DESIGNING AND ANALYSING A PETRI NET MODEL OF A SMALL HOSPITAL IN TIMES OF COVID.

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE
OF

MASTER OF SCIENCE(M.Sc.)

IN

MATHEMATICS

Submitted By:

Shivam gupta , Deepika chaudhary

(2K20/MSCMAT/28) (2K20/MSCMAT/09)

Under the supervision of

PROF. SANGITA KANSAL

DEPARTMENT OF APPLIED MATHEMATICS DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

Contents

ii

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We, Shivam gupta and Deepika chaudhary, Roll No.s 2K20/MSCMAT/28, 2K20/MSCMAT/09 of Master in Science (Mathematics), hereby declare that the project dissertation titled

DESIGNING AND ANALYSING A PETRI NET MODEL OF A SMALL HOSPITAL

IN TIMES OF COVID which is submitted by us to the Department of Applied Mathematics, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Science in Mathematics, is original and not copied from any source without proper citation.This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi   Shivam Gupta(2K19/MSCMAT/28)

Date: May 5, 2022

Deepika Chaudhary (2K19/MSCMAT/09)

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled DESIGNING AND ANALYSING A PETRI NET MODEL OF A SMALL HOSPITAL IN TIMES OF COVID which has been submitted by Shivam Gupta and Deepika Chaudhary , Roll No.s 2K20/MSCMAT/28 and 2K20/MSCMAT/09 of Department of Applied Mathematics, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Masters of Science in Mathematics, is a record of the project work carried out by the students under my supervision.

Place: Delhi

Prof. Sangita Kansal

Supervisor

Date: May 5, 2022

ACKNOWLEDGEMENT

We take the opportunity to express gratitude to our supervisor, *Dr. Sangita Kansal*, Professor, Department of Applied Mathematics, Delhi Technological University, for her constant encouragement, co-operation and valuable guidance throughout this project dissertation work. We are obliged to her for providing facilities to carry out the work smoothly.

Further, we extend our thanks to all the teachers of the Department for imparting in us the knowledge and understanding of mathematics for attaining better results, without whose kind efforts, we would not have reached this stage.

Shivam gupta (2k20/MSCMAT/28)

Deepika Chaudhary(2k20/MSCMAT/09)

ABSTRACT

The notion of Petri Net, formerly developed by Carl Adam Petri, is useful for modeling and analyzing a system's behavior. Petri Net is a graphical tool, defined as a bipartite graph consisting of two types of nodes, places (conditions) and transitions (activities).
In general, a discrete event dynamic system consists of activities that can model the system by consecutively listing its states; prior and after to the occurrence of these activities.

In this paper, a Petri Net model for a small hospital has been proposed, keeping in view the spread of the COVID-19 virus. Emphasis has been given to practicing social distancing and allowing a minimum number of people together at any stage. This model, which accounts for three beds (which can be occupied by new patients subsequently) and one staff
and their respective activities and a doctor for treatment, has been interpreted as a dynamic system. Furthermore, the model's design has been validated structurally and behaviorally using techniques from Linear Algebra, transitive matrices, and transition vectors. And defind the token flow with the help of petri net formulas and inference of properties like cyclic/acyclic nature, conflict, concurrency, boundedness, conservativeness, safeness, liveness, and deadlock has been interpreted physically with the proposed model

Chapter 1

Petri Nets : Overview

1.1 Introduction

Any system, in general, consists of a number of activities which can model the system by consecutively listing its states, prior and after to the occurrence of the activities. An *activity* is thus responsible for bringing change in the system from one state to another.

A simple example can be that of a switch where the activities including turning the switch off or on can alter the state of the system, i.e. breaking the circuit or allowing it to flow.

We can graphically represent all such state-transitions as state-transition diagrams. In mathematical modelling systems, there exist various such techniques including networking, queuing etc.

One of the eminent mathematical graphical tools used for modeling discrete event dynamic systems is a *Petri Net (PN)*; formally defined as a bipartite graph consisting of two types of nodes, the places, and the transitions.

1.2 Development of Petri Nets

Petri nets, originally developed by Carl Adam, formally specifies a model and helps to further derive properties and relations.

It is useful for modelling and analysing the functionality (behaviour) of systems, like computer networks, manufacturing units, scheduling areas etc.

1.3 Places and Transitions

Definition 1.3.1. (*Places*)

The places refer to certain set of conditions that are to be satisfied. In simple words, they can be thought of as a box which can hold something in it. These are denoted using circles ◦ in the PN structures.

Definition 1.3.2. (*Transitions*)
The transitions are the events or activities that occur and lead to the change in the state of the system. These are denoted using a vertical line | or a rectangular bar. The places and transitions are connected via *directed edges* or *arcs*.
1.4 Conditions

We have seen that we can bifurcate a Petri Net system into-

1. Events (Transitions)
2. Conditions (Places)

The occurrence of an event is determined when certain conditions hold valid which are known as *pre-conditions*. These eventually lead to cause other conditions known as *post-conditions.*

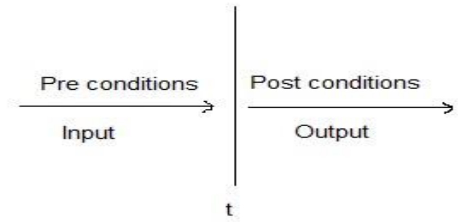


Figure 1.1: Conditions of the Petri Net

1.5 Tokens

Present in a system are some basic entities called *tokens* which get created and destroyed in the places (conditions) and can travel in a system under certain parameters that can change the state of the system.

