APPLICATIONS OF PETRI NETS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mathematics

by
Buket YILMAZ

October 2008
İZMİR

We approve the thesis of **Buket YILMAZ**

Assoc. Prof. Dr. Ünal UFUKTEPE
Supervisor

Assoc. Prof. Dr. Ahmet Hasan KOLTUKSUZ
Committee Member

Assoc. Prof. Dr. Durmuş ÖZDEMİR
Committee Member

15 October 2008

Prof. Dr. Oğuz YILMAZ
Head of the Mathematics Department

Prof. Dr. Hasan BÖKE
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my adviser Assoc. Prof. Dr. Ünal UFUKTEPE for his encouragement and endless support, for all that I have learnt from him, for all the opportunities he has given to me to travel and collaborate with other people, and most thanks for providing me his deep friendship.

I would like to thank Assoc. Prof. Dr. Durmuş ÖZDEMİR and Assist. Prof. Dr. Ahmet Hasan KOLTUKSUZ for being in my thesis committee, it was a great honor.

My special thanks go to my dear friend Belma ALKAN for her help and encouraging me during my thesis. I would also like to thank Alper YONTAR for his support and friendship.

Finally, I am deeply indepted to my family for always being with me and encouraging me in my life in many ways.

ABSTRACT

APPLICATIONS OF PETRI NETS

Petri nets are powerful formalism for modeling a wide range of dynamic systems and system behaviors. This thesis surveys the basic concept and application of Petri nets. The structure of Petri nets, their marking and execution and several examples of Petri net modeling. In this thesis we research into the analysis of Petri nets. Also we give the structure of Reachability graphs of Petri nets and their advantages for analyzing the Petri nets. The reachability problem for Petri nets is the problem of finding if $M_n \in R(M_0)$ for a given marking M_n in a net (N, M_0).

We present several different kinds of Petri nets, together with computer tools based on Mathematica. We give the Mathematica commands for Reachability problem and also we created Mathematica commands for Incidence matrix of Petri nets. We study the concept of Petri nets and applications of Petri nets. We especially focus on Biological applications on Petri nets and we work on modeling of Hashimoto's Thyroiditis in Petri Nets.

ÖZET

PETRİ AĞLARININ UYGULAMALARI

Petri Ağları dinamik sistem modellemesi ve sistem davranışları için çok güçlü bir yöntemdir. Bu tez, Petri Ağları ′ nın temel konsepti ve uygulamalarını araştırır. Petri Ağları ′ nın yapısı , işaretlemesi ve işletilmesi ve Petri Ağları modellemesine dair birkaç örnek verilmiştir. Bu tezde, Petri Ağları ′ nın analizini araştırdık. Ayrıca Petri Ağları ′ nın ulaşılabilirlik grafları ve bu grafların Petri Ağları ′ nın analizindeki avantajlarını da verdik. Petri Ağları ′ nda Ulaşılabilirlik problemi, (N, M_0) gibi bir ağda verilen M_n işaretlemesi için $M_n \in R(M_0)$ gibi bir M_n işaretlemesi bulmaktır.

Mathematica üzerinde temellendirilmiş birçok bilgisayar araçlarıyla birlikte, birçok farklı tip Petri Ağları′ ndan bahsettik. Ulaşılabilirlik problemi için yazılmış Mathematica komutlarını verdik. Petri Ağları ′ nın oran matrisleri için de Mathematica'da yeni komutlar geliştirdik. Bu tezde, Petri Ağları ′ nın genel yapısı ve Petri Ağları ′ ndaki uygulamaları inceledik. Özellikle, Petri Ağları ′ ndaki biyolojik uygulamalara yoğunlaştık ve Hashimoto tiroiditi hastalığının Petri Ağları′nda modellenmesi üzerine çalıştık.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Petri nets are graphical and mathematical modeling tool applicable to many systems. They are a promising tool for describing and studying information processing systems, that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a mathematical tool, a Petri net model can be described by a set of linear algebraic equations, or other mathematical models reflecting the behavior of the system. This opens a possibility for the formal analysis of the model. This allows one to perform a formal check of the properties related to the behavior of the underlying system, e.g., precedence relations amongst events, concurrent operations. Appropriate synchronization, freedom from deadlock, repetitive activities. and mutual exclusion of shared resources, to mention some. Petri nets are particularly suited to represent in a natural way logical interactions among parts of activities in a system. This theory originated from the doctoral thesis of C. A. Petri in 1962. Since then Petri nets have been developed and used in many theoretical as well as applicative.

A Petri net may be identified as a particular kind of bipartite directed graph populated by three types of objects. These objects are places, transitions, and directed arcs connecting places to transitions and transitions to places. Pictorially, places are depicted by circles and transitions as bars or boxes. A place is an input place to a transition if there exists a directed arc connecting this place to the transition. A place is an output place of a transition if there exists a directed arc connecting the transition to the place.

This thesis is organized as follows. Chapter two aimed of introducing the classical graph theory while the second part discusses the theory of the Petri nets and the execution rules, Structural and Behavioral properties of Petri net. In Chapter two, we analyzed through the generation of the reachability tree.

Chapter three is devoted to introduce the formalism of Petri nets with particular emphasis on the application of the methodology in the biological modeling. Basic definitions of Modified Petri nets and Biological Applications of Petri nets are given in Chapter three. We focus on Petri net models of non-linear gene interactions (Mayo 2005) in Chapter three.

The Chapter four includes Mathematica commands for Petri nets that we created in collaboration with Prof.Dr. Andres Iglesias.

In the last chapter, we give the basic information about Hashimotos Thyroiditis and Modeling of Hashimotos Thyroiditis in Petri nets. We create three different models for this disease.

CHAPTER 2

PRELIMINARIES

This Chapter will consist of some preliminary information about graph theory and short summary of (Murata 1989) and (Ye and Zhou 2003) about Petri nets and its' properties. One can find further information about Petri nets in (Murata 1989) and (Ye and Zhou 2003) about reachability of Petri nets.

2.1. Basic Definitions of Graph Theory

Definition 2.1 *A **graph** G=(V,E) is a mathematical structure consisting of two set V (vertices/nodes) and E (edges).*

Each edge has a set of one or two vertices associated to it, which one called its **endpoints**.

Definition 2.2 *A **loop** is an edge whose endpoints are equal. A non-simple graph with loops is depicted in Figure 2.1. (c).*

Definition 2.3 *A **multi-edge** is a collection of two or more edges having identical endpoints.*

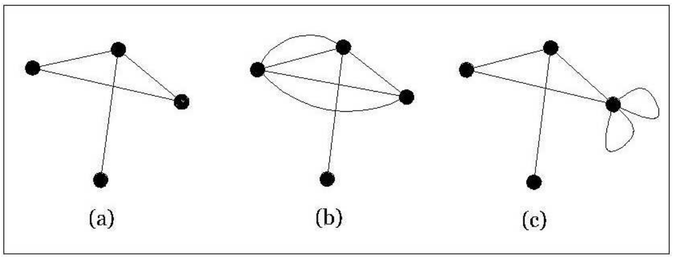


Figure 2.1. (a) A simple graph (b) A non-simple graph with multiple edges.

Definition 2.4 *A **simple graph** is a graph having no loops or multi-edges.*

Definition 2.5 *A **directed graph** is a graph each of whose edges is directed (Digraph).*

Definition 2.6 *A **weighted graph** is a graph in which each branch is given a numerical weight. A weighted graph is therefore a special type of labeled graph in which the labels are numbers (which are usually taken to be positive).*

Definition 2.7 *Graphs with labels attached to edges or vertices are called **labeled graph**. A graph in Figure 2.2. (a) is unlabeled graph.*

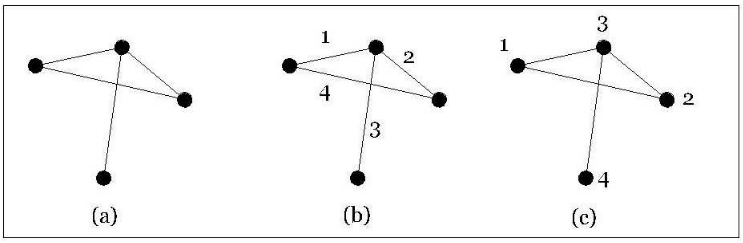


Figure 2.2. (b) An edge-labeled graph (c) A vertex-labeled graph.

Definition 2.8 *A **bipartite graph** G is a graph whose vertex set V can be partitioned into two subset U and W, such that each edge of G has one endpoint in U and one endpoint in W.*

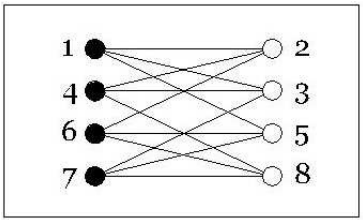


Figure 2.3. Bipartite graph

2.2. Introduction to Petri Nets

Petri nets are essentially weighted, labeled, directed graphs, with tokens that "move around" the graph as reactions take place. There are two types of nodes in a Petri net graph: places, depicted as circles, and transitions, which are rectangels, arcs may only be directed from place to transition (in which case they are referred to as input arcs) or transition to place (output arcs) . The implication of this is that a Petri net is always bipartite.

Definition 2.9 *A net is $PN = (P, T, F, W, M_0)$ where; $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of* ***places****,*

$T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of ***transitions****,*

$F \subseteq (P \times T) \cup (T \times P)$ is a set of ***arcs****,*

W is a ***weight function*** *of arcs, (default = 1)*

$M_0 : P \rightarrow \{0, 1, 2, \ldots\}$ is ***initial marking*** *where $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.*

Also; $k = P \rightarrow \{1, 2, 3, \ldots\} \cup \{\infty\} = partial\ capacity\ restriction\ (default = \infty)$.

Definition 2.10 *Let $X = P \cup T$ and $N = (P, T, F, W, M_0)$ be a PN, then:*

1. *$\bullet x = \{y \in X \mid (y, x) \in F\}$ is the pre-set (input set) of x,*

2. *$x\bullet = \{y \in X \mid (y, x) \in F\}$ is the pos-set (output set) of x,*

3. *$nbh[x] = \bullet x \cup x\bullet$ is called neighborhood of x,*

4. *If $Y \subseteq X$ then $\bullet Y = \cup \bullet x$ and $Y\bullet = \cup x\bullet$.*

Example 2.1 *The pre-set of t_2, $\bullet t_2 = \{p_4, p_3\}$ and the post-set of p_5, $p_5\bullet = \{t_4\}$.*

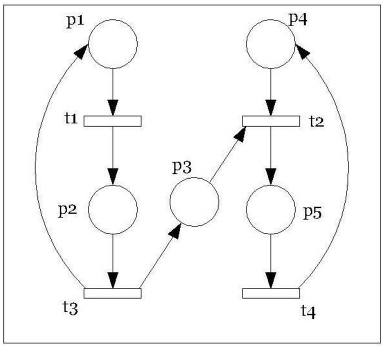


Figure 2.4. A Petri net in Example 2.1.

Definition 2.11 *Let $N = (P, T, F, W, M_0)$ be a PN then PN;*

1. *is* ***P-simple*** *iff $\forall\ x, y \in P, (\bullet x = \bullet y \wedge x\bullet = y\bullet \Longrightarrow x = y)$*

2. *is* ***T-simple*** *iff $\forall\ s, t \in T, (\bullet s = \bullet t \wedge s\bullet = t\bullet \Longrightarrow s = t)$*

3. *has no* ***isolated places*** *iff $\forall\ x \in X, nbh(x) \neq \emptyset$*

Definition 2.12 *A PN is;*

1. ***pure*** *iff $\forall\ x \in X, [\bullet x \cap x\bullet = \emptyset]$,*

2. ***simple*** *iff $\forall\ x, y \in X, [(\bullet x = \bullet y \wedge x\bullet = y\bullet) \Longrightarrow x = y]$*

2.2.1. Firing Rule

Let m(p) be the number of tokens in place p. For $t \in T$, a transition t is enabled at a marking m if,

every place $p \in \bullet t$ satisfies $M(p) \geq w(p,t)$ and

every place $p \in t\bullet$ satisfies $M(p) + w(p,t) \leq k(p)$

The occurrence of t leads to the **successor marking** M′, defined by

$$M'(p) = \begin{cases} M(p) & if\ p \notin \bullet t\, and\, p \notin t\bullet \\ M(p) - w(p,t) & if\ p \in \bullet t\, and\, p \notin t\bullet \\ M(p) + w(p,t) & if\ p \notin \bullet t\, and\, p \in t\bullet \\ M(p) + w(p,t) - w(p,t) & if\ p \in \bullet t\, and\, p \in t\bullet \end{cases}$$

$M[t >$ is enable under the marking M.

$M[t > M' = M\ t\ M'$

Example 2.2 *The following figure is illustrated, using the well-known chemical reaction:* $2H_2 + O_2 \rightarrow 2H_2O$. *Two tokens in each input place in first Petri net show that two units of* H_2 *and* O_2 *are available, and transition t is enabled. After firing t, the marking will change to the one shown in next Petri net, where the transition t is no longer enabled.*

$M_0 = \{3,\ 1,\ 1\} \rightarrow M_1\ =\ \{1,\ 2,\ 4\}$

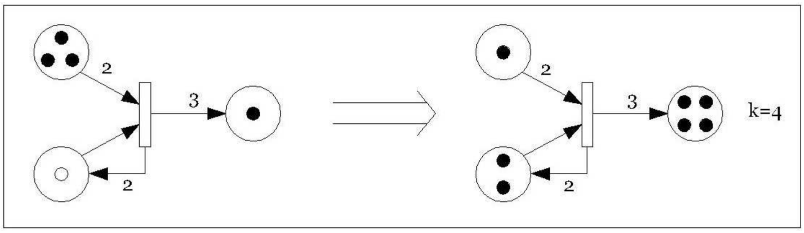


Figure 2.5. A Petri net of a chemical reaction.

