semiflow Y_i is a P-component. These component are defined as follows.

Definition 8 *Let X_i be a T-semiflow of a Petri Net $N = (G, M_0)$, the T-component $\mathbf{X}_i$ is a subnet of $G = (P, T, I, O)$ defined by:*
$\mathbf{X}_i = (P_{X_i}, T_{X_i}, I_{X_i}, O_{X_i})$
where $T_{X_i} = \{t_j | X_i(j) > 0\}$,
$I_{X_i} = \{I(p_k, t_j) \in I | X_i(j) > 0\}$,
$O_{X_i} = \{O(p_k, t_j) \in O | X_i(j) > 0\}$ *and*
$P_{X_i} = \{p_k \in P | I(p_k, t_j) > 0 \vee O(p_k, t_j) > 0, t_j \in T_{X_i}\}$. *For a P-semiflow Y_i the P-component $\mathbf{Y}_i$ is analogously defined.*

2.2.1 Petri nets properties

Petri nets properties can be divided in the following classes: dynamic and structural. Dynamic properties are those that depend on both, the PN marking and its structure. Meanwhile structural properties depend only on the structure. Some of these properties that result meaningful for this work will be now established. Further details on these and other properties can be found in [41] and [10].

Dynamic properties

Liveness and boundedness are two of the most important dynamic properties. A PN is said to be live if for every reached marking there exists a fireable sequence of transitions containing every transition. Boundedness implies that the amount of tokens in the PN cannot grow infinitely, even when it is not necessarily constant.

- **Liveness**. A transition $t_k \in T$ is live, for a marking M_0, if $\forall M_k \in R(G, M_0)$, $\exists M_n$ reachable from M_k such that $M_n \xrightarrow{t_k}$. A PN is live if all its transitions are live.
- **Boundedness**. A PN is k-bounded for a marking M_0, if $\forall p_i \in P \; \exists k \in \mathbb{N}$, such that $\forall M_j \in R(G, M_0)$, it holds that $M_j(p_i) \leq k$.

Safeness is an special case of boundedness. A PN is safe, or binary if it is $1-$bounded.

- **Safeness**. A PN is safe or binary if it is $1-$bounded, that is, if $\forall M_k \in R(G, M_0)$ and $\forall p_i \in P$, it holds that $M_k(p_i) \in \{0, 1\}$.

Structural properties

Among the properties that do not depend on the PN marking are those related to the T-semiflows and P-semiflows. For instance, repetitiveness implies that there exists a marking enabling a sequence containing all the PN transitions and depends on T-semiflows. Conservativeness implies that a weighted addition of tokens in the entire PN remains constant for any possible evolution and depends on P-semiflows. These properties are formally defined as follows.

- **Repetitiveness**. A PN is said to be repetitive if there exists a T-semiflow X_i such that $X_i \in \mathbb{N}^m$ with $m = |T|$.
- **Conservativeness**. A PN is said to be conservative if there exists a P-semiflow Y_j such that $Y_j \in \mathbb{N}^n$ with $n = |P|$.

Also structural versions of boundedness and liveness exist, they are established in the following definitions.

- Structural liveness. A PN structure is said to be structurally live if there is an initial marking such that the PN structure is live for that marking.

  Structural boundedness. A PN structure is said to be structurally bounded if it is bounded for any possible initial marking.

2.2.2 Petri nets classification

In order to study PN they have been divided according to the following classes that are based on the PN structure.

1. Marked Graphs. A marked graph is a PN where every place has one input transitions and one output transition, this is, $\forall p_i \in P$, $|^\bullet p_i| = |p_i{}^\bullet| = 1$.

2. State machine. A state machine is a PN where every transition has one input place and one output place, i.e. $\forall t_i \in T$, $|^\bullet t_i| = |t_i{}^\bullet| = 1$.

3. Free-choice nets. A free-choice net is a PN where if two transitions t_i and t_k share an input place p_i, then their sets of input places are equal (this is, $\forall p_i \in p$, if $|p_i{}^\bullet| > 1$, then $\forall t_k, t_j \in p_i{}^\bullet$, $^\bullet t_k = {}^\bullet t_j$).

4. Other nets. Those not included in previous definitions.

A PN is said to be strongly connected when there is a path of arcs joining every pair of vertices. Strongly connected marked graphs are covered by a T-semiflow, and strongly connected state machines by a P-semiflow. Therefore, most of previous PN properties follow directly from the structure for these PN.

Verifying these properties in free-choice and other nets is in the general case a computationally hard problem [10]. However, these PN are insteresting because they allow to describe more complex DES behavior in compact representations.

2.3 Interpreted Petri nets

Interpreted Petri Nets (IPN) are an extension to PN that allow to associate input and output signals to DES models. This is carried out by including an input alphabet, a transition labeling function that describes the input signals required by the DES for an event occurrence and an output function assigning output signals to PN markings.

Definition 9 *An Interpreted Petri Net (IPN) is the 4-tuple $Q = (N, \Sigma, \lambda, \varphi)$ where:*

- $N = (G, M_0)$ *is a PN.*

- $\Sigma = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ *is the input alphabet of the net, where α_i is an input symbol.*

- $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$ *is a labeling function of transitions with the following restriction: $\forall t_j, t_k \in T$, $j \neq k$ if $I(p_i, t_j) = I(p_i, t_k) \neq 0$ and both $\lambda(t_j)$, $\lambda(t_k) \neq \varepsilon$, then $\lambda(t_j) \neq \lambda(t_k)$. In this case ε represents an internal event of the system.*

- $\varphi : R(G, M_0) \rightarrow \{\mathbb{Z}^+\}^q$ *is an output function that associates to each marking in* $R(G, M_0)$ *an output vector and* q *is the number of available outputs associated to places.*

Remarks:

- In this work (Q, M_0) will be used instead of $Q = (N, \Sigma, \Phi, \lambda, \varphi)$ to emphasize the fact that there is an initial marking in an IPN.

- If not otherwise stated, it will be assumed that models for DES are ordinary IPN (i.e. $I(p_i, t_j), O(p_i, t_j) \in \{0, 1\}, \forall p_i, t_j$).

- This work focuses on the case where the function φ is a $q \times n$ matrix $\varphi = [\varphi_{ij}]$, where q is the number of places representing states with different output signals associated in the DES and n is the number of places in the model (G, M_0). Each column of this matrix is either an elementary or a null vector and $\varphi(i, j) = 1$ if the output corresponding to a state such that $M(p_j) = n$ contains n times the $i-th$ output signal.

- Identical transitions are not allowed, i.e., it is assumed that $\forall t_i, t_j$ such that $t_i \neq t_j$, $\lambda(t_i) = \lambda(t_j)$, it holds that $C(\cdot, i) \neq C(\cdot, j)$. This is not a major constraint because those transitions are redundant.

All definitions and results on PN previously described can be extended to IPN. However, in IPN the enabling and firing rules are defined as follows:

A transition $t_j \in T$ of an IPN is enabled at marking M_k if $\forall p_i \in P$, $M_k(p_i) \geqslant I(p_i, t_j)$. If $\lambda(t_j) = a_i \neq \varepsilon$ is present and t_j is enabled, then t_j must fire. If $\lambda(t_j) = \varepsilon$ and t_j is enabled then t_j can be fired. When an enabled transition t_j is fired in a marking M_k, then a new marking M_{k+1} is reached. This fact is represented as: $M_k \xrightarrow{t_j} M_{k+1}$. The new marking M_{k+1} can be computed using the PN state equation as $M_{k+1} = M_k + Cv_j$, with C and v_j defined as in a PN.

The state equation of an IPN is obtained including the $k-th$ output to the state equation of a PN:

$$\begin{aligned} M_{k+1} &= M_k + Cv_k \\ y_k &= \varphi M_k \end{aligned} \tag{2.1}$$

Note that by definition of λ, IPN are deterministic[18] over labeled transitions, i.e. two transitions with the same associated input symbol (different from symbol ε) cannot have the same input places. However, they can be nondeterministic[18] over unlabeled transitions (those t_j such that $\lambda(t_j) = \varepsilon$).

2.3.1 Manipulated transitions, measurable and computable places

According to functions λ and φ, transitions and places of an IPN (Q, M_0) can be classified as follows.

Definition 10 *If $\lambda(t_i) \neq \varepsilon$ the transition t_i is said to be manipulated. Otherwise it is non-manipulated.*

A place $p_i \in P$ is said to be measurable if the $i-th$ column vector of φ is not null (i.e., $\varphi(\cdot, i) \neq 0$). Otherwise it is non-measurable. If in addition to $\varphi(\cdot, i) \neq 0$, it holds that $\forall j$, such that $i \neq j$, $\varphi(\cdot, i) \neq \varphi(\cdot, j)$, then p_i is said to be computable, otherwise p_i is non-computable.

This means that a transition is said to be manipulated if its firing requires a given input signal to be present. A place is measurable if it has an output signal associated, and computable if that output signal is different from those associated to all other places. Notice that this implies that the marking of computable places can be directly obtained from the output signal.

Through this work, the measurable places of an IPN are depicted as clear circles and the non-measurable ones as dark circles. Analogously manipulated transitions are depicted as clear bars and non-manipulated transitions as dark bars.

2.3.2 Languages, sequences, and vectors

The set of all sequences of transition firings of an IPN is called the IPN firing language and every transition sequence has a vector associated.

Definition 11 *A firing sequence of an IPN (Q, M_0) is a sequence $\sigma = t_i t_j ... t_k ...$ such that $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ... M_w \xrightarrow{t_k} ....$ The set of all firing sequences is called the firing language $\mathcal{L}(Q, M_0) = \{\sigma | \sigma = t_i t_j ... t_k ... \text{ and } M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ... M_w \xrightarrow{t_k} ...\}$.*

Let $\sigma = t_i t_j t_k ...$ be a firing sequence. The Parikh vector $\overrightarrow{\sigma} : T \longrightarrow \mathbb{Z}^+$ of σ maps every transition $t \in T$ to its number of occurrences in σ.

By associating the input symbol present while the output of the IPN changes with the new output symbol, the sequences of input-output symbols are obtained.

Definition 12 *A sequence of input-output symbols of (Q, M_0) is a sequence $\omega = (\alpha_0, y_0)(\alpha_1, y_1) \cdots (\alpha_n, y_n)$, where $\alpha_j \in \Sigma \cup \{\varepsilon\}$ and α_{i+1} is the current input of the IPN when its output changes from y_i to y_{i+1}. It is assumed that $\alpha_0 = \varepsilon, y_0 = \varphi(M_0)$ and (α_{i+1}, y_{i+1}) belongs to the sequence ω when:*

- *(α_i, y_i) belongs to ω,*

- *$y_{i+1} \neq y_i$, and*
- *y_{i+1} is the output symbol occurring immediately after y_i.*

Notice that, according to the IPN firing rule for non-manipulated transitions, a change in the input while an input symbol α_i is present, might be caused by the firing of a transition t_j such that $\lambda(t_j) \neq \alpha_i$.

The relevance of these sequences resides in the fact that they, together with the IPN structure, provide all the required information used to compute an IPN marking in the observability problem that will be later discussed.

Later, the following sets of sequences of input-output symbols will be useful for the characterization of observable IPN.

Definition 13 *Let (Q, M_0) be an IPN. The set of all sequences of input-output symbols of (Q, M_0) will be denoted by $\Lambda(Q, M_0)$. In particular, the set of all input-output sequences of length greater or equal than n will be denoted by $\Lambda^n(Q, M_0)$, i.e., $\Lambda^n(Q, M_0) = \{\omega \in \Lambda(Q, M_0) : |\omega| \geq n\}$.*

Also the sequences of input-output symbols leading to blocked markings or marking that do only enable self-loop transitions are relevant in the observability analysis.

Definition 14 *The set of all input-output sequences leading to an ending marking in the IPN (markings enabling no transition or only self-loop transitions) is denoted by $\Lambda_B(Q, M_0)$, i.e., $\Lambda_B(Q, M_0) = \{\omega \in \Lambda(Q, M_0) \mid \exists \sigma \in \Omega(\omega)$ such that $M_0 \xrightarrow{\sigma} M_j$ and if $M_j \xrightarrow{t_i}$ then $C(\cdot, t_i) = 0\}$.*

According to the IPN definition any input signal can be applied at any marking. This implies that not every input signals does necessarily produce transition firings. Moreover, in consequence to the existence of non-manipulated transitions, the sequences of transitions that fire while a input signal is present are not uniquely determined by the input.

In addition, the fact that the same output signal might be associated to many different markings implies that the output changes produced by a transition sequence firing do not determine the transition sequence fired.

In resume, nor input signals, nor output signals can uniquely determine, in the general case the transition sequence fired in an IPN. Therefore, sequences of input-output symbols does not uniquely determine the sequences of transitions firing in the IPN.

Analogously, different firing sequences may be related to the same sequence of input-output symbols. The following definition introduces the set of firing sequences that may be produced by a given sequence of input-output symbols and the set of input-output symbols related to a firing sequence.

Definition 15 *If $\omega = (\alpha_0, y_0)(\alpha_1, y_1) \cdots (\alpha_n, y_n)$ is a sequence of input-output symbols, then the firing sequence $\sigma \in \pounds(Q, M_0)$ whose firing actually generates ω is denoted by σ_ω. The set of all possible firing sequences that could generate the word ω is $\Omega(\omega) = \{\sigma | \sigma \in \pounds(Q, M_0)$ and the firing of σ produces $\omega\}$.*

Inversely, if σ is a firing sequence, then $W(\sigma)$ is the set of all sequences of input-output symbols ω that could have been generated by σ, i.e., $W(\sigma) = \{\omega \in \Lambda(Q, M_0) \,|\sigma \in \Omega(\omega)\}$.

The following definition relates a sequence of input-output symbols ω with the sequences of markings that an IPN may reach by the occurrence of ω.

Definition 16 *Let $\omega = (\alpha_0, y_0)(\alpha_1, y_1)\cdots(\alpha_n, y_n) \in \Lambda(Q, M_0)$ be a sequence of input-output symbols. The marking sequence set corresponding to ω is defined as*

$$\begin{array}{c} S_\omega = \{M_0 M_1 \cdots M_k | M_i \in R(Q, M_0), \\ M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \cdots \xrightarrow{t_m} M_k \wedge \sigma_\omega = t_i t_j \cdots t_m \in \Omega(\omega)\} \end{array} \tag{2.2}$$

Consider the IPN depicted in figure 2.1. This is a strongly connected state machine IPN. Transitions t_1, t_2, t_3 and t_5 are manipulated and t_4 is non-manipulated. Places p_1 and p_2 are measurable and p_3 and p_4 are non-measurable. If the output signals associated to p_1 and p_2 are different, then they are also computable and otherwise non-computable. The IPN has only one elemental P-semiflow $Y_1 = [1111]^T$ whose support is the set of all places in the IPN and the corresponding P-component is the entire IPN. Meanwhile, there exist two elemental T-semiflows $X_1 = [11100]^T, X_2 = [00111]^T$ and, because of the existence of Y_1 and $X_3 = [11211]^T$ the IPN is both conservative and repetitive. The initial marking (depicted in the figure) is $M_0 = [0100]^T$ which makes the IPN live, since every transition can be infinitely often fired from that marking. The firing language is $\{t_1, t_4, t_1t_2, t_4t_5, t_1t_2t_3, t_4t_5t_3, t_1t_2t_3t_1, t_1t_2t_3t_4, ...\}$ and some sequences of input-output symbols are presented in the following table along with their corresponding transition sequence and marking sequence

Input-output symbols sequence	Transition sequence	Marking sequence
$(\varepsilon, [01])$	ε	$[0100]$
$(\varepsilon, [01])(a, [10])$	t_1	$[0100][1000]$
$(\varepsilon, [01])(\varepsilon, [00])$, $(\varepsilon, [01])(a, [00])$, $(\varepsilon, [01])(b, [00])$, $(\varepsilon, [01])(c, [00])$	$\left\{\begin{array}{l} t_4 \\ t_4t_5 \end{array}\right.$	$\left\{\begin{array}{l} [0100][0001] \\ [0100][0001][0010] \end{array}\right.$
$(\varepsilon, [01])(a, [10])(b, [00])$	t_1t_2	$[0100][1000][0010]$
$(\varepsilon, [01])(a, [10])(b, [00])(c, [01])$	$t_1t_2t_3$	$[0100][1000][0010][0100]$
$(\varepsilon, [01])(\varepsilon, [00])(c, [01])$, $(\varepsilon, [01])(a, [00])(c, [01])$, $(\varepsilon, [01])(b, [00])(c, [01])$, $(\varepsilon, [01])(c, [00])(c, [01])$	$t_4t_5t_3$	$[0100][0001][0010][0100]$
$\vdots$	$\vdots$	$\vdots$

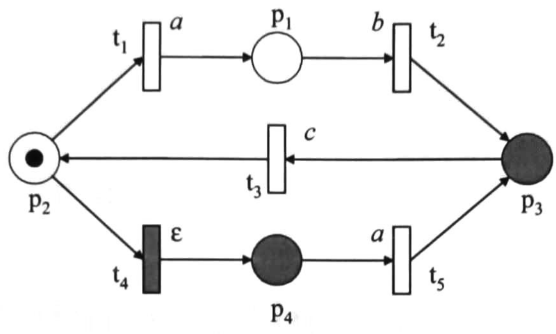

Figure 2.1: Binary, strongly connected state machine *IPN*.