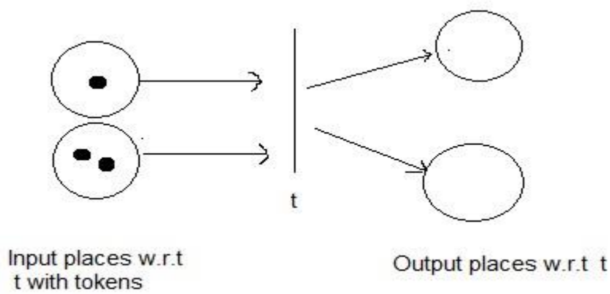


Figure 1.2: Tokens in a Petri Net

With respect to the above simple Petri Net structure, we can deduce that for the transition *t*, there exist total four places P_1, P_2, P_3, P_4. Further, it is evident that these places and the transition are connected respectively by arcs or the directed edges.

The tokens are labelled at the input places P_1 and P_2 for t, where P_1 has a single token and P_2 has 2 tokens.

Chapter 2

Petri Net Structure

2.1 Introduction to Bags

1. *Bag theory* is an extension of the set theory, which is a collection of elements from a certain predefined domain. However, multiple existence of elements is possible in bags unlike that in sets.

2. For any arbitrary element, $b \in B$ bag, we denote the number of times b occurs in B by $\#(b,B)$.

Remark 2.1.1. The concept of bag theory reduces down to set theory when the condition $0 \leq \#(b,B) \leq 1$ holds.
We can consider the following example for an illustrative comparison.

Example 2.1.2. *Consider a pre-defined domain as the S=*$\{w, x, y, z\}$.
Then, $B_1 = \{w,x,y\}$ *is a bag, also a set. Also,* $B_2 = \{w,x,w,x\}$ *and* $B_3 = \{w,w,x,x\}$ *are the same bags irrespective of the position of the elements where* $\#(w,B_2) = \#(w,B_3) = 2$ *and* $\#(x,B_2) = \#(x,B_3) = 2$.

2.2 Structural Description

A Petri Net structure is composed of four parts and is written as a four tupple,

$$PN = (P,T,I,O), \qquad \text{where}$$

P = *set of all places.*
T = *set of all transitions.*
I = *Matrix that explains the association of input places and the transitions.* O = *Matrix that explains the association of output places and the transitions*

i.e. for a PN consisting of say, M-places and N- transitions,

$$P = \{p_1,p_2,\cdots,p_M\}$$

$$T = \{t_1,t_2,\cdots,t_N\}$$

Matrices I,O can have the values $a_{ij} = 0$ *or* 1 *where we construct I,O as, say*

$$I = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & \\ 0 & 1 & 1 & \cdots & 1 & \\ \vdots & 1 & \cdots & \cdots & 0 & \\ 1 & 1 & 0 & \cdots & \cdots & 1 \end{pmatrix}$$

where the column vectors correspond to the places and the row vectors correspond to the transitions. Here,

$$a_{ij} = \begin{cases} 1 & P_i \text{ is an input place for transition } t_j \\ 0 & P_i \text{ is not an input place for transition } t_j \end{cases}$$

Similarly, the matrix O can be constructed and thus be interpreted easily.

Remark 2.2.1. The set of places and the set of transitions are disjoint; $P \cap T = \varphi$.

Remark 2.2.2. The *input function* is defined as $I : T \to P^{\infty}$ and the *output function* is defined as $O : T \to P^{\infty}$ where T represents the set of transitions and P^{∞} denotes the bag of the places.

2.3 Example of a Petri Net Structure

We have a general Petri Net structure defined as $PN = \{P,T,I,O\}$; let us now consider that for a particular PN structure, where $P = \{p_1,p_2,p_3,p_4,p_5,p_6\}$ and $T = \{t_1,t_2,t_3,t_4,t_5\}$ where $I : T \to P^{\infty}$ and $O : T \to P^{\infty}$.

Let us be given the defined input and output functions as

$I(t_1) = \{p_1\}$ $\quad O(t_1) = \{p_2,p_3\}$

$I(t_2) = \{p_3\}$ $\quad O(t_2) = \{p_3,p_5,p_5\}$

$I(t_3) = \{p_2,p_3\}$ $\quad O(t_3) = \{p_2,p_4\}$

$I(t_4) = \{p_4,p_5,p_5,p_5\}$ $\quad O(t_4) = \{p_4\}$

$I(t_5) = \{p_2\}$ $\quad O(t_5) = \{p_6\}$

The input and the output functions can be extended as $I : P \to T^{\infty}$ and $O : P \to T^{\infty}$ such that $\#(t_j,I(p_i)) = \#(p_i,O(t_j))$ and $\#(t_j,O(p_i)) = \#(p_i,I(t_j))$. The extended input and output functions are:

$I(p_1) = \{\}$ $\quad O(p_1) = \{t_1\}$

$I(p_2) = \{t_1,t_3\}$ $\quad O(p_2) = \{p_3,p_5\}$

$I(p_3) = \{t_2,t_2\}$ $\quad O(p_3) = \{p_2,p_3\}$

$I(p_4) = \{p_3,p_4\}$ $\quad O(p_4) = \{t_4\}$

$I(p_5) = \{t_2\}$ $\quad O(p_5) = \{t_4,t_4,t_4\}$

$I(p_6) = \{t_5\}$ $\quad O(p_6) = \{\}$

For the above, Petri Net structure can be drawn as below

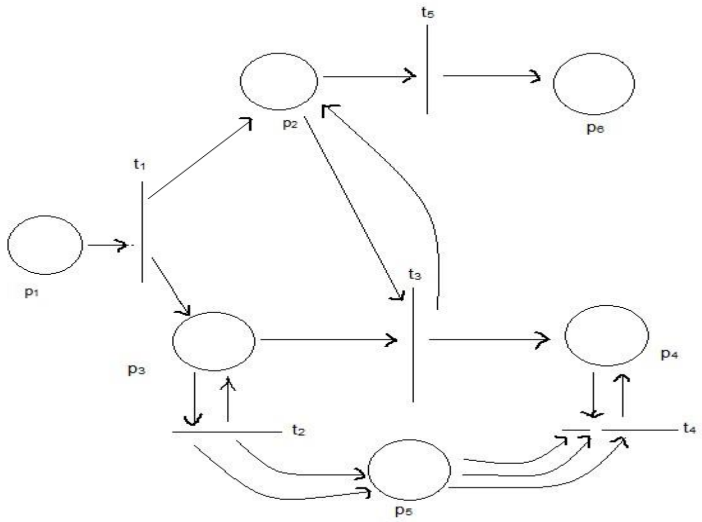


Figure 2.1: Structure of the Petri Net

2.4 Dual of a Petri Net

Since, both the vertex sets V_1,V_2 can be either of the two, places or the transitions, thus a Petri Net can be accordingly defined, with a resulting interchanged sets of places and transitions.

For a defined given Petri Net, $PN = (P,T,I,O)$, the *dual* of the Petri Net PN denoted by $\overline{PN} = (T,P,I,O)$.

2.5 Marking of a Petri Net

A *marking* of a Petri Net PN, at a certain given state t is the assignment of the tokens to the set of places. It is denoted by $M_t = M_1,M_2,...,M_m$ where M_i gives the number of tokens that are available at the place p_i at a certain state t.

A marking M is a function defined from P, the set of all places to the non-negative integers i.e., $M : P \rightarrow Z^+$, where clearly, $M(p_i) = M_i$. The marking at initial state (at t = 0) is called the initial marking $M_0 : P \rightarrow Z^+$. A marked Petri Net PN w.r.t M_0 is a 5-tuple structure where $PN = (P,T,I,O,M_0)$.

It is obvious to realise that the number of tokens which can be assigned to any place in a PN is not bounded, and thus, there are significantly infinite many number of markings possible for the PN.

2.6 Transition enabling and firing

Any transition, say t_j in a system, is enabled and can fire with one or multiple input places; if the number of tokens in all the input places is at least equal to the multiplicity of all the input arcs for t_j of those places respectively. We also call this the *triggering* of t_j. When t_j in a system triggers, a token gets deleted from its input places and eventually gets created in the respective output places,

i.e., a transition t in a marked Petri Net having marking M gets enabled to fire, if for all $p_i \in P$, i = 1 to m and

$$M(p_i) \geq \#(p_i, I(t_j)).$$

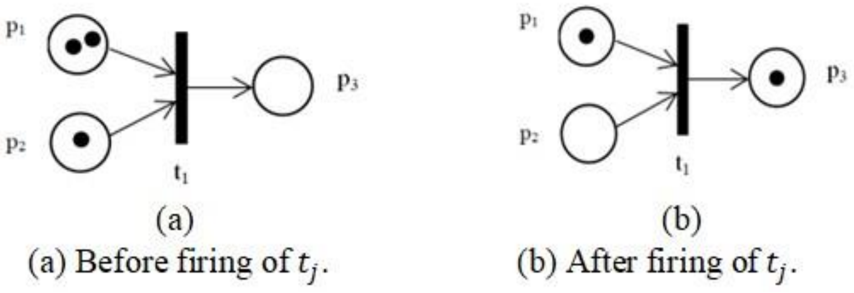


Figure 2.2: Transition enabling in a Petri Net

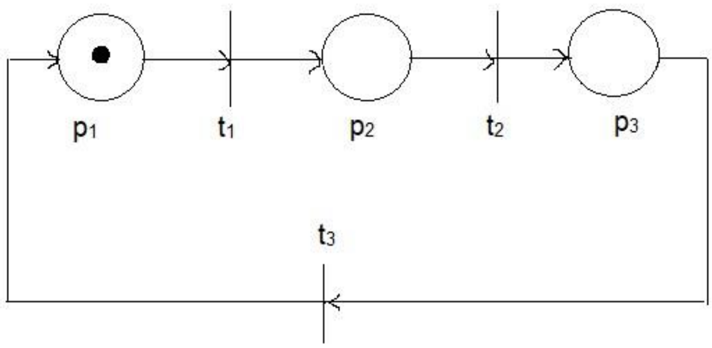


Figure 2.3: Transition enabling in a Petri Net

At time $t = t_0$ (initial time), let us have $M(t_0)$ = (1,0,0). Correspondingly,

$$I = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$O = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Since token is in p_1 and $M(p_1) \geq \#(p_i,I(t_j))$, j=1,2,3, thus t_1 can fire. A transition $t_j \in T$ in a marked PN with a marking M might be enabled to fire. Firing an enabled transition t_j will result in a new marking M defined by

$$M^0(p_i) = M(p_i) - \#(p_i,I(t_j)) + \#(p_i,O(t_j)).$$

It must be noted that transition firing can be in progress until there exists at least one enabled transition i.e. there exists one token in each input place for a transition. When there is no enabled transition, the execution *halts*.