(Source : Murata 1989)

Definition 2.13 *A transition without any input place is called a* ***source transition****, and one without any output place is called a* ***sink transition****. A source transition is unconditionally enabled, and that the firing of a sink transition consumes tokens, but does not produce any.*

Definition 2.14 *A pair of place p and a transition t is called a* ***self-loop*** *if p is both an input and output place of t. A Petri net is said to be* ***pure*** *if it has no self-loops.*

Definition 2.15 *A Petri net is said to be* ***ordinary*** *if all of its arc weights are 1's.*

2.2.2. Firing Sequences and Reachability

A finite sequence $\sigma = t_1\ t_2\ \ldots\ t_m$ of transitions is a **finite firing sequence** leading from M_0 to M_n if
$M_0\ t_1\ M_1\ t_2\ \ldots\ t_n\ M_n$

- A marking M is **reachable** (from M_0) if there is a firing sequence leading from M_0 to M.
  $[M_0\ >$ is the set of all reachable markings.

- An infinite sequence $\sigma = t_1\ t_2\ t_3 \ldots$ is an infinite firing sequence enabled at M_0 if $M_0\ t_1\ M_1\ t_2\ M_2\ t_3\ \ldots$

The **Marking graph** of a marked petri-net is an edge-labeled graph with initial vertex;
initial vertex - initial marking M_0
vertices - set of reachable markings $[M_0\ >$
labeled edges - set of triples (M, t, M') such that $M\ t\ M'$

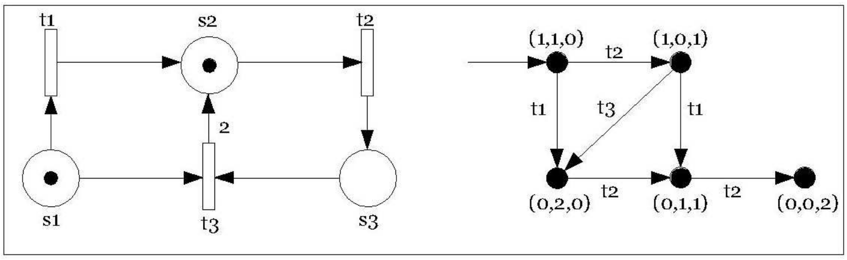


Figure 2.6. A Petri net and its' marking graph.

2.3. Behavioral Properties of Petri Nets

Two types of properties can be studied with a Petri net model: those which depend on the initial marking and those which are independent of initial marking.

The former type of properties is referred to as marking-dependent of **behavioral** properties, whereas the latter type of properties is called **structural** properties.

1. **Terminating**: A Petri net is terminating if there is no infinite firing sequence.

   Definition 2.16 *A marking* $M \in M_0[>$ *is called a dead marking if* $\forall t \in T$ *:* $M[t \geq$. *Under this marking, no transition can be fired (Ye and Zhou 2003). A PN is* ***deadlock-free*** *if each reachable marking enables a transition. A PN is in Figure 2.6. is not deadlock-free, because under the marking* (0, 0, 2) *no transition can be fired.*

2. **Reachability** : Reachability is a fundamental basis for studying in the dynamic properties of any system. The firing of an enabled transition will change the token marking in a net according to the firing rule. A sequence of firings will result in a sequence of markings. A marking M_n is said to be reachable from a marking M_0 if there exists a sequence of firings that transforms M_0 to M_n. *A* firing sequence is denoted by

   $\sigma = M_o\ t_1\ M_1\ t_2\ M_2\ ...\ i_n\ M_n$ or simply

   $\sigma = t_1 t_2\ ...\ t_n$

   In this case, M_n is reachable from M_0 by σ and we write $M_0[\sigma > M_n$. The set of all possible markings reachable from M_0 in a net (N, M_0) is denoted by *R(N,* M_0) or simply R(M_0) and the set of all possible firing sequence is denoted by *L(N,* M_0) or simply *L(M_0).*

   The **reachability problem** for Petri nets is the problem of finding if $M_n \in R(M_0)$ for a given marking M_n in a net *(N,* M_0).

   For the set of all possible reachable markings *R(N,* M_0), we communicated with Andres Iglesias (University of Cantabria) and created Mathematica commands for Reachability problem (See Chapter 4).

3. **Safeness** : A place $p \in P$ of a *PN,* is **safe** if $\forall M \in M_0[>$: $M(p) \leq 1$. A *PN* is safe if each place in the net is **safe.**

4. **Boundedness** : A Petri net is said to be **k-bounded** if the number of tokens in any place p, where $p \in P$, is always less or equal to k (k is a nonnegative

integer number) for every marking M reachable from the initial marking $M \in R(M_0)$. A Petri net is **safe** if it is $1 - \textit{bounded}$.

5. **Liveness** : A Petri net is said to be **live** if, no matter what marking has been reached from M_0, it is possible to ultimately fire any transition of the net by progressing through some further firing sequence.

6. **Reversibility and Home State** : A Petri net is said to be **reversible** if, for each marking $M \in R(M_0)$, M_0 is reachable from M. In many applications, it is not necessary to get back to the initial state as long as one get back to some (home) state. Therefore, we relax the reversibility condition and define a home state. A marking M' is said to be a **home state** if, for each marking $M \in R(M_0)$, M' is reachable from M.

7. **Coverability** : A marking M in a Petri net is said to be **coverable** if there exists a marking $M' \in R(M_0)$ such that $M'(p) \geq M(p)$ for each p in the net.

8. **Persistence** : A Petri net is said to be **persistent** if, for any two enabled transitions, the firing of one transition will not disable the other. A transition in a persistent net, once it is enabled, will stay enabled until it fires.

9. **Fairness** : Two transitions t_1 and t_2 said to be in a bounded-fair (or B-fair) relation if the maximum number of times that either one can fire while the other is not firing is bounded. A Petri net is said to be a B-fair net if every pair of transitions in the net are in a B-fair relation.

2.4. Analysis Methods

2.4.1. The Coverability Tree

Given a Petri net (N, M_0) from the initial marking M_0, we can obtain as many "new" markings as the number of the enabled transitions. From each new marking, we can again reach more markings. This process results in a tree representation of the markings. Nodes represent markings generated from M_0 (the root) and its successors, and each arc represents a transition firing, which transforms one marking to another.

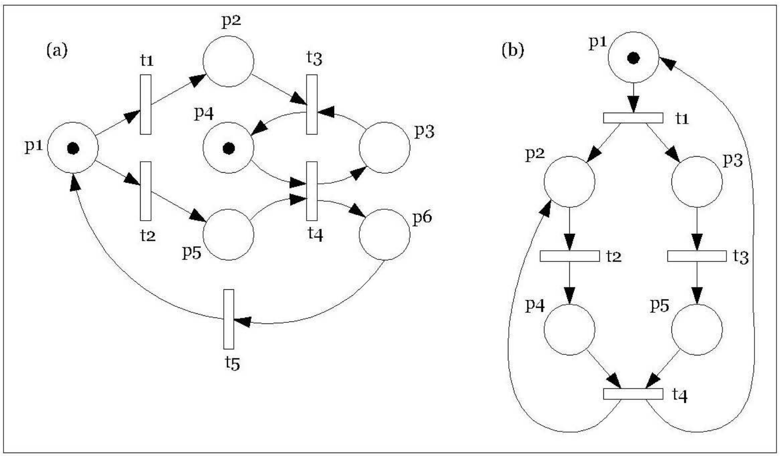


Figure 2.7. (a) A safe, non-live PN and (b) An unbounded, live, nonreversible PN.

(Source : Murata 1989)

The coverability tree for a Petri net (N, M_0) is constructed by the following algorithm.

Step 1. Label the initial marking M_0 as the root and tag it "new"

Step 2. While "new" markings exist, do the following:

Step 2.1 Select a new marking M.

Step 2.2 If M is identical to a marking on the path from the root to M, then tag M "old" and go to another new marking.

Step 2.3 If no transitions are enabled at M, tag M "dead-end".

Step 2.4 While there exist enabled transitions at M, do the following for each enabled transition t at M:

Step 2.4.1 Obtain the marking M' that results from firing t at M.

Step 2.4.2 On the path from the root to M if there exists a marking M'' such that $M'(p) \geq M''(p)$ for each place p and $M' \neq M''$, i.e., M'' is coverable, then replace $M'(p)$ by w for each p such that $M'(p) > M''(p)$

Step 2.4.3 Introduce M' as a node , draw an arc with label t from M to M', and tag M' "new".

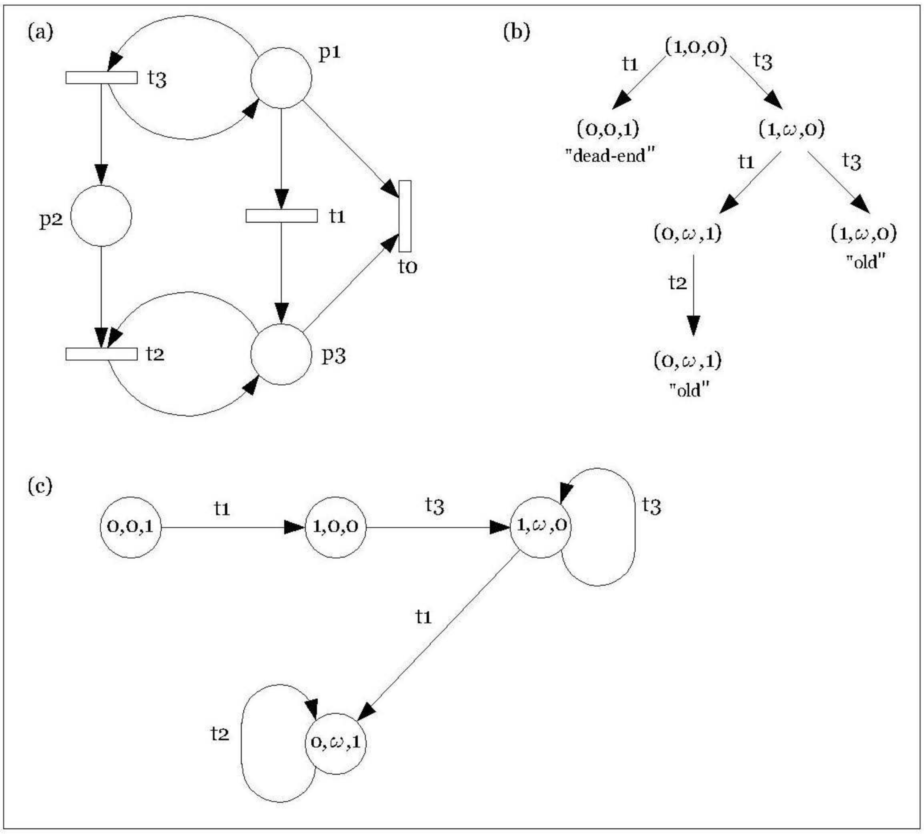


Figure 2.8. (b) The coverability tree and (c) The coverability graph for PN in (a).

(Source : Murata 1989)

The above tree representation, however will grow infinitely large if the net is unbounded. To keep the tree finite, we introduce a special symbol w, which can be thought of as "infinity". It has the properties that for each integer $n, w > n, w \mp n = w$ and $w \geq w$.

Some of the properties that can be studied by using the coverability tree T for a Petri net (N, M_0) are the following:

1. A net (N, M_0) is **bounded** and thus $R(M_0)$ is finite iff w does not appear in any node labels in T.

2. A net (N, M_0) is **safe** iff only $0's$ and $1's$ appear in node labels in T.

3. A transition t is **dead** iff it does not appear as an arc label in T.

4. If M is **reachable** from M_0, then there exists a node labeled M' such that $M \leq M'$

For a bounded Petri net, the coverability tree is called the reachability tree since it contains all possible markings. The disadvantage is that this is an exhaustive method. However, in general, because of the information lost by the use of symbol w (which may represent only even or odd numbers, increasing or decreasing numbers) the reachability and liveness problems can not be solved by the coverability tree method alone.

The **coverability graph** of Petri net (M, M_0) is labeled directed graph G = (V,E). Its node set V is the set of all distinct labeled nodes in the coverability tree and the arc set E is the set of arcs labeled with single transition t_k representing all possible single transition firings such that M_i $[t_k > M_j$, where M_i and M_j are in V. For a bounded Petri net, the coverability graph is referred to as the **reachability graph**, because the vertex set V becomes the same as the reachability set $R(M_0)$.

2.4.2. Reachability Graphs

A reachability graph of a PN is a directed graph $G = (V, E)$, where $v \in V$ represents a class of reachable markings; $e \in E$ represents a directed arc from a class of markings to the other class of markings. An example is shown in Figure 2.9. A reachability graph is also called **occurrence graph** or **state space**. The reachability graph demonstrates a better performance than the reachability tree.

Although a PN is finite, the set of its reachable markings is not always finite. For instance, when a PN is not safe or bounded, its number of tokens can be infinite, thus the set of reachable markings being infinite.

In a reachability graph, a (likely infinite) class of nodes can be abstracted as a node in order to obtain a finite representation of the reachability graph. Furthermore, the marking abstraction process used consistent denotations, i.e. the increasing or decreasing number of tokens in a marking, denoted by $weight \times n$. The obtained reachability graph is unique.

The reachability tree can not distinguish these cases because of the abstraction w. Instead, the reachability graph retains the appropriate level of abstraction by using the weight on the arcs.

Petri net analysis using reachability graphs (Ye and Zhou 2003)

1. *Safeness and boundedness*

A PN is safe if all places in the net are safe, i.e. the number of tokens in each place never exceeds one. This can be checked easily since each reachable marking is explicitly retained in the nodes of the reachability graph.

The boundedness of a PN can be determined by checking that *n (n* = 1, 2, 3, · ·) does not exist in any marking on the nodes of the reachability graph.

