University of La Rioja

Doctoral Thesis

Titulo de la tesis

Author:
Iñigo León

Supervisor:
Dr. Emilio Jimenez

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Research Group Name
Department of Electrical Engineering

June 2015

Declaration of Authorship

I, Iñigo León, declare that this thesis titled, 'Titulo de la tesis' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: June 2015

"'Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY OF LA RIOJA

Abstract

Faculty of Science, Agrifood Studies and Computing
Department of Electrical Engineering

Doctor of Philosophy

Titulo de la tesis

by Iñigo León

El abstract debe caber en esta pagina, centrada verticalmente. Debe comenzar o contener "My original contribution to knowledge is. . . "

Acknowledgements

//BORRAR The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Thanks to my years of personal study and my curiosity for new knowledge, I fought to combine my job, my family and studies. I has been a difficult work, but the result has been worth it.

//TODO

Contents

List of Figures

Abbreviations

PN	**P**etri **N**et
SN	**S**ub**N**et
...	//TODO

For/Dedicated to/To my...

Chapter 1

Introduction

Que sea extensa para que la lectura no sea pesada. Poner tambien las keywords

1.1 Background of the research

Petri nets are widespread for modeling many classes of systems, such as manufacturing logistics processes and services [1, 2], concurrent systems [3]. However, all these nets are described in a comprehensive way and must have the information of the entire net to determine its evolution.

1.2 Research problem

The problem occurs when somebody doesn't want to describe the whole subnet. Or, maybe, is wanted that one part of the process is only accessible for one specific person or entity.

The first approach to solve this problem is to take two Petri nets

- One Petri Net with the public information, extracting the private data. This is an incomplete model of the process
- Another Petri Net with the whole information for the interested person or entity.

As you can notice, this is not an efficient way to publish this kind of Petri Nets.

1.3 Justification of the research

It would be interesting to take a Petri net and hide a part of it. This can be useful, for example, distributing a process we want to be secret [4], or simply to be a part of the net to be complex and do not interested handle for any reason [4]. So here is my contribution. I have researched the possibilities of hiding a part of a Petri Net so that everybody can access the public information, maintaining the secret of the private data. This private data is accessible only for authorized people.

Some authors study the possibilities of Petri nets reduction [5], grouping in one place or transition a subnet, so that what happens on this subnet, is encapsulated in a single point of execution. However, we want to go further by defining parts of the net that are hidden (not clustered) and what are the implications, studied within network properties.

The main objective of this thesis is to hide parts of a Petri Net, maintaining visible other parts.

1.4 Methodology

In order to achieve this goal, I have defined three milestones:

1. Extend Petri Nets definition in order to define the public and the private information
2. Choose a lossless and extendible representation of this kind of Petri Nets
3. Define a hiding method for this representation

For the first milestone, I work for the creation of the theoretical basis for further study of Petri nets in which certain parts are hidden. So we setup a generic framework of definitions and notations that allow us to deepen in the study of the characteristics and properties of Petri nets and their subnets [6, 7]. Also mention work already carried by other researchers in which we rely for our goal (i.e. [3, 8, 9, 10]). All this will be necessary to create the framework that allows us to study occultation in Petri nets. We will expand the vision of Petri nets, providing them with greater functionality, such us attachable subnet.

The next step in this work is to choose (or define) a flexible representation of Petri nets that allows us to translate the previous extended nets. This representation has to be really extendible and flexible in order to be able to show actual and future characteristics

of Petri nets. I can advance you that the selected representation is the standard PNML and I have to define and extension for it in order to represent subnets that are going to be hidden.

Once selected this representation, the last step is the hiding method. Once more, I bet for standard protocols like XMLEncription.

This is a very basic investigation because I extend the very early definitions of Petri nets. Because of it, the results of this thesis is very probably extensible to any other development whose base are the classic Petri Nets. For example, I am not going to study colored Petri nets, neither timed Petri nets, etc. But it is very easy to see that the results achieved in this thesis can be applied to them with no problem.

1.5 Delimitations of scope and key assumptions

For this work we will always deal with ordinary and pure networks, unless otherwise expressly. This assumption is only for clarity reasons, because the protocols and methods described in this work are perfectly extensible to other kind of Petri nets, as long as these Petri nets are representable in PNML format.

Chapter 2

Literature review

//BORRAR

- G - Petri nets general
- SM - Simulacion de modelos
- EPN - Extensiones de Petri nets originales
- R - Reduccion de redes
- SN - Subredes
- REP - Representation
- PROP - Propiedades de PN
- HID - Ocultacion de partes
- PNML - PNML
- PNMLE - Extensiones PNML

//

In this chapter I am going to go over the ancient Petri net history. This work is a very basic investigation on Petri nets. This means that the most of the references are very general.

For this literature review, I will follow the structure of this thesis:

1. First of all, I am going to describe some generalities of Petri nets as an introduction

2. Then I will describe subnets and the process of splitting Petri nets into several subnets. Some of those subnets are going to be hidden
3. The next step is to explain some possible Petri net representations and my selection of PNML for the hiding process
4. Once selected PNML, I am going to extend this language to support the subnets defined before
5. The last step is the hiding properly speaking. To do this, I will use the standard XMLEncryption for ciphering the secret information

2.1 Introduction. Petri nets

In the 60's, Carl Adam Petri invented a new way to describe distributed systems called Petri nets [11, 12, 13]. Many net theories are based on those works[14]. Nowadays, Petri nets are really extended to represent discrete systems [15, 16, 17, 18]. There are lots of applications of Petri nets:

- Modelling of sequential processes[19], concurrent systems[20, 21], manufacturing [8, 22, 22, 23, 24], logistic processes [2],...
- Simulation of industrial applications [25, 26], logistic and production systems [1],...

//TODO Incluir mas aplicaciones de las redes de Petri

This work is not about of Petri net application, so I am not going to deepen this field. However I really mind the intrinsic structure of them. Since the definition of Petri nets, many authors investigated about them. The basic basis of my work are some of the best Petri net researchers, who are included in this review:

- Murata, T [6, 27, 28]
- Silva, M [7, 29, 30]
- Peterson, JL [10]

And not only general Petri nets. Any kind of extensions are well received

- Jensen, K [20, 31]
- ...

2.2 Subnets

2.3 Petri nets representation. PNML

2.4 PNML extension for hiding support

2.5 Hide information on Petri nets. XMLEncryption

Chapter 3

Private information in Petri nets. Subnets

3.1 Introduction

By default, a Petri net is described in a comprehensive and public way. In this chapter I am going to describe a way to declare private information of a Petri net. This information is candidate to be hidden.

First of all I have to create a like framework of definitions, properties and methodologies in order to achieve this goal. The main idea is the concept of subnet an its interaction front-end.

Once defined this subnets, the Petri net can be divided into public and private chunks only by ordering the places and transitions in one subnet or another.

3.2 Petri subnets

3.2.1 Definitions and properties

Let P and T the non-empty finite sets of places and transitions of a Petri net, respectively. Let $|P| = n$ (the number of places of the net) and $|T| = m$ (number of transitions of the net). Let α and β pre and post incidence matrices respectively. Let $N = \langle P, T, \alpha, \beta \rangle$ be a Petri net and let C the incidence matrix of N

Definition 3.1 (Subnet [7])**.** A subnet of $N = \langle P, T, \alpha, \beta \rangle$ is a net $\overline{N} = \langle \overline{P}, \overline{T}, \overline{\alpha}, \overline{\beta} \rangle$ so that $\overline{P} \subseteq P$ y $\overline{T} \subseteq T$, $\overline{\alpha}$ y $\overline{\beta}$ are restrictions of α and β over $\overline{P} \times \overline{T}$.

In other words, a subnet is a subset of places and transitions together with the arches that connect them together.

Let's look at the implications of the latter definition since it is one of the most important with regard to this work.

A subnet corresponds [4], in terms of matrices, with the resulting submatrix keeping only the rows and columns corresponding to places and transitions of the selected subnet.

Example 3.1. *Let's take the Petri net which has the following incidence matrix:*

$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\ \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \end{array}$$

If we select the places p_1, p_3, p_4 and p_5 and transitions t_1, t_2, and t_6 we have the subnet defined by this incidence matrix

$$C' = \begin{array}{c} \\ p_1 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_6 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

by simply erasing p_2 and p_6 rows and t_3, t_4 and t_5 columns.

In [4] is shown that the set of all possible permutations of rows and/or columns of a matrix of incidence corresponding to a Petri net, either the prev or post incidence matrix, make an equivalence relation. In other words, given an incidence matrix, both rows and columns can be rearranged and this rearrangement describes exactly the original Petri net.

In this way, we can study the incidence matrices reordering rows and columns as preferred one at any time, without loss of generality.

From all these definitions and proofs we can draw several trivial conclusions:

1. A subnet, like generic Petri net does not have to be square.

$$
\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{array}
\begin{array}{c}
\begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\
\left(\begin{array}{cccccc}
-1 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 1 & -1
\end{array} \right)
\end{array}
\equiv
\begin{array}{c} \\ p_8 \\ p_1 \\ p_3 \\ p_6 \\ p_4 \\ p_5 \\ p_2 \\ p_7 \end{array}
\begin{array}{c}
\begin{array}{cccccc} t_1 & t_6 & t_3 & t_5 & t_4 & t_2 \end{array} \\
\left(\begin{array}{cccccc}
0 & -1 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 0
\end{array} \right)
\end{array}
$$

FIGURE 3.1: Two equivalent incidence matrices to describe the same Petri net

2. If a row or column of the incidence matrix of the subnet is all zeros, it doesn't mean that that place or that transition is isolated. This occurs only with pure nets. //TODO EJEMPLO/FIGURA

3. It does not matter the number of places and/or transitions that are chosen for the subnet, as long as they are not empty sets.

3.2.2 Splitting a net into subnets

Let $N = \langle P, T, \alpha\beta \rangle$ a Petri net where $|P| = n$ and $|T| = m$. So $P = \{p_1, p_2 ... p_n\}$ and $T = \{t_1, t_2 ... t_m\}$. Select two subsets $P' \subseteq P$ and $T' \subseteq T$ so that $|P'| = r \leq n$ and $|T'| = s \leq m$. With these premises divide into two subnets the original one.

We have seen that we can identify a subnet simply removing rows and columns (places/-transitions) of an incidence matrix. Taking advantage of the equivalence relation defined in [4], we reorder the incidence matrix so that places and transitions of the subnet are in the top-left. Without loss of generality, and for convenience, places and transitions can be renamed so that the incidence matrix is as follows:

$$
C = \begin{array}{c}
\begin{array}{cccccc} & t_1 & \cdots & t_s & t_{s+1} & \cdots & t_m \end{array} \\
\left(\begin{array}{c|cc|cc}
p_1 & a_{11} & \cdots & a_{1s} & a_{1(s+1)} & \cdots & a_{1m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
p_r & a_{r1} & \cdots & a_{rs} & a_{r(s+1)} & \cdots & a_{rm} \\
\hline
p_{r+1} & a_{(r+1)1} & \cdots & a_{(r+1)s} & a_{(r+1)(s+1)} & \cdots & a_{(r+1)m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
p_n & a_{n1} & \cdots & a_{ns} & a_{n(s+1)} & \cdots & a_{nr}
\end{array} \right)
\end{array}
$$

We now have the net divided into two disjoint and complementary subnets. They are disjoint because there is no place and no common transition, and complementary because the union of the two we gives the complete net. At this point note that the incidence matrix is divided into four blocks $C = \begin{pmatrix} N_1 & PIM_{12} \\ TIM_{12} & N_2 \end{pmatrix}$. The interpretation is as follows:

- N_1 subnet made up of places $p_1..p_r$ and transitions $t_1..t_s$
- N_2 subnet that is complementary to N_1, made up of the places $p_{r+1}..p_n$ and transitions $t_{s+1}..t_m$
- PIM_{12} (Places Influence Matrix) is the matrix that defines the interaction of the N_1 places with N_2 transitions. Basically it is the matrix whose elements are those that are in the same rows of N_1 but outside of it (rows $1..s$ and columns $s+1..m$).
- TIM_{12} (Transitions Influence Matrix) is the matrix that defines the interaction of N_1 transitions with N_2 places. It is the matrix whose elements are in the same columns of N_1 elements but outside of it (rows $r+1..n$ and columns $1..s$).

We can notice that $PIM_{12} = TIM_{21}$ and $PIM_{21} = TIM_{12}$ by applying the definition.