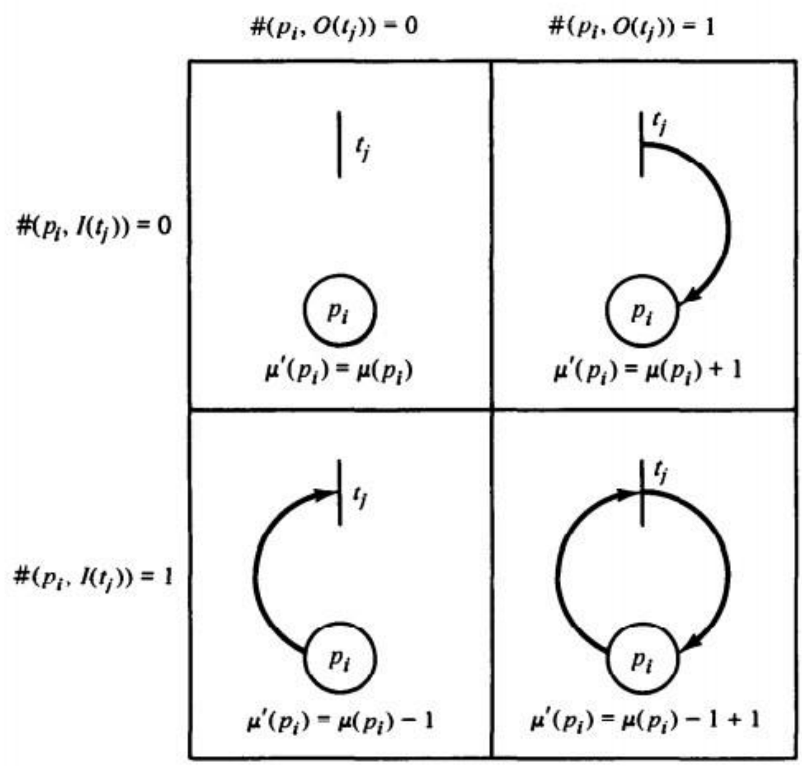


Figure 2.4: Illustration of the change in marking in a place when a transition fires.

Fig 2.4 is the illustration of how a marking of a place changes when a transition t_j is fired.

Consider a marked PN drawn below which shall help us illustrate the firing rules, where we have transitions,say, t_1,t_3,t_4 are enabled.

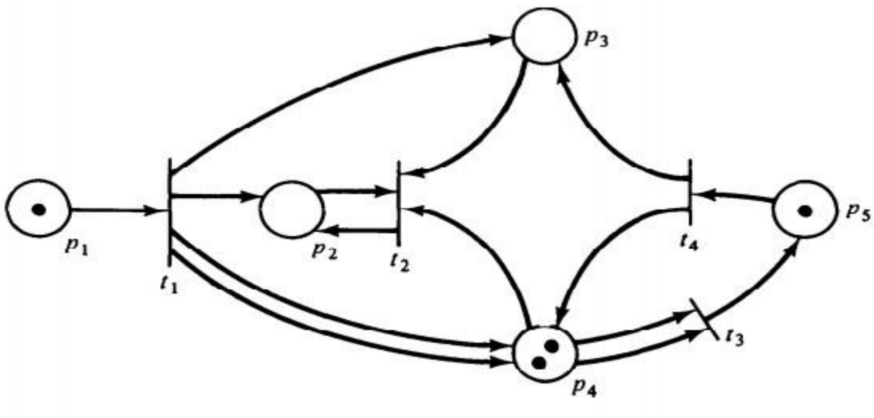


Figure 2.5: Transitions t_1,t_3,t_4 are enabled.

When a transition t_j of a system occurs or triggers, a token gets removed from all the input places and eventually gets added to the respective output places.
What must be noted here is that it is not necessary for the number of the input places to be equal to the number of output places w.r.t the triggered transition.