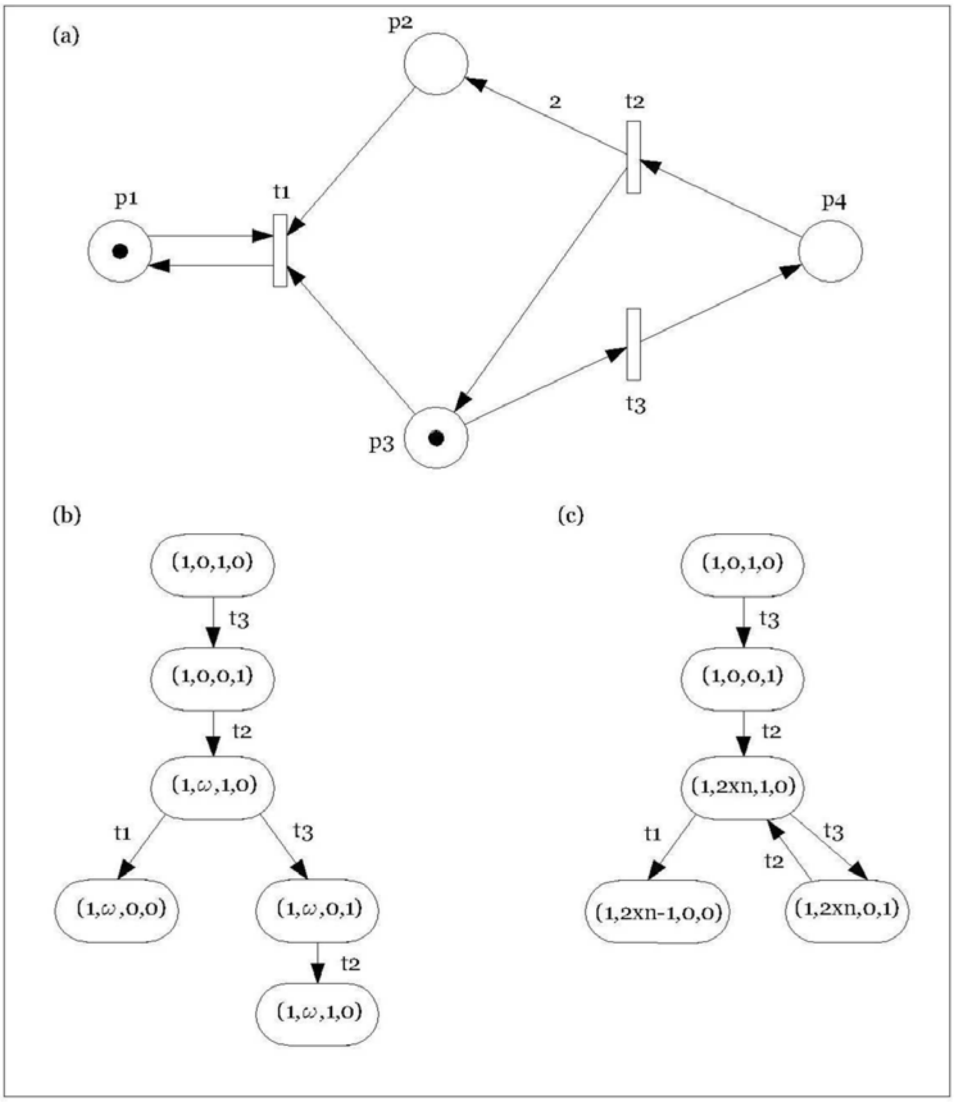


Figure 2.9. (b) The reachability tree and (c) The reachability graph for PN in (a). (Source : Ye and Zhou 2003)

2. *Conservation*

A PN is conservative if its tokens are neither created nor destroyed. If *n* exists in a reachability graph, then the weight w of each relative arc should be considered. If *w* equals 0, the PN is conservative, otherwise the PN is not conservative.

If there is no *n* in the reachability graph, then for each reachable marking M, the following equation is considered: $w_1M(p_1) + w_2M(p_2) + \cdots + w_mM(p_m) = K$, where K is a constant, $K = w_1M_0(p_1) + w_2M_0(p_2) + \cdots + w_mM_0(p_m)$. If the equation stands, the PN is conservative.

3. *Reachability and coverability*

The reachability problem is to decide if a given marking M belongs to the set M0[> or not. We examine the nodes in the graph one by one to find the node which includes the marking that equals M or contains M. For instance, in Figure 2.9., the marking (1, 6, 0, 1) is reachable, since the node (1, 2 × *n,* 0, 1) contains (1, 6, 0, 1), where *n* is a natural number. It is easy to see that the marking (1, 6, 0, 1) is reached after the transitions t_3, t_2, t_3, t_2, t_3, from the initial marking.

For the coverability problem we want to determine, for a given marking M, if a marking M′ belongs to the set M[> or not. Because of the resolution of the reachability problem, the coverability problem can be solved easily.

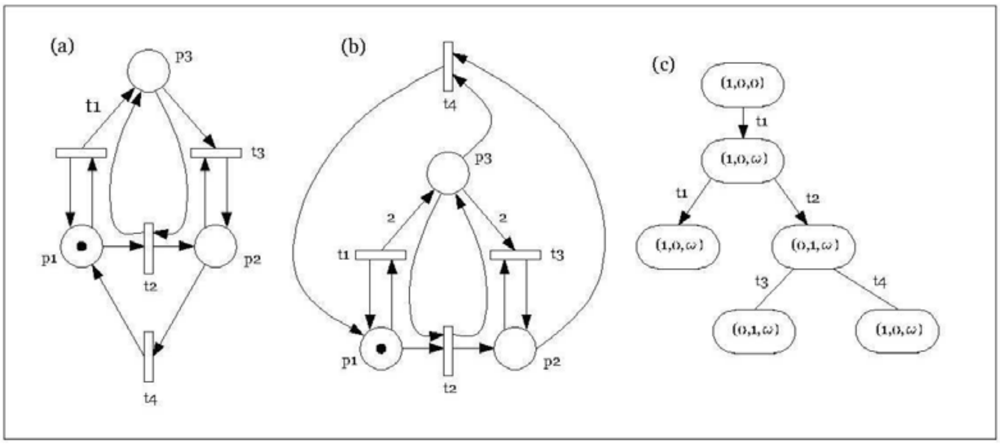


Figure 2.10. (a) A live PN. (b) A PN with deadlock. (c) Their reachability tree.
(Source : Ye and Zhou 2003)

4. *Liveness*

A PN is live if there is no deadlock in it. A marking is a *dead marking* if a PN is deadlocked.

A PN may ha ve a deadlock even if there is a terminal node in its reachability graph. For example, the node (1, 2 × *n,* 0, 0) is a terminal node in Figure 2.9.(b), so the PN in Figure 2.9. may be deadlocked. For the example shown in Figure 2.10.(a), there is no deadlock in the PN, however, there is no terminal node in the reachability graph.

We can use reachability graph to solve efficiently the liveness problem in a reachability tree, there are two kinds of nodes without successors: *terminal* and *frontier* nodes, while in a reachability graph, there is only one kind of nodes without successors: *terminal* node.

If you find a node has no successor, it is a terminal node, the PN must be deadlocked; otherwise, the PN is live at any time.

From a reachability tree, it is impossible to determine if its corresponding PN has deadlocks. For instance, Figure 2.10.(a) has no deadlock while Figure 2.10.(b) has a deadlock, but they have the same reachability tree. However, if we consider their corresponding reachability graphs in Figure 2.11., it is easy to draw the conclusion.

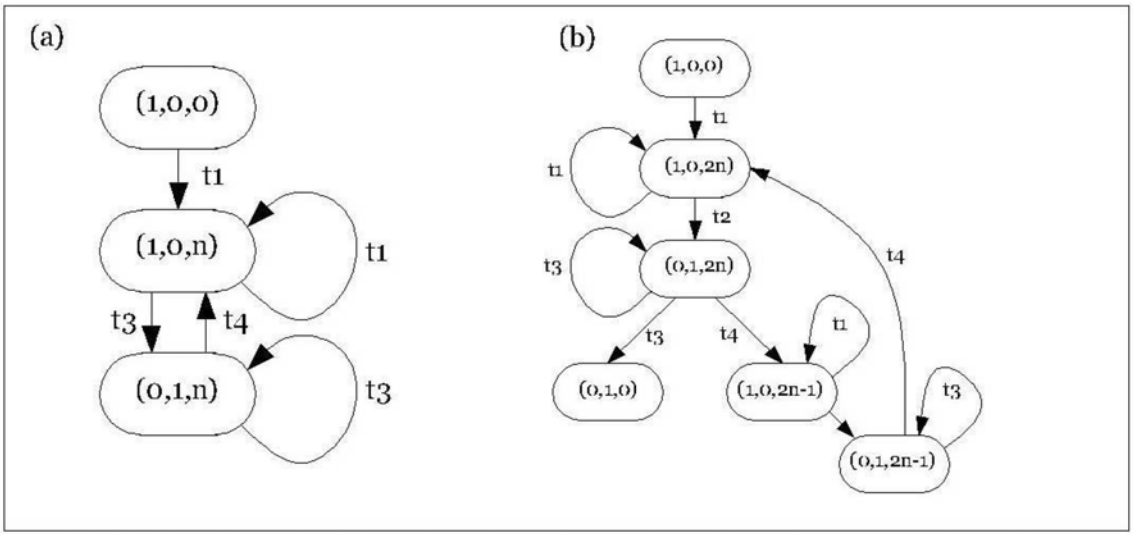


Figure 2.11. The reachability graphs of Figure 2.9.(a) and (b).

(Source : Ye and Zhou 2003)

2.4.3. Incidence Matrix and State Equation

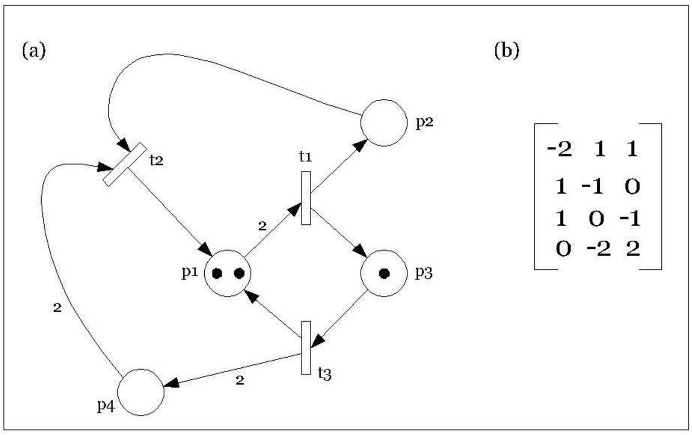
We present matrix equations to describe and analyze completely the dynamic behavior of Petri nets.

Incidence Matrix : For a Petri net *PN* with n transitions and m places, the incidence matrix $A = [a_{ij}]$ is an $n \times m$ matrix of integers and its typical entry is given by;

$$a_{ij} = {a_{ij}}^{+} - {a_{ij}}^{-}$$

where ${a_{ij}}^{+} = w(i, j)$ is the weight of the arc from transition i to its output place j and ${a_{ij}}^{-} = w(i, j)$ is the weight of the arc to transition i from its input place j. Transition i is enabled at marking M iff

$$ {a_{ij}}^{-} \leq M(j), j = 1, 2, \ldots , m$$



$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{bmatrix}$$

Figure 2.12. (b) The incidence matrix of a given Petri net in (a).

(Source : Murata 1989)

State Equation: In writing matrix equations, we write a marking M_k as an $m \times 1$ column vector. The jth entry of M_k denotes the number of tokens in place j immediately after the kth firing in some firing sequence. The ***k*th firing** or **control vector** u_k is an $n \times 1$ column vector of $n - 10's$ and one nonzero entry, a 1 in the ith position indicating that transition i fires at the kth firing. Since ith row of the incidence matrix A denotes the change of the marking as the result of firing transition i, we can write the following state equation for a Petri net (Murata 1977) :

$$M_k = M_{k-1} + A^T u_k, k = 1, 2, \dots . \tag{2.1}$$

Necessary Reachability Condition : Suppose that a destination marking M_d is reachable from M_0 through a firing sequence $\{u_1, u_2, \dots, u_d\}$. Writing the state equation (2.1) for $i = 1, 2, \dots, d$ and summing them, we obtain

$$M_d = M_0 + A^T \sum_{k=1}^{d} u_k \tag{2.2}$$

which can be rewritten as

$$A^T x = \triangle M \tag{2.3}$$

where $\triangle M = M_d - M_0$ and $x = \sum_{k=1}^{d} u_k$. Here x is an $n \times 1$ column vector of nonnegative integers and is called the*firing count vector*. The ith entry of x denotes the number of times that transition i must fire to transform M_0 to M_d. It is well known (Hohn 1958) that a set of linear algebraic equations (2.3) has a solution x iff $\triangle M$ is orthogonal to every solution y of its homogeneous system,

$$Ay = 0 \tag{2.4}$$

Let r be the rank of A, and partition A in the following form:

$$\begin{array}{cccc} & m-r & r & \\ & \longleftrightarrow & \longleftrightarrow & \\ A = & \left[\begin{matrix} A_{11} \\ A_{21} \end{matrix}\right. & \left.\begin{matrix} A_{12} \\ A_{22} \end{matrix}\right] & \begin{matrix} \updownarrow r \\ \updownarrow n-r \end{matrix} \end{array}$$

where A_{12} is a nonsingular square matrix of order r. A set of *(m - r)* linearly independent solutions y for (2.4) can be given as the *(m - r)* rows of the following $(m - r) \times m$ matrix B_f :

$$B_f = [I_\mu : -A_{11}^T (A_{12}^T)^{-1}] \tag{2.5}$$

where I_μ is the identity matrix of order $\mu = m - r$. Note that $AB_f^T = 0$. That is, the vector space spanned by the row vectors of A is orthogonal to the vector space spanned by the row vectors of B_f. The matrix B_f corresponds to the fundamental circuit matrix in the case of a marked graph. Now, the condition that $\triangle M$ is orthogonal to every solution for $Ay\ = 0$ is equivalent to the following condition:

$$B_f \triangle M = 0 \tag{2.6}$$

Thus, if M_d is reachable from M_0, then the corresponding firing count vector x must exist and (2.6) must hold. Therefore, we have the following necessary condition for reachability in an unrestricted Petri net (Murata 1977).

Theorem 2.1 *If M_d is reachable from M_0 in a Petri net (N, M_0), then $B_f \triangle M = 0$, where $\triangle M = M_d - M_0$ and B_f is given by* (2.5).

The contrapositive of this theorem provides the following sufficient condition for nonreachability.

Lemma 2.1 *In a Petri net (N, M_0) a marking M_d is not reachable from $M_0 (\neq M_d)$ if their difference is a linear combination of the row vectors of B_f, that is,*

$$\triangle M = B_f^T z \tag{2.7}$$

where z is a nonzero $\mu \times 1$ column vector.