Notice that transition sequence t_4 has associated the input-output sequences $(\varepsilon,[01])\,(\varepsilon,[00])$, $(\varepsilon,[01])\,(a,[00])$, $(\varepsilon,[01])\,(b,[00])$, $(\varepsilon,[01])\,(c,[00])$ because it may fire while any input symbol ε, a, b, or c is present. Sequence t_4t_5 has the same input-output sequences associated because the firing of t_5 causes no change in the output signal.

2.4 Conclusions

This chapter presented the formal definition of *PN* along with some basic properties that will be useful in this work. Also *IPN*, and extension to *PN* that allows to capture the physical meaning of the model within it has being formally introduced. Moreover, languages and sequences that arise from associating input and output signals to *IPN* evolution have being presented, as well as the relations existing among them.

Chapter 3

Modular synthesis

Summary: In this chapter a modeling methodology for *DES* in *IPN* terms is provided. This modeling methodology is based on separately describing the behavior of *DES* components in terms of state variables called modules and later composing the obtained modules to build the model for the entire system.

Also a transformation technique that allows to reduce the amount of places used to represent a *DES* is provided.

3.1 Introduction

A fine survey of the existing modeling methodologies for *DES* in *PN* terms is presented in [20]. Most of these works are extensions to modeling methodologies available for *FA*. They are based on separately representing parts of the system and later composing these parts using the synchronous product. Using these methodologies, the resulting model might contain state trajectories that are not feasible in the system. Therefore, a tuning stage where all system trajectories are compared with those that the model presents is necessary. This, besides of being subject to imprecisions, requires the complete enumeration of the states obtained, becoming prohibitive for large *DES*.

Herein a modeling methodology that allows to represent an existing *DES* in terms of *IPN* is presented. It is a bottom-up methodology which is also based in separately describing the behavior of *DES* components in terms state variables represented as simple *PN* called modules. Those modules are later composed using two basic operators that are herein defined: synchronous and permissive compositions. Synchronous composition, as the synchronous product of other methodologies, is used to represent that events of different modules, or components may be synchronously executed, or in other words, may be local representations of the same global event. The novel permissive composition allows to avoid the tuning stage necessary in other methodologies.

The derived modules representing bounded state variables with the methodology herein proposed are binary and ordinary *PN* state machine. This fact could leads to large *PN* models. To overcome this problem a transformation technique, used to reduce the model size, is also presented in this chapter. This size reduction is achieved by using non-binary markings as well as weighted arcs.

3.2 Modular description

Figure 3.1 shows a diagram of a clinic. There are four doctor offices in this clinic, and a service corridor at the back of the offices. Patients arrive to a reception area, waiting there to receive attention until a doctor office is available. For hygienic reasons, cleaning is mandatory before any patient enters to an office. In the cleaning process a sterilizing device (*SD*) is used. This *SD* should be left at the service corridor while it is not being used. However, from time to time, doctors keep it into their offices while they are attending patients. This causes a delay in the attention to patients because, meanwhile, none other office can be cleaned. The clinic administrator believes that because of this delay, annoyed of waiting, many patients leave before they receive medical attention. In order to determine if a second *SD*, or a better usage of the existing one may reduce the time that patients need to wait to be treated and in consequence, the amount of them leaving without attention, the clinic administrator is trying to obtain a

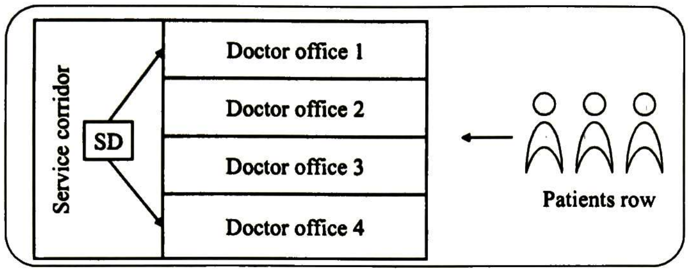

Figure 3.1: Clinic diagram.

model for the clinic. He places himself at the reception area from where he can see if an office is clean or if its occupied. However, he cannot determine if an unoccupied office is dirty or being cleaned. At this position, he can also decide the usage of available offices.

Describing the entire behavior of a large system (consider for instance a clinic with 40 offices) in a single step might become prohibitive. For that reason, the modeling methodology proposed in this work is based on a divide and conquer strategy. It is proposed to firstly determine those components of the system that are relevant for the model and then establish which are the aspects of those components that have to be captured by the model. Then, separated descriptions of the behavior of this aspects (herein called state variables) can be obtained, reducing the complexity in the modeling process.

In the clinic example, for instance, according to the administrator point of view, the patients waiting line, the doctor offices, and the SD are considered as the relevant components of the system. The length of the waiting line, the status of the offices and both, the location and the occupancy of the SD can be considered as the aspects of the components or the state variables that must be included in the model.

In order to obtain a representation of the behavior of each state variable, it is desirable to determine the range of values that variables may have. In other words, it has to be determined which are the values that will be given to the status of an office, to the SD location, and its occupancy, as well as to the line length.

According to the clinic description, four values are enough to describe the status of an office: ready, occupied, dirty and being cleaned. There are five different locations where the SD can be: at office 1, at office 2, at office 3, at office 4 and at the service corridor. Its occupancy can be simply described as idle and busy; and finally the length of the waiting line can be described by a nonnegative integer.

Once that a system's state variable and its range of values have been determined, its behavior can be represented in terms of a PN. In order to do this, each of the values that it can take has to be associated to a PN marking. This can be easily achieved in the following way: If the range of values is finite, a *PN* place for each value can be created and the marking holding one token in that place and none in the remaining can be associated to the corresponding value. When the range of values is not bounded, each value can be associated to a nonnegative integer and all be represented as the marking of a single place holding the corresponding amount of tokens.

In the clinic example four places are necessary to represent the corresponding possible values of each office, five more to describe the *SD* location, two for its occupancy and one for the row length.

The following step to represent the behavior of a state variable is to establish how the events change their values and represent them in terms of transitions that cause the corresponding change in the *PN* marking when firing. For instance, the entrance of a patient increases the value of the line length, therefore a transition representing this entrance, along with an arc going from the transition to the place that describes the row length must be created. Analogously, a transition representing the leaving of a patient from the row, either to be attended or to leave the clinic, must be created. The firing of this transition must decrease the amount of tokens in the place representing the line length, thus an arc going from the place to the transition should be added.

Transitions representing the movement of the *SD* along the clinic, and the beginning and ending of an office cleaning must be added to the model of the *SD* location and occupancy respectively. Also, in each model of the status of an office, transitions representing the entrance of a patient, the patient leaving and the beginning and ending of the cleaning must be added.

Finally, to obtain a complete description of each state variable, its initial value must be represented. This is, if the *SD* is currently at the service corridor then the corresponding place and no other in that model must be holding a token. If its idle, then the place representing this value should have a token, and so on.

The entire *PN* models for these state variables, hereafter called modules are depicted in figures 3.2 and 3.3. It has been assumed that the *SD* is currently idle at the service corridor, that there are three patients in the waiting line and that all but office 2, which is occupied, are ready to receive patients.

The algorithm that has been followed to obtain these modules is now formally established:

Algorithm 17 *Building PN modules*

Input: *A description of a system behavior.*
Output: *A set of isolated PN modules,* $\mathfrak{M}^i_j$.

---

1. ***System components.-*** *The system components must be identified and named. Then, a finite set* SYSTEM_COMPONENTS= $\{sc_1, sc_2, \ldots, sc_n\}$ *of these names must be created. A system component could be a valve, a motor, a system resource, etc.*

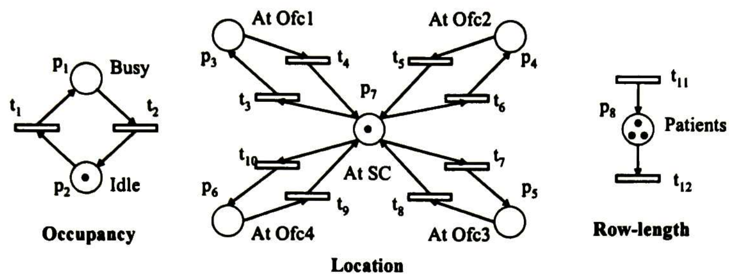

Figure 3.2: *PN* modules of state variables Occupancy, Location and Row-length.

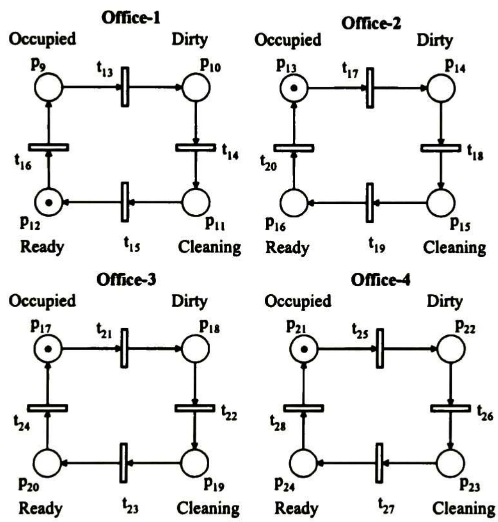

Figure 3.3: *PN* modules of state variables Office-1, Office-2, Office-3 and Office-4.

2. ***State variables.-*** *For each system component, the different variables needed to represent its behavior must be chosen. In other words, the finite set* STATE_VARIABLES$_i$ $= \{sv_1^i, sv_2^i, \ldots, sv_m^i\}$ *associated to the system component* $sc_i \in$ SYSTEM_COMPONENTS *must be built. These variables could represent the position (for instance a valve position), velocity, voltage, etc. of each system component.*

3. ***Set of values.-*** *For each state variable* $sv_j^i \in$ STATE_VARIABLES$_i$*, the range of possible values must be stated. When the number of possible values is finite, then the state variable is bounded, otherwise the state variable is unbounded. For each bounded state variable* sv_j^i $\in$STATE_VARIABLES$_i$ *determine the set* VALUE$_{sv_j^i}$ $= \left\{val_1^{ij}, val_2^{ij}, \ldots \; val_p^{ij}\right\}$ *of possible values of* sv_j^i*. For instance, the variable "valve_position" may take two values: "Open", "Closed". Create a set* UNBOUNDED$_i$ *containing all unbounded state variables* sv_j^i *of system component* sc_i*. Associate the set of non-negative integers to each unbounded state variable.*

4. ***Codification.-*** *The values in each set* VALUE$_{sv_j^i}$*, when* sv_j^i *is bounded, must be represented in terms of PN markings. This can be easily achieved if binary places are used. Thus, for each bounded state variable* sv_j^i $\in$STATE_VARIABLES$_i$ *a set* $P_{sv_j^i} = \left\{p_1^{ij}, p_2^{ij}, \ldots \; p_n^{ij}\right\}$ *of places such that* $\left|\text{VALUE}_{sv_j^i}\right| = \left|P_{sv_j^i}\right|$ *must be created. The marking of these places is binary and mutually exclusive. Then,* $M(p_z^{ij}) = 1$ *means that the variable* sv_j^i *takes the value* val_z^{ij}*. For each state variable* sv_j^i *in each* UNBOUNDED$_i$ *create a place* p^{ij} *to represent* sv_j^i*. The marking of each of these places will be a non-negative integer number and* $M(p^{ij}) = k$ *means that* sv_j^i *takes value related to* k*.*

5. ***Event modeling.-*** *For each pair of values* val_m^{ij}*,* val_n^{ij} *such that the state variable* sv_j^i *could change from value* val_m^{ij} *to value* val_n^{ij}*, a transition* t_{mn}^{ij} *must be created. Then, one arc going from place* p_m^{ij} *to transition* t_{mn}^{ij} *and one arc going from transition* t_{mn}^{ij} *to place* p_n^{ij} *must be added. If* sv_j^i *is an unbounded variable, then create a transition* t_{Ik}^{ij} *and an arc weighted* n *going from* t_{Ik}^{ij} *to the place* p^{ij} *representing* sv_j^i*, for each event* ev_k *that increases the value of* sv_j^i *in* n *units. Analogously, create a transition* t_{Dk}^{ij} *and an arc weighted* n *going from* p^{ij} *to* t_{Ik}^{ij} *for each event* ev_k *that decreases the value of* sv_j^i *in* n *units.*

6. ***Initial marking.-*** *For places representing bounded state variables the initial marking is defined as:* $M_0(p_m^{ij}) = 1$ *if the initial value of the variable* sv_j^i *is* val_m^{ij} *and* $M_0(p_m^{ij}) = 0$ *otherwise. For unbounded state variables the initial marking is defined as:* $M_0(p_m^{ij}) = k$ *if the initial value of the variable* sv_j^i *is the value related to* k*. The built PN, for the state variable* sv_j^i*, is named the PN module* $\mathfrak{M}_j^i$*.*

Applying this algorithm, isolated *PN* modules $\mathfrak{M}_j^i$ for each state variable sv_j^i are built. Notice that these models are state machines which makes them simple to analyze.

3.3 Module composition

In order to obtain the model for the entire system, the relations among state variables must be determined and represented by composing the *PN* modules.

Relations among state variables can be classified into synchronous and permissive relations. There exists a synchronous relation between two state variables when there are transitions in their respective modules representing the same event occurrence. For instance, in each office module there exists a transition that represents the entrance of a patient. Each of these transitions represent a different event occurrence and therefore, each one of them is related to an occurrence of the event represented by the transition that removes a patient from the row. Similarly, each office module contains a transition representing the beginning of the cleaning process. Each one of these transitions represents a different event occurrence and is related to an occurrence of that event in the *SD* occupancy module. The same situation holds for those transitions representing the cleaning process ending in the *SD* occupancy and office modules.

3.3.1 Composition operators

In order to capture a synchronous relation in the model, those transitions representing several different occurrences of the same event must be replicated to have a copy available for each occurrence. Then, transitions of different modules representing the same event occurrence must be merged into a single one. This transition merging achieves to represent with a single transition a single event occurrence and then, the transition firing produces the corresponding change in the value of the state variables involved.

A permissive relationship occurs when a specific value of a state variable is needed for the occurrence of an event, and that event occurrence does not change the value of the variable. For instance, assuming that the *SD* can only be retrieved from an office when the office is ready, then the occurrence of the event is tuned by the value of the state variable representing the office and, however, the removal of the *SD* from the office does not change the office status. In a similar way, assuming that the *SD* only gets into an office when it is dirty, the occurrences of the entrance of the *SD* to the offices are related with the value "*dirty*" of the office status. In order to capture these relationships into model, the place representing the required value and the transitions representing the event occurrences must be joined by a bidirectional arc. In this way, transitions can fire only while a token is held into the place and these firings do not change the token location.

Therefore, synchronous and permissive relations can be captured by the model by composing the PN modules that describe state variables. In order to formally establish these compositions, an arbitrary set of labels for places and transitions is introduced.