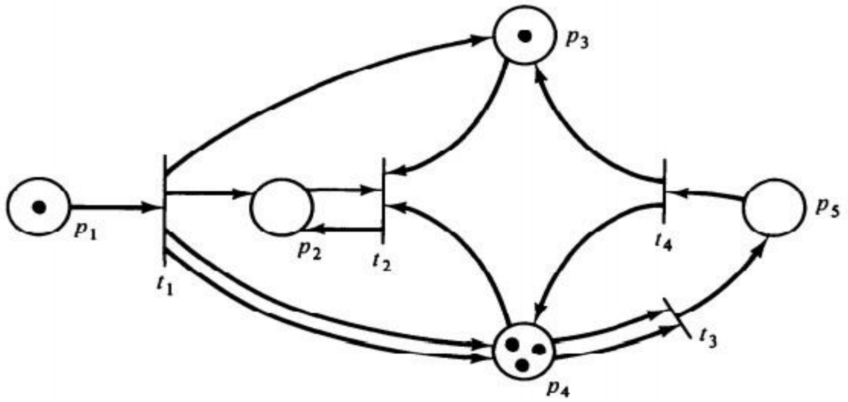


Figure 2.6: Transition t_4 fires

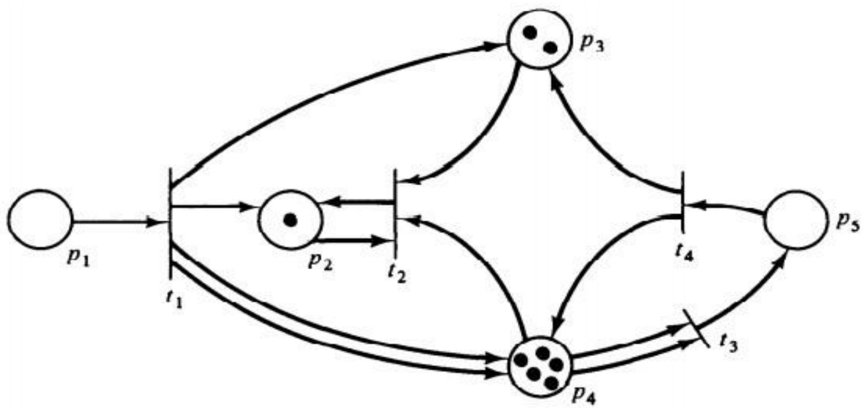


Figure 2.7: Transition t_1 fires

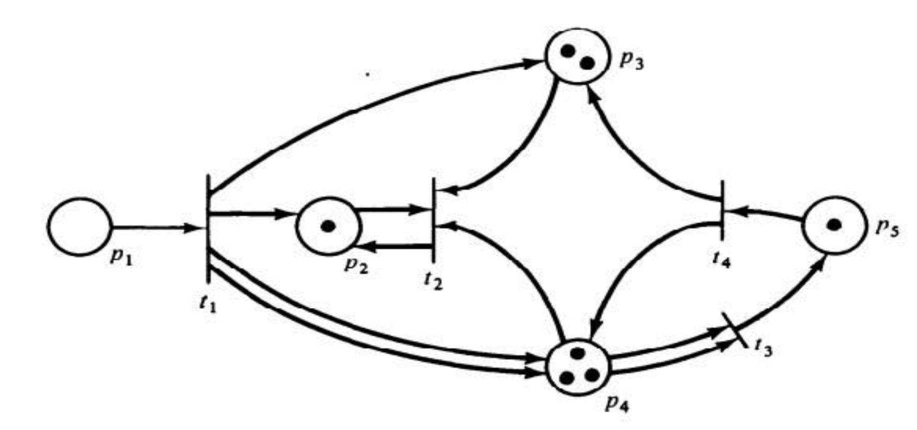


Figure 2.8: Transition t_3 fires

Chapter 3

Petri Net State Space and Components

3.1 Petri Net State Space

We define the *state* of a Petri Net by the corresponding markings at that time. The firing of a transition in a Petri Net represents an alteration in the state of the PN by changing the marking.

For a marking $M : P \to Z^+$ of a PN where $M(p_i) = M_i$ where $P = \{p_1, p_2, \cdots, p_m\}$ i.e. a Petri Net with m-places has a *state space*, or, a set of all markings which shall be equal to N^m.

The change in the state that occurs by firing an enabled transition is defined using a *change function* φ, which is known as the *next-state function.*

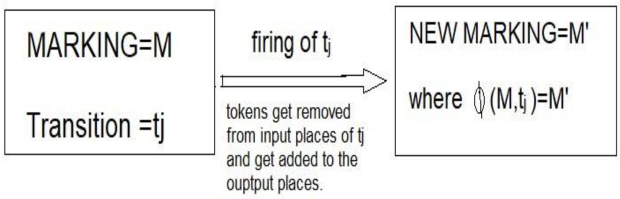


Figure 3.1: Petri Net marking.

Formally, if we define the *next-state function*, $\varphi : N^m \times T \to N^m$ for a Petri Net $PN = (P,T,I,O)$ with the marking M and a transition $t_j \in T$ is defined if and only if $M(p_i) \geq \#(p_i, I(t_j))$ for all $p_i \in P$. If $\varphi(M,t_j)$ is defined, then $\varphi(M,t_j) = M'$, where
$M'(p_i) = M(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)), \forall p_i \in P.$

For a given petri net $PN = (P,T,I,O)$ and an initial marking M_0, the PN can be then executed by successive transition firings. The two sequences which result from the PN execution are-

1. Sequence of markings: $(M_0, M_1, \cdots)$

2. Sequence of transitions: $(t_{j0}, t_{j1}, \cdots)$

These two above mentioned sequences are related as:

$$\varphi(M_k, t_{jk}) = M_{k+1}, \qquad k = 0,1,2,\cdots$$

The result of the firing of an enabled transition, say t_j is the change in the state from M to M' and we say that M' is *immediately reachable* from M i.e the transition of the state takes place from M to M'.

This concept can be extended to *reachability* which shall be discussed in detail in subsequent chapters. For the time being, we define it.

Definition 3.1.1. (Reachability)

We define the *reachability set* $R(PN,M)$ for a petri net $PN = (P,T,I,O)$ with the marking M as the smallest set of markings defined as:

1. $M \in R(PN,M)$
2. If $M' \in R(PN,M)$ and $M'' = \varphi(M',t_j)$ for some $t_j \in T$, then $M'' \in R(PN,M)$.

3.2 Components of a system

The following are the components [5] of a Petri Net structure PN.

1. *Events and Conditions*

We have discussed about events and conditions above.

A simple view using a PN structure where we have the tabular data is as follows:

PRE-CONDITION	EVENT	POST-CONDITION
-	1	q
p, q	2	r
r	3	s, p
s	4	-

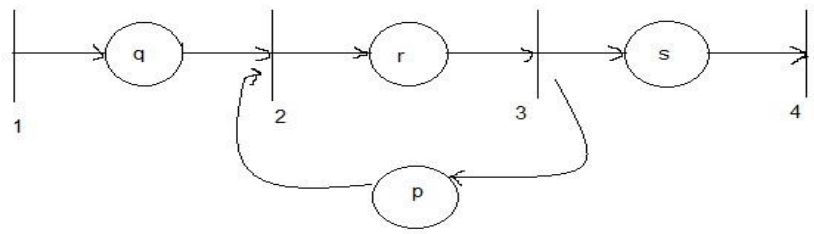


Figure 3.2: Events and conditions

2. *Concurrency*

We say that two transitions are concurrent if they are independent i.e. where one transition occurs independently of the other, either it can fire before, after, or in parallel to another enabled transition.

When we discuss about concurrency, we deal with the sharing of variables. Concurrency is a binary relation denoted by *co* which exhibits both reflexive and symmetric nature but not the transitive nature.

To understand this, we consider three events as follows: e_1
: Cooking food in kitchen e_2 : Singing e_3 : Riding a bicycle

Clearly, e_1 co e_1 (reflexive) , e_1 co $e_2 \Rightarrow e_2$ co e_1 (symmetric) but e_1 and e_3 are clearly not concurrent.(not transitive).

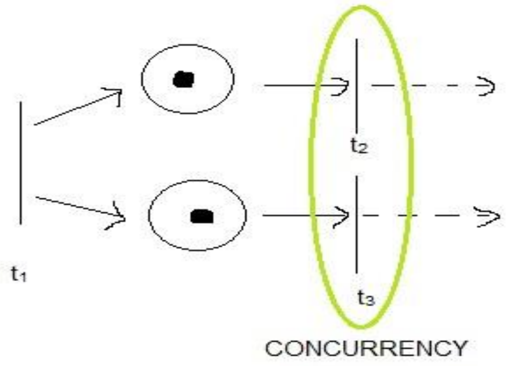


Figure 3.3: Transitions t_2 and t_3 executing concurrency.