Proof : If (2.7) holds, then $B_f \triangle M = B_f B_f^T z \neq 0$, since $z \neq 0$ and $B_f B_f^T$ is a $\mu \times \mu$ nonsingular matrix (because the rank of B_f is $\mu = m - r$). Therefore, by previous theorem, M_d is not reachable from M_0. An integer solution x of the homogeneous equation ($\triangle M = 0$ in (2.3))

$$A^T x = 0 \tag{2.8}$$

is called a **T-invariant**, and an integer solution y of the transposed homogeneous equation $Ay = 0$ is called **S-invariant**.

2.4.4. Simple Reduction Rules for Analysis

There exist many transformation techniques for Petri nets. In this section, we present only the simplest transformations, which can be used for analyzing liveness, safeness and boundedness. It is not difficult to see that the following six operations preserve the properties of liveness, safeness and boundedness. That is, let (N,M_0) and (N', M_0') be the Petri nets before and after one of the following transformations. Then (N', M_0') is live, safe or bounded iff (N,M_0) is live, safe or bounded, respectively.

1. Fusion of Series Places(FSP) as depicted in Figure 2.13.a.

2. Fusion of Series Transitions(FST) as depicted in Figure 2.13.b.

3. Fusion of Parallel Places(FPP) as depicted in Figure 2.13.c.

4. Fusion of Parallel Transitions(FPT) as depicted in Figure 2.13.d.

5. Elimination of Self-Loop Places(ESP) as depicted in Figure 2.13.e.

6. Elimination of Self-Loop Transitions(EST) as depicted in Figure 2.13.f.

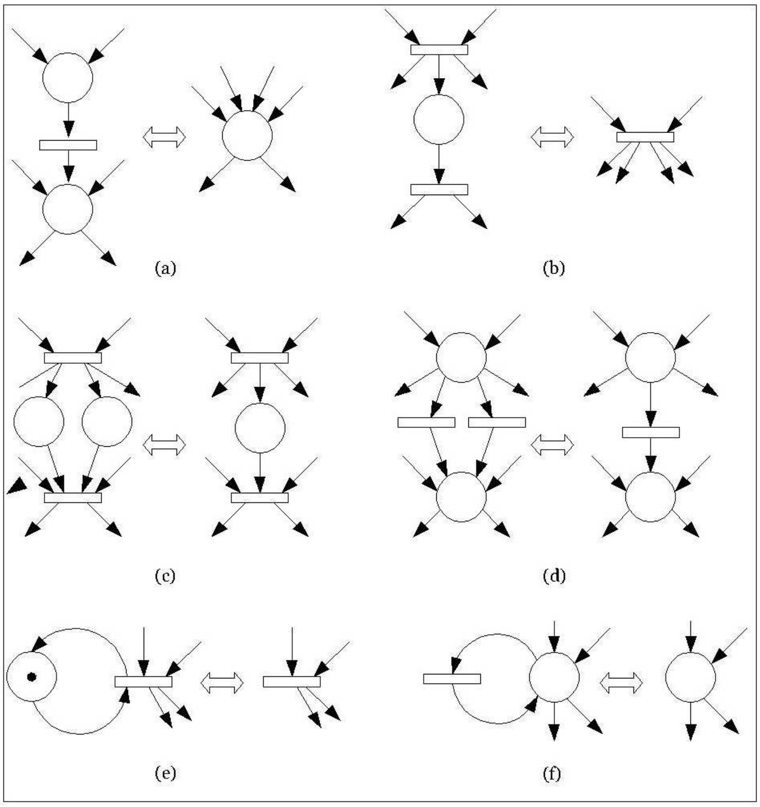


Figure 2.13. Six transformations preserving liveness, safeness and boundedness.

(Source : Murata 1989)

Example 2.3 *The net shown in Figure 2.14(a) can be reduced to the one shown in Figure 2.14.(b) after firing t_2 to remove the token in p_1 and then fusing t_1 and t_2 into t_{12} and*

t_3 and t_4 into t_{34}. The net in Figure 2.14.(b) can be reduced to the one shown in Figure 2.14.(c) after eliminating self-loop transition t_{12} and place p_3. It is easy to see that both nets shown in Figure 2.14.(a) and Figure 2.14.(c) are bounded and non-live.

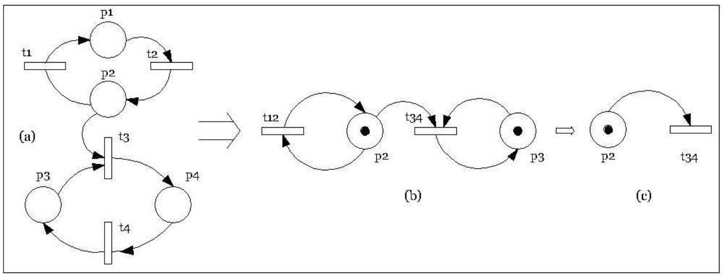


Figure 2.14. All the three nets are bounded, non-live, and nonreversible.

(Source : Murata 1989)

2.5. Characterizations of Liveness, Safeness

2.5.1. Subclass of Petri Nets

1. A **state machine(SM)** is an ordinary Petri net such that each transition t has exactly one input place and exactly one output place, i.e.,
   $| \bullet t| = |t \bullet | = 1$ for all $t \in T$

2. A **marked graph(MG)** is an ordinary Petri net such that each place p has exactly one input transition and exactly one output transition i.e.,
   $| \bullet p| = |p \bullet | = 1$ for all $p \in P$

3. An **extended free-choice net (EFC)** is an ordinary Petri net such that
   $p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet = p_2 \bullet$ for all $p_1, p_2 \in P$

4. An **asymmetric choice net (AC)** (also known as a simple net) is an ordinary Petri net such that
   $p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet \subseteq p_2 \bullet$ or $p_1 \bullet \supseteq p_2 \bullet$ for all $p_1, p_2 \in P$

5. A **free-choice net (FC)** is an ordinary Petri net such that every arc from a place is either a unique outgoing arc or a unique incoming arc to a transition, i.e.,

For all $p \in P, |p \bullet| \leq 1$ or $\bullet(p\bullet) = p$,equivalently,

For all $p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow |p_1 \bullet| = |p_2 \bullet| = 1$

2.5.2. Liveness and Safeness Criteria

Existence of Live-Safe Markings:

A place p (transition t) is said to be a **source place (source transition)** if $\bullet p = \emptyset(\bullet t = \emptyset)$.

A place p (transition t) is said to be a **sink place (sink transition)** if $p\bullet = \emptyset(t\bullet = \emptyset)$.

Theorem 2.2 *If a Petri net is live and safe, then there are no source or sink places and source or sink transitions.*

This theorem can be generalized and we can state that if a connected Petri net is live and safe, then N is **strongly-connected**.

2.6. Structural Properties

1. **Structural Liveness:**
   A Petri net N is said to be *structurally live* if there exists a live initial marking for N.

2. **Controllability:** A Petri net N is said to be *completely controllable* if any marking is reachable from any other marking.

3. **Structural Boundedness:** A Petri net N is said to be *structurally bounded* if it is bounded for any finite initial marking M_0.

4. **Conservativeness:** A Petri net N is said to be *conservative* if every $M \in R(M_0)$;
   $\sum_i w_i M(p_i) = \sum_i w_i M_0(p_i) = a$ constant.

5. **Repetitiveness:** A Petri net N is said to be *repetitive* if there exists a marking M_0 and a firing sequence from M_0 such that every transition occurs infinitely often in σ.

6. **Consistency:** A Petri net N is said to be *consistent* if there exists a marking M_0 and a firing sequence from M_0 back to M_0 such that every transition occurs at least once in σ.

CHAPTER 3

APPLICATIONS OF PETRI NETS

Petri nets are a relatively non-mathematical alternative to ODEs (Ordinary Differential Equations) for modeling time-dependent processes. Petri nets, which were originally developed in the 1960s, have long been used to model discrete distributed flow systems, such as data communications networks and manufacturing processes. It wasn't until 1993 that biologists realized that this modeling approach could be easily adapted to representing biological systems(Reddy and Mavrovouniotis 1993). Petri nets were originally designed to function as discrete automata, but later enhancements have added the ability to deal with continuous quantities (Matsuno 2003).

In this chapter we give summary of modifications and extensions made on Petri nets in the first section. And we focus on biological applications on Petri nets in second section.

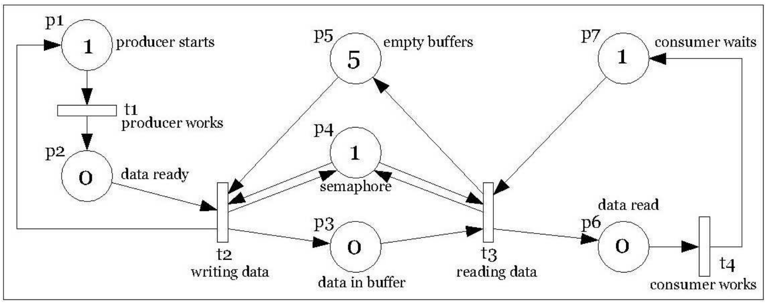


Figure 3.1. Producer and consumer problem.

First, we give two models for applications of Petri nets. The first model a cooperation between two processes called Producer and Consumer. The Producer prepares data and writes them to buffers. If there is no empty buffer, the Producer must wait. The Consumer reads data supplied by the Producer. The initial marking of the place "Empty buffers" is the total number of buffers available (initially all the buffers are empty). The semaphore ensures that only one process can work with data at a time. After reading the data the Consumer returns the empty buffer. This Petri net model in Figure 3.1. is 5-bounded and not safe.

Next example is a model of a simple one way message transmission system. The system is made of the sender's user, the sender, the receiver and the receiver's user. Markings of places represent these facts about system:

p_1 = a message has been generated by the user of the sender,

p_2 = a message has been transmitted to the receiver.

p_3 = the receiver is waiting for a message (its user has asked for it),

p_4 = the user of the receiver asks for a message,

p_5 = the sender is ready for transmission,

p_6 = the receiver is ready to accept a message,

p_7 = an acknowledgement passed from the sender to its user,

p_8 = an acknowledgement received by the sender,

p_9 = a message passed to the user of the receiver.

Petri net model is 1-bounded and safe.

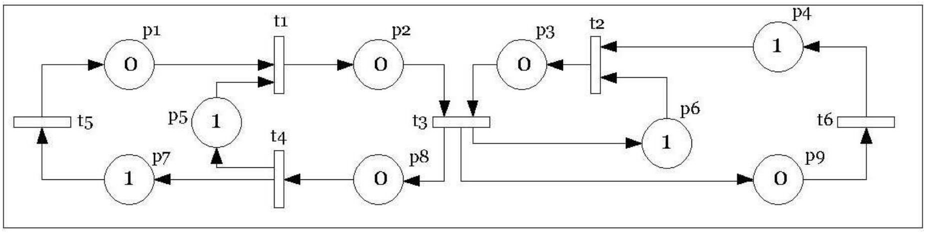


Figure 3.2. Transmission system.

The transitions represent the following activities in the system:

t_1 = the sender transmits a message to the receiver,

t_2 = the receiver accepts a request for a message from its user,

t_3 = the receiver accepts a message, passes it to its user, sends an acknowledgement and becomes ready for the next message,

t_4 = the sender accepts an acknowledgement, passes it to its user and becomes ready for the next message,

t_5 = the sender's user prepares the next message,

t_6 = the receiver's user processes a message.

3.1. Modified Petri Nets

3.1.1. Timed Nets

The concept of time is not explicitly given in the original definition of Petri nets. However, for performance evaluation and scheduling problems of dynamic systems, it is (at present) necessary and useful to introduce time delays associated with transitions and/or places in their net models. Such a Petri net model is known as a (deterministic) *timed net* if the delays are deterministically specified.

The aim is finding how fast each transition can initiate firing in a periodically operated timed Petri net, where a period τ is defined as the time to complete a firing sequence leading back to the starting marking after firing each transition at least once. τ is called a *cycle time*. Thus, it is assumed that the net is consistent, i.e.,

$$\exists x > 0, A^T x = 0. \tag{3.1}$$

Suppose there is a delay of at least $d_i sec$ associated with transition t_i, i = 1, 2, $\cdots$ n. This means that when t_i is enabled, a_{ij}^- tokens will be reserved in place p_j for at least $d_i sec$ before their removal by firing t_i, where a_{ij}^- is the weight of the arc from p_j to t_i. It's defined that the *resource-time product (RTP)* as the product of the number of tokens (resources) and the length of time that these tokens reside in a place. Thus , the RTP is given by $a_{ij}^- d_i x_i$, which can be written in matrix form

$$(A^-)^T Dx \tag{3.2}$$

where $A^- = [a_{ij}^-]_{nxm}$ and D is the diagonal matrix of d_i, i = 1, 2, $\cdots$ n. $(A^-)^T Dx$ represents the vector of m RTP's for m places, and each RTP considers only reserved tokens (Murata 1989).

3.1.2. Stochastic Nets

Suppose the delay d_i associated with transition t_i is a non-negative continuous random variable X with the exponential distribution function

$$F_x(x) = P[X \leq x] = 1 - e^{-\lambda_i x} \tag{3.3}$$

(or the probability density function, $f_x(x) = \lambda_i e^{-\lambda_i x}$).
Then, the average delay is given by

$$\overline{d_i} = \int [1 - F_x(x)]dx = \int e^{-\lambda_i x} dx = \frac{1}{\lambda_i} \tag{3.4}$$

where λ_i is the firing rate of transition t_i.

A *Stochastic Petri net* (SPN) is a Petri net where each transition is associated with an exponentially distributed random variable that expresses the delay from the enabling to the firing of the transition. In a case where several transitions are simultaneously enabled, the transition that has the shortest delay will fire first. Due to the memoryless property of the exponential distribution of firing delays, it has been shown (Goss and Peccoud 1988) that the reachability graph of a bounded SPN is isomorphic to a finite Markov Chain (Murata 1989).