Labeling of nodes

Definition 18 *Let (G, M_0) be an PN and* LABELS$=\{l_1, ..., l_w\}$ *be a set of labels arbitrarily defined. (G, M_0) is a labelled PN if there exists a function $tLab : T \longrightarrow$* LABELS *associating one label to each transition and a function $pLab : P \longrightarrow 2^{\text{LABELS}}$ associating a subset of labels to each place.*

Using this set of labels, synchronous composition can be defined as follows:

Synchronous composition

Definition 19 *Let (G_1, M_0^1) and (G_2, M_0^2) be two labelled PN, where* LABELS$_1$ *is not necessarily different from* LABELS$_2$. *The synchronous composition of (G_1, M_0^1) and (G_2, M_0^2) is the labelled PN $\left(G_{1\|2}, M_0^{1\|2}\right)$ given by*

$P_{1\|2} = P_1 \cup P_2,$

$T_{1\|2} = (T_1 - cT_1) \cup (T_2 - cT_2) \cup cT,$

$M_0^{1\|2} = M_0^1 \cup M_0^2,$

$pLab_{1\|2} = pLab_1 \cup pLab_2,$

$$tLab_{1\|2}(t_j) = \begin{cases} tLab_1(t_j) & \text{if } t_j \in T_1 \cup cT \\ tLab_2(t_j) & \text{if } t_j \in T_2 \end{cases}$$

$$I_{1\|2} = I_1|_{\{(p_i,t_j)|t_j \in T_1 - cT_1\}} \cup I_2|_{\{(p_i,t_j)|t_j \in T_2 - cT_2\}} \cup \left\{ ((p_i,t_j),1) \,\middle|\, \begin{array}{c} t_j \in cT_1 \text{ and } I_1(p_i,t_j) = 1 \\ or \\ t_j \in cT_2 \text{ and } I_2(p_i,t_j) = 1 \end{array} \right\}.$$

$$O_{1\|2} = O_1|_{\{(p_i,t_j)|t_j \in T_1 - cT_1\}} \cup O_2|_{\{(p_i,t_j)|t_j \in T_2 - cT_2\}} \cup \left\{ ((p_i,t_j),1) \,\middle|\, \begin{array}{c} t_j \in cT_1 \text{ and } O_1(p_i,t_j) = 1 \\ or \\ t_j \in cT_2 \text{ and } O_2(p_i,t_j) = 1 \end{array} \right\},$$

where

$cT_1 = \{t_k \in T_1 | \exists t_i \in T_2, tLab_1(t_k) = tLab_2(t_i)\},$

$cT_2 = \{t_k \in T_2 | \exists t_i \in T_1, tLab_2(t_k) = tLab_1(t_i)\},$

$cT = t_k | \exists t_i \in cT_1, \exists t_j \in cT_2, tLab_1(t_i) = tLab_2(t_j)$, *and*

$f|_A$ *means the function f restricted to the set A.*

From previous definition it follows immediately that synchronous composition is commutative and associative. The permissive composition uses the set of labels associated to places and transitions to create self-loops between places and transitions related. This is formally established as follows:

Permissive composition

Definition 20 *Let (G_1, M_0^1) and (G_2, M_0^2) be two labelled PN, where* LABELS$_1$ *is not necessarily different from* LABELS$_2$. *The permissive composition of (G_1, M_0^1)*

and (G_2, M_0^2) *is the labelled PN* $\left(G_{1\diamond 2}, M_0^{1\diamond 2}\right)$ *given by*

$P_{1\diamond 2} = P_1 \cup P_2$,
$T_{1\diamond 2} = T_1 \cup T_2$,
$M_0^{1\diamond 2} = M_0^1$,
$I_{1\diamond 2} = I_1 \cup I_2 \cup$

$$\left\{ ((p_k, t_j), 1) \;\middle|\; \begin{array}{c} p_k \in P_{1\diamond 2}, t_j \in T_{1\diamond 2} \text{ and} \\ tLab_1(t_j) \in pLab_2(p_k) \text{ or} \\ tLab_2(t_j) \in pLab_1(p_k) \end{array} \right\},$$

$O_{1\diamond 2} = O_1 \cup O_2 \cup$

$$\left\{ ((p_k, t_j), 1) \;\middle|\; \begin{array}{c} p_k \in P_{1\diamond 2}, t_j \in T_{1\diamond 2} \text{ and} \\ tLab_1(t_j) \in pLab_2(p_k) \text{ or} \\ tLab_2(t_j) \in pLab_1(p_k) \end{array} \right\},$$

$tLab_{1\diamond 2} = tLab_1 \cup tLab_2$, *and*
$pLab_{1\diamond 2} = pLab_1 \cup pLab_2$.

Clearly, also the permissive composition is both associative and commutative. By means of synchronous and permissive compositions, the algorithm that establishes the module compositions above described results as follows:

Algorithm 21 *PN Module composition*

Input: *A set of isolated PN modules,* $\mathfrak{M}_j^i$.
Output: *A labelled PN* (G, M_0) *representing the behavior of the DES.*

---

1. ***Labelling modules.-*** *For each PN module* $\mathfrak{M}_j^i$, *the functions* $tLab_j^i$ *and* $pLab_j^i$ *must be defined. These functions are arbitrary defined for each module;* $tLab_j^i$, *however, must be injective. The labelling process associates a physical meaning to PN nodes.*

   If a transition has a physical meaning that is tuned by the occurrence of other variables in the system, then this transition must be replicated as many times as its physical meaning is tuned and each instance of the transition must have a different label to preserve the injectiveness. For instance, if a machine can be unloaded by `robot 1` *or by* `robot 2`, *then the transition representing the unloading action must be duplicated.*

   The set of labels of a place p_m *of* $\mathfrak{M}_j^i$ *must contain the label of a transition* t_n *of* $\mathfrak{M}_l^k$ *if the occurrence of the event represented by* t_n *requires the value of* sv_j^i *to be equal to the value represented by the place* p_m.

2. ***Module composition.-*** *Obtain the synchronous composition* $\mathfrak{M}^{sc}$ *of all modules* $\mathfrak{M}_j^i$ *and then apply the permissive composition to* $\mathfrak{M}^{sc}$ *with itself to obtain* (G, M_0).

The output of previous algorithm is a *PN* model for the *DES*. Figure 3.4 shows the *PN* model obtained applying the *PN* module composition algorithm

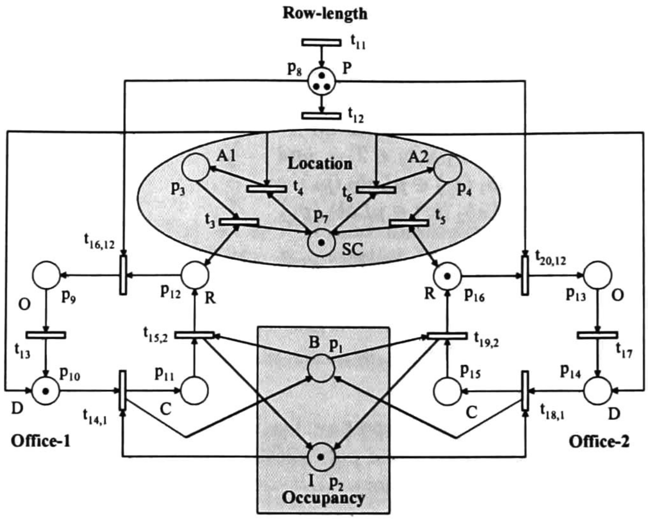

Figure 3.4: *PN* model for the clinic example considering only two doctor offices.

in the clinic example. In the interest of clearness, only two doctor offices have been considered.

3.4 Model interpretation

As mentioned earlier, the main advantage of *IPN* over *PN* is that input and output signals associated to event occurrences and the state values of the system, respectively, are included in an *IPN* model. Thus, by associating input and output signals to transitions and places, a *PN* model becomes an *IPN*. Such input and output signals association is made by defining functions λ and φ, respectively.

In order to introduce how functions λ and φ can be established, consider again the clinic example and remember that the clinic administrator has placed himself at the reception area. From this point he can determine the length of the patients queue, and if an office is ready or occupied. However, he cannot determine if it is dirty or being cleaned. Moreover, the only event occurrences that he can manipulate is the entrance of patients into the offices. Therefore, the place representing the row length, and those that represent the states ready and occupied of each office status should have output signals associated. Also, input signals must be assigned to the transitions representing the entrance of a patient to offices.

In the complete model for the clinic, transitions $t_{16,12}$, $t_{20,12}$, $t_{24,12}$ and $t_{28,12}$ represent the entrance of patients to office_1, office_2, office_3 and office_4, respectively; therefore function λ can be established as follows: $\lambda(t_{16,12}) =$ "*go_to_1*", $\lambda(t_{20,12}) =$ "*go_to_2*", $\lambda(t_{24,12}) =$ "*go_to_3*", $\lambda(t_{28,12}) =$ "*go_to* 4" and $\lambda(t_i) = \varepsilon$ for all other t_i. According to the *IPN* definition 9, function φ has to be established as a matrix such that all columns are either elementary or null vectors. Since the places representing those values that can be measured are p_8, p_9, p_{12}, p_{13}, p_{16}, p_{17}, p_{20}, p_{21} and p_{24}, matrix φ can be established as the 9×24 matrix $\varphi = [\underbrace{00}_{\text{Occup}}\ \underbrace{00000}_{\text{Locat}}\ \underbrace{e_1}_{\text{Row}}\ \underbrace{e_2 00 e_3}_{\text{Office-1}} \underbrace{e_4 00 e_5}_{\text{Office-2}} \underbrace{e_6 00 e_7}_{\text{Office-3}} \underbrace{e_8 00 e_9}_{\text{Office-4}}]$, where zeros represent null columns and e_i the vector such that its i-th entrance is 1 and all other are 0.

The algorithm that formally establishes the steps that have to be observed in order to obtain an *IPN* model from the *PN* model previously obtained.

Algorithm 22 *IPN building*

Input: *A labelled PN* (G, M_0).

Output: *An IPN* (G, M_0) *representing the behavior of the DES, including the input commands and the output signals.*

---

1. ***Defining*** λ. *Functions* λ *and* φ *must be specified to obtain* (Q, M_0). *Function* λ *is established by associating to each transition* t^{ij}_{mn} *in the model, the input symbol* α^{ij}_{mn} *(if any) that is used to change the state variable* sv^i_j *from value* val^{ij}_m *to value* val^{ij}_n.

2. ***Defining*** φ. *Function* φ *is defined as a* $q \times n$ *matrix, where* q *is the number of available output signals in the DES and* n *is the number of places in the model* (G, M_0). *Each column of this matrix is an elementary or null vector and* $\varphi(i,j) = 1$ *when the output* i *is present (turned on) every time that* $M(p_j) = 1$.

   According to this, φ *is linear and the output of the IPN contains* k *times the output signal related to place* p_j *if* $M(p_j) = k$.

Summarizing, the methodology to obtain an *IPN* model consist in applying algorithm `Building PN modules`, afterwards algorithm `PN Module composition` and finally algorithm `IPN building`. This methodology is illustrated through a case of study in the next subsection.

3.5 Application example

System description and scenario.- Consider the manufacturing cell depicted in figure 3.5. This cell is composed of a conveyors end, two work-sites, and a

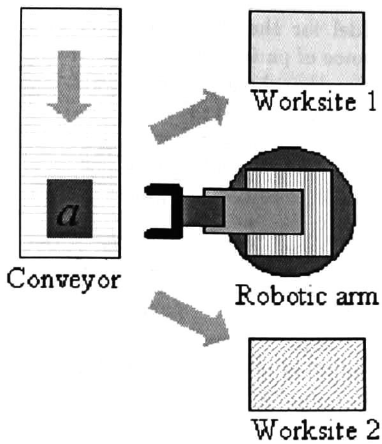

Figure 3.5: Manufacturing cell layout.

robotic arm. Two different types of parts (a and b) are loaded into the end of the conveyor. From that point, the robotic arm takes parts of type a and load them into work-site 1 and parts b into work-site 2. Afterwards external agents remove parts from the work-sites. In order to avoid an overflow in the work-sites, the robotic arm movement is restricted in the following way: the robotic arm cannot go to a work-site unless the work-site is empty.

Applying algorithm `Building PN`.- The identified system components are work-site 1, work-site 2, the conveyors end and the robotic arm. Therefore the following set is identified SYSTEM_COMPONENTS= {*worksite_1*, *worksite_2*, *conveyors_end*, *robotic_arm*}. State variables $W1$, $W2$ and CE will describe the occupancy of the work-sites 1 and 2 and the conveyor last site, respectively. The action that the robotic arm is performing will be described by state variable RA. The set {*empty*, *occupied*} describes the values that state variables $W1$ and $W2$ can take and {*empty*, *holding_a*, *holding_b*}, and {*idle*, *picking*, *holding_a*, *holding_b*, *delivering_a*, *delivering_b*} the values for CE, and RA respectively. Places p_1 and p_2 are created to represent the values of $W1$; p_3 and p_4 to describe the values of W_2 respectively. Places p_5, p_6 and p_7 represent the values of CE respectively; p_8, p_9 ,...,p_{13} represent the values of RA respectively.

For every state variable a set of transitions are created to represent the events that change the value of a given variable. For example t_1 represents the change *empty* – *occupied* and t_2 represents the change *occupied* – *empty*. The sets $\{t_3, t_4\}$, $\{t_5, t_6, t_7, t_8\}$, and $\{t_{12}, t_{10}, t_{13}, t_9, t_5, t_{11}, t_{14}\}$ are used to represent the changes in $W2$, $W2$, CE, and RA respectively. In order to obtain the initial marking of the modules we assume that the initial state for the system is $W1$ = *empty*, $W2$ = *empty*, CE = *empty* and RA = *idle*. The

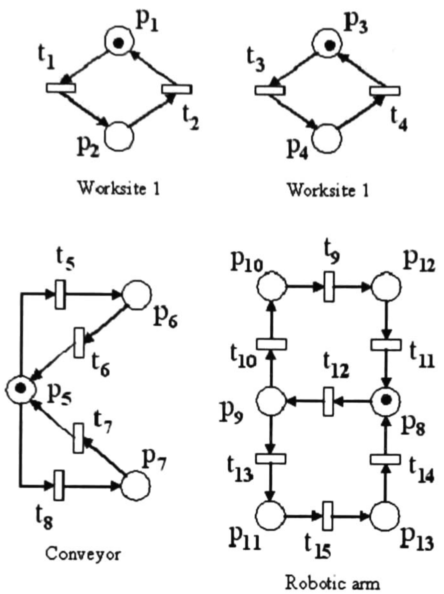

Figure 3.6: *PN* modules describing state variables.

PN modules obtained by this algorithm are depicted in figure 3.6.

Applying algorithm `PN Module composition`.- We assign the same label to the transitions representing the picking of a part a and the same to those representing the picking of a part b from the conveyors last site (i.e., $tLab\,(t_6) = tLab\,(t_{10})\,,\ tLab\,(t_7) = tLab\,(t_{13})$). Also the transitions representing the same part delivering at a work-site are given the same label ($tLab\,(t_1) = tLab\,(t_{11})\,,$ $tLab\,(t_3) = tLab\,(t_{14})$). Since t_9 is associated with the movement of the robotic arm to the work-site 1 and place p_1 represents that this work-site is empty, $pLab\,(p_1) = \{tLab\,(t_9)\}$. Analogously, $pLab\,(p_3) = \{tLab\,(t_{15})\}$ for work-site 2.

Figure 3.7 shows the *PN* resulting from performing the synchronous compositions of the modules representing the state variables $W1, W2, CE$ and RA and later performing the permissive composition of the resulting *PN* with itself. In the figure, the bidirectional arcs represent a self-loop around the related nodes.

Applying algorithm `IPN definition`.- Assuming that input symbol "*pick*" orders the robotic arm to pick the part at the conveyors end we make $\lambda\,(t_{6,10}) =$ *pick*, $\lambda\,(t_{7,13}) =$ *pick*. Null input signals (i.e. input signal ε) are associated to the remaining transitions. It is also assumed that signal "*at_conv*" is emitted for the robotic arm while it is at the conveyors end, and "*deliver*" while delivering a part. Signals "*hold_a*", "*hold_b*" are emitted by the conveyor while holding

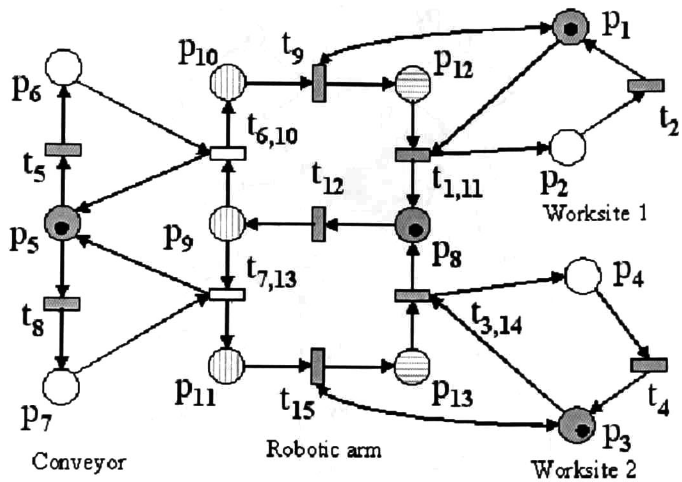

Figure 3.7: *IPN* model for the entire system.

such a part and work-sites 1 and 2 provide signals "*w1_o*" and "*w2_o*" while occupied, respectively. Thus, the places p_2, p_4, p_6, and p_7 have different output signals associated, while p_9, p_{10} and p_{11} have the same signal, as well as p_{12} and p_{13}. The remaining places have no output signal associated and thus, φ is given as the 6×13 matrix:

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

The *IPN* obtained applying this algorithm is given in figure 3.7, the places that have no output signals associated are depicted as dark circles, those that have different output signals are represented as clear circles and those with the same output signal as dashed circles. Similarly, transitions that have an input symbol associated are depicted as clear bars, and those without an input symbol are dark.