This can be generalized to multiple disjoint and complementary subnets without further to re-apply the same process to any of the subnets already defined. Thus, generically we can divide a network into i subnetworks, so we'll have a matrix of this style:

$$\left(\begin{array}{ccc|ccc|c|ccc}
a_{11} & \cdots & a_{1s} & a_{1(s+1)} & \cdots & a_{1t} & & a_{1u} & \cdots & a_{1m} \\
\vdots & \mathbf{N_1} & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\
a_{p1} & \cdots & a_{ps} & a_{p(s+1)} & \cdots & a_{pt} & & a_{pu} & \cdots & a_{pm} \\
\hline
a_{(p+1)1} & \cdots & a_{(p+1)s} & a_{(p+1)(s+1)} & \cdots & a_{(p+1)t} & & a_{(p+1)u} & \cdots & a_{(p+1)m} \\
\vdots & \ddots & \vdots & \vdots & \mathbf{N_2} & \vdots & \cdots & \vdots & \ddots & \vdots \\
a_{q1} & \cdots & a_{qs} & a_{q(s+1)} & \cdots & a_{qt} & & a_{qu} & \cdots & a_{qm} \\
\hline
 & \vdots & & & \vdots & & \ddots & & \vdots & \\
\hline
a_{r1} & \cdots & a_{rs} & a_{r(s+1)} & \cdots & a_{rt} & & a_{ru} & \cdots & a_{rm} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \mathbf{N_i} & \vdots \\
a_{n1} & \cdots & a_{ns} & a_{n(s+1)} & \cdots & a_{nt} & & a_{nu} & \cdots & a_{nm}
\end{array}\right)$$

In this situation, if we select two subnets N_j and N_k, we locate the zones of influence of each with respect to the other:

$$\left(\begin{array}{c|c|c|c|c} \ddots & \cdots & \cdots & \cdots & \cdots \\ \hline \vdots & SN_j & \cdots & PIM_{jk} = TIM_{jk} & \cdots \\ \hline \vdots & \cdots & \ddots & \cdots & \cdots \\ \hline \vdots & TIM_{jk} = PIM_{kj} & \cdots & SN_k & \cdots \\ \hline \vdots & \vdots & \cdots & \vdots & \ddots \end{array}\right)$$

Thus, the submatrix PIM_{jk} represents the arcs that connect places of the submatrix N_j with N_k transitions and the matrix TIM_{kj} represents the arcs that connect places of SN_k to SN_j transitions.

Notation. For simplicity, we will call

- PIM$_i$ to all the elements in the same rows of N_i but outside of it.
- TIM$_i$ to all the elements in the same columns of N_i but outside of it.

We can notice again that $PIM_{jk} = TIM_{kj}$ and $PIM_{kj} = TIM_{jk}$ by applying the definition.

Definition 3.2 (Partition of a Petri net into subnets). We say that a set $P = \{N_1 N_2 ... N_k\}$ is a partition into subnets of N if the following holds:

- $N_1 \cup N_2 \cup ... \cup N_k = N$
- $\forall i, j | 1 \leqslant i, j \leqslant k \Rightarrow \ N_i \cap N_j = \emptyset$ (pairwise disjoint)

3.2.3 Subnet classification

Depending on how each of the four pieces of matrix (N_1, N_2, PIM and TIM) are, we can see some special cases.

3.2.3.1 Disjointed subnets

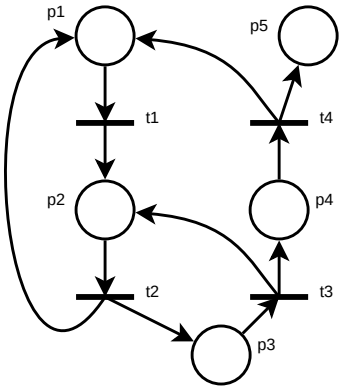
Definition 3.3 (Disjointed subnet). Pure subnet said disjointed if there is no arc between its own places and transitions

Suppose that in the incidence matrix divided into the four pieces explained, are N_1 or N_2 be a null matrix. In this case the interpretation is that there arcs between places and transitions of the subnet, which would simply places and/or no transitions related to each other but with the additional subnetwork. Subnet talk then disjointed.

Proposition 3.4. *A subnet is disjoint iif its incidence submatrix is the null matrix.*

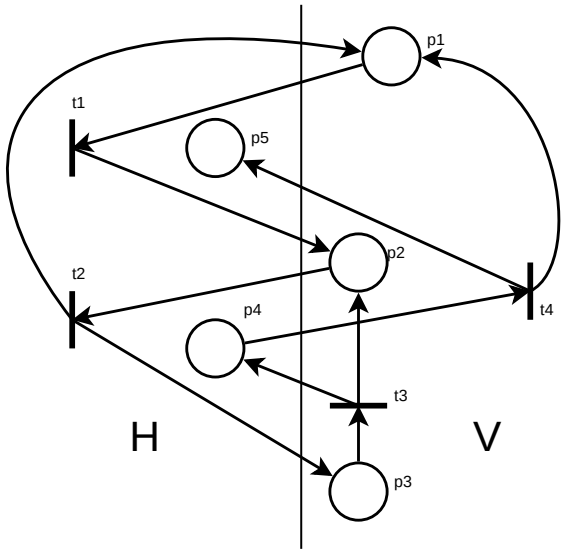
//TODO demostracion??

Example 3.2. *Consider the next well-printed Petri net and its incidence matrix:*



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

We assume that we select as subnet the one formed by 4th and 5th places and transitions 1 and 2. Then the graph and the incidence matrix are thus:



$$C = \begin{array}{c} \\ p_4 \\ p_5 \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ \hline -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{array}\right) \end{array}$$

Here we can see that although really p_4, p_5, t_1 and t_2 are not isolated, there is no arc that connects them together. In the incidence matrix, the corresponding submatrix is the zero matrix. Therefore, whether or not there are elements isolated in the net, total subnet formed by p_4, p_5, t_1 and t_2 is a disjointed subnet.

We can extend this definition to a subnet that is part of a bigger net. It doesn't matter in how many subnets it is separated: if one subnet is the null matrix, this subnet is disjointed. This characteristic is implicit to the selected subnet and it doesn't depend on the rest of the net.

3.2.3.2 Macroplace

//TODO Introduccion

Definition 3.5 (Macroplace). A macroplace is a subnet that meets the following:

1. arcs entering any node of the subnet from an external node come from a transition.
2. arcs leaving any node on the subnet to an external node go to a transition.

Proposition 3.6. *A subnet N_1 is a macroplace iif its transition influence matrix (TIM) is the null matrix. In the same way, N_2 is a macroplace iif its place influence matriz (PIM) is the null matrix*

$$N_1 \textit{ is macroplace} \iff \forall a_{ij} \in TIM_1, a_{ij} = 0$$

$$N_2 \textit{ is macroplace} \iff \forall a_{ij} \in PIM_1, a_{ij} = 0$$

TODO demostracion??

Suppose that the incidence matrix divided into the four pieces explained, TIM appears to be the zero matrix. Then we conclude that the subnet N_1 is only related by arcs with places of subnet N_2. All arcs entering N_1 come from transitions of N_2 and all arcs coming out from N_1 go to transitions of N_2. Stated another way, the subnet N_1 behaves like a place, but may contain places and transitions.

Note that this is not really a place, and that the subnet has not marked as such. The marking is on the places within the subnet and depends on the arches of arrival.

Example 3.3 (Macroplace). *//TODO*

We can extend this definition to a subnet that is part of a bigger net too. However, in this time, it does matter the way in which the bigger net is separated. This characteristic is not implicit to the selected subnet and depends on the rest of the net.

//TODO explicar macroplace dependiendo del resto de subredes. Una subred puede comportarse como macrolugar sobre otra subred, pero no sobre una tercera.

//TODO macroplace absoluta si da igual el resto de subredes

//TODO Caracterizacion de macroplace absoluta

3.2.3.3 Macrotransition

Definition 3.7 (Macrotransition). A macrotransition is a subnet that meets with the following:

1. arcs entering any node of the subnet from an external node come from a place.
2. arcs leaving any node on the subnet to an external node go to a place.

Proposition 3.8. *A subnet is a macrotransition iif its place influence matrix (PIM) is the null matrix.*

//TODO demostracion??

This is other option that can happen is that in the incidence matrix: PIM appears to be the zero matrix. Then we conclude that the subnet N_1 is only related by arcs with places of subnet N_2. All arcs entering N_1 come from places of N_2 and all arcs coming out from N_1 go to places of N_2. Stated another way, the subnet N_1 behaves like a transition, but may contain places and transitions.

Like macroplaces, macrotransitions are not transitions as such. It is not necessary that all entries are marked to fire the macrotransition, and not all output places are marked after entering it. Everything depends on the inner workings of the macrotransition.

Example 3.4 (Macrotransition). *//TODO*

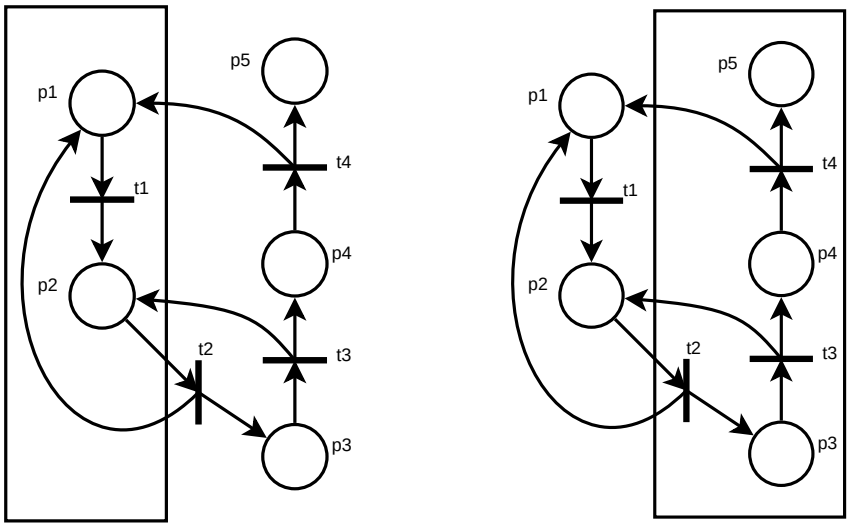
We can extend this definition again to a subnet that is part of a bigger net too. Like with macroplaces, it does matter the way in which the bigger net is separated.

//TODO explicar macrotransition dependiendo del resto de subredes. Una subred puede comportarse como macrotransition sobre otra subred, pero no sobre una tercera.

//TODO macrotransition absoluta si da igual el resto de subredes

//TODO Caracterizacion de macrotransicion absoluta

//TODO Explicar relacion entre una macrotransicion y un macrolugar en la misma red, con subredes interrelacionadas por esta caracteristica. Una subred es macrolugar sobre otra si y solo si esta ultima es macroplace sobre la primera. Comprobarlo y demostrarlo en su caso



a) Macroplace b) Macrotransition

FIGURE 3.2: Macroplace y macrotransition //TODO quitar esta figura y poner en los ejemplos de macrolugar y macrotransicion

3.2.3.4 Sinkhole subnet

Another thing that can happen is that the hidden subnet reach only arcs. We then find that you can not leave the subnet. We speak then of a sinkhole subnet.

Definition 3.9 (Sinkhole subnet). It is said that a subnet is a sinkhole subnet if no arc has its origin in an internal node (place or transition) of the subnet.

It is easy to see that a subnet is sinkhole if and only if all elements of PIM are greater or equal to zero and all elements of TIM are less than or equal to zero.

$$N_1 \text{ is sinkhole} \iff \forall a_{ij} \in PIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in TIM_1, a_{pq} \leq 0$$

//TODO demostracion??

Example 3.5 (Sinkhole Subnet). *//TODO*

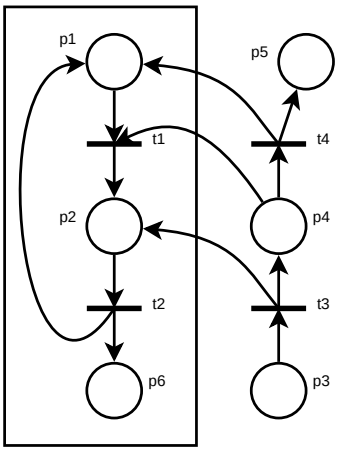
If we can extend this definition to a subnet that is part of a bigger net, in this case, it does matter the way in which the bigger net is separated. This characteristic is not implicit to the selected subnet and depends on the other subnets.

//TODO reescribir este parrafo anterior para que no sea igual que los anteriores

//TODO explicar sinhole dependiendo del resto de subredes. Una subred puede comportarse como sinhole sobre otra subred, pero no sobre una tercera.

//TODO Sinkhole absoluta si da igual el resto de subredes

//TODO Caracterizacion de Sinhole subnet absoluta



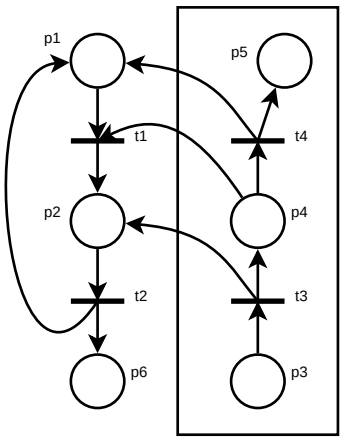
$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_6 \\ \hline p_3 \\ p_4 \\ p_5 \end{array} \left(\begin{array}{cc|cc} t_1 & t_2 & t_3 & t_4 \\ -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

FIGURE 3.3: Sinkhole subnet

3.2.3.5 Source subnet

If instead of this what happens is no arc gets into the subnet, we have a source subnet. In a source subnet we can not enter.

Definition 3.10 (Source subnet). It is said that a subnet is a source subnet if no arc has its destination in an internal node (place or transition) of the subnet.



$$C = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \\ \hline p_1 \\ p_2 \\ p_6 \end{array} \left(\begin{array}{cc|cc} t_3 & t_4 & t_1 & t_2 \\ -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

FIGURE 3.4: Source subnet

It is easy to see that a subnet is source if and only if all elements of TIM are greater than or equal to zero and all elements of PIM are less than or equal to zero.

$$N_1 \text{ is source} \iff \forall a_{ij} \in TIM_1, a_{ij} \geq 0 \wedge \forall a_{pq} \in PIM_1, a_{pq} \leq 0$$

3.2.4 Matrix parts description once defined the subnets

As places and transitions can be reordered smoothly, we study a network N divided into 2 subnets, for simplicity and without loss of generality.