3.1.3. Colored Petri Nets

A *Colored Petri net* (CPN) model is a description of the modeled system, and it can be used as a specification of a system that we want to build or as a presentation of a system that we want to explain. By creating a model we can investigate a new system before we construct it. This is an obvious advantage, in particular for systems where design errors may jeopardize security or be expensive to correct. Furthermore, the behavior of CPN model can be analyzed,either by means of simulation or by means of more formal analysis methods (Zhang and Hong 2006).

The colored Petri net N is defined by the n-tuple (P, T, Pre, Post, C, M) where:
(P, T, Pre, Post, C, M) is a Petri net and the tokens of M are identified by a color; $C = \{C_1, C_2, ...\}$, a set of colors. The incidence mappings Pre and Post are functions of the token colors (Hardy and Robillard 2004).

Colored Petri nets are frequently used in many applications. (Fantia and Giuab 2006) can be checked for further information.In this paper, colored Petri nets (CPN) were used to model the dynamics of a railway system: places represent tracks and stations, tokens are trains. Using digraph tools, deadlock situations are characterized and a strategy is established to define off-line a set of constraints that prevent deadlocks. They show that these constraints limit the weighted sum of colored tokens in subsets of places.

3.2. Biological Applications on Petri nets

Petri nets can serve to model, analyze and simulate biological processes. The use of Petri nets in biology was suggested for the first time by Reddy, who qualitatively analyzed metabolic pathways (Reddy and Mavrovouniotis 1993). Since then, several types of biological processes have been modeled and simulated with Petri nets, mainly molecular biology systems, but also in epidemic and ecologic modeling (Hardy and Robillard 2004).

Traditional Petri nets were originally suggested for biological pathway modeling by Reddy, and the bridging of molecular species and chemical reactions with Petri net places and transitions was achieved for the first time by them (Reddy and Mavrovouniotis 1993). The association of places with molecular species and transitions with chemical reactions is used for all types of Petri net model presented in this review. However, special situations necessitate more than one place for one species, for example, when distinguishing between an enzyme in an activated or a deactivated state, or a metabolite in various sites of the cell. The number of tokens indicates the quantity of substance and it corresponds to a predefined measure unit according to the scale of the model, such as the exact number of molecules, mole, millimole, etc. Reddy demonstrated that the Petri net approach was an appropriate tool for a preliminary qualitative analysis of biopathways. Behavioral and structural properties of Petri nets, like liveness, boundedness and invariants were used to identify some characteristics of models. This analysis approach was applied to the erythrocyte pentose phosphate pathway and to the main glycolytic pathway (Reddy and Mavrovouniotis 1996). The analysis of these pathways showed boundaries for certain molecular species, conservation properties, regenerative reactions and situations leading to a deadlocking of the system (Hardy and Robillard 2004).

The following figure shows two snapshots of a simple Petri net, modeling just one chemical reaction, given by its stoichiometric equation. The resulting metabolic Petri nets describe the set of all paths from the input to the output compounds respecting the given stoichiometric relations (Heiner, et al. 2004).

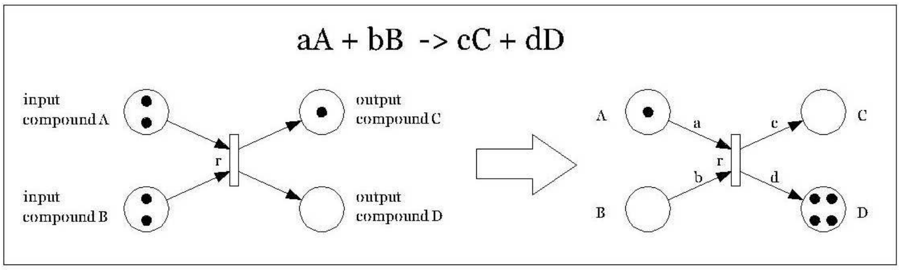


Figure 3.3. Petri net model of a single chemical reaction.

(Source : Heiner 2004)

In a molecular model, each place is a species of molecule with some number of tokens inside, representing the number of molecules or concentration of that species, whereas transitions represent reactions. Places are connected to transitions by arrows (or directed arcs) either from source (input) places into the transition or from the transition to product (output) places. The stoichiometry of a reaction is indicated by a weight on the arc. Because Petri nets are a discrete system, they are driven in stepwise fashion by implicit time increments. A transition fires (i.e. the reaction occurs) when the markings at all its input places are greater than the weights on its input arcs (i.e. when there are enough source molecules), producing the product of the appropriate weights on its output arcs (which are subsequently stored in the product places).

Petri nets have been used to model a wide range of biological processes, including qualitative modeling of apoptosis (Heiner, et al. 2004), iron homeostasis (Sackmann, et al. 2006) and the yeast mating response (Sackmann, et al. 2006). A particularly interesting application of Petri nets was recently demonstrated with the modeling and biomedical profiling of metabolic disorders (Chen and Hofestadt 2006). Using the urea cycle as an example, Chen and Hoefstadt (Chen and Hofestadt 2006) built a hybrid Petri net that qualitatively modeled metabolite levels, transcription factor activity and signaling pathway changes for this complex pathway. This model successfully predicted the elevated arginine levels, hyperammonaemia, and mild increases in urine orotic acid found in patients with ornithine transcarbamalase deficiency (one of the key enzymes in the urea cycle). This model also enabled them to rationalize the potential therapeutic treatments for this disorder, including limited protein intake and the supplementation of the diet with arginine or citrulline. The authors argue that similar Petri net models could be used to assist with the diagnosis, drug development and treatment of a wide variety of metabolic disorders (Materi and Wishart 2007).

Also Petri nets were used to modeling of protein production (Barjis J. and Barjis I. 1999). They demonstrated two models, one of them was compact and the other one was detailed. Constructing a detailed Petri net model of the protein production, it could be easily analyzed the behavior of petri nets and they could get correct and adequate results.

3.2.1. Biological Applications on Stochastic Nets

The random nature of molecular interactions at low concentration has been observed in several experiments. However, the Kolmogorov equations of the stochastic model corresponding to a biological system rapidly become impossible to solve analytically. Goss and Peccoud used stochastic Petri nets (SPN) (Ajmane, et al. 1991) as a tool for biological modeling of stochastic models (Goss and Peccoud 1988). They implied that the Petri net formalism and its modeling power can reduce model implementation delays. With their model, they successfully analyzed the stabilizing effect of the ROM protein on the genetic network controlling the replication of ColEl plasmid replication (Goss and Peccoud 1999).

In the SPN model of a system composed of molecular interactions, each place corresponds to a particular molecular species. Tokens represent molecules and transitions between places are chemical reactions involving reactants (input places) and products (output places). At any time, the marking of the system indicates the number of molecules of each species involved. The values of arcs originating from input places and ending at output places are the equivalent of stoichiometric coefficients. As in traditional place/transition nets, if the number of tokens at input places is higher than the weight of all the input arcs of a transition, this transition can fire. In molecular terms, the firing of a transition means that a chemical reaction is occurring. The particularity of SPN is that the firing of a transition is not instantaneous. There is a delay following a probabilistic distribution, thus the delay is a random variable. In SPN biological models, this delay is interpreted as the reaction rate, and it is given by the weight function of the corresponding transition. The delay mean time is obtained by the transition reaction rate, which is a function of a stochastic rate constant and the quantity of each molecular species involved as a reactant or a catalyst. This constant takes into account volume, temperature, pH and other environmental factors. It is also related to the deterministic rate of the reaction (Hardy and Robillard 2004).

3.2.2. Biological Applications on Colored Petri Nets

The differentiation between categories of tokens when modeling large systems with Petri nets was considered in order to reduce the size of models. Thus, Petri nets were enhanced with this new feature by adding colors. The resulting high-level net, colored Petri net (CPN), is composed of tokens identified by a color. With this augmentation of the formalism, it is possible to represent, in the same model, different dynamic behaviors modeled by different token colors (Jensen 1992).

Genrich et al. modeled an enzymatic reaction with a colored Petri net transition (Genrich and Kuffner 2001). This transition is connected to places representing substrates like re-actant, product, enzyme and inhibitor. In this model, tokens are identified by two colors, one associated with the substance name and the other with its concentration. The CPN used for this modeling also has functional features because an execution model is called upon, after every firing of the transition, to calculate and modify substrate concentration. These reaction rate calculations are performed according to the MichaelisMenten biochemical equation, augmented by an additional term for the free reaction energy. The specific constants associated with each enzyme needed for these calculations are extracted from the BRENDA biochemical database (Brenda 1970). This transition is, in fact, a sub-model integrated into the glycolysis and citric acid metabolic models. A chain of enzymatic reactions constitutes the metabolic network to be quantitatively simulated. Another interesting part of the Genrich et al. paper is to propose rules for automatic pathway identification from databases, after which the pathways are modeled as Petri nets for simulation purposes (Hardy and Robillard 2004).

3.2.3. Learning Petri net models of non-linear gene interactions

In biological science, identifying the DNA sequence variations in human populations that cause genetic disease, is very important. But, for most common genetic diseases such as sporadic breast cancer, this problem is difficult because gene interactions are non-linear. Biochemists are also interested in finding " biochemically plausible " models of the casual influence of genes on disease. If a model is biochemically plausible to a degree, then it may reveal characteristic of the actual biochemical pathways in humans that can aid understanding of the disease.

Petri net models are considered biochemically plausible. They are ideal for this purpose, because Petri nets can be used to parallel, interacting processes.In this paper, the practical problem of automatically finding Petri net models of biochemical interactions is addressed.

Micheal Mayo has created a new approach to work of (Moore and Hahn 2003). He used Petri nets to model biological interactions.

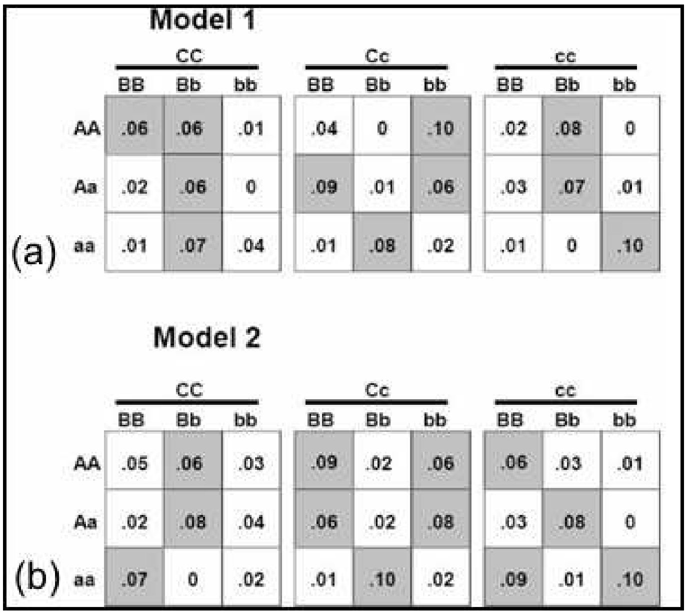
For modeling the non-linear gene interactions that caused disease we need genetic background:

A genotype is defined as a combination of two alleles, one inherited from the mother, the other from the father. For example, if *A* and *a* are the alleles in a biological population, then the possible genotypes in individuals are *AA, Aa* or *aa* (the ordering of the alleles is unimportant). A consecutive ordering of genotypes such as these form a DNA sequence.

A disease is considered genetic if the presence of one or more genotypes is statistically correlated to the presence of a disease in a population. That is, there is a non-trivial conditional probability $P(D|G)$ derived from frequency data of an individual having a disease D given a genotype G. This probability function is also called the penetrance (Moore and Hahn 2003).

The aim of this paper is finding a perfect Petri nets for the models of three-gene or four-gene non-linear interactions, which were developed on the Penetrance function values of different DNA sequence variations. Moore and Hahn discovered models for Penetrance function values of hypertension and sporadic breast cancer risk, using a genetic algorithm to mine human genetic data obtained from an experiment.

In Figure 3.4., two competing non-linear models of the influence of genotypes on the risk of essential hypertension are depicted. The risk of disease in these examples is determined by the presence of three genes in various combinations, but the average penetrance of each single gene is approximately equal. Dark shaded cells represent high risk (more than 5 per cent penetrance) while light unshaded cells represent low risk (less than 5 per cent penetrance). Since each of the three genotypes can have three different values, there are a total of 3^3 or 27 different DNA sequence variations that a single individual could have.



Model 1

(a)

	CC			Cc			cc		
	BB	Bb	bb	BB	Bb	bb	BB	Bb	bb
AA	.06	.06	.01	.04	0	.10	.02	.08	0
Aa	.02	.06	0	.09	.01	.06	.03	.07	.01
aa	.01	.07	.04	.01	.08	.02	.01	0	.10

Model 2

(b)

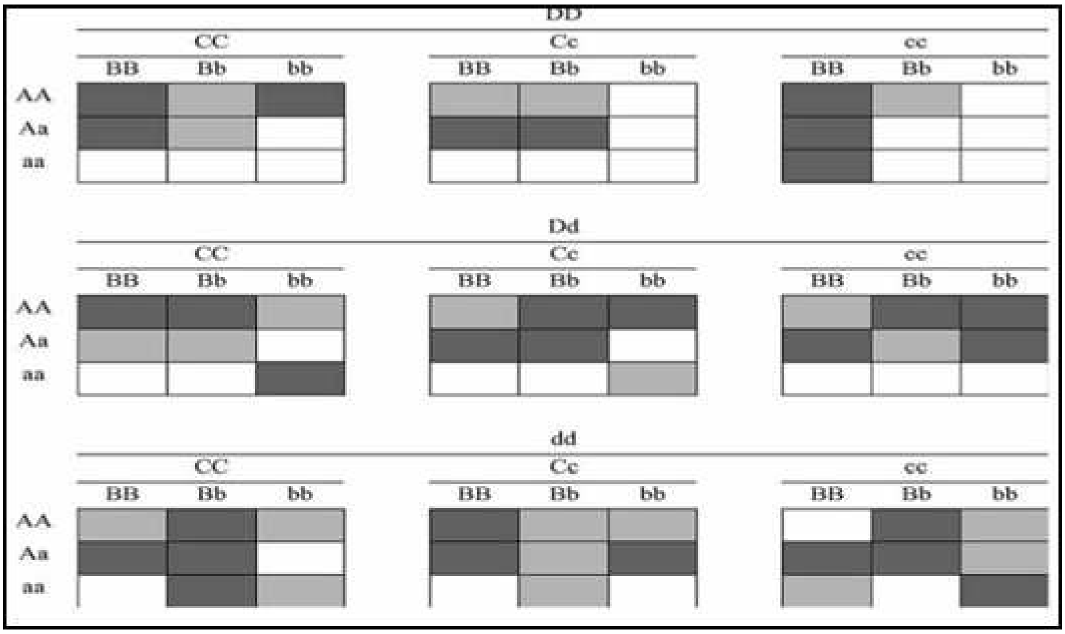
	CC			Cc			cc		
	BB	Bb	bb	BB	Bb	bb	BB	Bb	bb
AA	.05	.06	.03	.09	.02	.06	.06	.03	.01
Aa	.02	.08	.04	.06	.02	.08	.03	.08	0
aa	.07	0	.02	.01	.10	.02	.09	.01	.10

Figure 3.4. Two three-gene non-linear genetic models of essential hypertension.