3.6 Module transformation

The output of algorithm **PN Module-composition**, is a binary state machine when the state variable represented is bounded. This section presents a technique that allows to reduce the amount of places of that module by transforming it into a non-binary state machine. Moreover, a reduction of an entire *IPN* model can be obtained by defining establishing the reductions corresponding to its modules.

Definition 23 *Let* sv_j^i $\in$STATE_VARIABLES$_i$ *be a state variable and* $\mathfrak{M}_j^i = (G, M_0)$ *its binary PN model obtained with algorithm* PN Module-composition. *Any PN* $\mathfrak{M}_j^{i\prime} = (G', M_0')$ *model such that the reachability graphs of both nets are equivalents (i.e. the number of states is the same and they are connected by the same labelled arcs) is a codification of* $\mathfrak{M}_j^i$.

In the following, $\mathfrak{M}_j^i$ will be named the canonical codification of sv_j^i and if the number of places in $\mathfrak{M}_j^{i\prime}$ is less than the number of places in $\mathfrak{M}_j^i$ it will be said that $\mathfrak{M}_j^{i\prime}$ is a place reduction codification of sv_j^i.

Definition 24 *Let* $\mathfrak{M}_j^i$ *be a canonical codification. The matrix* $\mathfrak{T} = [v_1, ..., v_k]$ *where* v_i *is a* $q \times 1$ *vector of non-negative integers,* $q \geqslant 1$ *and* $k = \left|\text{VALUE}_{sv_j^i}\right|$ *is named a code transformation matrix if* $\forall v_i, \sum_{i=1}^{k} v_i(j) = \xi$ *for some constant value* ξ, *and* $v_m \neq v_l$.

The following result shows how a code transformation matrix $\mathfrak{T}$ can be used to obtain a codification $\mathfrak{M}_j^{i\prime}$ of a *PN* module $\mathfrak{M}_j^i$.

Proposition 25 *Let* $\mathfrak{M}_j^i$ *be a PN module and* M_0, C^+ *and* C^- *be the initial marking, post and pre incidence matrices of* $\mathfrak{M}_j^i$, *respectively. Let* $\mathfrak{T}$ *be a code transformation matrix, then* $\mathfrak{T}M_0$, $\mathfrak{T}C^+$ *and* $\mathfrak{T}C^-$ *describe the initial marking, post and pre incidence matrices of a codification* $\mathfrak{M}_j^{i\prime}$ *of* $\mathfrak{M}_j^i$.

Proof. Any marking in the reachability graph of $\mathfrak{M}_j^i$ has a unique marking in the reachability graph of $\mathfrak{M}_j^{i\prime}$. This fact follows from the following observations.

Let $M_0 + (C^+ - C^-)\, v_k$ be a reachable marking of $\mathfrak{M}_j^i$. Since $\mathfrak{M}_j^i$ is an ordinary binary state machine *PN*, then the columns of C^+ and C^- are elemental vectors, thus they are markings of the reachability graph of $\mathfrak{M}_j^i$. Moreover, any reachable marking of $\mathfrak{M}_j^i$ is an elemental vector.

On the other hand, a reachable marking in $\mathfrak{M}_j^{i\prime}$ is $\mathfrak{T}M_0 + (\mathfrak{T}C^+ - \mathfrak{T}C^-)\, v_k = \mathfrak{T}[M_0 + (C^+ - C^-)\, v_k]$, since any reachable marking of $\mathfrak{M}_j^i$ is an elemental vector, then the reachable markings of $\mathfrak{M}_j^{i\prime}$ are equal to the columns of $\mathfrak{T}$. Moreover, since the columns of $\mathfrak{T}$ are different from each other, then to each different marking of $\mathfrak{M}_j^i$ corresponds a different marking of $\mathfrak{M}_j^{i\prime}$.

When $M_i \xrightarrow{t_a} M_j$ occurs in $\mathfrak{M}^i_j$ then, from the state equation of $\mathfrak{M}^{i\prime}_j$, it follows that $\mathfrak{T}M_i \xrightarrow{t_a} \mathfrak{T}M_j$, thus both reachability graphs are equivalent.

Also, any marking in the reachability graph of $\mathfrak{M}^{i\prime}_j$ has a unique marking in the reachability graph of $\mathfrak{M}^i_j$. This can be proved in a similar way.

Since the reachability graphs of both nets are equivalent, then $\mathfrak{M}^{i\prime}_j$ is a codification of $\mathfrak{M}^i_j$. ■

By creating a block diagonal matrix of code transformation matrices, a code transformation matrix for and entire IPN can be obtained.

Proposition 26 *Let $\mathfrak{S} = \{\mathfrak{M}_1, ..., \mathfrak{M}_n\}$ be a set of PN modules and (Q, M_0) be the IPN derived when synchronous and permissive compositions are applied to the set $\mathfrak{S}$ and the input and output functions are defined. If there exists $\mathfrak{T}_1$, ...,$\mathfrak{T}_n$ code transformation matrices for the respective modules in $\mathfrak{S}$, then the matrix*

$$\mathfrak{T} = \begin{bmatrix} \mathfrak{T}_1 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \mathfrak{T}_n \end{bmatrix}$$

is a code transformation matrix for (Q, M_0).

Proof. Note that the places of (Q, M_0) are the union of the places of the modules. Then, by a similar reasoning to that applied in previous proposition, this new matrix is a code transformation matrix. ■

The more interesting codifications are those with a lower number of places. In order to obtain this kind of codifications, it is enough to select a code transformation matrix $\mathfrak{T}$ with less rows than the number of places of $\mathfrak{M}^i_j$.

Figure 3.8 shows a place reduction codification of the IPN module representing the state variable that describes the dynamics of the robotic arm in the example of section 3.5, whose binary PN representation is depicted in figure 3.6. It can be verified that the reachability graphs of both nets are equivalent and that the incidence matrices of the place reduction codification, as well as its initial marking can be obtained using the code transformation matrix

$$\mathfrak{T} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

For instance

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

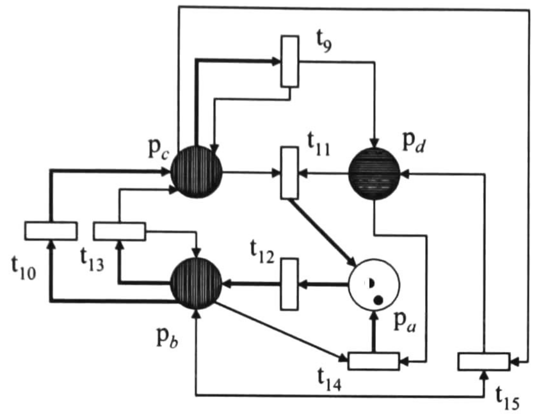

Figure 3.8: Place reduction codification of the *IPN* module representing the robotic arm.

and

$$\mathfrak{T}C^- = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

which is the pre-incidence matrix of the reduced model.

3.7 Conclusions

A novel modeling methodology that allows to obtain *IPN* for *DES* has been presented. This modeling methodology is based on establishing the amount of state variables needed to describe the behavior of *DES* components along with the set of possible values that each of those state variables may take. The behavior of the state variables have been represented in terms of *PN* called modules, and later, in order to obtain the model for the entire system, the relation among those modules has been represented using two basic operators: the synchronous and the permissive compositions. Synchronous composition allows to represent those event that commonly affect the state of different modules and permissive composition to represent the need of an event occurrence for the state of a different module to have a specific value.

Also a reduction technique that allows to transform binary models into non-binary *IPN* with weighted arcs, and equivalent reachability graphs has been presented.

Chapter 4

Observability

Summary: A formal definition of observability in IPN, along with a novel geometric characterization of observability based on input-output equivalence relations on the marking sequences sets, are presented in this chapter. That characterization is applied to a linear continuous system, leading to the well known structural conditions necessary for observability in such systems; showing that the observability characterization is well posed and that can be applied to any dynamic system. Since this observability definition is based on input-output sequences, directly applying it to verify observability in IPN results computationally complex. Therefore, using a similar process to the one that lead to the structural characterization of observability in linear systems, conditions necessary for observability in IPN are obtained. These conditions motivate the division of the observability problem into the sequence- and marking-detectability. Event-detectability, a stronger property than sequence-detectability is defined and structural characterized. Also a structural characterization of a broad class of marking-detectable IPN is presented. Thus, polynomial algorithms to verify observability in a broad class of IPN can be obtained.

4.1 Introduction

Observability is an important dynamic property that determines if the state of a system can be computed using the input and output signals and the structure of the system.

As mentioned earlier, there exists many observability definitions in DES literature, however, most of those definitions are either special cases of what in systems theory is considered observability or conditions for the existence of system controllers under partial event observation.

In this chapter, a formal definition of observability for IPN, that arises from the generalization of observability to the case when the system is non-deterministic, is presented. Moreover, a theorem that characterizes observable systems is introduced.

In order to show that this theorem, and therefore, the observability definition presented, capture the meaning of this property as it is understood in systems theory, by applying the theorem to a linear continuous system, the well known conditions for observability in this kind of systems, are derived.

Since the theorem mentioned earlier is based on input-output sequences and its relation with the sequences of transitions that fire in the IPN, its importance results only theoretical and cannot be directly applied to determine if a given IPN is observable. Therefore, following an analogous procedure to the one that lead to the observability conditions in the linear continuous case, observability conditions for IPN are derived. These conditions lead to the division of the observability property into sequence-detectability and marking-detectability. Both of this properties are herein formally defined and analyzed.

As the name suggests, sequence-detectability directly depends on the sequences of transitions fired in an IPN and analyzing all possible sequences that may fire in an IPN results, in the general case, a highly complex problem. Fortunately, there exists a stronger property than sequence-detectability, named event-detectability, which can be fully characterized in terms of the IPN structure.

Moreover, for a broad class of IPN that exhibit event-detectability, also marking-detectability can be characterized in terms of the structure. Therefore, polynomial algorithms to determine if an IPN is observable can be obtained using the results herein presented.

4.2 Observability formal definition

Loosely speaking, a dynamic system model is said to be observable if the knowledge of its inputs, outputs and structure suffices to uniquely determine its state.

A deterministic and time-driven dynamic model, for instance $\dot{x} = Ax + Bu; y = Cx$, is said to be observable if there exists a finite time t_1 such that the knowledge of the model structure (A, B, C), the input signal $(u(t))$ and the output signal $(y(t))$ over the interval $t_0 \leq t \leq t_1$ suffices to uniquely determine the initial state $(x(t_0))$. A remarkable fact in deterministic models is that the knowledge of $x(t_0)$ and $u(t)$ uniquely determines $x(t)$ for all $t \geq t_0$.

When the system is not deterministic (i.e., when the solution of the model is not unique [33]), however, the knowledge of $x(t_0)$ and $u(t)$ does not guarantee the computation of $x(t_0)$ for all $t \geq t_0$. Thus, for the general case, the observability condition must be changed to provide the knowledge of the initial state and all states $x(t), t > t_0$.

A non-deterministic dynamic model, for instance an IPN, is observable if there exists a finite integer n such that the knowledge of the model structure (C, λ, φ), the input sequence (v_k) and the output sequence (y_k) for any $k \geq n$ suffices to uniquely determine the state sequence over $0 \leq l \leq k$ $(M_0 \ldots M_k)$.

Using the input-output sequences introduced in chapter 2, the observability definition in IPN can be formally stated as follows.

Definition 27 *An IPN given by (Q, M_0), where M_0 may be unknown, is observable if there exists an integer $k < \infty$ such that $\forall \omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ it holds that the information provided by ω and (Q, M_0) suffices to uniquely determine the initial marking M_0 and the marking M_i reached by the firing of the underlaying firing sequence σ_ω.*

Therefore an IPN is observable when for any sequence of input-output signals of length equal or greater than k or any blocking sequence, the marking reached by the system can be uniquely determined.

4.3 Geometric characterization

Since the set S_ω contains all the marking sequences corresponding to the same input-output sequence $\omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$, then when $|S_\omega| = 1$ there exists only one marking sequence for the word ω. Thus the initial and the actual marking can be computed from these marking sequence. If this is the case for every $\omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$, then any marking reached by those ω can be computed and therefore (Q, M_0) is observable. This fact is formalized in the following theorem.

Theorem 28 *An IPN given by (Q, M_0) is observable if and only if there is an integer $k < \infty$ such that $\forall \omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ it holds that $|S_\omega| = 1$, where S_ω is the marking sequence corresponding to ω.*

Proof. *(Sufficiency) Assume that there is an integer $k < \infty$ such that $\forall \omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ it holds that $|S_\omega| = 1$, then, there exists a function $\Psi : \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0) \longrightarrow \mathbf{R}(Q, M_0) \times \mathbf{R}(Q, M_0)$ such that $\forall \omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ it holds that $\Psi(\omega, (Q, M_0)) = (M_0, M_i)$, where*

M_0 is the initial marking and M_i is the marking reached by the firing of the underlaying firing sequence σ_ω.

(Necessity) Suppose that there is no integer $k < \infty$ such that $\forall\omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ it holds that $|S_\omega| = 1$, then for any k there is at least one $\omega \in \Lambda^k(Q, M_0) \cup \Lambda_B(Q, M_0)$ such that $|S_\omega| \neq 1$, therefore $|S_\omega| > 1$. Assume without lost of generality that $S_\omega = \{\gamma_1 = M_i M_j \cdots M_k \cdots M_n, \gamma_2 = M_i' M_j' \cdots M_k' \cdots M_n'\}$. Since these sequences are different, then there must exist markings M_k and M_k' in γ_1 and γ_2, respectively, such that $M_k \neq M_k'$. Notice then that in the case when the initial marking of (Q, M_0) is M_k or M_k', there exist two different values to assign to M_0, or alternatively, that function Ψ cannot be obtained. Thus (Q, M_0) is not observable. ■

It is worth noting that previous theorem uses the input an output sequences of the model, but does not use the structure of the model. Thus, this result can be applied to any dynamic system model. For instance, when the model is the linear differential state equation:

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad x(t_0) = x_0 \\ y &= Cx \end{aligned} \tag{4.1}$$

the model is observable if and only if there exists a finite t_1 such that any pair of different state trajectories $x_1(t), x_2(t)$ can be distinguished using the input signals $u_1(t), u(t_2)$, or the output signals $y_1(t), y_2(t)$ for $t_0 \leq t \leq t_1$. To clarify this fact notice that the solution to equation (4.1) is:

$$x(t) = e^{At}x_0 + \int_0^{t_0} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{4.2}$$

If the same input is given for two different initial conditions $x_1(t_0)$ and $x_2(t_0)$, then two possible solutions are found:

$$\begin{aligned} x_1(t) = e^{At}x_1 + \textstyle\int_0^{t_0} e^{A(t-\tau)}Bu(\tau)d\tau \\ x_2(t) = e^{At}x_2 + \textstyle\int_0^{t_0} e^{A(t-\tau)}Bu(\tau)d\tau. \end{aligned} \tag{4.3}$$

Now, if the output signals $y_1(t)$ and $y_2(t)$ generated by $x_1(t)$ and $x_2(t)$, respectively, are the same for every $t \geq t_0$, then

$$Ce^{At}x_1 + C\int_0^{t_0} e^{A(t-\tau)}Bu(\tau)d\tau = Ce^{At}x_2 + C\int_0^{t_0} e^{A(t-\tau)}Bu(\tau)d\tau \tag{4.4}$$

or

$$Ce^{At}x_1 - Ce^{At}x_2 = 0. \tag{4.5}$$

Thus

$$Ce^{At}(x_1 - x_2) = 0 \tag{4.6}$$

which can be written as

$$C(I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + ... + \frac{1}{(n-1)!}A^{n-1}t^{n-1} + ...)(x_1 - x_2) = 0. \tag{4.7}$$

using Taylor series. Since

$$e^{At} = (I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + ... + \frac{1}{(n-1)!}A^{n-1}t^{n-1} + ...)$$

is non-singular [7], then the system is non-observable when there exist at least two vectors x_1, x_2 belonging to $\ker(C) \cap \ker(CA) \cap ... \cap \ker(CA^{n-1})$, or equivalently the system is non-observable when the rank of the matrix

$$O = \begin{bmatrix} C \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{4.8}$$

is less than n. Thus the continuous system is non-observable when $\ker(C) \cap \ker(CA) \cap ... \cap \ker(CA^{n-1}) \neq \{0\}$ or equivalently, when the rank of O is less that n. The converse can also be proved [43].

Notice then that, using theorem 28, a structural characterization of the observability for those systems described by equation (4.1) was derived. Now, using a similar reasoning, a structural characterization for observability in IPN models will be derived.