For consistency with [4] we will follow this notation:

$$\left(\begin{array}{c|c} H & HP \\ \hline HT & V \end{array}\right)$$

where

- H (Hidden Subnet) is the subnet wanted to be hidden.
- V (Visible Subnet) is the subnet that is visible.
- HT (Hidden Transitions Submatrix) are the relationships between places of V and H transitions
- HP (Hidden Places Submatrix) are the relations between transitions of V and H sites

Note. Following this notation can be convenient because it is clear what is each of the submatrices. However, elsewhere in the document be referenced as N_1 and N_2 for be more clarifying or being something generic and independent networks concealment. However, using N_1 and N_2 the notation of subnets of influence with respect to the other is more diffuse.

Example 3.6. *Consider the Petri net of the figure 3.5 with the next incidence matrix:*

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cccc} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

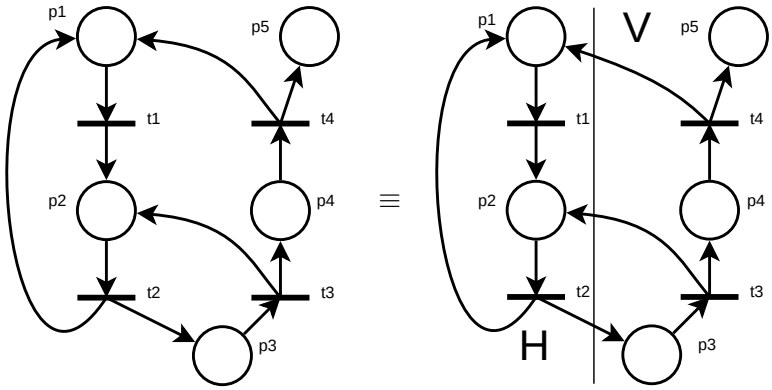


FIGURE 3.5: Selecting subnet to hide

The subnet we want to hide is formed by sites 1 and 2 and 1 and 2 transitions. Graphically, separate places and transitions to hide (H) from the rest of the network (V)

The incidence matrix is already sorted by the places and transitions to the top of it. Here are the four parts described above.

$$\begin{array}{c|cc|cc} & t_1 & t_2 & t_3 & t_4 \\ p_1 & -1 & 1 & 0 & 1 \\ p_2 & 1 & -1 & 1 & 0 \\ \hline p_3 & 0 & 1 & -1 & 0 \\ p_4 & 0 & 0 & 1 & -1 \\ p_5 & 0 & 0 & 0 & 1 \end{array}$$

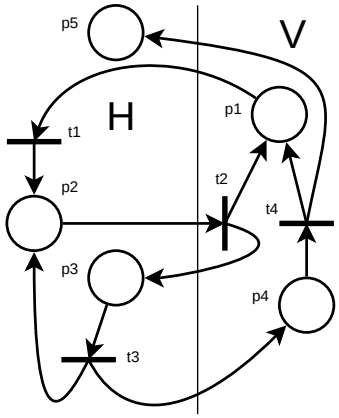
In this matrix we can see:

- $H = \begin{array}{c} \\ p_1 \\ p_2 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \end{array}$ *is the subnet we want to hide. It is the same as* SN_1
- $V = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc} t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \end{array}$ *is the subnet that is visible. It is the same as* SN_2
- $HP = \begin{array}{c} \\ p_1 \\ p_2 \end{array} \begin{array}{c} \begin{array}{cc} t_3 & t_4 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$ *are the relationships between transitions of V and H places. It is the equivalent to* PIM_1 *or* TIM_2

- $HT = \begin{array}{c} \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{array}$ *are the relationships between places of V and H transitions. It is the equivalent to* TIM_1 *or* PIM_2

At this moment, it is easy to see that we can choose any subset of places and transitions in order to hide it, simply reordering rows and columns.

Example 3.7. *In the previous example we have seen a fairly simple option selection subnet and we have chosen the locations 1 and 2 and the transitions 1 and 2. However, we can choose any other subset of places and transitions. In this example we will select locations 2, 3 and 5 and the transitions 1 and 3. Thus, in the graph of the previous example move the locations and transitions to hide on one side and the rest on the other.*



Although more confusing, can be seen that the graph is the same as the incidence matrix is the same (not just part of the equivalence class, it is exactly the same). Now, in this matrix move places 2, 3 and 5, and 1 and 3 transitions at the beginning of the matrix:

$$\begin{array}{c} \\ p_2 \\ p_3 \\ p_5 \\ \hline p_1 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cc|cc} t_1 & t_3 & t_2 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline -1 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 \end{array}\right) \end{array}$$

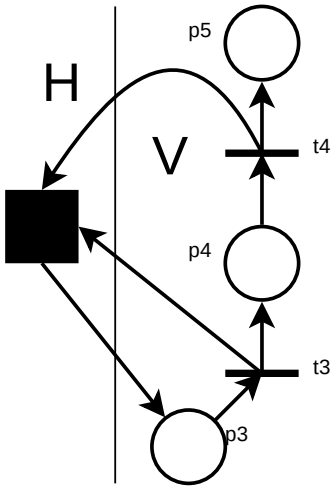
Interpreting each of the chunks of the matrix is similar to the previous example.

3.2.5 Private information. Hiding a subnet

Once you select the private subnet we proceed to the occultation as such [4]. Graphically, it seems simple. Just replace the subnet to hide by a black box and modify some arcs according to the following rules:

1. The arcs originating in a place or transition within the black box, and target a place or transition out of it will have the black box as the source.
2. The arcs originating in a place or transition out of the black box, and target a place or transition within it, are replaced by the black box as a destination.

Example 3.8. *We consider the Petri net of the Figure 3.5. The result of hiding the part of the graph H is the following:*



In the associated incidence matrix also replace the H elements by a black box:

$$\begin{array}{c|cc|cc} & t_1 & t_2 & t_3 & t_4 \\ p_1 & \blacksquare & \blacksquare & 0 & 1 \\ p_2 & \blacksquare & \blacksquare & 1 & 0 \\ \hline p_3 & 0 & 1 & -1 & 0 \\ p_4 & 0 & 0 & 1 & -1 \\ p_5 & 0 & 0 & 0 & 1 \end{array}$$

However, in this matrix notation is given information should also be hidden: it gives us information about the number of places and transitions of the hidden subnet, besides indicating hidden places and transitions with which it interacts. To solve this problem we proceed as follows. We can group all rows for the screened subnet into one. In each

row position examine all elements of the original rows corresponding to that position, and will put:

- If all these elements are zero, in the grouped row will be a zero.
- If one and only one of those elements is nonzero, will put that item.
- If there are several non-zero elements, we will post a list of these items separated by commas, creating a d-dimensional element (in d dimensions).

In the same way we have done with the rows, proceed with columns. Thus, if the hidden subnet has i columns and j rows, we will get a matrix like this:

$$\left(\begin{array}{c|ccc} \blacksquare & a_{1(i+1)} & \cdots & a_{1m} \\ \hline a_{(j+1)1} & & & \\ \vdots & & V & \\ a_{n1} & & & \end{array}\right)$$

where $\forall p, \forall q | i+1 \leq p \leq m \wedge j+1 \leq q \leq n$

$$a_{1p} = \begin{cases} 0 & \text{if } \forall r | 1 \leq r \leq j, c_{rp} = 0 \\ c_{rp} & \text{if } \exists ! r, 1 \leq r \leq j | c_{rp} \neq 0 \\ (c_{r_1 p}, c_{r_2 p}, ...) & \text{if } \exists r_1 \neq r_2 \neq ..., 1 \leq r_1, r_2, ... \leq j \\ & | c_{r_1 p}, c_{r_2 p}, ... \neq 0 \end{cases}$$

$$a_{q1} = \begin{cases} 0 & \text{if } \forall s | 1 \leq s \leq i, c_{qs} = 0 \\ c_{qs} & \text{if } \exists ! s, 1 \leq s \leq i | c_{qs} \neq 0 \\ (c_{qs_1}, c_{qs_2}, ...) & \text{if } \exists s_1 \neq s_2 \neq ..., 1 \leq s_1, s_2, ... \leq s \\ & | c_{qs_1}, c_{qs_2}, ... \neq 0 \end{cases}$$

So we hide the number of places and transitions of the hidden subnet and their relationships. Yes, some information is given about the hidden network. Really if this resulting matrix some node that is d-dimensional, at least in the hidden network must exist d nodes of this type.

Example 3.9. *We consider the Petri net defined by the following incidence matrix, separated into H,V,HT and HP.*

$$\left(\begin{array}{cccccc} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right)$$

After applying the above steps for the group, we would have the following:

$$\left(\begin{array}{c|ccc} \blacksquare & (1,-1,1) & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ (1,-1) & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

Here we see that the information about the number of hidden places and transitions is minimized. So we know that at least there is a hidden transition and at least three hidden places (there is a transition of dimension 3). However, we do not know the exact number of either.

//TODO estudiar mas en profundidad esta matriz. Puede haber bastante a anadir en cuestion de interfaz. La fila y columna ocultas indican puertas de entrada/salida a la subred.

3.2.6 Front-end interaction with the subnet. Input and output functions

Once you have defined all this environment, we will try to go a little further. Let's assume that we want to export a subnet we have hidden in another network, like a black box. Our intention is to connect this hidden network to another network, and can thus be reused subnets. For example, let's assume that we have a process modeling with Petri net modeling and in this there is a subnet we want to hide, but, at the same time, we want to reuse it in other Petri nets.

In this case we have a problem, and once hidden network disappears half the information input or output arcs of the same. In particular, we do not know the source nodes and arcs that leave the target nodes of the arcs that enter the network until no visible again. But if we want to reuse it on other networks, can not wait to make it visible. Should remain hidden, but should be able to connect to other networks.

We will try to solve this problem. This way we can reuse hidden networks like plug-in modules on other networks. However, we will not need the actual implementation of the source or destination nodes of the arcs that leave or enter the network, respectively. The solution is to define a facade or front-end input and output of the network. This front-end will contain the information needed to interact with the network hidden, but hide the specifics of implementation. To define this behavior going from some assumptions.

3.2.6.1 Previous definitions

Let $R = \langle P, T, \alpha, \beta \rangle$ be a Petri net and let $P = \{R_1, R_2\}$ be a partition of R.

Definition 3.11 (Input place). Let p_i a place of R_1. p_i is an input place of R_1 if it is the destination of an arc coming from a R_2 transition, ie,

$$p_i \text{ is an input place of } R_1 \text{ if } \exists t_j \in R_2 \ |c_{ij} > 0$$

Definition 3.12 (Input transition). Let t_i a transition of R_1. t_i is an input transition of R_1 if it is the destination of an arc coming from a R_2 place, ie,

$$t_i \text{ is an input transition of } R_1 \text{ if } \exists p_j \in R_2 | c_{ji} < 0$$

Definition 3.13 (Input node). An input node of R_1 is an input place or transition of R_1.

Definition 3.14 (Output place). Let p_i be a place of R_1. p_i is an output place of R_1 if an arc leaves it towards a transition of R_2, ie,

$$p_i \text{is an output place of } R_1 \text{ if } \exists t_j \in R_2 | c_{ij} < 0$$

Definition 3.15 (Output transition). let t_i be a transition of R_1. t_i is an output transition of R_1 if an arc leaves it towards a place of R_2, ie,

$$t_i \text{ is an output transition of } R_1 \text{ if } \exists p_j \in R_2 | c_{ji} > 0$$

Definition 3.16 (Output node). An output node of R_1 is an output place or transition of R_1.

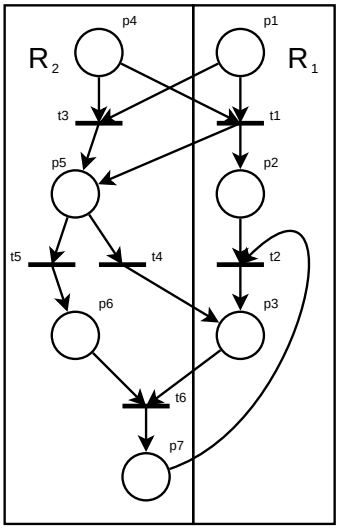
After defining these concepts, we can define the sets thereof.

Notation. We denote the sets of the elements defined above:

- Let $IP(R) \subseteq \overline{P}$ (Input Places) be the set of input places of a subnet.
- Let $IT(R) \subseteq \overline{T}$ (Input Transitions) be the set of input transitions of a subnet.
- Let $IN(R) \subseteq \overline{P} \cup \overline{T}$ (Input Nodes) be the set of input nodes of a subnet.
- Let $OP(R) \subseteq \overline{P}$ (Output Places) the set of output places of a subnet.
- Let $OT(R) \subseteq \overline{T}$ (Output Transitions) be the set of output transitions of a subnet.
- Let $ON(R) \subseteq \overline{P} \cup \overline{T}$ (Output Nodes) be the set of output nodes of a subnet.

Note. Recall that a node in a Petri net can be both a place and a transition, depending on the context.

Notation. Denote as n_i to a node of a Petri net.



$$C = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ \hline p_4 \\ p_5 \\ p_6 \\ p_7 \end{array} \begin{array}{c} \begin{array}{cc|cccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\ \left(\begin{array}{cc|cccc} -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ \hline -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{array} \right) \end{array}$$

FIGURE 3.6: Subnets with input and output nodes

As we have generic definitions, no problem in applying to a network divided into H, V, HN and HT, as the set $\{H, V\}$ is a partition of R.

3.2.6.2 Subnet Front-end

Once all these concepts, we create the front-end input/output of a Petri subnet. A front-end of the Petri net will be a intermediate facade that allows us to physically divide that subnet from the rest of the net. Thus, in order to enter or leave the subnet, you need to make it through this front-end.