(Source : Mayo 2005)

Figure 3.5. depicts a similar non-linear model of sporadic breast cancer risk. In contrast to the previous two models, which were three-gene models, this is a four-gene model. The data from which this model was derived was recently acquired after a controlled study involving 200 female subjects, and is believed to be the first four-gene non-linear model identified (Ritchie, et al. 2001). Again, shaded cells indicate high risk and unshaded cells indicate low risk. The empty cells indicate a sequence variation not found in the sample,and of the 3^4 or 81 possible variations, 52 different sequences were present in the experimental sample.

Mayo created a software for finding Petri nets which are matched to Figure 3.4. and Figure 3.5., based on genetic algorithm. He used *Multi-start random hill climbing* strategy for searching in algorithm.



DD

	CC			Cc			cc		
	BB	Bb	bb	BB	Bb	bb	BB	Bb	bb
AA	■	▨	■	▨	▨		■	▨	
Aa	■	▨		■	■		■		
aa							■		

Dd

	CC			Cc			cc		
	BB	Bb	bb	BB	Bb	bb	BB	Bb	bb
AA	■	■	▨	▨	■	■	▨	■	■
Aa	▨	▨		■	■		■	▨	■
aa			■			▨			

dd

	CC			Cc			cc		
	BB	Bb	bb	BB	Bb	bb	BB	Bb	bb
AA	▨	■	▨	■	▨	▨		■	▨
Aa	■	■		■	▨	■	■	■	▨
aa		■	▨		▨		▨		■

Figure 3.5. A four-gene non-linear genetic models of sporadic breast cancer.

(Source : Mayo 2005)

An analogy between Petri nets and biochemical networks is described in detail by (Goss and Peccoud 1988). The idea is that places represent molecular species, and tokens correspond to individual molecules in a biochemical network. A marking, therefore, is the distribution or concentration of molecules over molecular species at a particular point in time. Transitions correspond to chemical reactions, implying that input places are the reactants and output places are the products of the reactions. The arc weights are reaction rates, and an enabled transition means that a reaction is possible although it may not necessarily occur. In this context Petri nets can be considered biochemically plausible.

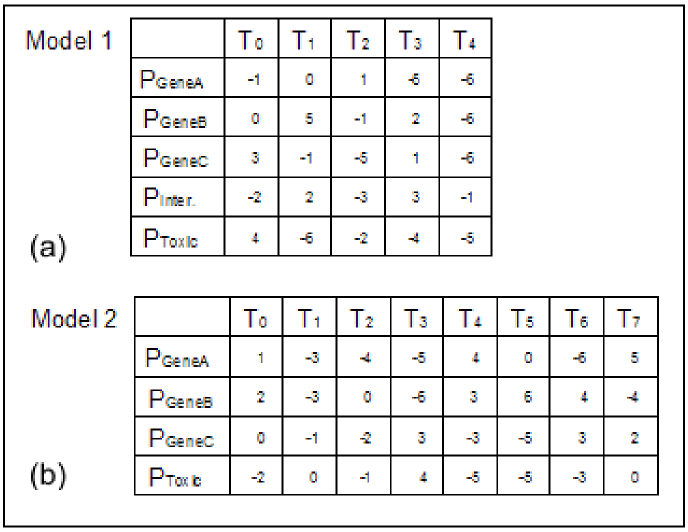
Petri net components can be gene dependent or not for modeling of gene interactions. And , different genotypes could , determine different arc weights, connectivity, place capacities, so on.

Moore and Hahn decided that the best approach was let to the search algorithm itself select how variables and their dependencies would be determined. In their approach; concentrations of a particular toxic substance that is produced by biochemical reactions, directly related to the risk of disease.In particular, if the concentration of this toxic substance exceeds a certain amount, then a high-risk assignment is made. In terms of the analogy with Petri nets, this implies that one place is a ″target″ or output place (representing the toxic substance), and if the number of tokens present at the place is greater than some discrete fixed threshold, then a high-risk assessment is made. Petri nets that make identical assessments for each DNA sequence variation as specified by the nonlinear models in Figure 3.4. and Figure 3.5. can be considered perfect.

Mayo's approach starts from the assumption that genes may only influence a very small portion of petri nets, with the rest of the network being independent of genetic variation. He chose only the initial marking (initial distribution of tokens or molecules) be genotype dependent, with the remainder of the variables (such as the structure of the network and arc weights) being genotype independent.

The basic value function to be maximized is the performance of the Petri net on the task of modeling an accumulation of a particular toxic substance that causes a disease. This is represented simply as the number of tokens at a designated place. If the number of tokens exceeds a fixed threshold, defined as 5% of the maximum capacity of the place after a certain amount of time (the same rule as used by (Moore and Hahn 2003)), then the risk assignment is high; otherwise it is low. For the model to be perfect, therefore, all high-risk genetic variations should lead to high-risk assignments; all low risk variations should avoid such an assignment. For the three-gene models, the maximum value is therefore 27 (being 27 correct assignments) and for the four-gene model it is 52.

He decided to assign a quantity of tokens to each place for each genetic variation for the initial marking. For example, if a gene can have values *AA, Aa or aa*, and the maximum place capacity is 10 tokens, then the quantity for *AA* could be 10 tokens, the quantity for *Aa* might be 5 tokens, and the quantity for *aa* may be 0 tokens. The quantities are completely arbitrary and different values could have been used; the key point is to distinguish between different genotypes by assigning them different quantities of tokens for the initial marking. To take this a step further, if a similar encoding scheme is used for all the genes, then a complete DNA sequence such as *AabbCC* leads to an initial marking of (5, 0,10) over the first three places of the network, uniquely specifying the entire portion of the genotype being considered.



Model 1

(a)

	T_0	T_1	T_2	T_3	T_4
P_{GeneA}	-1	0	1	-5	-6
P_{GeneB}	0	5	-1	2	-6
P_{GeneC}	3	-1	-5	1	-6
$P_{Inter.}$	-2	2	-3	3	-1
P_{Toxic}	4	-6	-2	-4	-5

Model 2

(b)

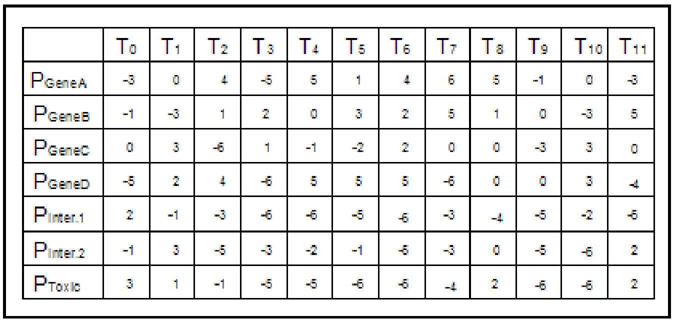
	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
P_{GeneA}	1	-3	-4	-5	4	0	-6	5
P_{GeneB}	2	-3	0	-6	3	6	4	-4
P_{GeneC}	0	-1	-2	3	-3	-5	3	2
P_{Toxic}	-2	0	-1	4	-5	-5	-3	0

Figure 3.6. Perfect solutions for models 1 and 2 from Figure 3.4.

(Source : Mayo 2005)

One additional place (which, in these experiments, is not one of the places used to encode the input) is designated to be the output place representing the ”toxic” or disease-causing substance. After firing the network a number of times given the initial marking, this will be the place where the concentration of tokens is measured in order to determine the risk assessment. There are, therefore, a minimum of four required places for the three-gene models in Figure 3.4. (being three places encoding the input and one output place), and five required places for the four-gene models. This contrasts to Moore and Hahns approach, which found Petri net models with only one place.

Mayo chose incidence matrix for the representation of perfect Petri nets. After several times of running the software, the maximum place capacity was set to 10, and the minimum and maximum weights for each arc were set to −6 to 6. His software found two different Petri net models for three-gene interaction models. In Figure 3.6., solutions for both models in Figure 3.4. are shown in incidence matrix form. And Figure 3.7., the perfect Petri net model for four-gene interactions model consists of 7 places and 12 transitions (Mayo 2005).



	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
P_{GeneA}	-3	0	4	-5	5	1	4	6	5	-1	0	-3
P_{GeneB}	-1	-3	1	2	0	3	2	5	1	0	-3	5
P_{GeneC}	0	3	-6	1	-1	-2	2	0	0	-3	3	0
P_{GeneD}	-5	2	4	-6	5	5	5	-6	0	0	3	-4
$P_{Inter.1}$	2	-1	-3	-6	-6	-5	-6	-3	-4	-5	-2	-6
$P_{Inter.2}$	-1	3	-5	-3	-2	-1	-6	-3	0	-5	-6	2
P_{Toxic}	3	1	-1	-5	-5	-6	-6	-4	2	-6	-6	2

Figure 3.7. A perfect PN for the four-gene problem depicted in Figure 3.5.

(Source : Mayo 2005)

CHAPTER 4

PETRI NETS WITH MATHEMATICA

We get the following results during my stay in University of Cantabria in fall 2007. We would like to thank you Prof. Andres Iglesias for his valuable helps.

4.1. Mathematica Commands for Reachability problem

We start the definition of Petri nets with Mathematica. In the following code the set defines places and their capacities,the second one shows the transition set, the last set defines the arcs between places and transition and their capacities.

```
In[1] := a = {{{p1, 6}, {p2, 4}, {p3, 3}, {p4, 2}, {p5, 3}}, {t1, t2, t3, t4, t5, t6},
{{t1, p1, −1}, {t1, p2, 2}, {t2, p2, −1}, {t2, p3, 3}, {t3, p3, −2}, {t4, p1, −1},
{t4, p4, 1}, {t5, p4, −2}, {t5, p5, 1}, {t6, p5, −2}}};
```

This command determines the elements of the first list are all positive or less than the corresponding elements of the second list:

```
In[1]:= FeasibleList[l1_,l2_] := Union[MapThread[0 ≤ ♯1 ≤ ♯2&, {l1, l2}]]//First
In[2] := FeasibleList[{3, 5}, {3, 7}]
Out[2] = True
In[3] := FeasibleList[{1, 3}, {2, 1}]
Out[3] = False
```

In the following module, **FireInstanceAux** computes the output of firing a transition in a net (give an initial marking). The output is any integer number; even it might be negative. So, we need to check this point (done in the **FireInstance** module).

```
In[1] :=FireInstanceAux[pn_,mark_,trans_]:=
Module[{a, b, h, t = Table[0, {Length[mark]}], pl = Transpose[Part[pn, 1]]},
{a, b} = Drop[Transpose[Select[Part[pn, 3], MemberQ[♯, trans]&]], 1];
h = Position[First[pl], ♯]&/@a//Flatten; (t[[h[[♯]]]] =
b[[♯]]&/@Range[Length[a]]; v = mark + t]
```

Notes:

m: length of marking.

pl: stores the list of places and capacities. Ex: {{p1, p2}, {2, 1}}

a,b: list of places connected to transitions and weights of those connections

h: position of places in *a*

t: initially, it is comprised of zeroes. Then, it stores the ordered weights of the connections to the fired transition.

v: computes the final output.

FireInstance computes the output of firing a transition in a net (given an initial marking). The output is checked to determine whether or not such a transition is feasible.

```
In[1] := FireInstance[pn_, mark_, trans_]:=Module[{v}, If[!MemberQ[Part[pn, 2],
trans], Message[FireInstance :: "Improper transition", trans, pn],
If[FeasibleList[(v = FireInstanceAux[pn, mark, trans]), Last[Transpose[Part[pn, 1]]]],
v, Message[FireInstance :: "Enabled transition", trans]; mark]]]
In[2] := FireInstance[a, {3, 0, 0, 0, 0}, t4]
Out[2] = {2, 0, 0, 1, 0}
```

The goal now is to determine the all enabled transitions at given marking (applied as above) automatically.

```
In[1]:= ApplyInstance[pn_,mark_]:=
Module[{tr = First/@Part[pn, 3]//Union, pl = Transpose[Part[pn, 1]]},
Part[tr, Position[FeasibleList[♯, Last[pl]]&/@
(FireInstanceAux[pn, mark, ♯]&/@tr), True]//Flatten]]
In[2] := ApplyInstance[a, {1, 0, 0, 2, 0}]
Out[2] = {t1, t5}
```

This command computes the list comprised of all markings that can be obtained from a given one by applying the enabled transitions from it only once:

```
In[1] := ListFireInstance[pn_, mark_]:=FireInstance[pn, mark, ♯]&/@
ApplyInstance[pn, mark]
In[2] := ListFireInstance[a, {3, 0, 0, 0, 0}]
Out[2] = {{2, 2, 0, 0, 0}, {2, 0, 0, 1, 0}}
```

An interesting extension of this command computes the list comprised of all markings that can be obtained from a given one by applying the enabled transitions from it only once and the initial marking itself.

```
In[1] := ListFireInstanceExtended[pn_, mark_]:=Append[FireInstance[pn, mark, ♯]
&/@ApplyInstance[pn, mark], mark]
In[2] := ListFireInstanceExtended[a, {3, 0, 0, 0, 0}]
Out[2] = {{2, 2, 0, 0, 0}, {2, 0, 0, 1, 0}, {3, 0, 0, 0, 0}}
```