Assume that there exist two initial markings M_0^1, M_0^2 (not necessarily different because IPN are in the general case non-deterministic) and the same input γ is given in both cases, generating two different sequences of markings $s_1^{n_1} = M_0^1 M_1^1 \cdots M_{n_1}^1$ and $s_2^{n_2} = M_0^2 M_1^2 \cdots M_{n_2}^2$, reached by the firing sequences $\sigma_1 = t_1^1 t_2^1 \cdots t_{n_1}^1$ and $\sigma_2 = t_1^2 t_2^2 \cdots t_{n_2}^2$, respectively. Now, according to theorem 28, if both of them generate the same output word ϕ for every $n_1, n_2 \geq 0$, then the system is non-observable. In this case, it holds that for all $\sigma_i' \in \overline{\sigma_1}$ there exists a $\sigma_j' \in \overline{\sigma_2}$, such that

$$\varphi(M_0^1 + C\overrightarrow{\sigma_1'}) = \varphi(M_0^2 + C\overrightarrow{\sigma_2'}) \tag{4.9}$$

or

$$\varphi(M_0^1 + C\overrightarrow{\sigma_1'}) - \varphi(M_0^2 + C\overrightarrow{\sigma_2'}) = 0. \tag{4.10}$$

Since both marking sequences generate the same output word, in the case when φ and C are linear, the conditions for non-observability are reduced to the following ones:

$$\begin{aligned} \varphi(M_0^1 - M_0^2) &= 0 \\ \varphi C\left(\overrightarrow{\sigma_1'} - \overrightarrow{\sigma_2'}\right) &= 0 \end{aligned} \tag{4.11}$$

and other combinations are not possible.

Hence, if the model is non-observable, it holds that:

$$\begin{aligned} M_0^1 - M_0^2 &\in \ker \varphi \\ \left(\overrightarrow{\sigma_1'} - \overrightarrow{\sigma_2'}\right) &\in \ker \varphi C \end{aligned} \tag{4.12}$$

The first condition is held if $M_0^1 = M_0^2$ or if $\varphi M_0^1 = \varphi M_0^2$ with $M_0^1 \neq M_0^2$; and the second one holds if $\sigma_1' = \sigma_2'$, or if $C\overrightarrow{\sigma_1'} = C\overrightarrow{\sigma_2'}$ with $\sigma_1' \neq \sigma_2'$, or if $\varphi C\overrightarrow{\sigma_1'} = \varphi C\overrightarrow{\sigma_2'}$ with $C\overrightarrow{\sigma_1'} \neq C\overrightarrow{\sigma_2'}$.

The condition $M_0^1 = M_0^2$ with $\sigma_1' = \sigma_2'$, however, is not fulfilled when the IPN is non-observable, since it generates only one marking sequence.

Condition $C\overrightarrow{\sigma_1'} = C\overrightarrow{\sigma_2'}$ with $\sigma_1' \neq \sigma_2'$ implies that those transitions that are different in σ_1' and σ_2' are indeed equivalent. Since this redundancy of transitions is not considered in this work, the non-observability of an IPN is reduced to the following cases:

1. $M_0^1 = M_0^2$ and $\varphi C\overrightarrow{\sigma_1'} = \varphi C\overrightarrow{\sigma_2'}$ with $C\overrightarrow{\sigma_1'} \neq C\overrightarrow{\sigma_2'}$.
2. $\varphi M_0^1 = \varphi M_0^2$ with $M_0^1 \neq M_0^2$ and $\sigma_1' = \sigma_2'$.
3. $\varphi M_0^1 = \varphi M_0^2$ with $M_0^1 \neq M_0^2$ and $\varphi C\overrightarrow{\sigma_1'} = \varphi C\overrightarrow{\sigma_2'}$ with $C\overrightarrow{\sigma_1'} \neq C\overrightarrow{\sigma_2'}$.

Now, these conditions will be studied to derive structural conditions to characterize IPN exhibiting the observability property.

The first case characterizes when a marking enables different firing sequences σ_1, σ_2 such that $\sigma_1, \sigma_2 \in \Omega(\omega)$ for some input-output symbol word ω.

The second case characterizes when two different markings enable the same firing transition sequence, thus two marking sequences generating the same input-output symbol sequence are obtained.

Finally, the third case characterizes when two different markings enable two different firing transition sequences σ_1, σ_2, such that the input-output symbol sequence generated is ω and $\sigma_1, \sigma_2 \in \Omega(\omega)$.

Thus, if these cases are avoided the IPN becomes observable. For instance, the first case is avoided when any pair of firing transition sequences σ_1, σ_2 can be detected since in this case for each input-output symbol word generated ω, $|\Omega(\omega)| = 1$. Thus according to definition 16, $|S_\omega| = 1$, and by theorem 28 the net is observable.

The second case is avoided when there exists a marking M_k that can be computed after the firing of $k < \infty$ transitions. In this case, we claim that either $M_1^0 \xrightarrow{\sigma} M_k$ or $M_2^0 \xrightarrow{\sigma} M_k$, but not both M_1^0, M_2^0 reach M_k for the same firing sequence. Assume that this is not true, i.e. $M_1^0 \xrightarrow{\sigma} M_k$ and $M_2^0 \xrightarrow{\sigma} M_k$. Since the firing transition sequence is the same in both cases, using the IPN state equation it follows that M_{k-1} is the same in both marking sequences and then, by recursively applying this idea, we obtain that $M_1^0 = M_2^0$, a contradiction. Then, when σ is known, the knowledge of M_k allows to detect the marking sequence, i.e. $|S_\omega| = 1$.

The third case is a combination of previous cases. Thus, the knowledge of the fired transition sequence σ_ω and the reached marking M_k are enough to avoid this case.

In order to determine the conditions that guarantee that both, the fired transition sequence σ_ω and the reached marking M_k, can be determined, it is important to consider that even when the precise location or state of the entities (resources, machines, buffer capacities, etc.) that constitute the DES may be unknown, in most of the cases, the amount of those entities is known. This concept is analogous to that of "macro-markings" used in [15] and is formally posed in the next definition.

Definition 29 *Let Q be an IPN structure and $M(p_j)$ be any marking of a place p_j in Q. The set of equations $CML = \{\sum_{j=1}^{n} \gamma_j^i \cdot M(p_j) = k_i | i \in [1, ..., s], \gamma_j^i \in \mathbb{Z}^+\}$ form a set of conservative marking laws (CML) if $\forall \gamma_k^i \neq 0$ it holds that k_i/γ_k^i is an integer value and all non-computable places (i.e., all p_n such that, $\varphi(\cdot, n) = 0$ or such that there exists a p_m such that $p_m \neq p_n$ and $\varphi(\cdot, m) = \varphi(\cdot, n)$) are contained in at least one equation (i.e., if p_n is non-computable then $\exists i$ such that $\gamma_n^i \neq 0$). A CML is said to be a binary CML ($BCML$) if it holds that $\forall \gamma_j^i \in \{0, 1\}$ and $\forall k_i = 1$. In a matrix form, a CML can be written as:*

$$\Gamma M = K \tag{4.13}$$

where $\Gamma_{i,j} = \gamma_j^i, M_j = M(p_j)$ and $K_i = k_i$.

Remarks:

- Hereafter $\mathcal{M}_0$ will denote the set of all possible initial markings fulfilling the stated CML; that is, $M_0 \in \mathcal{M}_0$ means that any marking $M \in R(Q, M_0)$ fulfills the CML constraints.
- Notation $p_n \in e_i$, where $e_i \in CML$ (or $e_i \in BCML$) means that there exists an equation $\sum_{j=1}^{n} \gamma_j^i \cdot M(p_j) = k_i$, named e_i in the CML, such that $\gamma_n^i \neq 0$.
- Also, $(Q, \mathcal{M}_0)$ will denote an IPN where $M_0 \in \mathcal{M}_0$; the initial marking M_0 could be, however, unknown. Notation $\mathcal{M}_0^B$ will be used when the CML is binary.

Using the CML concept, the following definitions can be posed. They establish when is that the transition sequence σ_ω, and when the reached M_k can be determined.

Definition 30 *An IPN given by $(Q, \mathcal{M}_0)$ is sequence-detectable if there exists an integer $k < \infty$ and a function $\Psi_S : \Lambda^k(Q, \mathcal{M}_0) \times (Q, \mathcal{M}_0) \longrightarrow \pounds(Q, \mathcal{M}_0)$ such that $\forall \omega \in \Lambda^k(Q, \mathcal{M}_0)$ it holds that $\Psi_S(\omega, (Q, \mathcal{M}_0)) = \sigma_\omega$.*

Definition 31 *An IPN given by $(Q, \mathcal{M}_0)$ is marking-detectable if there exists an integer $k < \infty$ and a function $\Psi_M : \Lambda^k (Q, \mathcal{M}_0) \times (Q, \mathcal{M}_0) \longrightarrow \mathbf{R}(Q, \mathcal{M}_0)$ such that $\forall \omega \in \Lambda^k (Q, \mathcal{M}_0)$ it holds that $\Psi_M (\omega, (Q, \mathcal{M}_0)) = M_i$, where M_i is the marking reached by the firing of the underlaying firing sequence σ_ω.*

According to these definitions and previous discussion on the conditions that lead to observable IPN it follows that sequence- and marking-detectability are necessary for observability in the case when it is assumed that $\forall t_i, t_j$ such that $t_i \neq t_j$, $\lambda(t_i) = \lambda(t_j)$, it holds that $C(\cdot, i) \neq C(\cdot, j)$. Moreover, it can be proved that sequence- and marking-detectability are necessary and sufficient for observability in this case. This is formally posed in the following lemma.

Lemma 32 *An IPN given by $(Q, \mathcal{M}_0)$ is observable if and only if it is both sequence- and marking-detectable.*

Proof. (Sufficiency) Let $(Q, \mathcal{M}_0)$ be an IPN and assume that there exists an integer $k < \infty$ and functions $\Psi_M : \Lambda^k (Q, \mathcal{M}_0) \times (Q, \mathcal{M}_0) \longrightarrow R(Q, \mathcal{M}_0)$ and $\Psi_S : \Lambda^k (Q, \mathcal{M}_0) \times (Q, \mathcal{M}_0) \longrightarrow \pounds (Q, \mathcal{M}_0)$ such that $\forall \omega \in \Lambda^k (Q, \mathcal{M}_0)$ it holds that $\Psi_M (\omega, (Q, \mathcal{M}_0)) = M_i$ and $\Psi_S (\omega, (Q, \mathcal{M}_0)) = \sigma_\omega$ where M_i is the marking reached by the firing of the underlaying firing sequence σ_ω. Then, a function $\Psi (\omega, (Q, \mathcal{M}_0)) = (M_0, M_i)$ can be built as $\Psi (\omega, (Q, \mathcal{M}_0)) = (M_S, M_M)$where $M_M = \Psi_M (\omega, (Q, \mathcal{M}_0))$ and $M_S = M_M - C\vec{\sigma_\omega}$. Clearly $M_0 = M_S$.
(Necessity) Follows from previous discussion on the conditions that lead to observable IPN. ■
In the following sections a characterization, based on the IPN structure, of those exhibiting sequence- and marking-detectability (and therefore observability) is addressed.

4.4 Sequence-detectable *IPN*

Sequence-detectability implies the knowledge of all firing sequences of an IPN, thus the problem of determining when an IPN is sequence-detectable is highly complex. Using event-detectability [40], a stronger property that establishes when each individual transition firing can be uniquely determined, the complexity can be overcame. This is because event-detectability can be proved using polynomial algorithms. Event-detectability is formally defined as follows:

Definition 33 *An IPN given by $(Q, \mathcal{M}_0)$ is event-detectable if any transition firing can be uniquely determined by the knowledge of the input given to $(Q, \mathcal{M}_0)$ and the output signals that it produces.*

The following lemma provides a structural characterization of the IPN exhibiting event-detectability.

Lemma 34 *A live IPN given by $(Q, \mathcal{M}_0)$ is event-detectable if and only if*

1. *$\forall t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j)$ or $\lambda(t_i) = \varepsilon$ it holds that $\varphi C(\cdot, t_i) \neq \varphi C(\cdot, t_j)$ and*

2. *$\forall t_k \in T$ it holds that $\varphi C(\cdot, t_k) \neq 0$.*

Proof. (Sufficiency) Assume that $(Q, \mathcal{M}_0)$ is an IPN where $\forall t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j)$ or $\lambda(t_i) = \varepsilon$ it holds that $\varphi C(\cdot, t_i) \neq \varphi C(\cdot, t_j)$ and $\forall t_k \in T$ it holds that $\varphi C(\cdot, t_k) \neq 0$. Let $M_m, M_n \in \mathcal{M}_0$ and a transition $t_p \in T$ such that $M_m \xrightarrow{t_p} M_n$ fire while the input symbol α is given to $(Q, \mathcal{M}_0)$. From state equation (2.1) $y_n - y_m$ can be computed as $y_n - y_m = \varphi(M_n) - \varphi(M_m) = \varphi(M_m + C(\cdot, t_p)) - \varphi(M_m) = \varphi C(\cdot, t_p)$. Since $\forall t_k \in T$ it holds that $\varphi C(\cdot, t_k) \neq 0$ the change in the output produced by the firing of t_p is not null, that is $y_n - y_m \neq 0$. Now there are two possibilities.

1. Suppose that the input symbol is ε, since $\forall t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j) = \varepsilon$ it holds that $\varphi C(\cdot, t_i) \neq \varphi C(\cdot, t_j)$ there is no transition $t_q \in T$ with $t_q \neq t_p$ such that $\lambda(t_q) = \varepsilon$ and $\varphi C(\cdot, t_q) = \varphi C(\cdot, t_p)$. Thus, the firing of transition t_p is the only one that could produce the change $y_n - y_m = \varphi C(\cdot, t_p)$ while the null input word $\alpha = \varepsilon$ was given to the system.

2. Suppose now that the input symbol is $\alpha \neq \varepsilon$, since $\forall t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j) = \alpha$ (or $\lambda(t_i) = \varepsilon, \lambda(t_j) = \alpha$) it holds that $\varphi C(\cdot, t_i) \neq \varphi C(\cdot, t_j)$ there is no transition $t_q \in T$ with $t_q \neq t_p$ such that $\lambda(t_q) = \alpha$ (or $\lambda(t_q) = \varepsilon$) and $\varphi C(\cdot, t_q) = \varphi C(\cdot, t_p)$. Thus, the firing of transition t_p is the only one that could produce the change $y_n - y_m = \varphi C(\cdot, t_p)$ while the non-null input word α was given to the system.

Then, in both cases, the transition t_p that fired can be uniquely determined and $(Q, \mathcal{M}_0)$ is event-detectable.

(Necessity) Suppose first that there exist two transitions $t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j) = \alpha$ and $\varphi C(\cdot, t_i) = \varphi C(\cdot, t_j)$. Then for an input word α there are two transitions t_i, t_j that may fire, therefore the input symbol given to $(Q, \mathcal{M}_0)$ does not provide information to distinguish the firings of t_i and t_j. Since $\varphi C(\cdot, t_i) = \varphi C(\cdot, t_j)$ the changes in the output that those firings produce are equal and no further information is provided. Therefore, there is no way to distinguish the firings of t_i and t_j.

Now suppose that there exist two transitions $t_i, t_j \in T$ such that $\lambda(t_i) \neq \varepsilon, \lambda(t_j) = \varepsilon$ and $\varphi C(\cdot, t_i) = \varphi C(\cdot, t_j)$. Assume that $\lambda(t_i) = \alpha$. Then for an input word α both transitions t_i and t_j may fire, again the input symbol does not help to distinguish the firings of t_i and t_j. If also $\varphi C(\cdot, t_i) = \varphi C(\cdot, t_j)$ then both firings produce the same change in the output and once more the firings of those transitions cannot be distinguished.

Finally assume that $\exists t_k \in T$ such that $\varphi C(\cdot, t_k) \neq 0$ Then the firing of t_k has no effect in the output and no matter what is the input symbol α given to $(Q, \mathcal{M}_0)$ there is no way to determine if transition t_k fires.

Thus, in all those cases the firing of the transitions cannot be uniquely determined and therefore, if any of those conditions holds $(Q, \mathcal{M}_0)$ is not event-detectable.

Therefore, in all those cases the firing of the transitions cannot be uniquely determined and if any of those conditions holds $(Q, \mathcal{M}_0)$ is not event-detectable. ■

4.5 Marking-detectable *IPN*

As mentioned in definition 31, marking-detectability deals with the possibility of computing the current marking of an IPN. This property is strongly related with structural properties of IPN and can be analyzed using the following place classification.