Let IA (input arcs) the set of arcs that enter the subnet R_1 and let OA (output arcs) the set of arcs leaving R_1.

Definition 3.17 (Input gate of a net). Let $a_i \in IA$ an arc of entrance to R_1. We define an input gate to R_1, and denote by ig_i, as a new logical node that is identified with an arc of entrance to the net. For each input arc, defines an input gate, regardless of the origin and destination of the arc. If the source is a transition, we denote igt_i and if a place, igp_i.

Definition 3.18 (Output gate of a net). Let a_i $inOA$ output arc R_1. We define an output gate of R_1, and denote by og_i, as a new logical node that is identified with an exit arc of the net. For each exit arc is defined an output gate, regardless of the origin and destination of the arc. If the source is a transition, we denote ogt_i and if it is a place, ogp_g.

In this way we can divide the input arcs and output into two parts: a R_1 internal and external to R_1. If we take an arc of entrance a_i that has an origin in n_j and destination in n_k, we define an input gate through a point of entry so that the original arc a_i is divided into two parts.

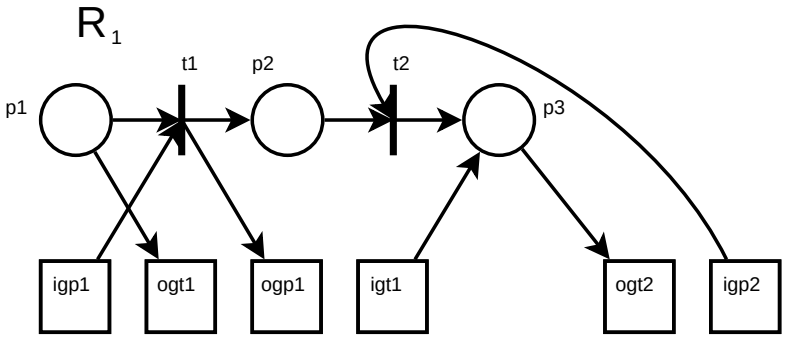
- a_{i1} (external to R_1) with origin in n_j and destination in igt_i or igp_i depending on if n_j is a transition or a place.
- a_{i2} (internal to R_1) with destination in igt_i or igp_i depending on if n_j is a transition or a place respectively.

Similarly, if we take a exit arc a_i that has an origin in n_k and destination in n_j, we define an output gate og_i so that the original arc a_i is divided into two parts:

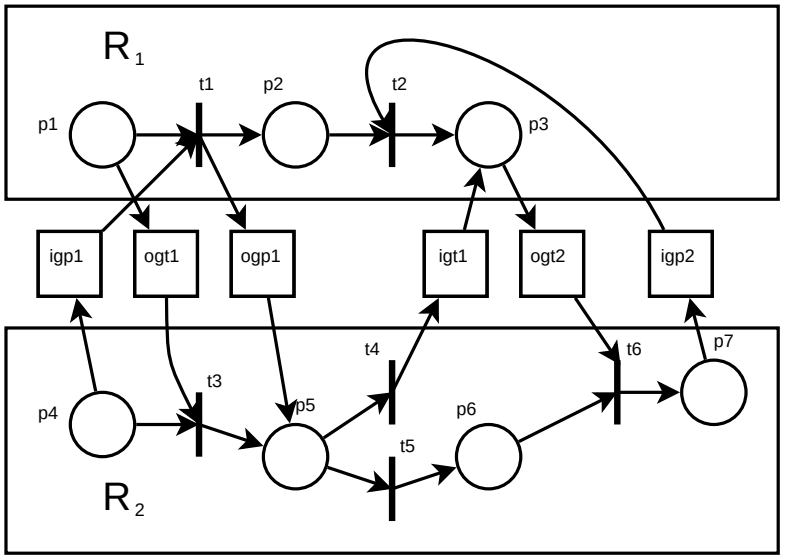
- a_{i1} (internal to R_1) with origin in n_k and destination in igt_i or igt_i depending on if n_j is a transition or a place .
- a_{i2} (external to R_1) with destination in n_j and origin in igt_i or igp_i depending on if n_j is a transition or a place respectively.

Example 3.10. *Consider the net in figure 3.6. In this network we have three arcs entering and leaving three arcs. For each of those emerging define output gates and each coming,*

we define input gates. The subnet R_1 becomes:



and in the complete net, arcs entering and leaving are divided into two pieces:



Definition 3.19 (Input Front-end of a net). The input front-end (or input interface) of a subnet R_1 is the set of all input gates of R_1. We denote by IF of R_1.

Definition 3.20 (Output Front-end of a net). The output front-end (or output interface)of a subnet R_1 is the set of all output gates of R_1. We denote by OF of R_1.

Definition 3.21 (Front-end of a net). The front-end (or interface) of a net R_1 is the pair of IF and OF of R_1. We denote by F of R_1.

$$F = \langle IF, OF \rangle$$

Example 3.11. *Taking the net of the example 3.10 and applying these new definitions, we would have R_1 net along with its front end as shown in Figure 3.7.*

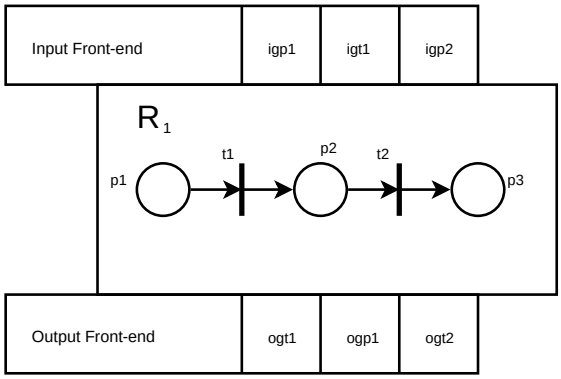


FIGURE 3.7: Front-end of a net

3.2.6.3 Input/output functions

Once all these input and output concepts defined, we will introduce a few key concepts for our purpose.

Let R a Petri net and let $\{R_1, R_2\}$ a partition of R. Let $F = \langle IF, OF \rangle$ the front-end of R_1.

Definition 3.22 (Petri net Input function). We define the input function f_i of R_1 as:

$$f_i : F \longrightarrow IN$$

such that for each input gate igt_i you mapped one or no input place R_1 and each input gate igp_j you mapped one or no input transition R_1.

Definition 3.23 (Petri net Output function). We define the output function f_o of R_1 as:

$$f_o : ON \longrightarrow F$$

such that each output place R_1 you mapped one or no output gate ogt_i of R_1 and each output transition R_1 you mapped one or no output gate ogp_j to R_1

The input function can be defined for all the input gates and the output function should be surjective because if not, some door would not be connected. Anyway that is not essential. If a front-end door is not connected with any element of your network, simply by solving the final network, the arcs connected to that door disappear. Note also that the input function is not necessarily injective: Multiple input gates can be associated to the same node of R_1.

Example 3.12. *Consider the net* R_1 *in figure 3.6 with its front-end in figure 3.7. The input and output functions are:*

- *Input function:*

F	igp_1	igt_1	igp_2
IN	t_1	p_3	t_2

- *Output function:*

ON	p_1	t_1	p_3
F	ogt_1	ogp_1	ogt_2

//TODO peso de los arcos y colores

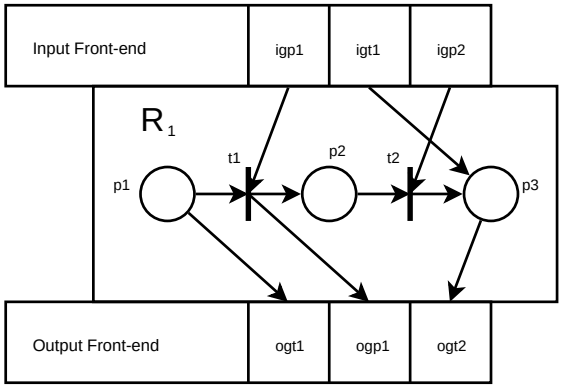
3.2.6.4 Attachable net

By joining the subnet R_1 along with its front-end and its input and output functions f_i and f_o we grouped both the internal network with external communication. This way we can " extract" a subnet and "implant" it in another net. You only need this destination network is to communicate with the front-end. So naturally appears the following definition.

Definition 3.24 (Attachable Petri net). An Attachable Petri net is a quadruple $R_a = \langle R, F, f_i, f_o \rangle$

From these definitions, it is clear that you can create attachable subnets taking a subnet of another given and applying the whole process we have defined. But it is also possible to create from scratch, starting from a network, defining a front end for that network and declaring the input and output functions. So you can create Petri nets modules providing functionality and out through a front-end without requiring the actual implementation.

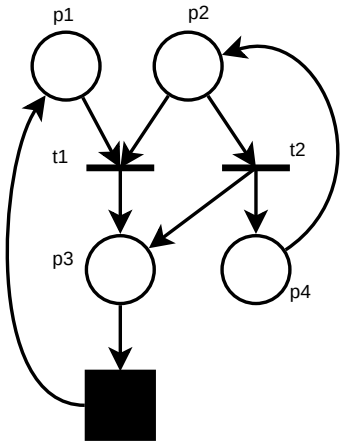
Example 3.13. *The attachable net in figure 3.6 would be the next:*



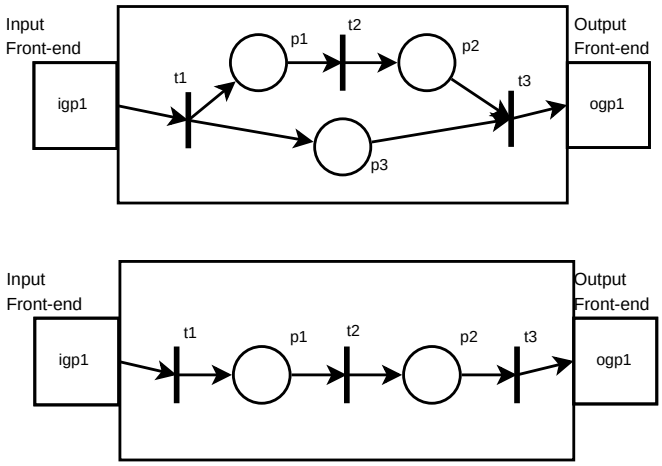
It can be seen as a private black box with visible input and output connectors that are "plugged" to other networks. In a attachable net, the private part would be R, f_i and f_o. The public part of the front-end would be F. All a net need to know is the input/output front-end.

A utility of these nets is that its definition is simple, since only the front-end is needed to define its operation. This makes possible to create nets using attachable nets in certain areas where they do not know their actual implementation, but its behavior. Additionally, it is possible to use different implementations of "network providers" of the same attachable nets, using at each moment the most appropriate one.

Example 3.14. *Consider now the following Petri net*



to which we want to connect an attachable net in the black box. Let's assume we have two equivalent alternatives described in Example 3.15:



We Can "plug" either because their front ends are equivalent and remains in figure 3.8.

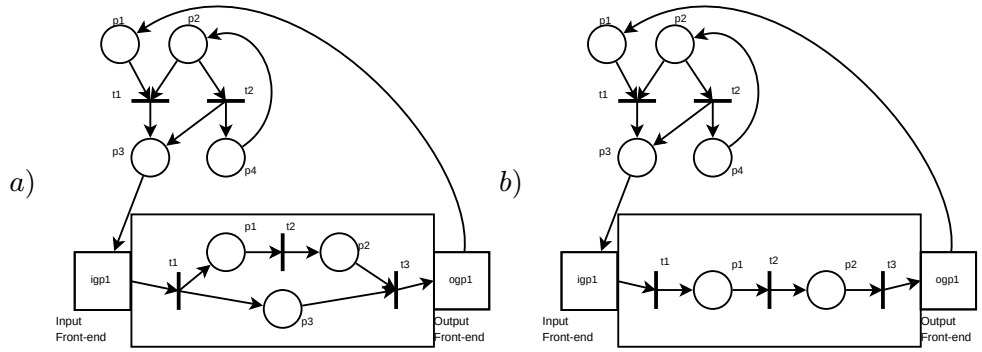


FIGURE 3.8: Two different implementations of attachable nets

In this case, the behavior of the net will be the same, but does not have to be. That will decide who connects nets. For example, you could create a" silly" net that does nothing at first and replace it later by the real one.

3.3 Hiding vs. Reduction

Both Silva works [7, 8] as in the article by Xia [5] discusses possible Petri nets reductions for grouping and simplifying, under certain circumstances, places and / or transitions. These reductions can be structural (only dependent on the structure and initial marking of the net) or depending on the interpretation of the Petri net.

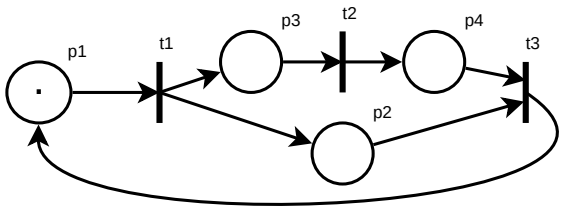
Should be clear that these reductions are not the same thing we are describing. We do not try to simplify the network together elements to have more or fewer places or transitions or to make it easier. What we want is to hide part of the network, regardless of how simple or complicated it is.

Here we have an example of what a reduction is.