3. *Conflict*

Two transitions are said to show conflict when the activities are in parallel i.e. either of the possible transition can occur/fire but both cannot simultaneously. Two transitions have a common input place and exhibit non-determinism.

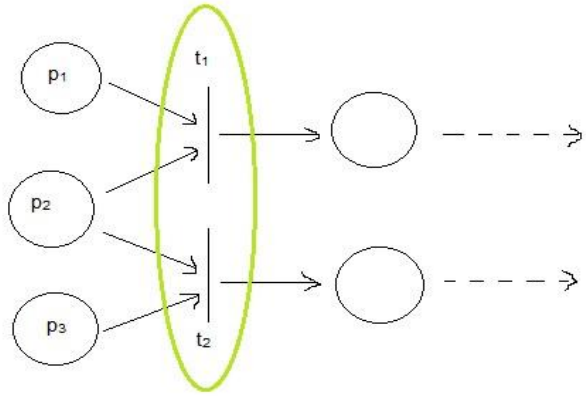


Figure 3.4: Conflict w.r.t p_2

Here in the above structure, we have three places p_1,p_2,p_3 and two transitions t_1,t_2 where both the transitions have p_2 as the common input place. If p_1 and p_2 has at least one token each and p_3 does not, then clearly t_1 can fire.

Similarly, if p_2 and p_3 has at least one token each and p_3 does not, then clearly t_2 can fire.

However, if all p_1, p_2 and p_3 have at least one token each, then both t_1,t_2 can fire and this will give rise to the situation of conflict.

To resolve this conflict, a selection is made amongst the possible enabled transitions based on some predefined characteristics/ probabilistic measures or policies.

Remark 3.2.1. When conflict and confusion occur together, it gives rise to *confusion.* Confusions can be as follows:

(a) *Symmetric Confusion* : Here t_1,t_3 are concurrent but are in symmetric conflict with t_2.

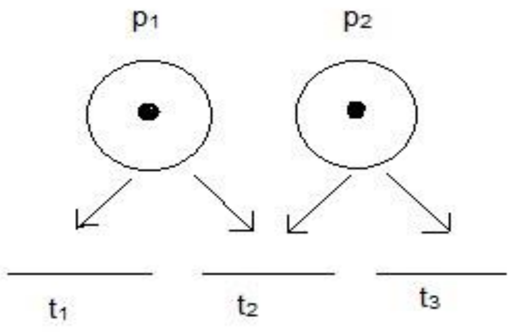


Figure 3.5: Symmetric Confusion

(b) *Asymmetric Confusion* : Here t_1,t_2 are concurrent. If t_2 fires initially, then there is a conflict between t_1 and t_3.

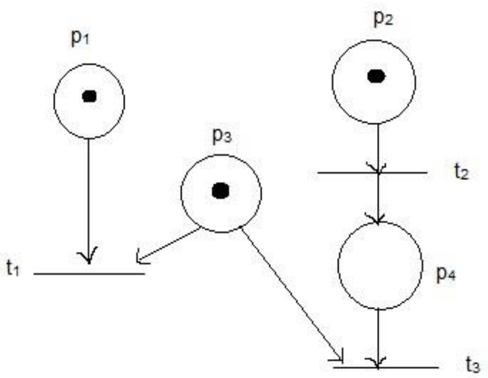


Figure 3.6: Asymmetric Confusion

4. *Sequential Action*
It is a straight forward component, as the name suggest where data comes in, gets processed and then goes out.

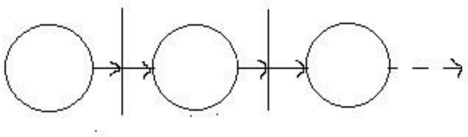


Figure 3.7: Straightforward sequence

It is also possible to have a transition that has no input place.
A simple model can be with two transitions and one intermediate place. The transition will thus, eventually keep on triggering, like a while loop continuing infinitely.

If we have a simple such sequential model, where let the time spent on each of the transitions is exponential with parameters, λ and μ for t_1 and t_2 respectively.

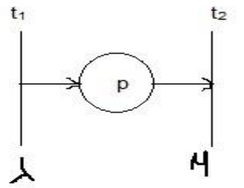


Figure 3.8: M/M/1 model

Then this system can represent a M/M/1 queuing model and the markings of this system can range from $0,1,\cdots\infty$, simply representing the number of customers in the queue.

5. *Resource Sharing*
Let us consider two systems under processing. Also, let there be a shared token in each of these systems.

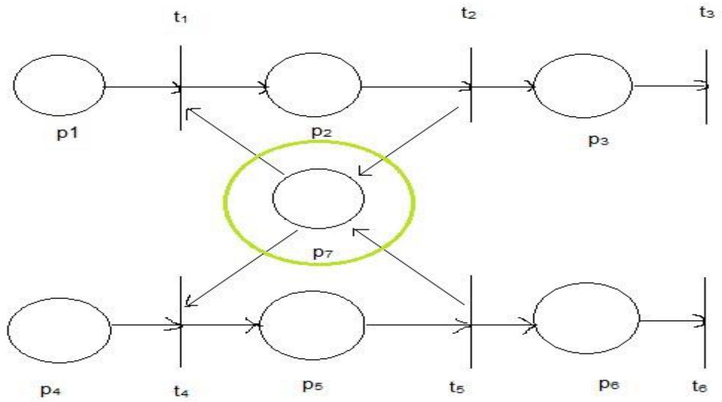


Figure 3.9: Resource Sharing

Here the first processor consists of the places p_1, p_2, p_3 and t_1, t_2, t_3 and the second processor consists of the places p_4, p_5, p_6 and t_4, t_5, t_6.