Definition 35 *Let $(Q, \mathcal{M}_0^B)$ be an IPN. The set of output-computable places is defined by $S_{oc} = \{p_i \in P | p_i$ is a computable place$\}$*

Notice that the set S_{oc} can be computed from the knowledge of φC; that is, $p_i \in S_{oc}$ when the column $\varphi C(\cdot, i)$ is not null and is different from all other columns of φC.

Definition 36 *Let $(Q, \mathcal{M}_0^B)$ be an IPN, $X = \{x_1, \ldots, x_r\}$ be the set of elemental T-components [10], [11] of Q, P_i be the set of places belonging to x_i, and $E = \{e_1, \ldots, e_s\}$ be the $BCML$ defined for $(Q, \mathcal{M}_0^B)$. The set of $BCML$-computable places is defined by $S_{Bc} = \{p_i \in P | \forall x_j \in X$ there exists an $e_k \in E$ with $p_i \in e_k$ and $\left(\bigcup_{p_n \in e_k} \{p_n\}\right) \cap P_j \neq \emptyset\}$.*

Notice that $p_i \in S_{Bc}$ if the following linear programming problem has no solution.

$$\begin{array}{c} \nexists P_X \\ s.t. \\ CX = 0; X \gneq 0 \\ X^T C^{-T} = P_X \\ \Gamma|_{p_i} P_X = 0 \end{array} \tag{4.14}$$

where C is the incidence matrix of Q, C^- is its pre-incidence matrix, X is a T-semiflow [10], P_X are the places contained in the T-component generated by X (this places are computed by equation $X^T C^{-T} = P_X$), and $\Gamma|_{p_i} P_X = 0$ means that there exists T-components whose places are disjoint from the set of places contained in those $BCML$ equation that contain place p_i. Hence, if previous problem has no solution then $p_i \notin S_{Bc}$.

Definition 37 *Let $(Q, \mathcal{M}_0^B)$ be an IPN, $E = \{e_1, \ldots, e_s\}$ be the BCML defined for $(Q, \mathcal{M}_0^B)$ and S_{oc}, S_{Bc} be the sets of output- and BCML-computable places of $(Q, \mathcal{M}_0^B)$, respectively. The set of transitive-computable places is defined by $S_{tc} = \bigcup_{i=0} S_c^i$ where:*

$$S_c^0 = S_{oc} \cup S_{Bc}$$
$$S_c^i = S_c^{i-1} \cup \{p_i \in P | p_i \in e_k, \text{ and every } p_j \in e_k, j \neq i, \text{ fulfills that } p_j \in S_c^{i-1}\} \tag{4.15}$$

Notice that also S_{tc} can be computed with a polynomial algorithm

As mentioned earlier, the marking of output-computable places can be directly obtained from the output signal, this is formally posed by the following lemma.

Lemma 38 *Let $(Q, \mathcal{M}_0^B)$ be a binary, live and event-detectable IPN. There is an integer $k < \infty$ such that $\forall \omega \in \Lambda^k (Q, \mathcal{M}_0^B)$ the marking $M_k(p_i)$ can be computed if $p_i \in S_{oc}$.*

Proof. Let $\omega \in \Lambda(Q, \mathcal{M}_0^B)$ be a word of finite length $k \geq 0, \omega = (\alpha_0, y_0) \cdots (\alpha_k, y_k)$. Then the current output of the IPN is the vector $y_k = \varphi M_k$. Since p_i is computable $\varphi(\cdot, i)$ is different from all other columns of φ. Moreover, the columns of φ are elemental vectors, thus $M_k(p_i) = y_k(q)$, where q is the number such that $\varphi(q, i) = 1$. ■

Now, the marking of a place belonging to the $BCML$-computable places set can be computed as established by the following lemma.

Lemma 39 *Let $(Q, \mathcal{M}_0^B)$ be a binary, live and event-detectable IPN, $X = \{x_1, \ldots, x_r\}$ be the set of elemental T-components of Q and $E = \{e_1, \ldots, e_s\}$ be the BCML defined for $(Q, \mathcal{M}_0^B)$. There is an integer $k < \infty$ such that $\forall \omega \in \Lambda^k (Q, \mathcal{M}_0^B)$ the marking $M_k(p_i)$ can be computed if $p_i \in S_{Bc}$.*

Proof. If p_i is computable, then its marking can be computed by previous lemma. If p_i is a non computable place, then it belongs to an equation $e_k \in E$. Moreover, since $(Q, \mathcal{M}_0^B)$ is a live IPN, then there exists a finite integer k' such that $\forall \sigma \in \pounds(Q, \mathcal{M}_0^B)$ fulfilling $|\sigma| \geq k'$ contains the firing of every transition in a T-component $x_i \in X$.

By definition, there exists a place p_n such that $p_i, p_n \in e_k$ and $p_n \in x_i$, thus the firing of a prefix σ' of σ will add a token in p_n. Let M_k be the first marking adding tokens to p_n. The firing of the transition t_j such that $M_{k-1} \xrightarrow{t_j} M_k$ is detected because the IPN is event-detectable and since the set $t_j^\bullet$ is known and $(Q, \mathcal{M}_0^B)$ is binary, it can be determined that $M_k(p_n) = 1$. Then, because of equation e_k it holds that $M_k(p_i) = 0$. Moreover, since Q is event detectable, then every time that tokens come in or out of p_i its marking can be computed. Thus the marking of p_i will be known for any marking M_l such that $l \geq k$. ■

Finally, we prove that the marking into the places belonging to S_{tc} is computable.

Lemma 40 *Let $(Q, \mathcal{M}_0^B)$ be a binary, live and event-detectable IPN. There is an integer $k < \infty$ such that $\forall \omega \in \Lambda^k (Q, \mathcal{M}_0^B)$ the marking $M_k(p_i)$ can be computed if $p_i \in S_{tc}$.*

Proof. According to previous lemmas the marking of p_j can be computed for some k_j if $p_j \in S_{oc} \cup S_{Bc}$. Thus for $k = \max\{k_i\}$ the marking of any $p_i \in S_{oc} \cup S_{Bc}$ is uniquely determined. Focus now in places $\{p_a, p_b, ..., p_g\} = S_{tc} - (S_{oc} \cup S_{Bc})$. Assume that they are given in the following order. The leading places in this set are those belonging to S_c^1 , after them are those belonging to S_c^2 and so on.

By definition of S_c^1 for each $p_x^1 \in S_c^1$ there exists a set of places $S_{p_x^1} = \{p_w, ..., p_s\} \subseteq (S_{oc} \cup S_{Bc})$ such that places $\{p_x^1\} \cup S_{p_x^1}$ are the only places belonging to the same $BCML$ equation $e_{p_x^1}$.

Thus, for any marking $M_q \in R(Q, \mathcal{M}_0^B)$ it holds that

$$M_q(p_x^1) + \sum_{p_r \in S_{p_x^1}} M_q(p_r) = 1 \tag{4.16}$$

which uniquely determines the marking $M_q(p_x^1)$ if the markings of all places in $S_{p_x^1}$ are known. This holds if $q \geq k$, therefore, the marking $M_q(p_x^1)$ reached by some $\omega \in \Lambda^k (Q, \mathcal{M}_0^B)$ will be determined for all places $p_x^1 \in S_c^1$. Inductively, the marking $M_q(p_y^2)$ will be uniquely determined because the marking of all places in $S_{p_y^2} \subseteq (S_{oc} \cup S_{Bc} \cup S_{p_x^1})$ are known and so on. Thus the marking of any place $p_i \in S_{tc}$ can be computed with $k = \max\{k_j\}$. ■

Finally, previous results lead to the following theorem that provides a structural characterization of observable IPN.

Theorem 41 *An IPN given by $(Q, \mathcal{M}_0^B)$ is observable if $P = S_{tc}$ and $(Q, \mathcal{M}_0^B)$ is event-detectable.*

Proof. If $(Q, \mathcal{M}_0^B)$ is event-detectable, it is sequence-detectable. According to Lemma 40 if $(Q, \mathcal{M}_0^B)$ is event-detectable and $P = S_{tc}$ then $(Q, \mathcal{M}_0^B)$ is marking-detectable. Thus, because of Lemma 32 $(Q, \mathcal{M}_0^B)$ is observable. ■

The IPN depicted in figure 4.1 results of relabeling some of the transitions of the IPN model obtained in section 3.5 for the manufacturing system. Notice that there are no null columns its corresponding matrix

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

However, transitions t_9 and t_{11} are non-manipulated and their corresponding columns in φC are equal. Therefore, this IPN is not event-detectable. This is because the firing of this transitions cannot be distinguished using the input and output signals that they produce. However, the IPN is sequence-detectable. This is because after the firing of t_9 transition t_1 has to fire (at most after the firing of t_5 or t_8), this firing is uniquely determined and cannot be held after the firing of t_{11} without firing t_9 before. Analogously, after the firing of t_{11} transition t_3 has to be fired and it uniquely determines t_{11}.

In order to illustrate the use of lemma 40 and theorem 41 suppose that the output signals associated to places p_{12} and p_{13} are different and therefore it holds that

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

resulting in an event-detectable IPN. Moreover, the set of computable places would be $S_{oc} = \{p_2, p_4, p_6, p_7, p_{12}, p_{13}\}$.

From the modeling methodology, it is known that $M_i(p_1) + M_i(p_2) = 1$, $M_i(p_3) + M_i(p_4) = 1$, $M_i(p_5) + M_i(p_6) + M_i(p_7) = 1$ and $M_i(p_8) + M_i(p_9) + M_i(p_{10}) + M_i(p_{11}) + M_i(p_{12}) + M_i(p_{13}) = 1$ for every reachable M_i. This is, the modeling methodology provides a $BCML$. From this equations and the fact that $p_2, p_4, p_6, p_7 \in S_{oc}$, also the markings of places p_1, p_4 and p_5 can be uniquely determined. Moreover, every infinitely long transition sequence contains at least one transition in the set $T_r = \{t_1, t_3, t_6, t_7, t_9, t_{10}, t_{11}\}$. Thus one of these transitions will eventually fire and because of event-detectability, this firing is uniquely determined. If it is not previously known, the first firing of a transition in T_r allows to compute the marking of places p_8, p_9, p_{10}, p_{11}. This is because the IPN is event-detectable and therefore that firing is uniquely determined. Then, since the post-incidence set of the fired transition is known, the set of places marked by the firing is known and therefore the marking of at least one those places. For instance, the firing of t_6 allows to establish that $M_j(p_{10}) = 1$. Moreover, since they share the same equation, they all become known. This is, if $M_j(p_{10}) = 1$ the marking of the remaining places in the equation is null.

Thus, assuming that p_{12} and p_{13} have different output signals associated, the IPN is marking-detectable and therefore observable. In fact, even without that assumption the IPN is marking-detectable, this can be verified constructing the reachability graph for the IPN, but this is, in the general case, a highly complex problem.

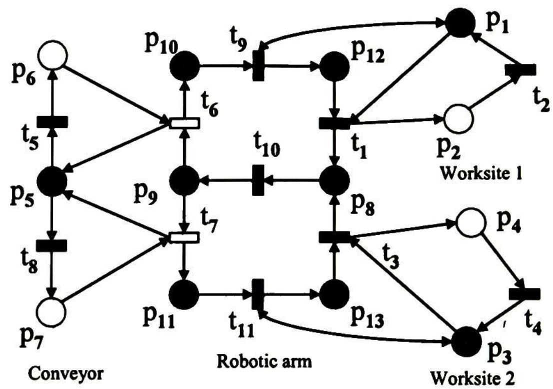

Figure 4.1: Relabeled IPN model of flexible manufacturing system.

4.6 Conclusions

In this chapter observability of IPN has been formally defined, also a geometric characterization of the IPN exhibiting observability has been presented. This characterization has been applied to a linear continuous system and the well known structural conditions for observability in this class of systems were obtained, proving that the observability characterization for IPN is well posed. Moreover, a detailed analysis of the process that leads to the structural characterization for observability in linear continuous system provided the conditions necessary for IPN to exhibit observability. From this analysis, the need for dividing the observability problem in sequence- and marking-detectability was obtained. Sequence detectability is a highly complex problem that was skipped by using event-detectability, a stronger property that can be proved by polynomial algorithms. Also polynomial algorithms to verify marking-detectability in event-detectable IPN have been presented. Therefore, polynomial algorithms to determine if a broad class of IPN exhibit observability have been proposed.

Chapter 5

Observability at the modeling stage

Summary: This chapter presents a study of the observability of a system at the modeling stage. Necessary and sufficient conditions for the modules obtained applying the modeling methodology presented in Chapter 3 to be observable are derived. Later, conditions on the composition of observable modules in order to guarantee that the composed model will result observable, are proposed. Finally, in this chapter it is proved that observability is invariant under the transformation technique that allows to obtain reduced models, presented above.

5.1 Introduction

In previous chapter, it was shown that the observability problem can be divided into the problems of sequence-detectability and marking-detectability. They deal with uniquely determining the transition sequence fired in an IPN and a marking reached by this transition sequence firing, respectively. Both sequence and marking-detectability require the explicit description of all transition sequence that can fire from any possible initial marking. Even when this has been avoided by the introduction of event-detectability and, using that concept, structural characterizations for a broad class of IPN were posed, many other IPN have been left aside. The modeling methodology herein proposed provides an alternative to characterize some of those IPN and a useful tool in the system design process. This is because the PN modules representing the behavior of the system state variables are binary, strongly connected state machines. For this class of PN it results relatively simple to obtain the set of all transition sequences that can be fired [10]. Therefore, it is easier to determine if the PN modules, transformed into IPN are observable. Moreover, if conditions for observable modules to generate observable models are available, the observability analysis is reduced to the analysis of the modules and the observation of some rules in the module composition.

This chapter presents conditions for strongly connected, binary state machines IPN to be observable and later, conditions on the module composition to generate observable IPN models for the entire system.

In addition, it is proved in this chapter that observability is invariant under the transformation technique proposed above, which allows to reduce the amount of states used to represent a DES in IPN terms.

5.2 Observable compositions

In order to introduce the definition of a relation that will be fundamental in the discussion presented in this section, consider the strongly connected state machine IPN structure depicted in figure 5.1. Suppose that this IPN is known to be binary and assume that functions λ and φ are given as follows.

Function λ	Function φ
$\lambda(t_1) = \varepsilon$ $\lambda(t_2) = \varepsilon$ $\lambda(t_3) = a$ $\lambda(t_4) = b$ $\lambda(t_5) = \varepsilon$ $\lambda(t_6) = c$ $\lambda(t_7) = d$	$\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$

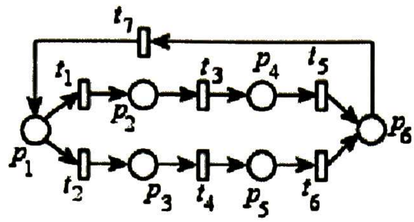

Figure 5.1: A strongly connected state machine.

Notice that under this assumptions, the possible initial markings $M_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$ and $M_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$ generate the input-output sequences given in the following tables.

$M_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$	
Input-output sequence	Transition sequence
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)$	*null*
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\left(c, \begin{bmatrix}0\\0\\0\end{bmatrix}\right)$	t_5
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\left(c, \begin{bmatrix}0\\0\\0\end{bmatrix}\right)\left(d, \begin{bmatrix}1\\0\\0\end{bmatrix}\right)$	t_5t_7
...	...

$M_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$	
Input-output sequence	Transition sequence
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)$	*null*
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\left(c, \begin{bmatrix}0\\0\\0\end{bmatrix}\right)$	t_6
$\left(\varepsilon, \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\left(c, \begin{bmatrix}0\\0\\0\end{bmatrix}\right)\left(d, \begin{bmatrix}1\\0\\0\end{bmatrix}\right)$	t_6t_7
...	...

Observe that since $\lambda(t_5) = \varepsilon$, it may fire while c is given as input signal. Therefore, even when the initial markings and the transition sequences fired are different, the input-output sequences can be equal. Moreover, after the firing of the first transition (t_5 in the case of M_4 and t_6 in the case of M_5) the marking reached is the same $M_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$ and thereafter, the same sequence of transitions and of input-output signals, can be generated from any of both initial markings.

According to that, the IPN of the example is not observable. In order to provide a structural characterization for observability, a relation on places that, like in this case p_4 and p_5, when marked can generate the same sequence of input-output signals will be established, i.e., the observability analysis presented in this chapter is based on defining a relation between two places p_i, p_j when the same input-output word can be generated by a marking M_i such that $M_i(p_i) = 1$ and a marking M_j such that $M_j(p_j) = 1$.