Example 3.15 (Reduction of an implicit place [7]). *In a marked Petri net, an implicit place is one that meets the following:*

1. *its marking can be calculated from other points marking*
2. *never is the only place that prevents the enabling of its output transitions*

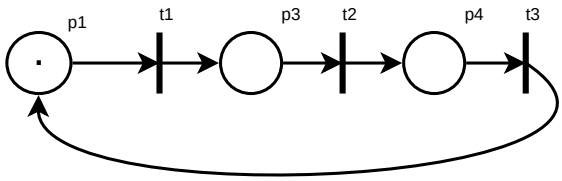
If we consider the following Petri net



we can notice that p_2 is an implicit place because its marking can be calculated as a function of p_3 y p_4:

$$M(p_2) = M(p_3) + M(p_4)$$

Moreover, by this same formula, it is clear that $M(p_2) \geq M(P_4)$ (marking cannot be negative) so the only place that can prevent enabling of T_3 is P_4. Thus eliminating p_2 does not alter the behavior of the network, which would be as follows:



In this network elements have been removed, no hidden. This example helps us to see the difference between hiding and a reduction.

3.4 Conclusions

As we have seen, any Petri net can be divided into any number of subnets, only limited by the number of places and transitions. Furthermore, each one of this Petri subnets has its own input and output front-ends in order to connect with the rest of the Petri net. Of course, the main application of this definitions in this thesis is that the private information of this Petri net is stored in one or more of these defined subnets.

So I have reached the first milestone: Extend Petri Nets definition in order to define the public and the private information.

Chapter 4

Petri net representation for subnets and hiding support. PNML

4.1 Introduction

Once explained how to split a net in several subnets, the next step is to define a way to hide one or more of those subnets.

There is not literature about this topic. Because of that I have to do a previous work about Petri net representation. First of all a way to represent subnet must be defined. Depending on the selected representation, the way to occult subnets may be different or even impossible.

4.2 Petri net representations

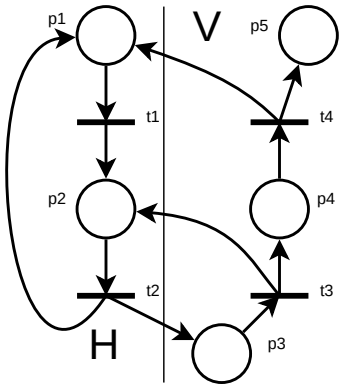
There are four standard ways to represent Petri nets. Each one of them have their properties, advantages and disadvantages. But I want to select one that I am able to represent any kind of Petri net, its subnets and allow to hide information without erasing it.

4.2.1 Graphic representation

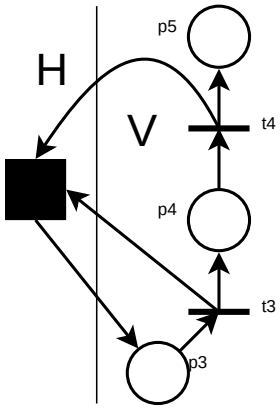
This is the clearest and extended way to represent Petri nets. It has a very important advantage and it is that a picture is worth a thousand words.

Subnets can be defined simply drawing a vertical line. The right part is one subnet and the left part is other subnet. Places and transitions can be moved from one location to another depending on the subnet they are situated. This is only an example. Other way would be to use colors for the nodes (same color indicates same subnet) or use rectangles, etc.

Example 4.1 (Graphic representation of a hidden subnet). *Let's take the following Petri net.*



I want to occult the part marked with an H, so the result is this graphic



And now, how can I maintain the H subnet information on the black box? If I want to distribute this Petri net I should send something more to that people I want to access the hidden subnet. I have not found a way to embed that information in the graphic so that some people can access it but other people can't.

So this representation is useful in order to show at one sight the Petri net structure, but I can't choose it for my goals. I have not been able to discover a way to show some people the hidden information (the hidden subnet). However I will continue using it where a clear idea of the Petri structure if necessary.

4.2.2 Matrix representation

This representation is very useful to study properties and evolution of a Petri net, independently of its graphic representation. As we have seen before (in chapter 3), we can reorder rows and columns and define subnets in the matrix.

Example 4.2 (Matrix representation of a hidden subnet). *This is the matrix representation of the Petri net in the previous example 4.1.*

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

As we can see before, If I occult the part marked with an H, so the result is:

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \left(\begin{array}{cc|cc} \blacksquare & \blacksquare & 0 & 1 \\ \blacksquare & \blacksquare & 1 & 0 \\ \hline 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

or this other one, grouping the places and transitions of the hidden subnet as seen in chapter 3

$$\begin{array}{c} \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{cccc} & & t_3 & t_4 \end{array} \\ \left(\begin{array}{c|cc} \blacksquare & 1 & 1 \\ \hline 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{array}\right) \end{array}$$

And now I have the same problems as in graphic mode: how can I keep information in the black box?

With this representation it is possible to study properties and it may be really important as a complement to graphic mode. With both representations together, everyone has a clear idea of the Petri net structure and properties. But I haven't found a way to store information inside the black box.

4.2.3 Equation representation

The third representation way for Petri nets is the equation representation. Basically, transitions are selected and, for each one, the tokens of the places connected to that transition are modified. This is very useful to compute the evolution of a Petri net, choosing the transition fired.

However it is difficult to find a way for representing subnets with this notation. And, of course, if a subnet cannot be represented, it cannot be hidden.

Example 4.3 (Equation representation of a hidden subnet). *Let's take again the Petri net from the previous examples 4.1. This is It's equation representation:*

```
if (p1>0) then
        p1 <- p1 - 1
        p2 <- p2 + 1
if (p2>0) then
        p2 <- p2 - 1
        p1 <- p1 + 1
        p3 <- p3 + 1
if (p3>0) then
        p3 <- p3 - 1
        p2 <- p2 + 1
        p4 <- p4 + 1
if (p4>0) then
        p4 <- p4 - 1
        p1 <- p1 + 1
        p5 <- p5 + 1
```

As we can see, we should hide several lines. In particular all that includes places p_1 and p_2, resulting something like this (strikethrough text would be part of the subnet):

~~if (p1>0) then~~
~~p1 <- p1 - 1~~
~~p2 <- p2 + 1~~
~~if (p2>0) then~~
~~p2 <- p2 - 1~~
~~p1 <- p1 + 1~~
p3 <- p3 + 1
if (p3>0) then
p3 <- p3 - 1
~~p2 <- p2 + 1~~
p4 <- p4 + 1
if (p4>0) then
p4 <- p4 - 1
~~p1 <- p1 + 1~~
p5 <- p5 + 1

This is probably the strangest way to represent Petri nets and it is difficult to define subnets over it.

I have tried to think about these ways of representation but, in my opinion, no one of them is suitable enough to represent subnets in a clear way than can be occulted. Because of that I have chosen the fourth representation, which is PNML, and that is explained in the next section.

4.3 PNML. Petri Net Marked Language

Originally, with basic Petri nets, the structure of a Petri net was fully provided. The only thing that is not supported in comparison with graphic mode is the graphical appearance of Petri nets: the position of nodes and transitions was not important, but with the arrival of High level Petri nets and Petri nets design software, is necessary to store this kind of information.

After several years of study, in 2004 the ISO/IEC 15909-1 [32] appeared to define conceptually and mathematically a xml representation of Petri nets: PNML[33].

So all the Petri nets that I am able to draw can be stored in PNML. Because of that, it is one of the most supported formats in almost every program that draw Petri nets.

PNML [34] is an implementation defined by the standard ISO/IEC 15909-2:2014 [35]. The goal of this ISO standard is to define a transfer format of Place/Transition nets, High-level Petri nets and Symmetric nets. However, it is designed in such way that it can be easily extended, so that other versions of Petri nets can be supported later. There is a way to define these extensions in a graphical way, using eclipse as intermediate [36, 37], but this is not the main goal of my work. My intention is to describe this extension in a theoretical way. Once described, it could be implemented with that kind of tools.

In this work, I am going to study only basic Petri nets, but that concept and the method is easily exportable to other kind of nets, such as Symmetric nets and High Level Petri nets (that are representable in PNML format too.)

4.3.1 Scope

The scope of this work in this section is basically bounded by the original and basic Petri nets, that is:

- there are places, transitions and arcs between them

- places can have tokens on them (but this does not have influence in the hidding process)
- transitions can be fired, but there is only one type of transition (not messages, not time, etc.)

Furthermore, I am going to explain several concepts for a specific graphical design of the Petri net (that will not influence the process either).

There are other options such as specific information for the design tool that I am not going to discuss because the tools have no interest in this work

4.3.2 Description

Petri Net Marked Language is an xml language created to represent Petri Nets. With this language we can take a Petri Net and store it into an xml file without loss of information.

One of the best properties of PNML is that, as it is an xml based schema, it can be extended with more functionality extending the grammar. Virtually, any extension over Petri nets can be translated into PNML in a logical and natural way. Moreover, this extension is defined by Petri net type definition [35, 38].

In this case PNML hasn't got a way to represent subnets. There is something named <page> that is used to represent several nets in the same PNML file. But, by default, a node inside a page cannot connect with a node of other page. So it cannot be used as "subnets". So I am going to extend the language in order to get several goals:

1. Represent subnets of a Petri Net.
2. Include input and output interfaces for every subnet.

As we can think, definition of several subnets of a Petri net is possible and the connection over them are always through their respective interfaces.

4.3.3 PNML grammar

As PNML is an xml based language it has to be described by an schema that define the creation rules of the PNML representation of a Petri net.

The grammar is defined since 2009 and updated until 2012, which is the most recent revision. I am not going to do and extensive explanation of all the possibilities of the

grammar, but the most important. As we can see later, anything we think is useful can be added to the process with little effort. So I am going to study only the most basic elements of a Petri net. The rest of the element can be attached later with facility.

4.3.3.1 PNML basics

In this section I am going to explain several characteristics of PNML files. With these explanations it is going to be easier the understanding of PNML structure.

First of all, as PNML files are xml files, there several things to know:

1. A xml file normally starts with a line defining some characteristics of the file, like the version and the encoding type. It has an aspect like this[1]:

```
<?xml version="1.0" encoding="utf-8"?>
```

2. A root node must exist. In this case, the root node is <pnml>. So every PNML files has to start with the tag <pnml> and end with the tag </pnml>. Below this tag, there is a new tag <net> that can contain

   - Type: the type of the Petri net as an attribute. In this case, as I am going to study only Place/Transition nets, it will be ptnet
   - Name of the net: New tag <name><text>...</text></name>
   - Pages: one page is an invention to store several Petri nets inside an unique PNML file, but usually there is only one page for file. It is nested inside a <page> tag.

3. Each element in PNML has to have a unique id inside the net to be identified unambiguously. So there cannot be two elements with the same id.

With this three observations, we can have an idea about how a PNML file is structured:

```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      .......
    </page>
  </net>
</pnml>
```

[1]For clarity, in the following examples, this line can be deleted.

LISTING 4.1: Example of general PNML file

Once this structure is defined, I am going to explain the next stage, that is the most important one in this work.

4.3.3.2 Places, transitions and arcs in PNML

As Petri nets has three main elements, places, transitions and arcs, and PNML has them too. These three elements have several things in common in an PNML file:

- They are all nested in a page tag
- Places and transitions (not arcs) can contain a tag <name> with its name. This tag has been defined before for the name of the net. It can store information about the text of the name and the graphical position of this label in this way:

```
<name>
  <text> Element Name </text>
  <graphics>
    <offset x="22" y="-10"/>
  </graphics>
</name>
```

- They can contain a tag <graphics> with information about its position and dimension:

```
<graphics>
  <position x="100" y="200"/>
  <dimension x="40" y="40"/>
</graphics>
```

These are the common properties of places, transitions and arcs. Now let's go on the particular characteristics of each one of them.

Places are represented with the tag <place> and the can have a marking with the label <initialMarking>. Here we have an example of a place with two tokens in PNML:

```
<place id="p1">
  <name>
    <text> Place number one </text>
    <graphics>
      <offset x="130" y="130"/>
    </graphics>
  </name>
  <graphics>
```

```
    <position x="130" y="90"/>
    <dimension x="40" y="40"/>
  </graphics>
  <initialMarking>
    <text> 2 </text>
  </initialMarking>
</place>
```

Listing 4.2: PNML representation for places

Transitions are represented with the tag <`transition`>. This is an example of a transition in PNML:

```
<transition id="t1">
  <name>
    <text> Transition number one </text>
    <graphics>
      <offset x="270" y="140"/>
    </graphics>
  </name>
  <graphics>
    <position x="270" y="100"/>
    <dimension x="40" y="40"/>
  </graphics>
</transition>
```

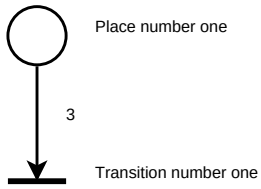
Listing 4.3: PNML representation for transitions

Arcs are represented with the tag <`arc`>. Arc have to have a source and a target, which are defined by the attributes `source` and `target` that have to point to a transition and a place, identified by their id. Furthermore, the arc weight can be fixed by the <inscription> tag. If the weight is one, the tag inscription is not necessary because this is the default value. This is an example of the arc with weight 3 that connects the place and the transition of the previous examples in PNML:

```
<arc id="a1" source="p1" target="t1">
  <inscription> 3 </inscription>
</arc>
```

Listing 4.4: PNML representation for arcs

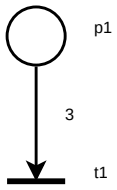
If we take the last examples all together, we can represent the following Petri net:




```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <name>
      <text> My new net </text>
    </name>
    <page id="page1">
      <place id="p1">
        <name>
          <text> Place number one </text>
          <graphics>
            <offset x="130" y="130"/>
          </graphics>
        </name>
        <graphics>
          <position x="130" y="90"/>
          <dimension x="40" y="40"/>
        </graphics>
        <initialMarking>
          <text> 2 </text>
        </initialMarking>
      </place>
      <transition id="t1">
        <name>
          <text> Transition number one </text>
          <graphics>
            <offset x="270" y="140"/>
          </graphics>
        </name>
        <graphics>
          <position x="270" y="100"/>
          <dimension x="40" y="40"/>
        </graphics>
      </transition>
      <arc id="a1" source="p1" target="t1">
        <inscription> 3 </inscription>
      </arc>
    </page>
  </net>
</pnml>
```

Listing 4.5: Complete PNML representation for a basic Petri net

For clarity, I am going to obviate several options. I am not going to put the graphic information and the names, but the id. So this last example is as follows:




```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
  <net id="myNet" type="http://www.pnml.org/version-2009/grammar/ptnet">
    <page id="page1">
      <place id="p1">
        <initialMarking>
          <text> 2 </text>
        </initialMarking>
      </place>
      <transition id="t1"/>
      <arc id="a1" source="p1" target="t1">
        <inscription> 3 </inscription>
      </arc>
    </page>
  </net>
</pnml>
```

Listing 4.6: Basic PNML representation for a basic Petri net

I think that this last listing is clear enough to understand the rest of the process. But not only that. For even more clarity, the tags <?xml>, <pnml>, <net> and <page> are going to be obviated too. So in many of the later examples they will not be present. Applying this criterium, the listing 4.6 change to:

```
<place id="p1">
  <initialMarking>
    <text> 2 </text>
 </initialMarking>
</place>
<transition id="t1"/>
<arc id="a1" source="p1" target="t1">
  <inscription> 3 </inscription>
</arc>
```

Listing 4.7: Simplified PNML representation for a basic Petri net

4.3.4 PNML examples

In this section I will present several examples of Petri nets represented with PNML. This examples are extracted directly form www.pnml.org. These examples will be processed later in order to hide parts of them.