This auxiliar function will allow us to obtain the list of all possible markings from a given one.

```
In[1] := ListInstancesAux[pn_, x_]:=Union[Flatten[ListFireInstanceExtended[pn, ♯]
&/@x, 1]]
```

This command computes all possible markings reachable from an initial marking by applying all possible transitions as many times as needed. It is defined by using the **FixedPoint** command so that the result no longer changes over the time.

```
In[1] := ListReachableMarkings[pn_, mark_]:=
FixedPoint[ListInstancesAux[pn, ♯]&, ListFireInstanceExtended[pn, mark]]
In[2] := lm= ListReachableMarkings[a, {3, 0, 0, 0, 0} ]
Out[2] = {{0, 0, 0, 1, 1}, {0, 1, 1, 0, 1}, {0, 1, 1, 2, 0}, {0, 1, 3, 0, 1}, {0, 1, 3, 2, 0},
{0, 2, 0, 0, 1}, {0, 2, 0, 2, 0}, {0, 3, 1, 1, 0}, {0, 3, 3, 1, 0}, {0, 4, 0, 1, 0}, {1, 0, 0, 0, 1},
{1, 0, 0, 2, 0}, {1, 1, 1, 1, 0}, {1, 1, 3, 1, 0}, {1, 2, 0, 1, 0}, {1, 3, 1, 0, 0}, {1, 3, 3, 0, 0},
{1, 4, 0, 0, 0}, {2, 0, 0, 1, 0}, {2, 1, 1, 0, 0}, {2, 1, 3, 0, 0}, {2, 2, 0, 0, 0}, {3, 0, 0, 0, 0}}
```

For each of those markings, the corresponding transitions yielding them are:

```
In[3] := lt = ApplyInstance[a, ♯]&/@lm
Out[3] = {{ }, { }, {t5}, {t3}, {t3,t5}, {t2}, {t2,t5}, { }, {t3}, {t2}, {t1,t4},{t1,t5}, {t1,t4},
t1,t3,t4}, {t1,t2,t4}, {t4}, {t3,t4}, {t2,t4}, {t1,t4}, {t1,t4}, {t1,t3,t4},{t1,t2,t4}, {t1,t4}}
```

This list indicates the firable transitions for each reachable marking of the net:

```
In[4] := lmt = MapThread[List[♯1, ♯2]&, {lm, lt}]
Out[4] = {{{0,0,0,1,1}, { }}, {{0,1,1,0,1}, { }}, {{0,1,1,2,0}, {t5}}, {{0,1,3,0,1},{t3}},
{{0,1,3,2,0}, {t3,t5}}, {{0,2,0,0,1}, {t2}}, {{0,2,0,2,0}, {t2,t5}} ,{{0,3,1,1,0}, { }},
{{0,3,3,1,0}, {t3}}, {{0,4,0,1,0}, {t2}}, {{1,0,0,0,1}, {t1,t4}}, {{1,0,0,2,0}, {t1,t5}},
{{1,1,1,1,0}, {t1,t4}}, {{1,1,3,1,0}, {t1,t3,t4}}, {{1,2,0,1,0}, {t1,t2,t4}}, {{1,3,1,0,0},
```

{t4}}, {{1,3,3,0,0}, {t3,t4}}, {{1,4,0,0,0}, {t2,t4}}, {{2,0,0,1,0}, {t1,t4}}, {{2,1,1,0,0}, {t1,t4}}, {{2,1,3,0,0}, {t1,t3,t4}}, {{2,2,0,0,0}, {t1,t2,t4}}, {{3,0,0,0,0}, {t1,t4}}

4.2. Incidence Matrix with Mathematica

First module is for **Pure** Petri nets.

In[1] := a = {{{p1, 2}, {p2, 1}}, {t1, t2, t3, t4}, {{t1, p1, 1}, {t3, p2, 1}, {t2, p1, 1}, {t4, p2, -1}}};

In[2] := IncMatrixPetri[*pn*_] :=

Module[{*posp* = *Position*[*First*[♯]&/@*First*[*pn*], ♯]&/@*Part*[*Flatten*[*Last*[*pn*], {2}], 2] //*Flatten*, *post* = *Position*[*Part*[*pn*, 2], ♯]&/@*First*[*Flatten*[*Last*[*pn*], {2}]]//*Flatten*,

posw = Last[Flatten[Last[pn], {2}]]},

AA = Table[0, {Length[Part[pn, 2]]}, {Length[First[pn]]}];

For[s = 0; m = 0; i = 0, i ¡ Length[posw], i++; = Part[post, i]; m = Part[posp, i];

AA[[s, m]] = Part[posw, i]]; AA]

In[3] := k = IncMatrixPetri[a]

Out[3] = {{1, 0}, {1, 0}, {0, 1}, {0, -1}}

In[4] := MatrixRank[k]

Out[4] = 2

In[5] := z = Transpose[k]

Out[5] = {{1, 1, 0, 0}, {0, 0, 1, -1}}

In[6] := MatrixForm[k]

Out[6] := $\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$

In[7] := MatrixForm[z]

Out[7] = $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$

The followings are for **Not Pure** nets.

In[1] := a = {{{p1, 2}, {p2, 1}}, {t1, t2, t3, t4}, {{t1, p1, 1}, {t3, p2, 1}, {t2, p1, 1}, {t4, p2, -1}, {t1, p1, -2}, {t2, p1, -3}}}

```
In[2] := IncidenceMatrixPetri[pn_] :=
Module[{s, m, EE = Table[0, {Length[Part[pn, 2]]}, {Length[First[♯]&/@First[pn]]}],
FF = Table[0, {Length[Part[pn, 2]]}, {Length[First[♯]&/@First[pn]]}],
posp = Position[First[♯]&/@First[pn], ♯]&/@Part[Flatten[Last[pn], {2}], 2]//Flatten,
post = Position[Part[pn, 2], ♯]&/@First[Flatten[Last[pn], {2}]]//Flatten,
posw = Last[Flatten[Last[pn], {2}]]},
For[s = 0; m = 0; i = 0, i < Length[posw], i + +;
s = Part[post, i]; m = Part[posp, i];
If[Part[posw, i] < 0, EE[[s, m]] = Part[posw, i], FF[[s, m]] = Part[posw, i]]];
EE + FF]
In[3] := k=IncidenceMatrixPetri[a]
Out[3] = {{-1, 0}, {-2, 0}, {0, 1}, {0, -1}}
In[4] := MatrixForm[k]
```

Out[4] =
$$\begin{pmatrix} -1 & 0 \\ -2 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$$

CHAPTER 5

MODELING OF HASHIMOTO’S THYROIDITIS ON PETRI NETS

In this chapter, we studied on the modelling of Hashimato’s Thyroiditis. We designed three different models for healthy, sick and healthy - on treatment situations.

5.1. Hashimato’s Thyroiditis

Hashimoto’s thyroiditis or chronic autoimmune thyroiditis is the most common thyroid disease and a common cause of goiter. It was described by Dr. Hashimoto in 1912 as a lymphocytic infiltration of the thyroid gland with goiter. It has since been well established as an autoimmune disease and circulating antibodies against the thyroid can be found in the majority of cases. Like most thyroid diseases, it is much more common in women than in men, with a sex ratio of 6 or 7 to 1.

Hashimoto’s thyroiditis is characterized by the production of immune cells and autoantibodies by the body’s immune system, which can damage thyroid cells and compromise their ability to make thyroid hormone. Hypothyroidism occurs if the amount of thyroid hormone which can be produced is not enough for the body’s needs. The thyroid gland may also enlarge in some patients, forming a goiter. Hashimoto’s thyroiditis results from a malfunction in the immune system. When working properly, the immune system is designed to protect the body against invaders, such as bacteria, viruses, and other foreign substances. The immune system of someone with Hashimoto’s thyroiditis mistakenly recognizes normal thyroid cells as foreign tissue, and it produces antibodies that may destroy these cells. Although various environmental factors have been studied, none have been positively proven to be the cause of Hashimoto’s thyroiditis.

Increased TSH (thyroid-stimulating hormone) level in the blood is the most accurate indicator of hypothyroidism. TSH is produced by another gland, the pituitary, which is located in the center of the head behind the nose. The level of TSH rises dramatically when the thyroid gland even slightly underproduces thyroid hormone, so a normal level of TSH reliably excludes hypothyroidism in patients with normal pituitary function.

Free T4 (thyroxine) - the active thyroid hormone in the blood. A low level of free T4 is consistent with thyroid hormone deficiency. However, free T4 values in the ″normal range″ may actually represent thyroid hormone deficiency in a particular patient, since a high level of TSH stimulation may keep the free T4 levels ″within normal limits″ for many years (AACE 1970).

During the diagnosis, the level of Free T3 (triiodothyronine) in the blood is tested. Cause of the high level TSH, Free T3 level is commonly low. When the thyroid gland is damaged, it can't produce thyroid hormones like T4 and T3 efficiently.

For treatment; in theory, giving T4 alone should be enough, since the body naturally converts some T4 to the more potent T3. However, some studies suggest that patients report feeling better, symptomatically, when given T3 together with their T4. More studies are needed on this at present, but it might be worth a try in some patients. However, taking T3 alone is not recommended, because T3 has a much shorter half-life than T4.

5.2. Modeling of Hashimato's Thyroiditis on Petri Nets

In modeling of Hashimoto's Thyroiditis on Petri Nets, Colored Petri nets are more acceptable for detailed models. In this thesis, we focused on creating a compact models to show the behavior of disease.

5.2.1. Petri net Model of Healthy Human

We used places to show hormones and the organs. Transitions define the connections between the organs and the hormones that are produced by that organs.

In healthy human body, TSH alerts the Thyroid gland, when it gets this alert, it produces thyroid hormones. The rate of T4/T3 should be 13. In the model in Figure 5.1., the place p_1 sends the TSH to Thyroid gland to make it produces T3 and T4. t_2 defines that the Thyroid gland works well and T3 and T4 are produced efficiently. In healthy human body, T3 is produced by T4 occasionally. The transition t_3 shows the production of T3 by T4. The path from place p_3 to the transition t_6 (p_3, t_5, p_6, t_6) is for the path of Free-T3 in the blood. The path from place p_4 to the transition t_6 (p_4, t_4, p_5, t_6) has the same duty for Free-T4. t_6 sends a message to p_7 (Hypothalamus) that means, the levels of T3 and T4 are normal and Thyroid gland works well. t_7 shows that Hypothalamus is alerted and it produces TRH. t_8 sends a message to p_9 (TRH) from p_8 (PG) and TRH alerts PG to produce TSH. The last transition t_9 sends a message to p_1 from p_9 (Pituiatory Gland) and TSH is produced again.

$\{1,0,0,0,0,0,0,0,0\}$ is the initial marking. One token is enough for place p_1 to fire the system.

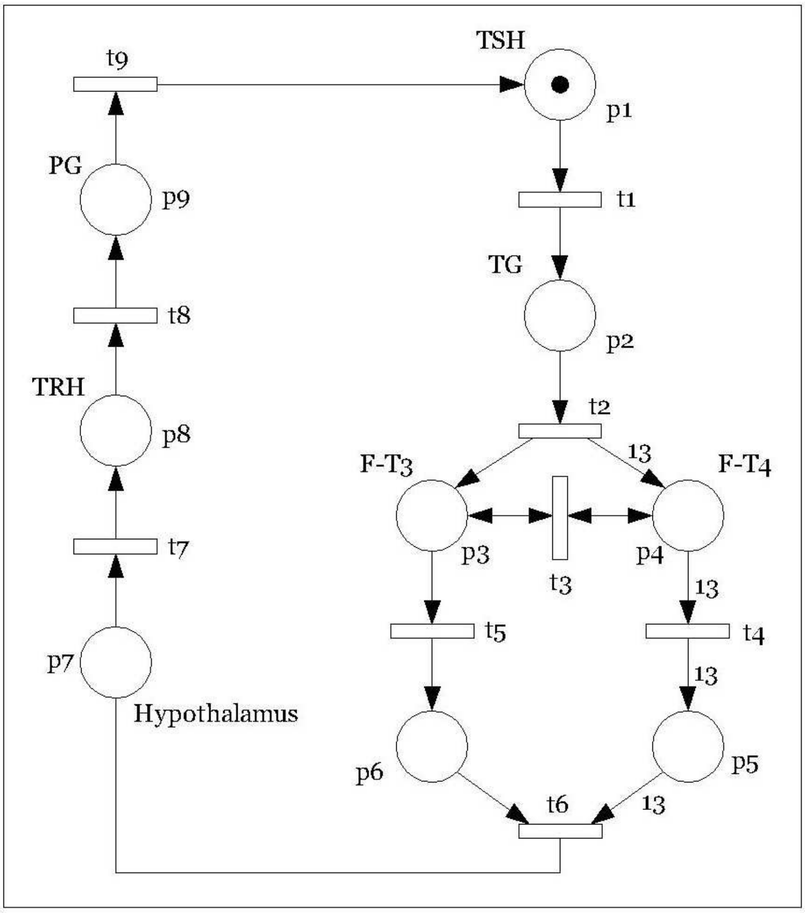


Figure 5.1. A Petri net model for healthy human

The set of places: (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9)

p_1 : The TSH hormone.

p_2 : Thyroid gland.

p_3 : Free-Triiodothyronine (T3-Thyroid hormone).

p_4 : Free-Thyroxine (T4-Thyroid hormone).

p_5 : The value of Free-T4 in the blood.

p_6 : The value of Free-T3 in the blood.

p_7 : The Hypothalamus.

p_8 : The TRH hormone.

p_9 : The Pituiatory Gland.

The set of transitions: $(t_1,\ t_2,\ t_3,\ t_4,\ t_5,\ t_6,\ t_7,\ t_8,\ t_9)$

t_1 : It ensures that TSH can send a message to Thyroid gland to produce thyroid hormones.

t_2 : It produces T3 and T4.

t_3 : T3 is produced by T4.

t_4 : It determines the value of Free-T4 in the blood.

t_5 : It determines the value of Free-T3 in the blood.

t_6 : It sends a message to Hypothalamus that the levels of Free-T3 and Free-T4 are normal or not.

t_7 : It sends a message from Hypothalamus to produce TRH.

t_8 : It alerts the Pituiatory Gland to produce TSH.

t_9 : It sends a message from Pituiatory Gland to produce TSH.