Definition 42 *Two places p_i, p_j of the IPN given by $(Q, \mathcal{M}_0)$ are* SIGNAL *related (i.e. (p_i, p_j) ∈*SIGNAL*) when there exist firing sequences $z_i^q = (p_i)^\bullet\,(p_i)^{3\bullet}\cdots(p_i)^{2q+1\bullet}$, $z_j^q = (p_j)^\bullet(p_j)^{3\bullet}\cdots(p_j)^{2q+1\bullet}$, (here $(p_i)^{k\bullet}$ represents an element of the set $(\ldots\underbrace{(p_i)^\bullet)^\bullet)\ldots^\bullet}_{k})$) that for all $q \geqslant 0$ fulfill the following two conditions*

- *$\varphi C(\bullet, z_i^q(r)) = \varphi C(\bullet, z_j^q(r))$, i. e. the $r-th$ elements of both sequences generate the same output change and*
- *$\lambda(z_i^q(r)) = \lambda(z_j^q(r))$ or $\lambda(z_i^q(r)) = \varepsilon$ or $\lambda(z_j^q(r)) = \varepsilon$, i. e. the $r-th$ elements of both sequences have the same input symbol or one of them has the null symbol associated.*

For the example given above, (p_4, p_5) ∈SIGNAL because of the existence of firing sequences $t_5t_7t_1t_3t_5t_7\cdots$ and $t_6t_7t_1t_3t_5t_7\cdots$ that, as previously discussed, generate the same input-output words and start with output transitions of the respective places. Moreover, notice that, although SIGNAL relation is given for any value of q, the test can be truncated when a T-component is detected.

Remember now that, as was shown in Chapter 4, the existence of a transition whose firing does not produce any change in the output signal implies non-observability. Likewise, the existence of a marking enabling two different transitions that have associated the same input-output signal also implies that the IPN is not observable. This is because those markings reached by the transitions firings are confused. The interested reader can as an example verify that for the IPN given above, the fact that transitions t_1 and t_2 are non-manipulated and that places p_2 and p_3 have the same output signal associated, imply that each time that one of these transitions fires from the marking $M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ it cannot be immediately determined which one has fired and if it was $M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ or $M_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$ the marking reached.

The following lemma formally establishes that the three conditions that have being discussed are not only necessary but also sufficient for an strongly connected state machine to be observable.

Lemma 43 *A live, strongly connected state machine given by $(Q, \mathcal{M}_0^B)$ is observable if and only if*

1. *$\forall t_k \in T$ it holds that $\varphi C(\bullet, t_k) \neq 0$, and*

2. *For all $p_i, p_j \in P$ it holds that $(p_i, p_i) \in$SIGNAL and $(p_i, p_j) \notin$SIGNAL, $i \neq j$ and*

3. *For all marking $M_i \in \mathcal{M}_0^B$, if $M_i \xrightarrow{t_j} M'_{i+1}$, $M_i \xrightarrow{t_k} M''_{i+1}$ and $M'_{i+1} \neq M''_{i+1}$, then $W(t_j) \cap W(t_k) = \emptyset$.*

Proof. (Sufficiency) Assume that $(Q, \mathcal{M}_0^B)$ is strongly connected state machine that is not observable, then, according with Theorem 28 there is no integer $k < \infty$ such that $\forall \omega \in \Lambda^k(Q, \mathcal{M}_0)$ it holds that $|S_\omega| = 1$. Then, for any k, there is at least one $\omega \in \Lambda^k(Q, \mathcal{M}_0)$ such that $|S_\omega| \geqslant 2$. Without lost of generality, assume that $S_\omega = \{M'_0 M'_1 \cdots M'_{n_1},\ M''_0 M''_1 \cdots M''_{n_2}\}$. Then, the following cases arise:

Case 1.- Sequences $M'_0 M'_1 \cdots M'_{n_1}$, $M''_0 M''_1 \cdots M''_{n_2}$ have different length. Assume that $|M'_0 M'_1 \cdots M'_{n_1}| < |M''_0 M''_1 \cdots M''_{n_2}|$. Then $\varphi(M'_0) = \varphi(M''_0)$ and there is a first marking M'_i in the sequence $M'_0 M'_1 \cdots M'_{n_1}$ such that $\varphi(M'_i) = \varphi(M''_i)$ and $\varphi(M'_i) = \varphi(M''_{i+1})$. Therefore $\varphi(M''_i) = \varphi(M''_{i+1})$ and then, for the transition t_j such that $M''_i \xrightarrow{t_j} M''_{i+1}$ it holds that $\varphi C(\bullet, t_j) = 0$.

Case 2.- Sequences $M'_0 M'_1 \cdots M'_{n_1}$, $M''_0 M''_1 \cdots M''_{n_2}$ have equal length. Since the sequences are different, there exists three possible cases.

Case 2.1.- Sequences have equal prefixes and different suffixes.

Case 2.2.- Sequences have different prefixes and equal suffixes and

Case 2.3.- Sequences are completely different.

In case 2.1. there are markings M'_i, M''_i such that $M'_i = M''_i$ and $M'_{i+1} \neq M''_{i+1}$ while $\varphi(M'_{i+1}) = \varphi(M''_{i+1})$. Therefore there are transitions t_j, t_k such that $M'_i \xrightarrow{t_j} M'_{i+1}$, $M'_i \xrightarrow{t_k} M''_{i+1}$ and, since both marking sequences belong to the same S_ω it holds that $W(t_j) \cap W(t_k) \neq \emptyset$.

In case 2.2. there are markings M'_i, M''_i such that $M'_i = M''_i$ and $M'_{i-1} \neq M''_{i-1}$ while $\varphi M'_{i-1} = \varphi M''_{i-1}$. Since $(Q, \mathcal{M}_0^B)$ is a binary, live and strongly connected state machine, there are places p_a, p_b, p_c such that $M'_{i-1}(p_a) = 1$, $M'_{i-1}(p_b) = 0$, $M''_{i-1}(p_a) = 0$, $M''_{i-1}(p_b) = 1$, $M'_i(p_c) = M''_i(p_c) = 1$ and transitions t_j, t_k such that $t_j \in (p_a)^\bullet \cap {}^\bullet(p_c), t_k \in (p_b)^\bullet \cap {}^\bullet(p_c)$ and $W(t_j) \cap W(t_k) \neq \emptyset$. Then, according to the definition of relation SIGNAL it holds that $(p_a, p_b) \in$SIGNAL.

Finally, in case 2.3., since $M'_0 \neq M''_0$, there are places p_a, p_b such that $M'_0(p_a) = 1$, $M'_0(p_b) = 0$, $M''_0(p_a) = 0$, $M''_0(p_b) = 1$, and infinitely long firing sequences $\sigma' = t'_1 t'_2 \cdots, \sigma'' = t''_1 t''_2 \cdots$, such that $W(\sigma') \cap W(\sigma'') \neq \emptyset$ and

$M_0' \xrightarrow{\sigma'}$, $M_0'' \xrightarrow{\sigma''}$. Therefore, according to the definition of relation SIGNAL it holds that $(p_a, p_b) \in$SIGNAL.

(Necessity) Suppose first that there exists a transition $t_k \in T$ such that $\varphi C(\bullet, t_k) = 0$. Then, since $(Q, \mathcal{M}_0^B)$ is a binary, live and strongly connected state machine, for every k there is a transition sequence σ such that $|\sigma| > k$ and a marking M_i such that $M_i \xrightarrow{t_k} M_{i+1} \xrightarrow{\sigma}$. Since $\varphi C(\bullet, t_k) = 0$ it holds that $\varphi(M_i) = \varphi(M_{i+1})$, therefore $W(t_k\sigma) \cap W(\sigma) \neq \emptyset$ and then for any $\omega \in W(t_k\sigma) \cap W(\sigma)$ it holds that $|S_\omega| > 1$. Assume now that there are places p_a, p_b such that $p_a \neq p_b$ and $(p_a, p_b) \in$SIGNAL. Again, because $(Q, \mathcal{M}_0^B)$ is a live and strongly connected state machine, there are markings $M_0', M_0'' \in \mathcal{M}_0^B$ such that $M_0'(p_a) = 1, M_0'(p_b) = 0$ and $M_0''(p_a) = 0, M_0''(p_b) = 1$. Since $(p_a, p_b) \in$SIGNAL there are infinitely long firing sequences $\sigma' = t_1't_2' \cdots, \sigma'' = t_1''t_2'' \cdots$, such that $W(\sigma') \cap W(\sigma'') \neq \emptyset$ and $M_0' \xrightarrow{\sigma'}$, $M_0'' \xrightarrow{\sigma''}$. Therefore it holds that $W(\sigma') \cap W(\sigma'') \neq \emptyset$ and again for any $\omega \in W(\sigma') \cap W(\sigma'')$ it holds that $|S_\omega| > 1$. Finally assume that there is a marking M_i such that $M_i \xrightarrow{t_j} M_{i+1}'$, $M_i \xrightarrow{t_k} M_{i+1}''$ with $M_{i+1}' \neq M_{i+1}''$, and $W(t_j) \cap W(t_k) \neq \emptyset$. Since $(Q, \mathcal{M}_0^B)$ is live, for any M_0 and any k there is a firing sequence σ such that $|\sigma| > k$ and $M_0 \xrightarrow{\sigma} M_i$. Then, if $W(t_j) \cap W(t_k) \neq \emptyset$ it holds that $W(\sigma t_j) \cap W(\sigma t_k) \neq \emptyset$ and again for any $\omega \in W(\sigma t_j) \cap W(\sigma t_k)$ we have that $|S_\omega| > 1$. Thus, if any of the conditions established in the theorem does not hold there exists an infinitely long sequence of input-output signals ω such that $|S_\omega| > 1$ which, according to Theorem 28, implies that $(Q, \mathcal{M}_0^B)$ is not observable. ■

Notice that the modules obtained by the algorithm `Building PN` are binary state machine PN with a $BCML$ and that by applying algorithm `IPN definition` these modules can be transformed into IPN. Thus, previous result provides sufficient and necessary conditions for these modules to be observable.

Consider again the IPN structure depicted in figure 5.1 and notice that it could indeed be obtained as a module representing a state variable for a given system. Assume now that places p_2 and p_6 share output symbol A and places p_3 and p_4 have output symbols B and C associated, respectively. Also consider all transitions in the module as non-manipulated, therefore having the null input symbol ε associated. The resulting IPN with these input and output functions is depicted in figure 5.2.

Notice that all columns in the corresponding matrix

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

are different from zero.

Also notice that even when places p_2 and p_6 have the same output symbol associated, since $\{p_4\} = ((p_2)^\bullet)^\bullet, \{p_1\} = ((p_6)^\bullet)^\bullet$ and p_4 and p_1 have different output symbols associated, then $(p_2, p_6) \notin$SIGNAL.

Now, it holds that $p_2 \in ((p_1)^\bullet)^\bullet, p_6 \in ((p_5)^\bullet)^\bullet$ and places p_2 and p_6 have the same output symbol, as well as p_1 and p_5. However, $(p_1, p_5) \notin$SIGNAL because

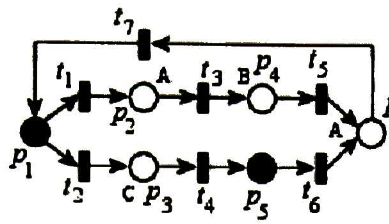

Figure 5.2: An observable *IPN* module.

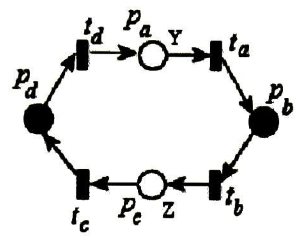

Figure 5.3: An observable *IPN* module.

$(p_2, p_6) \notin$SIGNAL.

Moreover, notice that (being the *IPN* binary) the only possible marking enabling two different transitions, namely t_1 and t_2, is $M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. Besides markings $M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ and $M_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$ reached by the firing of these transitions generate different output symbols.

In resume, all the conditions established in previous lemma hold. Therefore, the *IPN* is observable. This is because condition 1 in the lemma guaranties that all transition firings generate a change in the output and therefore an input-output signal. Condition 2 guaranties that all markings generate different input-output signals, which implies that they will be distinguished and condition 3 guaranties that once that the marking has been distinguished it will not be confuse in any future evolution of the *IPN* marking.

Now that the conditions for an *IPN* module to be observable have been established, we focus on determining conditions for a global model, obtained by composing observable *IPN* modules to be observable.

In order to introduce these results consider the binary *IPN* depicted in figure 5.3. The input and output functions for this net are resumed in the following table.

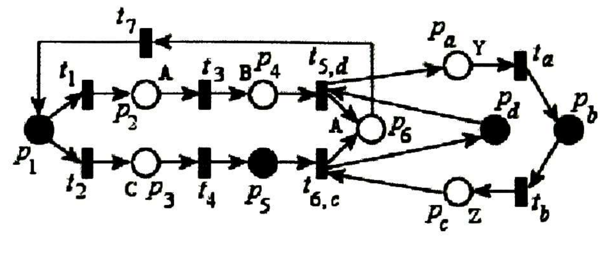

Figure 5.4: An observable IPN obtained by composing two observable modules.

<table>
<tr><th>Function λ</th><th>Function φ</th></tr>
<tr><td>$\lambda(t_a) = \varepsilon$
$\lambda(t_b) = \varepsilon$
$\lambda(t_c) = \varepsilon$
$\lambda(t_d) = \varepsilon$</td><td>$\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$</td></tr>
</table>

Notice that all columns in the resulting matrix

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

are no-null and different from each other. This implies that all transitions generate different input-output signals when firing and therefore, the IPN is observable.

Consider now the binary IPN depicted in figure 5.4. This is the result of composing the IPN depicted in figure 5.2 with the IPN depicted in 5.3 by synchronizing transition t_5 with t_d and transition t_6 with t_c.

In order to verify the observability of the obtained IPN notice that all transition when fired produce different and no-null input-output signals, which can be easily seen by constructing the corresponding φC matrix. However, this condition does not longer guarantee observability. This is because, it is still necessary to verify if any transition firing sequence that may fire will generate sufficient information to compute the IPN marking. Fortunately, in this example, the IPN has only one elemental T-component that contains all the transitions. Therefore, all the sequences of transitions that can be fired in this IPN contain at least one transition belonging to each of the composed modules which, in the case when transition firings of one module cannot be confused with transition firings of the other, is sufficient to guarantee that its entire marking can be computed.

Only in the case when equal output signals are associated to the observable modules composed the transition firings of one of them can be confused with transition firings in others. Fortunately, in practice, it is not common to mix the instrumentation devoted to measure variables in different system components in such a way that their output signals may be confused.

The following theorem formally establishes previous discussion to characterize the observability of the global model obtained by composing observable modules.

Theorem 44 *Let* $\left\{\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)\right\}$ *be a the set of binary, live and observable IPN modules and* $\left(Q, \mathcal{M}^B_0\right)$ *be the binary and live IPN obtained by composing modules* $\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)$. *The IPN given by* $\left(Q, \mathcal{M}^B_0\right)$ *is observable if the following conditions are held:*

1. *For every pair of measurable places* $p_n \in \mathfrak{M}^i_j, p_m \in \mathfrak{M}^k_l$ *where* $\mathfrak{M}^i_j \neq \mathfrak{M}^k_l$, *it holds that* $\varphi(\bullet, n) \neq \varphi(\bullet, m)$ *and*

2. *If there are places* $p_n, p_m \in \mathfrak{M}^i_j$ *such that* $\varphi(\bullet, n) = \varphi(\bullet, m)$ *then all T-component* $\mathbf{X}_a$ *of* $\left(Q, \mathcal{M}^B_0\right)$ *contains at least one transition of the module* $\mathfrak{M}^i_j$.