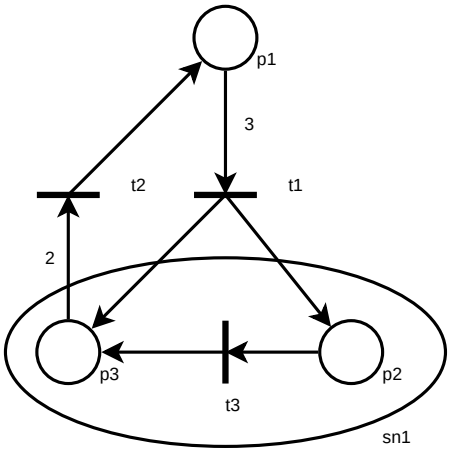
As first example, we have the Dining Philosophers...

//TODO terminar

4.3.5 PNML extension for representing subnets

In this section I am going to define new tags and structures in PNML. At this point, I have developed all the necessary to extend PNML in order to represent subnets.

Let's take the following simple Petri net that will serve us to explain the method to achieve a subnet representation and the PNML extension associated to it:



The PNML code for this net is:

```
<place id="p1"/>
<place id="p2"/>
<place id="p3"/>
<transition id="t1"/>
<transition id="t2"/>
<transition id="t3"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
```

```
  <inscription>
    <text>2</text>
  </inscription>
</arc>
<arc id="a5" source="t3" target="p3"/>
<arc id="a6" source="p2" target="t3"/>
<arc id="a7" source="t2" target="p1"/>
```

I want the ellipse region to be a subnet, so I have to specify a subnet with the elements inside the ellipse.

The first step is to define a new tag <**subnet**>. This tag will have an id, as the rest of PNML elements. And now we proceed in this way:

1. The places and transitions inside the subnet are moved into the <subnet> tag
2. The arcs joining subnet elements will be included inside the tag
3. The arcs entering or leaving the subnet will be copied inside the tag. This mean that there are arcs duplicated inside and outside the tag

If we apply these rules to the example:

1. $p2$, $p3$ and $t3$ are moved into the tag <**subnet**>
2. $a5$ and $a6$ are put inside the tag
3. $a2$, $a3$ and $a4$ are copied inside the tag

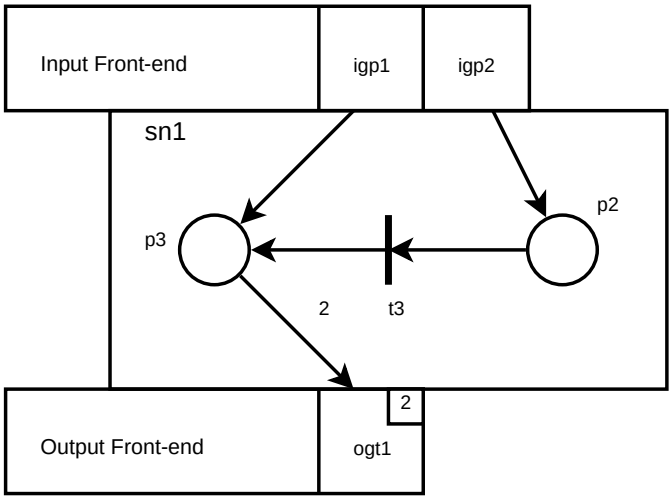
and we have this other PNML extended code:

```
<subnet id="sn1">
  <place id="p2"/>
  <place id="p3"/>
  <transition id="t3"/>
  <arc id="a2" source="t1" target="p2"/>
  <arc id="a3" source="t1" target="p3"/>
  <arc id="a4" source="p3" target="t2">
    <inscription>
      <text> 2 </text>
    </inscription>
  </arc>
  <arc id="a5" source="t3" target="p3"/>
  <arc id="a6" source="p2" target="t3"/>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
```

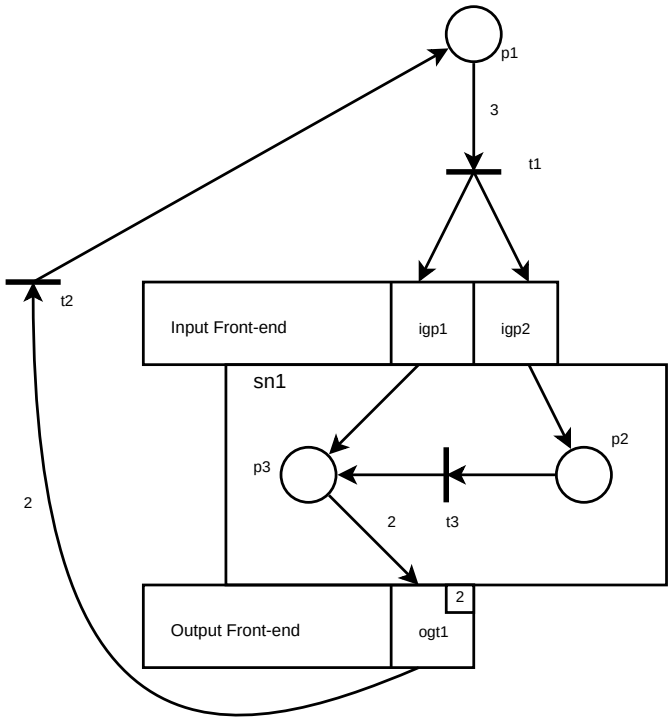
```
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

Now this is one of the most important moment in this work: I will separate the inside and the outside of the subnet completely. Taking advantage of the process described in chapter 3, I have to extract the front-end from this subnet.

In this case I have two igp (input gate to a place) and an ogt (output gate to a transition). This graphic illustrates the interface associated:



And this is the complete net including the subnet and its front-end



And now that I have the graphic, how can I represent it in PNML? To answer this question I will define four new tags: <`interface`>, <`gate`>, <inscription> and <`content`>. Let's explain them.

As its name says, <`interface`> is the tag name for encapsulate the front-end. This tag has no attributes (only the id, of course) but it has embedded the gates inside of it. This gates are represented by <`gate`>. This tag has two new attributes: `action` and `type`. These two attributes have information about the gates. The attribute `action` can take two different values: `input` and `output`. It indicates whether the gate is an input gate or an output gate. The other attribute, `type`, can take other two values: `place` and `transition`. As arcs have weight, gates have it too. For being in accordance, I define the tag <`inscription`> embedded in the tag <`gate`>. It has the weight of the arc associated.

There is one other tag <`content`> that probably at this moment seems useless, but it is necessary for the rest of the process, as we will see in the next chapter. So I am going to introduce it now. This tag is used to encapsulate the rest of the subnet outside the interface. That is, <`subnet`> has two children: <`interface`> and <`content`>, that have the input/output gates and the rest of the elements, respectively.

Applying this definition to the example, we will have this code

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
```

```
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="t1" target="p2"/>
    <arc id="a3" source="t1" target="p3"/>
    <arc id="a4" source="p3" target="t2">
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="p2"/>
<arc id="a3" source="t1" target="p3"/>
<arc id="a4" source="p3" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

At this moment I have to do only one thing more. The last step is to modify the arcs that are repeated inside and outside the net changing their source or target, depending on where is it:

- If the arc is **entering** the subnet
  - For the <`arc`> tag inside the tag <`subnet`>, the **source** attribute of the arc is changed by the **input** gate associated
  - For the <`arc`> tag outside the tag <`subnet`>, the **target** attribute of the arc is changed by the **output** gate associated
- If the arc is **leaving** the subnet

- For the <arc> tag inside the tag <subnet>, the **target** attribute of the arc is changed by the **input** gate associated
- For the <arc> tag outside the tag <subnet>, the **source** attribute of the arc is changed by the **output** gate associated

Applying again this rule to the example we have the definitive code for this Petri net:

```
<subnet id="sn1">
  <interface id="sn1-interface">
    <gate id="igp1" action="input" type="place"/>
    <gate id="igp2" action="input" type="place"/>
    <gate id="ogt1" action="output" type="transition">
      <inscription>
        <text> 2 </text>
      </inscription>
    </gate>
  </interface>
  <content id="sn1-content">
    <place id="p2"/>
    <place id="p3"/>
    <transition id="t3"/>
    <arc id="a2" source="igp2" target="p2"/>
    <arc id="a3" source="igp1" target="p3"/>
    <arc id="a4" source="p3" target="ogt1">
      <inscription>
        <text> 2 </text>
      </inscription>
    </arc>
    <arc id="a5" source="t3" target="p3"/>
    <arc id="a6" source="p2" target="t3"/>
  </content>
</subnet>
<place id="p1"/>
<transition id="t1"/>
<transition id="t2"/>
<arc id="a1" source="p1" target="t1">
  <inscription>
    <text> 3 </text>
  </inscription>
</arc>
<arc id="a2" source="t1" target="igp2"/>
<arc id="a3" source="t1" target="igp1"/>
<arc id="a4" source="ogt1" target="t2">
  <inscription>
    <text> 2 </text>
  </inscription>
</arc>
<arc id="a7" source="t2" target="p1"/>
```

Once this is done, the only way to enter or leave the subnet is crossing the front-end.

4.4 Conclusions

Chapter 5

XMLEncryption

5.1 Introduction

I have to do this in order to develop a standard way to hide information from indiscrete eyes. But this information should be accessible to authorized people without necessity of supplying any other kind of data.

5.2 XMLEncryption revision

5.3 XMLEncryption with Extended PNML

5.4 Examples

Chapter 6

Conclusions

Intentar que sea un resumen de todo lo tratado, haciendo hincapie en lo aportado y en futuras lineas de investigacion. Que no sea corta. Lo suficientemente larga para que alguien que se lea la tesis pueda recordarla completamente

Throughout this paper we have presented Petri nets with definitions and basic properties. From this initial presentation, have been building a series of elements as a basis for further investigation. We defined subnets, subnets classifications have been studied, we have defined front-ends (interfaces) for those subnets, etc.. From this point is possible a further study of these subnets (their properties, utilities,....)

The contribution of this research is to establish the basis for the methodological study of hiding parts of Petri nets. We present the definitions and some basic properties and from here is to complete it by preparing a doctoral thesis.

Chapter 7

Ejemplos Latex

7.1 Codigo XML

Para poner codigo xml dentro del documento

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="points">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="point">
          <xs:complexType>
            <xs:attribute name="x" type="xs:unsignedShort" use="required" />
            <xs:attribute name="y" type="xs:unsignedShort" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

7.2 varias columnas

Para poner texto en varias columnas

Datos en dos columnas fndio fpeowu fpieuwpu piud pqeudp oquewp duqwipud pqwudpqwiodu pqwidu pqwidu pqwiud piqwud pqwu dpqwoud pqwod qpwodu qwpdu qwpd upou pqowdu wpqo po uqwpod u

Appendix A

PNML grammar

The official PNML grammar is available in www.pnml.org. It is written in RELAX NG format. This are the main parts that I use in this work.

A.1 RELAX NG implementation of PNML Core Model

```
<?xml version="1.0" encoding="UTF-8"?>

<grammar ns="http://www.pnml.org/version-2009/grammar/pnml"
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

  <a:documentation>
    Petri Net Markup Language (PNML) schema.
    RELAX NG implementation of PNML Core Model.