Now, if we want to trigger the transition t_1, then we will have to satisfy the condition where t_1 can get an input from both p_1 and p_7. The result that will be obtained from this will get stored in p_2. Now as long as there exists at least one token in p_2, the triggering of t_2 is possible which eventually stores the replica of this token in p_3.

However, it must be noted that the original data in the form of the token gets returned to the place p_7, following to which, we have the transition t_4 which is enabled and can fire.

6. *Buffers*
Every place in the system can accommodate some number of tokens which can be both finite or infinite. Such a number denotes the *capacity* of the PN.

Clearly, as the name suggests, the *finite capacity PN* implies a finite upper bound to the number of tokens and the *infinite capacity PN* can accommodate infinite number of tokens.

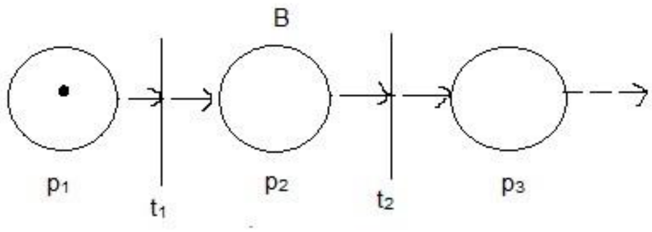


Figure 3.10: Buffering

For an example to be considered, let us consider that the Buffer B denoting the maximum number of tokens acts on place p_2 such that in p_2 part of the system, the condition exists that there have to be, say, exactly or at most B buffers.
So, now what we do is, introduce another place p_4 when p_2 finishes, that feeds back to t_1. Now, if we impose the condition that $p_2 + p_4 \leq B$.

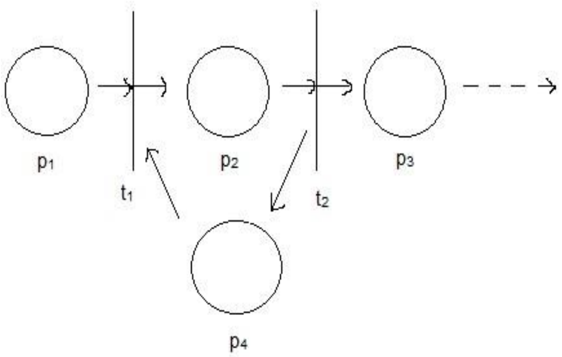


Figure 3.11: Adding Conditions in Buffering

Here, we have input p_1 and let us take the value B = 3. This means that at most 3 tokens can be accommodated in both p_2 and p_4 together.

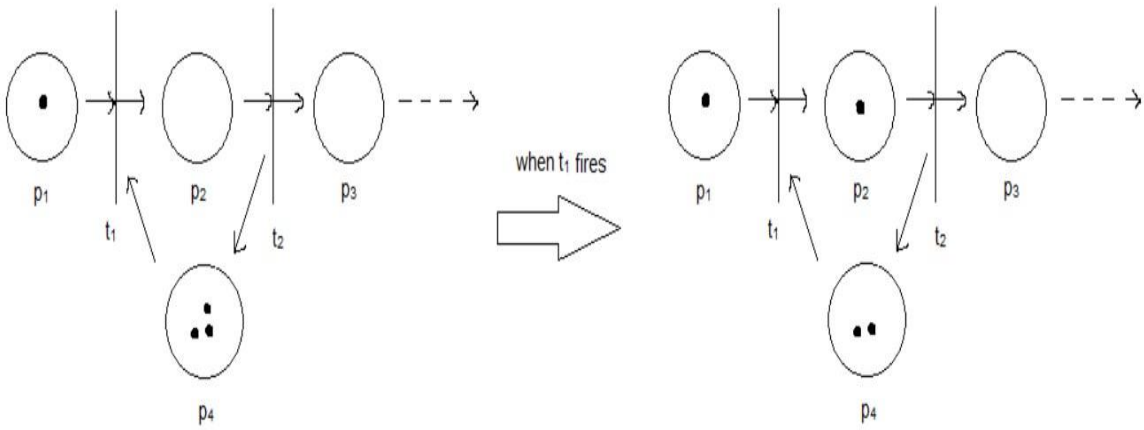


Figure 3.12: Petri Net representation

In a scenario, let us suppose that at some initial state t_0, all the three tokens are located in p_4. After this, the transition t_2 can be triggered where a single token goes to output p_3 and also returns to p_4 i.e. one process in t_2 finishes and comes back to p_3 and we have to model the system with the defined buffer.

Chapter 4

Properties of Petri Net

4.1 Safeness

Definition 4.1.1. A place $p_i \in P$ of a Petri Net $PN = (P,T,I,O)$ with an initial marking M_0 is safe [5] if for all $M^0 \in R(PN,M_0)$,$M^0(p_i) \leq 1$. A Petri Net is said to be Safe if all the places in that Petri Net are safe.

Remark 4.1.2. When modelling a Petri Net as a real hardware device, safeness property of a Petri Net can be useful for its analysis.