5.2.2. Petri net Model of Sick Human

This model describes the sick human body system. The reasons of Hashimoto's Thyroiditis are unknown. The human body produces antibodies to thyroid cells cause of these unknown reasons. If there exists these reasons t_1 fires and the place p_2 has tokens (t_1 is a source transition). That means antibodies are produced. Besides of the antibodies p_1 (TSH) tries to alert thyroid gland normally. But, in this model when t_4 gives tokens to p_3, t_3 or t_5 can fire and thyroid gland (TG) loses some of its' tokens because of antibodies. So thyroid gland can not produces T3 and T4 enough.

The initial marking is $\{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$. At the beginning, one token in p_1 is enough to fire the system. In Hashimoto's Thyroiditis, value of TSH is high while values of T3 and T4 are low. To show that higher TSH value, we used t_2 to explain that if antibodies exist, TG is damaged and value of TSH is high, we give two to the weight of arc between t_2 and p_1.

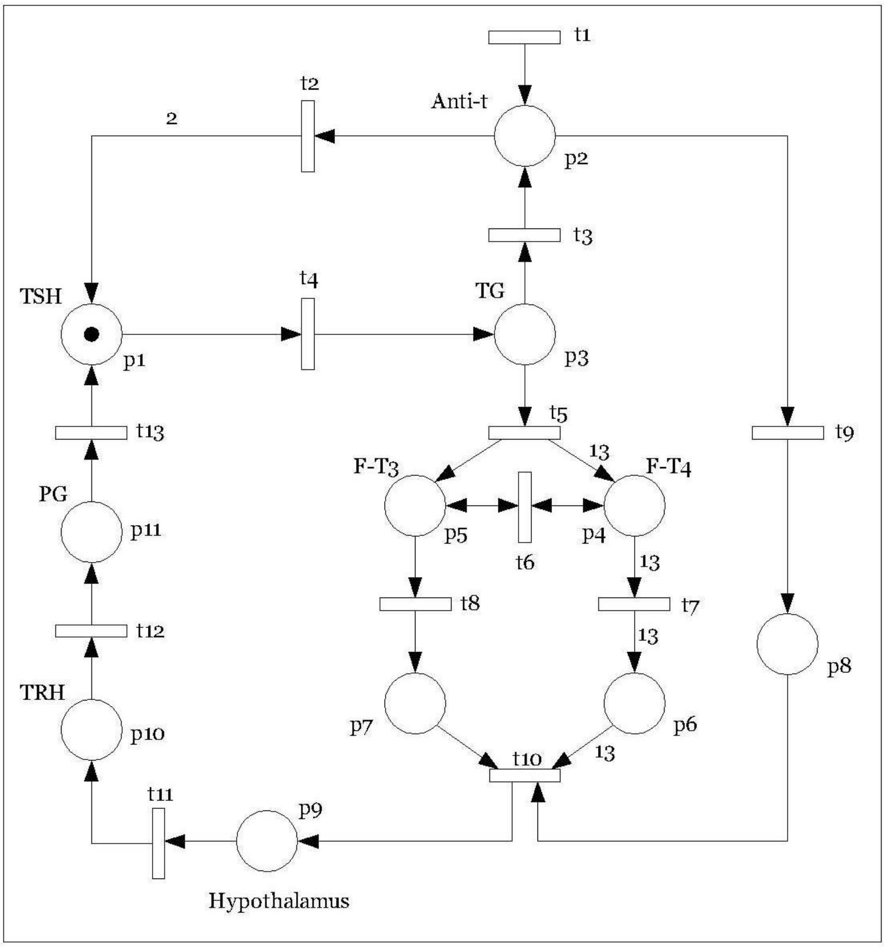


Figure 5.2. A Petri net model for sick human

The set of places: (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11})

p_1 : The TSH hormone.

p_2 : Antibodies.

p_3 : The Thyroid gland.

p_4 : Free-Thyroxine (T4-Thyroid hormone).

p_5 : Free-Triiodothyronine (T3-Thyroid hormone).

p_6 : The value of Free-T4 in the blood.

p_7 : The value of Free-T3 in the blood.

p_8 : The value of antibodies in the blood.

p_9 : The Hypothalamus.

p_{10} : The TRH hormone.

p_{11} : The Pituiatory Gland.

The set of transitions: (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13})

t_1 : It defines the unknown reasons and it makes the human body produces antibodies to the thyroid cells.

t_2 : It sends the effects of antibodies and makes the value of TSH high.

t_3 : It defines the damage in the Thyroid gland because of the antibodies.

t_4 : It ensures that TSH can send a message to Thyroid gland to produce thyroid hormones.

t_5 : It produces T3 and T4.

t_6 : T3 is produced by T4.

t_7 : It determines the value of Free-T4 in the blood.

t_8 : It determines the value of Free-T3 in the blood.

t_9 : It sends a message to p_8 that antibodies are produced.

t_{10} : It sends a message to Hypothalamus that the levels of Free-T3 and Free-T4 are normal or not.

t_{11} : It sends a message from Hypothalamus to produce TRH.

t_{12} : It alerts the Pituiatory Gland to produce TSH.

t_{13} : It sends a message from Pituiatory Gland to produce TSH.

5.2.3. Petri net Model of Healthy-Sick and on Treatment

This model combines three different situations. System starts with t_1, after t_1 gives a token to p_2 (TG), there are two different firing sequences. One of them is for healthy human. This sequence is almost same with first model in Figure 5.1. The other firing sequence includes treatment. If t_3 fires, that means; Thyroid gland is damaged and thyroid hormones aren't produced efficiently. For treatment we used place p_5 as a medicine. When t_5 fires, T3 and T4 hormones are taken as a medicine.

The set of places: (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10})

p_1 : The TSH hormone.

p_2 : The Thyroid gland.

p_3 : Free-Triiodothyronine (T3-Thyroid hormone).

p_4 : Free-Thyroxine (T4-Thyroid hormone).

p_5 : It defines the treatment.

p_6 : The value of Free-T4 in the blood.

p_7 : The value of Free-T3 in the blood.

p_8 : The Hypothalamus.

p_9 : The TRH hormone.

p_{10} : The Pituiatory Gland.

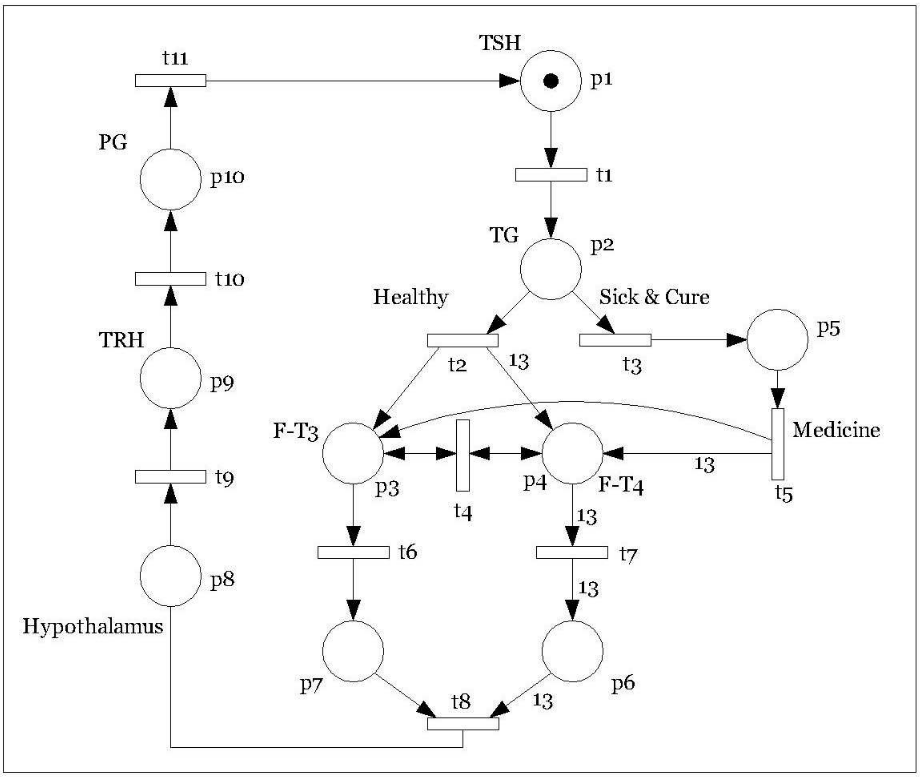


Figure 5.3. A Petri net model for healthy-on treatment

The set of transitions: (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11})

t_1 : It ensures that TSH can send a message to Thyroid gland to produce thyroid hormones.

t_2 : It defines that the body works well and TG can produce T3 and T4.

t_3 : It defines the damage in the Thyroid gland because of the antibodies and it sends a message to the treatment place.

t_4 : T3 is produced by T4.

t_5 : It produces T3 and T4.

t_6 : It determines the value of Free-T3 in the blood.

t_7 : It determines the value of Free-T4 in the blood.

t_8 : It sends a message to Hypothalamus that the levels of Free-T3 and Free-T4 are normal or not.

t_9 : It sends a message from Hypothalamus to produce TRH.

t_{10} : It alerts the Pituiatory Gland to produce TSH.

t_{11} : It sends a message from Pituiatory Gland to produce TSH.

We used ″Thomas Braunl′s S/T Petri-Net Simulation System″ and ″Luis Alejandro Cortes′ SimPRES″ to test these models. They can be found at (Informatik 2005) .

CHAPTER 6

CONCLUSION

In this thesis, we gave the brief review of Petri Nets and their applications. In the last two chapter, we gave an original biological application models and wrote down our modules for reachability and incidence matrix in Petri Nets with Mathematica. Similarly, many biological and medical problems can be modeled by Petri Nets. We used a classical Petri Nets for modeling of basic behavior of Hashimato's Thyroiditis. Timed-colored and Stochastic Petri Nets are more acceptable for having more efficient models for Hashimato's Thyroiditis. We can analyze the disease and have efficient results for human body with using detailed Timed-colored Petri Nets models.

REFERENCES

AACE. http://www.aace.com.

Ajmane M. M., Balbo G., and Chiola G., eds. 1991. An introduction to generalized stochastic Petri nets . *Microelectron Reliab* 31 (4): 699 - 725.

Barjis J. and Barjis I. 1999. Formalization of the Protein Production by Means of Petri Nets. *IEEE*.

BRENDA. http://www.brenda.uni-koeln.de:80/.

Chen M. and Hofestadt R. 2006. A medical bioinformatics approach for metabolic disorders: biomedical data prediction, modeling, and systematic analysis . *J. Biomed. Inform* 39 : 147 - 159.

Dennis B. J. 1970. *Record Project MAC Conference. Concurrent Systems and Parallel Computaion* 199.

Fantia M. P. and Giuab A., eds. 2006. Monitor design for colored Petri nets: An application to deadlock prevention in railway networks . *Control Engineering Practice* 14 : 1231 - 1247.

Genrich H. and Kuffner R. eds. 2001. Executable Petri net models for the analysis of metabolic pathways . *Int J STTT* 3 (4) : 394 - 404.

Goss P. and Peccoud J. 1999. Analysis of the stabilizing effect of Rom on the genetic network controllin ColE1 plasmid replication. *Pac Symp Biocomput* 4 : 6576.

Goss P. and Peccoud J. 1988. Quantitative modelling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc.Nat. Acad. Sci* 95 : 6750 - 6755.

Hardy S. and Robillard N. P. 2004. Modeling and simulation of moleculer biology systems using Petri Nets: Modeling goals of various approaches. *Journal of Bioinformatics and Computational Biology* 2 (4) : 619 - 637.

Heiner M., eds. 2004. Model validation of biological pathways using Petri nets demonstrated for apoptosis. *Biosystems* 75 : 15 - 28.

Hohn E. F. 1958. *Elementary Matrix Algebra*. New York : Macmillian.

INFORMATIK. http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/java/.

Jensen K. 1992. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. *Monographs on Theoritical Computer Science*. SpringerVerlag.

Materi W. and Wishart D. S. 2007. Computational systems biology in drug discovery and development: methods and applications. *Drug Discovery Today* 12 (7/8).

Matsuno H. 2003. Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biol* 3 : 389 - 404.

Mayo M. 2005. Learning Petri net models of non-linear gene interactions . *BioSystems* 82 : 74 - 82.

Molloy K. M. 1982. Performance analysis using stochastic Petri nets . *IEEE Trans. Computers* C-31 (9) : 913 - 917.

Moore H. J. and Hahn W. L. 2003. Petri net modeling of high-order genetic systems using grammatical evolution . *BioSystems* 72 : 177 - 186.

Murata T. 1989. Petri Nets: Properties, Analysis and Applications . *Proceedings of the IEEE* 20 (4).

Murata T. 1977. State equation, controllability, and maximal matchings for Petri nets . *IEEE Trans. Automat. Contr.* Ac-22 (3) : 412 - 416.

Petri A. C. 1962. Kommunikation mit Automaten , Bonn: Institut fr Instrumentelle Mathematik, Schriften des IIM.3 (1962), Also,(English translation) Communication with Automata, New York: Griffiss Air Force Base. Tech. Rep. RADC-TR. 1 (1).

Reddy V. N. and Mavrovouniotis M. L. eds. 1993. Petri net representation in metabolic pathways . *Proc. Int.Conf. Intell. Syst. Mol. Biol.* 1 : 328 - 336.

Reddy V. N. and Mavrovouniotis M. L., eds. 1996. Qualitative analysis of bio chemical reaction systems. *Comput Biol Med* 26 (1) : 9 - 24.

Ritchie M., Hahn L. and Roodi N., eds. 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am. J. Hum. Genet. 69 : 138 - 147.

Sackmann A., eds. 2006. Application of Petri net-based analysis techniques to signal transduction pathways . *BMC Bioinformatics* 7 : 482.

Ye Z. and Zhou J., eds. 2003. On reachability graphs of Petri nets . *Computers and Electrical Engineering* 29 : 263 - 272.

Zhang Z. and Hong F.,eds. 2006. Modeling Chinese Wall Policy Using Colored Petri Nets . *Proceedings of The Sixth IEEE International Conference on Computer and Information Technology (CIT'06).*