    File name: pnmlcoremodel.rng
    Version: 2009
    (c) 2001-2009
    Michael Weber,
    Ekkart Kindler,
    Christian Stehno,
    Lom Hillah (AFNOR)
    Revision:
    July 2008 - L.H
  </a:documentation>

  <include href="http://www.pnml.org/version-2009/grammar/anyElement.rng"/>

  <start>
    <ref name="pnml.element"/>
  </start>

  <define name="pnml.element">
```

```
  <element name="pnml">
    <a:documentation>
      A PNML document consists of one or more Petri nets.
      It has a version.
    </a:documentation>
    <oneOrMore>
      <ref name="pnml.content"/>
    </oneOrMore>
  </element>
</define>

<define name="pnml.content">
  <ref name="net.element"/>
</define>

<define name="net.element">
  <element name="net">
    <a:documentation>
      A net has a unique identifier (id) and refers to
      its Petri Net Type Definition (PNTD) (type).
    </a:documentation>
    <ref name="identifier.content"/>
    <ref name="nettype.uri"/>
    <a:documentation>
      The sub-elements of a net may occur in any order.
      A net consists of at least a top-level page which
      may contain several objects. A net may have a name,
      other labels (net.labels) and tool specific information in any order.
    </a:documentation>
    <interleave>
      <optional>
        <ref name="Name"/>
      </optional>
      <ref name="net.labels"/>
      <oneOrMore>
        <ref name="page.content"/>
      </oneOrMore>
      <zeroOrMore>
        <ref name="toolspecific.element"/>
      </zeroOrMore>
    </interleave>
  </element>
</define>

<define name="identifier.content">
  <a:documentation>
    Identifier (id) declaration shared by all objects in any PNML model.
  </a:documentation>
  <attribute name="id">
    <data type="ID"/>
  </attribute>
</define>

<define name="nettype.uri">
  <a:documentation>
```

```
      The net type (nettype.uri) of a net should be redefined in the grammar
      for a new Petri net Type.
      An example of such a definition is in ptnet.pntd, the grammar
      for P/T Nets. The following value is a default.
    </a:documentation>
    <attribute name="type">
      <value>http://www.pnml.org/version-2009/grammar/pnmlcoremodel</value>
    </attribute>
  </define>

  <define name="net.labels">
    <a:documentation>
      A net may have unspecified many labels. This pattern should be used
      within a PNTD to define the net labels.
    </a:documentation>
    <empty/>
  </define>

  <define name="basicobject.content">
    <a:documentation>
      Basic contents for any object of a PNML model.
    </a:documentation>
    <interleave>
      <optional>
        <ref name="Name"/>
      </optional>
      <zeroOrMore>
        <ref name="toolspecific.element"/>
      </zeroOrMore>
    </interleave>
  </define>


  <define name="page.content">
    <a:documentation>
      A page has an id. It may have a name and tool specific information.
      It may also have graphical information. It can also have many arbitrary labels.
      Note: according to this definition, a page may contain other pages.
      All these sub-elements may occur in any order.
    </a:documentation>
    <element name="page">
      <ref name="identifier.content"/>
      <interleave>
        <ref name="basicobject.content"/>
        <ref name="page.labels"/>
        <zeroOrMore>
          <ref name="netobject.content"/>
        </zeroOrMore>
        <optional>
          <element name="graphics">
            <ref name="pagegraphics.content"/>
          </element>
        </optional>
      </interleave>
    </element>
```

```
</define>

<define name="netobject.content">
  <a:documentation>
    A net object is either a page, a node or an arc.
    A node is a place or a transition, a reference place of
    a reference transition.
  </a:documentation>
  <choice>
    <ref name="page.content"/>
    <ref name="place.content"/>
    <ref name="transition.content"/>
    <ref name="refplace.content"/>
    <ref name="reftrans.content"/>
    <ref name="arc.content"/>
  </choice>
</define>

<define name="page.labels">
  <a:documentation>
    A page may have unspecified many labels. This pattern should be used
    within a PNTD to define new labels for the page concept.
  </a:documentation>
  <empty/>
</define>

<define name="place.content">
  <a:documentation>
    A place may have several labels (place.labels) and the same content
    as a node.
  </a:documentation>
  <element name="place">
    <ref name="identifier.content"/>
    <interleave>
      <ref name="basicobject.content"/>
      <ref name="place.labels"/>
      <ref name="node.content"/>
    </interleave>
  </element>
</define>

<define name="place.labels">
  <a:documentation>
    A place may have arbitrary many labels. This pattern should be used
    within a PNTD to define the place labels.
  </a:documentation>
  <empty/>
</define>

<define name="transition.content">
  <a:documentation>
    A transition may have several labels (transition.labels) and the same
    content as a node.
  </a:documentation>
  <element name="transition">
```

```
      <ref name="identifier.content"/>
      <interleave>
        <ref name="basicobject.content"/>
        <ref name="transition.labels"/>
        <ref name="node.content"/>
      </interleave>
    </element>
  </define>

  <define name="transition.labels">
    <a:documentation>
      A transition may have arbitrary many labels. This pattern should be
      used within a PNTD to define the transition labels.
    </a:documentation>
    <empty/>
  </define>

  <define name="node.content">
    <a:documentation>
      A node may have graphical information.
    </a:documentation>
    <optional>
      <element name="graphics">
        <ref name="nodegraphics.content"/>
      </element>
    </optional>
  </define>

  <define name="reference">
    <a:documentation>
      Here, we define the attribute ref including its data type.
      Modular PNML will extend this definition in order to change
      the behavior of references to export nodes of module instances.
    </a:documentation>
    <attribute name="ref">
      <data type="IDREF"/>
    </attribute>
  </define>

  <define name="refplace.content">
    <a:documentation>
      A reference place is a reference node.
    </a:documentation>
    <a:documentation>
      Validating instruction:
      - _ref_ MUST refer to _id_ of a reference place or of a place.
      - _ref_ MUST NOT refer to _id_ of its reference place element.
      - _ref_ MUST NOT refer to a cycle of reference places.
    </a:documentation>
    <element name="referencePlace">
      <ref name="refnode.content"/>
    </element>
  </define>

  <define name="reftrans.content">
```

```
    <a:documentation>
      A reference transition is a reference node.
    </a:documentation>
    <a:documentation>
      Validating instruction:
      - The reference (ref) MUST refer to a reference transition or to a
      transition.
      - The reference (ref) MUST NOT refer to the identifier (id) of its
      reference transition element.
      - The reference (ref) MUST NOT refer to a cycle of reference transitions.
    </a:documentation>
    <element name="referenceTransition">
      <ref name="refnode.content"/>
    </element>
  </define>

  <define name="refnode.content">
    <a:documentation>
      A reference node has the same content as a node.
      It adds a reference (ref) to a (reference) node.
    </a:documentation>
    <ref name="identifier.content"/>
    <ref name="reference"/>
    <ref name="basicobject.content"/>
    <ref name="node.content"/>
  </define>

  <define name="arc.content">
    <a:documentation>
      An arc has a unique identifier (id) and
      refers both to the node's id of its source and
      the node's id of its target.
      In general, if the source attribute refers to a place,
      then the target attribute refers to a transition and vice versa.
    </a:documentation>
    <element name="arc">
      <ref name="identifier.content"/>
      <attribute name="source">
        <data type="IDREF"/>
      </attribute>
      <attribute name="target">
        <data type="IDREF"/>
      </attribute>
      <a:documentation>
        The sub-elements of an arc may occur in any order.
        An arc may have a name, graphical and tool specific information.
        It may also have several labels.
      </a:documentation>
      <interleave>
        <optional>
          <ref name="Name"/>
        </optional>
        <ref name="arc.labels"/>
        <optional>
          <element name="graphics">
```

```
          <ref name="edgegraphics.content"/>
        </element>
      </optional>
      <zeroOrMore>
        <ref name="toolspecific.element"/>
      </zeroOrMore>
    </interleave>
  </element>
</define>

<define name="arc.labels">
  <a:documentation>
    An arc may have arbitrary many labels. This pattern should be used
    within a PNTD to define the arc labels.
  </a:documentation>
  <empty/>
</define>

<define name="pagegraphics.content">
  <a:documentation>
    A page graphics is actually a node graphics
  </a:documentation>
  <ref name="nodegraphics.content"/>
</define>

<define name="nodegraphics.content">
  <a:documentation>
    The sub-elements of a node's graphical part occur in any order.
    At least, there may be one position element.
    Furthermore, there may be a dimension, a fill, and a line element.
  </a:documentation>
  <interleave>
    <ref name="position.element"/>
    <optional>
      <ref name="dimension.element"/>
    </optional>
    <optional>
      <ref name="fill.element"/>
    </optional>
    <optional>
      <ref name="line.element"/>
    </optional>
  </interleave>
</define>

<define name="edgegraphics.content">
  <a:documentation>
    The sub-elements of an arc's graphical part occur in any order.
    There may be zero or more position elements.
    Furthermore, there may be a line element.
  </a:documentation>
  <interleave>
    <zeroOrMore>
      <ref name="position.element"/>
    </zeroOrMore>
```

```
    <optional>
      <ref name="line.element"/>
    </optional>
  </interleave>
</define>

<define name="simpletext.content">
  <a:documentation>
    This definition describes the contents of simple text labels
    without graphics.
  </a:documentation>
  <optional>
    <element name="text">
      <a:documentation>
        A text should have a value.
        If not, then there must be a default.
      </a:documentation>
      <text/>
    </element>
  </optional>
</define>

<define name="annotationstandard.content">
  <a:documentation>
    The definition annotationstandard.content describes the
    standard contents of an annotation.
    Each annotation may have graphical or tool specific information.
  </a:documentation>
  <interleave>
    <optional>
      <element name="graphics">
        <ref name="annotationgraphics.content"/>
      </element>
    </optional>
    <zeroOrMore>
      <ref name="toolspecific.element"/>
    </zeroOrMore>
  </interleave>
</define>

<define name="simpletextlabel.content">
  <a:documentation>
    A simple text label is an annotation to a net object containing
    arbitrary text.
    Its sub-elements occur in any order.
    A simple text label behaves like an attribute to a net object.
    Furthermore, it contains the standard annotation contents which
    basically defines the graphics of the text.
  </a:documentation>
  <interleave>
    <ref name="simpletext.content"/>
    <ref name="annotationstandard.content"/>
  </interleave>
</define>
```

```
<define name="Name">
  <a:documentation>
    Label definition for a user given name of an
    element.
  </a:documentation>
  <element name="name">
    <ref name="simpletextlabel.content"/>
  </element>
</define>

<define name="annotationgraphics.content">
  <a:documentation>
    An annotation's graphics part requires an offset element describing
    the offset the center point of the surrounding text box has to
    the reference point of the net object on which the annotation occurs.
    Furthermore, an annotation's graphic element may have a fill, a line,
    and font element.
  </a:documentation>
  <ref name="offset.element"/>
  <interleave>
    <optional>
      <ref name="fill.element"/>
    </optional>
    <optional>
      <ref name="line.element"/>
    </optional>
    <optional>
      <ref name="font.element"/>
    </optional>
  </interleave>
</define>

<define name="position.element">
  <a:documentation>
    A position element describes Cartesian coordinates.
  </a:documentation>
  <element name="position">
    <ref name="coordinate.attributes"/>
  </element>
</define>

<define name="offset.element">
  <a:documentation>
    An offset element describes Cartesian coordinates.
  </a:documentation>
  <element name="offset">
    <ref name="coordinate.attributes"/>
  </element>
</define>

<define name="coordinate.attributes">
  <a:documentation>
    The coordinates are decimal numbers and refer to an appropriate
    xy-system where the x-axis runs from left to right and the y-axis
    from top to bottom.
```

```
    </a:documentation>
    <attribute name="x">
      <data type="decimal"/>
    </attribute>
    <attribute name="y">
      <data type="decimal"/>
    </attribute>
  </define>

  <define name="dimension.element">
    <a:documentation>
      A dimension element describes the width (x coordinate) and height
      (y coordinate) of a node.
      The coordinates are actually positive decimals.
    </a:documentation>
    <element name="dimension">
      <attribute name="x">
        <ref name="positiveDecimal.content"/>
      </attribute>
      <attribute name="y">
        <ref name="positiveDecimal.content"/>
      </attribute>
    </element>
  </define>

  <define name="positiveDecimal.content" ns="http://www.w3.org/2001/XMLSchema-datatypes">
    <a:documentation>
      Definition of a restricted positive decimals domain with a total digits
      number of 4 and 1 fraction digit. Ranges from 0 to 999.9
    </a:documentation>
    <data type='decimal'>
      <param name='totalDigits'>4</param>
      <param name='fractionDigits'>1</param>
      <param name='minExclusive'>0</param>
    </data>
  </define>

  <define name="fill.element">
    <a:documentation>
      A fill element describes the interior colour, the gradient colour,
      and the gradient rotation between the colors of an object.  If an
      image is available the other attributes are ignored.
    </a:documentation>
    <element name="fill">
      <optional>
        <attribute name="color">
          <ref name="color.type"/>
        </attribute>
      </optional>
      <optional>
        <attribute name="gradient-color">
          <ref name="color.type"/>
        </attribute>
      </optional>
      <optional>
```

```
      <attribute name="gradient-rotation">
        <choice>
          <value>vertical</value>
          <value>horizontal</value>
          <value>diagonal</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="image">
        <data type="anyURI"/>
      </attribute>
    </optional>
  </element>
</define>

<define name="line.element">
  <a:documentation>
    A line element describes the shape, the colour, the width, and the
    style of an object.
  </a:documentation>
  <element name="line">
    <optional>
      <attribute name="shape">
        <choice>
          <value>line</value>
          <value>curve</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="color">
        <ref name="color.type"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="width">
        <ref name="positiveDecimal.content"/>
      </attribute>
    </optional>
    <optional>
      <attribute name="style">
        <choice>
          <value>solid</value>
          <value>dash</value>
          <value>dot</value>
        </choice>
      </attribute>
    </optional>
  </element>
</define>

<define name="color.type">
  <a:documentation>
    This describes the type of a color attribute. Actually, this comes
```

```
      from the CSS2 (and latest versions) data type system.
    </a:documentation>
    <text/>
  </define>