Proof. Assume that $\left(Q, \mathcal{M}^B_0\right)$ is binary and live and not observable. Then, there exist at least two places $p_n, p_m \in P$ such that $\varphi(\bullet, n) = \varphi(\bullet, m)$ since otherwise any marking can be directly obtained from $y_k = \varphi M_k$ and $\left(Q, \mathcal{M}^B_0\right)$ would be observable. Because of condition 1 if places p_n and p_m are measurable then they belong to the same IPN module $\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)$. Notice then that, since $\left(Q, \mathcal{M}^B_0\right)$ is live and bounded every long enough firing sequence σ contains all the transitions in a T-component $\mathbf{X}_a$ of $\left(Q, \mathcal{M}^B_0\right)$. Then, because of condition 2, the T-component $\mathbf{X}_a$ contains a transition of $\mathfrak{M}^i_j$ and therefore a T-component $\mathbf{X}_b$ of $\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)$. Then, there exists an infinitely long firing sequence in the module $\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)$ which is composed of the transitions in $\mathbf{X}_b$ and whose firing does not provide enough information to compute the markings of p_n and p_m. This implies that $\left(\mathfrak{M}^i_j, \mathcal{M}^{Bij}_0\right)$ is not observable which contradicts the hypothesis. Now, if places p_n and p_m are not measurable they might belong in different modules. However, if there are no other places p'_n and p'_m respectively belonging in the same modules that p_n and p_m then their markings would be uniquely determined by the $BCML$ of each module. Moreover, the knowledge of the markings of places p_n and p'_n does not depend on the knowledge of the markings of places p_m and p'_m because it is sufficient to determine which place of the module containing p_n and p'_n is marked. Therefore, this case reduces to the case when there are two different places with the same output signal in a single module, which has been already addressed and lead to a contradiction. Thus, it is concluded that under conditions 1 and 2 $\left(Q, \mathcal{M}^B_0\right)$ is observable. ■

The first condition in the theorem guaranties that the output signals produced by the firing of transitions in one module are not confused with transition

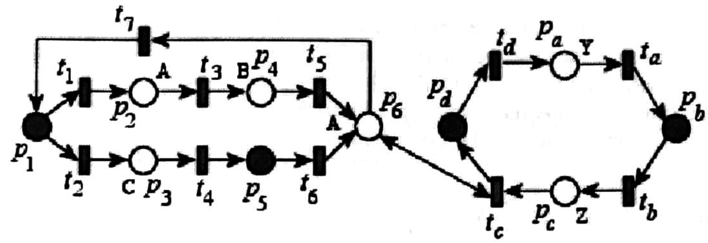

Figure 5.5: A non-observable composition of two observable *IPN* modules.

firings in some other, and the second condition guaranties that any sequence of transition firings will eventually contain transition firings of all modules, which, given that modules are observable, guaranties that the markings of all modules will be known.

According to the way that the permissive composition has been defined, if an *IPN* module $(\mathfrak{M}^1, \mathcal{M}_0^{B1})$ is binary and live, it is covered by a T-component $\mathbf{X}_a$. Now, if $(\mathfrak{M}^1, \mathcal{M}_0^{B1})$ is related with another module $(\mathfrak{M}^2, \mathcal{M}_0^{B2})$ only by permissive compositions then $\mathbf{X}_a$ still is a T-component of the *IPN* resulting by the composition of $(\mathfrak{M}^1, \mathcal{M}_0^{B1})$ and $(\mathfrak{M}^2, \mathcal{M}_0^{B2})$ which does not include any transition in the module $(\mathfrak{M}^2, \mathcal{M}_0^{B2})$. Therefore and because of previous result the next corollary can be established.

Corollary 45 *Let* $\left\{\left(\mathfrak{M}_j^i, \mathcal{M}_0^{Bij}\right)\right\}$ *be a the set of binary, live and observable IPN modules and* $(Q, \mathcal{M}_0^B)$ *be the binary and live IPN obtained by composing modules* $\left(\mathfrak{M}_j^i, \mathcal{M}_0^{Bij}\right)$. *The IPN given by* $(Q, \mathcal{M}_0^B)$ *is not observable if there exists a module* $\left(\mathfrak{M}_j^i, \mathcal{M}_0^{Bij}\right)$ *with places* p_n, p_m *such that* $\varphi(\bullet, n) = \varphi(\bullet, m)$ *and* $\left(\mathfrak{M}_j^i, \mathcal{M}_0^{Bij}\right)$ *is related with the remaining modules only by permissive compositions.*

The *IPN* depicted in figure 5.5 results from composing the *IPN* modules of figures 5.2 and 5.3 described above by a permissive composition between place p_6 and transition t_c. Notice that places p_b and p_d have the same output symbol associated, the null one. Thus, according to previous corollary, the *IPN* resulting from this composition is not observable. This is because, if only permissive compositions exist, then the *IPN* module of figure 5.2 can infinitely evolve without needing the *IPN* of figure 5.3 to evolve. Thus, there is no way to guarantee that the marking of this module will ever be known.

5.3 Observability invariance

In this section, it is shown that observability is invariant under the transformation technique given in Chapter 3. This observability invariance arises from the fact that the reachability graphs of both, the original and the transformed model are the same and also that the input-output sequences are not affected with the transformation. Therefore, any dynamic property of the model which is based on the sets of transition firings and the corresponding input-output sequences is invariant under the transformation technique herein proposed. The following theorem proves this fact for the case of observability. Proving that other properties such as controllability are also invariant under the transformation technique is beyond the scope of this work.

Theorem 46 *Let $\left(Q, \mathcal{M}_0^B\right)$, $(Q', \mathcal{M}_0)$ be two IPN where Q' is a place reduction codification obtained from Q when a code transformation matrix $\mathfrak{T}$ is used, then $(Q', \mathcal{M}_0)$ is observable if $\left(Q, \mathcal{M}_0^B\right)$ is observable.*

Proof. Assume that $\left(Q, \mathcal{M}_0^B\right)$ is an observable *IPN*, then there exists an integer $k < \infty$ and a function Ψ such that $\forall \omega \in \Lambda^k \left(Q, \mathcal{M}_0^B\right)$ it holds that $\Psi\left(\omega, \left(Q, \mathcal{M}_0^B\right)\right) = (M_0, M_i)$. Thus, for the same k, a function $\Psi'\left(\omega, (Q', \mathcal{M}_0)\right) = (M_0', M_i')$ can be obtained as $\Psi'\left(\omega, (Q', \mathcal{M}_0)\right) = (\mathfrak{T}M_0, \mathfrak{T}M_i)$. ■

Figure 5.6 *a*) shows a reduced *IPN* module for module depicted in figure 5.2, obtained with the code transformation matrix

$$\mathfrak{T} = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Notice that places p_a and p_d have output symbols R and T associated, respectively, and that places p_b and p_c have no output symbol associated.

Figure 5.6 *b*) presents the reachability graph for the reduced *IPN* module which can be easily verified to be equivalent to the reachability graph corresponding to the original *IPN*. Notice that transitions t_a and t_f in the reduced module produce the same output change when firing, which is in accordance with the fact that the firings of transitions t_1 and t_6 generate the same output change in the original module. Also notice that all other transition firings in the reduced model generate different output changes, and that this is the same that holds for the remaining transitions in the original module. That is mainly the reason why observability is invariant under the transformation, i.e., the reachability graph is not changed and the sequences of input-output signals generated by the resulting module are equivalent to those generated by the original one. This indeed guaranties that all dynamic properties that only depend on those sequences and the reachability graph are preserved by the transformation technique.

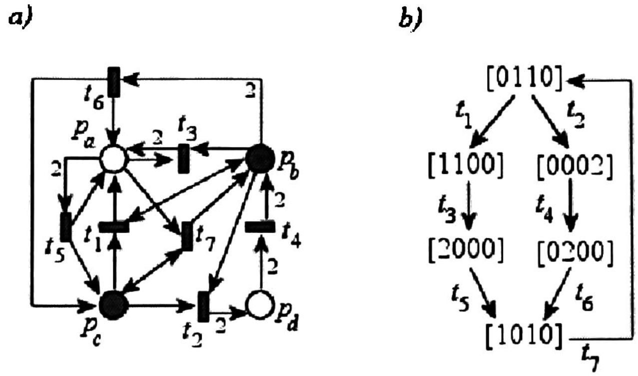

Figure 5.6: *a*) Reducted module, *b*) reachability graph.

5.4 Conclusions

Necessary and sufficient conditions that can be verified in polynomial time, for binary state machine *IPN* to be observable have been derived in this chapter. Since the modules obtained by applying the modeling methodology presented in chapter 3 belong to this class of *IPN*, then it a polynomial algorithm to verify observability in the modules is available. Moreover, conditions that may be applied in the *DES* design to obtain observable systems from observable modules have been presented. Finally, it has been proved that the reduction technique previously given preserves observability.

Chapter 6

Concluding remarks

This work included three major contributions: the first one is a modeling methodology which is modular and allows to describe *DES* in terms of *IPN* from a verbal description of the system and avoids the tuning stage that other modeling methodologies include in order to eliminate the state trajectories included in the model, that are not feasible in the system. Along with the modeling methodology, a reduction technique which has been shown to preserve dynamic properties was presented. The observability definition, along with characterizations of this property for a broad class of *IPN* models it the second major contribution of this work. The observability concept, as herein presented, can be applied to any dynamic system represented by a state variable model and the characterizations of this property can be verified in polynomial time. The third contribution of these work is an analysis of the observability of *IPN* models obtained applying the proposed modeling methodology which provide conditions can be applied for *DES* design to guarantee that the obtained system will exhibit observability.

Bibliography

[1] L. Aguirre, A. Ramírez, O. Begovich. "Design of asymptotic observers for discrete event systems". *Proc. IASTED International Conference on Intelligent Systems and Control*, pp. 188-193, Santa Barbara, USA, Oct. 1999.

[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 1990.

[3] J. Billington. "Development of an international standard for High-leve Petri nets" *Third IEEE International Software Engineering Standards Symposium and Forum 97, 'Emerging International Standards'.* pp. 155-162. June 1997.

[4] P.E. Caines, R. Greiner, S. Wang. "Dynamical logic Observers for Finite Automata", *Proc. 27th Conf. on Desicion and Control*, pp. 226-233, Austin, Texas, Dec. 1988.

[5] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers. 1999.

[6] V. Chandra and R. Kumar, "A DiscreteEvent Systems Modeling Formalism Based on Event Occurrence Rules and Predicates". IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pp.785-794. Dec. 2001.

[7] C. T. Chen. *Linear system theory and design.* Harcourt Brace Jovanovich Inc. Sanders College Publishing. 1970.

[8] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. "Supervisory control of discrete event processes with partial observation". *IEEE Transaction on Automatic Control*, vol. 33, no. 3, pp. 249-260, 1988.

[9] M. Cüneyt Özveren and A. S. Willsky. "Observability of discrete event dynamic systems". *IEEE Transaction on Automatic Control*, vol. 35, no. 7, pp. 797-806, Jul. 1990.

[10] J. Desel and J. Esparza. *Free choice Petri nets.* Cambridge University Press. 1995.

[11] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva and F.B. Vernadat. *Practice of Petri Nets in Manufacturing.* Chapman & Hall Ed. Great Britain, 1993.

[12] P. Freedman. "Time, Petri Nets, and Robotics" *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 417-433, Aug. 1991.

[13] A. Fanni, A. Giua, N. Sanna, "Control and error recovery of Petri net models with event observers", *Proc. 2nd Int. Work. on Manufacturing and Petri Nets*, France, pp. 53-68, Jun. 1997.

[14] A. Fanni, and A.Giua. "Discrete Event Representation of Qualitative Models Using Petri Nets" *IEEE Transactions on Systems, Man and Cybernetics-Part B:Cybernetics*, vol. 28, no. 6, pp. 770-780, Dec. 1998.

[15] A. Giua, "Petri net state estimators based on event observation", *Proc. of the IEEE 36th CDC*, pp. 4086-4091, Dec. 1997.

[16] A. Giua, C. Seatzu, "Observability Properties of Petri Nets", *Proc. IEEE 39th CDC*, pp. 2676-2681, Sydney, Australia, Dec. 2000.

[17] L.E. Holloway and S. Chand. "Time templates for discrete event fault monitoring in manufacturing systems" *Proc. of the 1994 American Control Conference*, Baltimore, USA, Jun. 1994.

[18] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages and computation.* Addison-Wesley. 1979.

[19] A. Ichikawa and K. Hiraishi. "Analysis and control of discrete event systems with irregular observations". *Lecture notes in control and information sciences*, vol. 103, Springer Verlag, pp. 115-134, 1987.

[20] M. Jeng, F.DiCesare. "A review of synthesis techniques for Petri nets with applications to Automated Manufacturing systems". *IEEE Transactions on Systems, Man, and Cybernetics.* vol. 23, no. 1, pp. 301-312, Jan/Feb 1993.

[21] S. Jiang and R. Kumar. "Decentralized Control of Discrete Event Systems with Specializations to Local Control and Concurrent Systems". *IEEE Transactions on Systems, Man and Cybernetics-Part B:Cybernetics*, vol. 30, no. 5, pp. 653-660. Oct 2000.

[22] G. Juhás and R. Lorenz. "Modelling with Petri Modules". Synthesis and Control of Discrete Event Systems. pp. 125-138. Kluwer Academic Publishers. Edited by B. Cailland et al. 2001.

[23] H.K. Khalil. *Nonlinear systems.* Second edition. Prentice-Hall. 1996.

[24] I. Koh and F. DiCesare. "Synthesis rules for colored Petri nets and their applications to automated manufacturing systems". *Proc. of the 1991 IEEE International Symposium on Intelligent Control.* pp. 152-157. Arlington, USA. Aug. 1991.

[25] R. Kumar, M.A. Shayman. "Formulae relating controllability, observability and co-observability" *Automatica*, pp. 211-215, vol. 2, Mar. 1998.

[26] Y. Li and W.M. Wonham. "Controllability and observability in the state-feedback control of discrete event systems". *Proc. IEEE. 27th CDC*, Dec. 1988.

[27] Y. Li and W.M. Wonham. "Control of Vector Discrete-Event Systems I - The Base Model" *IEEE Transactions on Automatic Control*, vol. 38, no. 8, pp. 1214-1227, Aug. 1993.

[28] F. Lin and W.M. Wonham. "Decentralized control and coordination of discrete-event systems with partial observation" *IEEE Transactions on Automatic Control*, vol. 35, no. 12, pp. 1330-1337, Dec. 1990.

[29] D.G. Luenberger. *Introduction to dynamic systems: Theory, Models & Applications*. John Wiley & Sons. Quinn-Woodbine, Inc. 1979.

[30] M. E. Meda and A. Ramírez. "Identification in discrete event systems" *IEEE International Conference Systems, Man and Cybernetics*, pp. 740-745, San Diego, USA, Oct. 1998.

[31] T. Murata. "Petri nets: Properties, analysis and applications". *Proc. IEEE*, vol. 77, no. 4, pp. 541-580, Apr. 1989.

[32] K. Ogata. *Modern control engineering*. Second Edition. Prentice Hall. 1990.

[33] L. Perko. *Differential Equations and Dynamic Systems*. Texts in applied Mathematics, Springer 1996.

[34] J.H. Prosser, M. Kam, and H.G.Kwatny. "On-line supervisor synthesis for partially observed discrete-event systems". *IEEE Transaction on Automatic Control*, vol. 43, no. 11, pp. 1630-1634, Nov. 1998.

[35] P.J. Ramadge. "Observability of discrete event systems". *Proc. 25th IEEE Conf. Desicion and Control*, pp. 1108-1112, Athens, Greece, Dic. 1986.

[36] P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206-230, 1987.

[37] A. Ramírez-Treviño, I. Rivera-Rangel and E. Lopez-Mellado. "Observer Design for Discrete Event Systems Modeled by Interpreted Petri Nets" *Proc. IEEE ICRA 2000*, pp. 2871-287, San Francisco, USA, Apr. 2000.

[38] A. Ramírez-Treviño, I. Rivera-Rangel, E. López-Mellado. "Observability of Discrete Event Systems Modeled by Interpreted Petri Nets". *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 557-565, Aug. 2003.

[39] G. Ramírez-Prado, A. Santoyo, A. Ramírez-Treviño and O. Begovich. "Regulation problem in discrete event systems using Petri nets" *Proc. IEEE Systems Man and Cybernetics*, vol. 3, pp. 2174-2179, Nashville, USA, 2000.

[40] I. Rivera-Rangel, L. Aguirre-Salas, A. Ramírez-Treviño and E. Lopez-Mellado. "A Petri Net Scheme for DES State Estimation" *Proc. IEEE 39th CDC*, pp. 2260-2265, Sydney, Australia, Dec. 2000.

[41] M. Silva. *Petri nets on automatica and data processing.* Madrid, Spain: AC Ed., 1985, [in Spanish].

[42] T. Ushio. "On the existence of finite-state supervisors under partial observations". *IEEE Transactions on Automatic Control*, vol. 42, no. 11, pp. 1577-1581, 1997.

[43] W. M. Wonham. Linear Multivariable Control. *A geometric approach.* New York, USA: Spring-Verlag, 1985.

[44] Z. Zhang and W.M. Wonham. "STCT: An efficient algorithm for supervisory control design". Synthesis and Control of Discrete Event Systems. pp. 77-100. Kluwer Academic Publishers. Edited by B. Cailland et al. 2001.

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Observabilidad y síntesis modular de modelos en redes de Petri de Sistemas de Eventos Discretos

del (la) C.

José Israel RIVERA RANGEL

el día 14 de Octubre de 2004.

Dr. Bernardino Castillo Toledo
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Ofelia Begovich Mendoza
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. José Javier Ruíz León
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

A. Ramírez

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dr. Manuel Silva Suárez
Catedrático
Centro Politécnico Superior de la
Universidad de Zaragoza