  <define name="font.element">
    <a:documentation>
      A font element describes several font attributes, the decoration,
      the alignment, and the rotation angle of an annotation's text.
      The font attributes (family, style, weight, size) should be conform
      to the CSS2 and latest versions data type system.
    </a:documentation>
    <element name="font">
      <optional>
        <attribute name="family">
          <text/>  <!-- actually, CSS2 and latest versions font-family -->
        </attribute>
      </optional>
      <optional>
        <attribute name="style">
          <text/>  <!-- actually, CSS2 and latest versions font-style -->
        </attribute>
      </optional>
      <optional>
        <attribute name="weight">
          <text/>  <!-- actually, CSS2 and latest versions font-weight -->
        </attribute>
      </optional>
      <optional>
        <attribute name="size">
          <text/>  <!-- actually, CSS2 and latest versions font-size -->
        </attribute>
      </optional>
      <optional>
        <attribute name="decoration">
          <choice>
            <value>underline</value>
            <value>overline</value>
            <value>line-through</value>
          </choice>
        </attribute>
      </optional>
      <optional>
        <attribute name="align">
          <choice>
            <value>left</value>
            <value>center</value>
            <value>right</value>
          </choice>
        </attribute>
      </optional>
      <optional>
        <attribute name="rotation">
          <data type="decimal"/>
        </attribute>
```

```
      </optional>
    </element>
  </define>

  <define name="toolspecific.element">
    <a:documentation>
      The tool specific information refers to a tool and its version.
      The further substructure is up to the tool.
    </a:documentation>
    <element name="toolspecific">
      <attribute name="tool">
        <text/>
      </attribute>
      <attribute name="version">
        <text/>
      </attribute>
      <zeroOrMore>
        <ref name="anyElement"/>
      </zeroOrMore>
    </element>
  </define>

</grammar>
```

Listing A.1: RELAX NG implementation of PNML Core Model

A.2 RELAX NG implementation of Petri Net Type Definition for Place/Transition nets

```
<?xml version="1.0" encoding="UTF-8"?>

<grammar ns="http://www.pnml.org/version-2009/grammar/pnml"
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0">

  <a:documentation>
    RELAX NG implementation of Petri Net Type Definition for Place/Transition nets.
    This PNTD re-defines the value of nettype.uri for P/T nets.

    File name: ptnet.pntd
    Version: 2009
    (c) 2007-2009
    Lom Hillah (AFNOR)
    Revision:
    July 2008 - L.H
  </a:documentation>

  <a:documentation>
    The PT Net type definition.
    This document also declares its namespace.
    All labels of this Petri net type come from the Conventions document.
    The use of token graphics as tool specific feature is possible.
  </a:documentation>

  <include href="http://www.pnml.org/version-2009/grammar/conventions.rng"/>

  <!--
  <include href="http://www.pnml.org/version-2009/grammar/pnmlextensions.rng"/>
    We do not need to include this, because the pnmlcoremodel.rng covers any
    toolspecific extension.
  -->

  <include href="http://www.pnml.org/version-2009/grammar/pnmlcoremodel.rng"/>

  <define name="nettype.uri" combine="choice">
    <a:documentation>
      The URI value for the net type attribute,
      declaring the type of P/T nets.
    </a:documentation>
    <attribute name="type">
      <value>http://www.pnml.org/version-2009/grammar/ptnet</value>
    </attribute>
  </define>


  <define name="PTMarking">
    <a:documentation>
      Label definition for initial marking in nets like P/T-nets.
      <contributed>Michael Weber</contributed>
      <date>2003-06-16</date>
```

```
      <reference>
        W. Reisig: Place/transition systems. In: LNCS 254. 1987.
      </reference>
    </a:documentation>
    <element name="initialMarking">
      <ref name="nonnegativeintegerlabel.content"/>
    </element>
  </define>

  <define name="PTArcAnnotation">
    <a:documentation>
      Label definition for arc inscriptions in P/T-nets.
      <contributed>Michael Weber, AFNOR</contributed>
      <date>2003-06-16</date>
      <reference>
        W. Reisig: Place/transition systems. In: LNCS 254. 1987.
      </reference>
    </a:documentation>
    <element name="inscription">
      <ref name="positiveintegerlabel.content"/>
    </element>
  </define>

  <define name="place.labels" combine="interleave">
    <a:documentation>
      A place of a P/T net may have an initial marking.
    </a:documentation>
    <optional><ref name="PTMarking"/></optional>
  </define>

  <define name="arc.labels" combine="interleave">
    <a:documentation>
      An arc of a P/T net may have an inscription.
    </a:documentation>
    <optional><ref name="PTArcAnnotation"/></optional>
  </define>

</grammar>
```

Listing A.2: RELAX NG implementation of PNTD for Place/Transition nets

Bibliography

[1] E. Jiménez Macías and M. Pérez de la Parte. Simulation and optimization of logistic and production systems using discrete and continuous petri nets. Simulation, 80 (3):143–152, 2004.

[2] T. Guasch, M A. Piera, J. Casanovas, and J. Figueras. Modelado y Simulación. Aplicación a procesos logísticos de fabricación y servicios. Edicions UPC, Barcelona, Spain, 2002.

[3] K. Jensen and L.M. Kristensen. Coloured Petri Nets. Modelling and Validation of Concurrent Systems. Springer, Berlin, Germany, 2009.

[4] I. León Samaniego. Seguridad y protección en envío y almacenamiento de datos. firmado y cifrado. aplicación a redes de petri y gestión de residuos con e3l. Master's thesis, Universidad de La Rioja. Logroño, Spain, 2011.

[5] C Xia. Analysis and application of petri subnet reduction. Procs. of the IEEE, 6 (8):1662–1669, 2011.

[6] Tadao Murata. Petri nets: properties, analysis and applications. Proceedings of the IEEE, 77(4):541–580, 1989. doi: 10.1109/5.24143. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0024645936&partnerID=40&md5=43ee2546042ef10b55eed792c6aeb409. cited By 4705.

[7] M Silva. Las Redes de Petri: en la Automática y en la Informática. Ed. AC, Madrid, Spain, 1985.

[8] M Silva. In Practice of Petri Nets in Manufacturing. Chapman and Hall, London, UK, 1993.

[9] R. David and H. Alla. Discrete, Continuous and Hybrid Petri Nets. Springer, Berlin, Germany, 1st ed., 2004 edition, 2010.

[10] JL Peterson. Petri Net Theory and the Modeling of Systems. Prentice Hall, Englewood Clifs, NJ, 1981.

[11] Carl Adam Petri. Kommunication mit Automaten. PhD thesis, Technischen Hoschule Darmstadt, 1962.

[12] Carl Adam Petri. Communication with automata. Technical Report 65-377, Rome Air Devolopement Center, 1966.

[13] Carl Adam Petri. Interpretations of net theory. Technical Report 75-07, Gessellschaft fur Mathematik und Datenverarbeitung, Bonn, 1976.

[14] Carl Adam Petri and E; Smith. The pragmatic dimension of net theory. In In procs. of the 8th European workshop on Applications and Theory of Petri Nets, 2007.

[15] L.E.a Holloway, B.H.b Krogh, and A.c Giua. A survey of petri net methods for controlled discrete event systems. Discrete Event Dynamic Systems: Theory and Applications, 7(2):151–190, 1997. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0031118870&partnerID=40&md5=6f083862d531c836dbf57418b3cdd7ce. cited By 220.

[16] J.I.a Latorre, E.b Jiménez, M.c Pérez, J.c Blanco, and E.c Martínez. The alternatives aggregation petri nets as a formalism to design discrete event systems. International Journal of Simulation and Process Modelling, 6(2):152–164, 2010. doi: 10.1504/IJSPM.2010.036019. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77958113541&partnerID=40&md5=de863be257a873f1cc403539c0739439. cited By 8.

[17] J.-I.a Latorre, E.b Jiménez, and M.c Pérez. Coloured petri nets as a formalism to represent alternative models for a discrete event system. pages 247–252, 2010. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84871324694&partnerID=40&md5=ef4fa8b5c804dd69148278197cc3587a. cited By 4.

[18] M. Silva, J. Júlvez, C. Mahulea, and C.R. Vázquez. On fluidization of discrete event models: Observation and control of continuous petri nets. Discrete Event Dynamic Systems: Theory and Applications, 21(4):427–497, 2011. doi: 10.1007/s10626-011-0116-9. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-80855132233&partnerID=40&md5=1373ddba02ca88d4580dda36b855f8ab. cited By 34.

[19] L.b Recalde, E.b Teruel, and M.a b b Silva. Modeling and analysis of sequential processes that cooperate through buffers. IEEE Transactions on Robotics and Automation, 14(2):267–277, 1998. doi: 10.1109/70.681245. URL http://www.scopus.com/inward/record.url?eid=2-s2.

0-0032048263&partnerID=40&md5=e26881591411e7172062e4c52606b95b. cited By 15.

[20] K. Jensen, L.M. Kristensen, and L. Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer, 9(3-4):213–254, 2007. doi: 10.1007/s10009-007-0038-x. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-34249933531&partnerID=40&md5=30c94bf82a91bc34487db18fa572dfca. cited By 428.

[21] L.M. Kristensen and K. Jensen. Teaching modelling and validation of concurrent systems using coloured petri nets. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5100 LNCS:19–34, 2008. doi: 10.1007/978-3-540-89287-8-2. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-58449092374&partnerID=40&md5=8cc7aa4643b2ebf3e720c7785186fe1a. cited By 0.

[22] R SILVA, M; VALETTE. Petri nets and flexible manufacturing. Advances in Petri Nets, 424:374–417, 1989.

[23] A. Desrochers and R.Y. Al-Jaar. Applications of Petri Nets in Manufacturing Systems. Modeling, Control ad Performance Analysis. IEEE Press, New York, USA, 2010.

[24] M.a b Silva and E.a Teruel. Petri nets for the design and operation of manufacturing systems. European Journal of Control, 3(3):182–199, 1997. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0031380765&partnerID=40&md5=301506cd728cf333b8411856d17b6e75. cited By 46.

[25] E. Jiménez, M. Pérez, and I. Latorre. Industrial applications of petri nets: System modelling and simulation. pages 159–164, 2006. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84870231099&partnerID=40&md5=c5bd5375e5b74e79e291daac8035ee73. cited By 5.

[26] J.I.a Latorre, E.b Jiménez, and M.b Pérez. The optimization problem based on alternatives aggregation petri nets as models for industrial discrete event systems. Simulation, 89(3):346–361, 2013. doi: 10.1177/0037549712464410. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84876047833&partnerID=40&md5=49380e89dca5af641e4e2108f351eeb2. cited By 12.

[27] Tadao Murata. State equation, controllability, and maximal matchings of petri nets. IEEE Transactions on Automatic Control, AC-22(3):412–416, 1977. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0017504301&partnerID=40&md5=55405f2bd4e03e2139f167355a60fef2. cited By 48.

[28] Tadao Murata. Petri nets, marked graphs, and circuit-system theory. Circuits Syst, 11(3):2–12, 1977. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0017500512&partnerID=40&md5=aec1a2ecfd640b95fe535c63fb4a0509. cited By 10.

[29] Manuel Silva. Introducing Petri nets. Chapman and Hall, 1993.

[30] M. Silva. 50 years after the phd thesis of carl adam petri: A perspective. pages 13–20, 2012. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84881038063&partnerID=40&md5=a9ebe7fa6a955a01c357a3420ed3e117. cited By 0.

[31] Kurt Jensen. Introduction to high-level petri nets. pages 723–726, 1985. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0022285861&partnerID=40&md5=3f7da33332c8d293e110262536d36563. cited By 5.

[32] Iso/iec 15909-1:2004 - systems and software engineering – high-level petri nets – part 1: Concepts, definitions and graphical notation, 2004. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=43538. [Online; accessed 21-May-2015].

[33] Lom Hillah, Fabrice Kordon, Laure Petrucci-Dauchy, and Nicolas Trèves. PN standardisation: A survey. In Formal Techniques for Networked and Distributed Systems - FORTE 2006, 26th IFIP WG 6.1 International Conference, Paris, France, September 26-29, 2006., pages 307–322, 2006. doi: 10.1007/11888116_23. URL http://dx.doi.org/10.1007/11888116_23.

[34] Pnml.org - pnml reference site, 2009. URL http://www.pnml.org/. [Online; accessed 21-May-2015].

[35] Iso/iec 15909-2:2011 - systems and software engineering – high-level petri nets – part 2: Transfer format, 2011. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=43538. [Online; accessed 21-May-2015].

[36] L.-M.a Hillah, F.b Kordon, C.c Lakos, and L.d Petrucci. Extending pnml scope: A framework to combine petri nets types. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and

Lecture Notes in Bioinformatics), 7400 LNCS:46–70, 2012. doi: 10.1007/978-3-642-35179-2_3. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84870041075&partnerID=40&md5=5ef5e22c0a2490324ffa2441bfc1a406. cited By 2.

[37] E. Kindler. The epnk: An extensible petri net tool for pnml. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6709 LNCS:318–327, 2011. doi: 10.1007/978-3-642-21834-7_18. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-79960086227&partnerID=40&md5=695ea1e592fda71522303206030789a8. cited By 3.

[38] J.a Billington, S.b Christensen, K.c Van Hee, E.d Kindler, O.e Kummer, L.f Petrucci, R.c Post, C.g Stehno, and M.h Weber. The petri net markup language: Concepts, technology, and tools. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2679:483–505, 2003. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-35248854880&partnerID=40&md5=82ea4c455b0683b147cbf9f8a5b1a557. cited By 99.

Faltan las referencias de paginas web