Remark 4.1.3. Example of a Safe-Petri Net

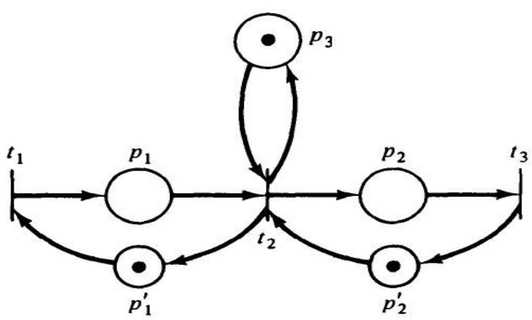


Figure 4.1: Safe-Petri Net

In the Fig. 4.1, when the transition t_1 is fired it removes a token from the place $p_1'$and adds a token in the place p_1 , this enables the transition t_2 and when it is fired a $_0$ token is added to the place p_1 and a token is deleted from the place p_1.Thus firing of any transition results in one token in either p_1 or p_1' at a time. Moreover firing of t_2 removes and adds a token in p_3 and hence there is only one token in p_3.Also a token is added in p_2 and removed from p_2'.Hence a similar dynamics takes place in p_2 and p_2' as of p_1 and p_1'.The total number of tokens at any time in a place after firing of any transition is either 1 or 0, hence the Petri Net is Safe.

Remark 4.1.4. If we interpret a place in a Petri Net as a logical condition then we know that a logical condition is either true or false. The logical condition being true means a single token in that place and a logical condition being false means no token in that place . Hence multiple tokens have no interpretation and the marking is safe under this assumption for all the places.

4.2 Boundedness

Safeness is a special case of the more general Boundedness property of Petri Net.

Definition 4.2.1. A place $p_i \in P$ of a Petri Net $PN = (P,T,I,O)$ with an initial marking M_0 is n-safe or n-bounded [5] if the number of tokens in that place cannot exceed an integer n

$$i.e.\ \forall M \in R(PN,M_0),\ M(p_i) \leq n$$

Remark 4.2.2. If a place is bounded , it is n-safe for some n.A Petri Net is bounded if all the places in that net are bounded.

Remark 4.2.3. If the number of tokens keeps on increasing in any place , the Petri Net would become unbounded.The system which is modeled by such kind of Petri Net would become unstable ,hence boundedness is a relevant property on which analysis techniques are performed.

4.3 Conservation

Definition 4.3.1. (Strictly-Conservative Petri-Net)
A Petri Net $PN = (P,T,I,O)$ with initial marking M_0 is said to be strictly-Conservative if for all $M \in R(PN,M_0)$

$$\sum_{p_i \in P} M'(p_i) = \sum_{p_i \in P} M_0(p_i)$$

Consider the petri-net in the figure given below ,

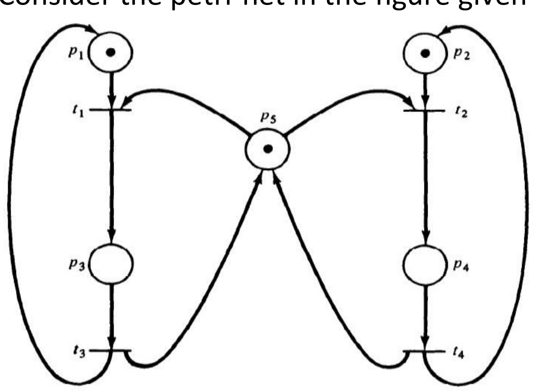


Figure 4.2: Not Strictly Conservative.

Here the enabled transitions are t_1 and t_2 .

$$M_0 = (1,1,0,0,1)\ (initial\ marking)$$
$$M_1 = (0,1,1,0,0)\ (On\ firing\ t_1)$$
$$M_2 = (1,1,0,0,1)\ (On\ firing\ t_3) \qquad (4.3.1)$$
$$M_3 = (1,0,0,1,0)\ (On\ firing\ t_2)$$
$$M_4 = (1,1,0,0,1)\ (On\ firing\ t_4)$$

The Petri Net in Fig 4.2 is not strictly conservative since the number of tokens in each of the markings are either increased or decreased by one, on consecutive firing of transitions and hence the token count is not constant.

Remark 4.3.2. When Petri Net are modelled in such a way that the tokens are represented as resources which are neither created nor destroyed, conservation becomes an important property to monitor.

Remark 4.3.3. A Petri Net is said to be partially conservative [2], [6] if the token count is constant for few markings, then changes to another positive integer and remains constant for the next few markings and eventually becomes constant. For a partially conservative Petri Net, the tokens in a place will never become unbounded.

4.4 Conservative with respect to weighing vector.

Definition 4.4.1. A Petri Net $PN = (P,T,I,O)$ with an initial marking M_0 is conservative with respect to weighing vector u,where $u = (u_1,u_2,u_3,u_4......,u_m)$ and $|P| = m$, $u > 0$ (positive non-zero vector),if for all $M^0 \in R(PN,M_0)$,

$$\sum_i u_i.M^{'}(p_i) = \sum_i .u_i.M_0(p_i)$$

Remark 4.4.2. A Strictly Conservative Petri Net is conservative with respect to the weighing vector $u = (1,1,1,1,1....,1)$.

Remark 4.4.3. The weighing vector is important concept since the tokens in the places need not be identical in nature , that is some tokens might be of larger relevance to us and thus would be assigned a larger weight , whereas some tokens might be of no importance and thus can be assigned a lesser or 0 weight.Hence in modelling of Petri Net conservation is an important property which can be investigated with respect to the importance of tokens in the model.

Remark 4.4.4. For the example of conservation in Fig 4.2 , the Petri Net is conservative with respect to the weighing vector $u = (1,1,2,2,1)$.
If we consider the resultant markings after considering the weights of the token, we get

$$(1,1,0,0,1).(1,1,2,2,1) = (1,1,0,0,1)$$

$$(0,1,1,0,0).(1,1,2,2,1) = (0,1,2,0,0) \qquad (4.4.1)$$
$$(1,0,0,1,0).(1,1,2,2,1) = (1,0,0,2,0)$$

Hence the total token number count which is 3, is constant for all the markings after considering the weights of the tokens.Hence the model is conservative with respect to the weighing vector $u = (1,1,2,2,1)$.

Remark 4.4.5. Conservation and Safeness are special cases of boundedness.

4.5 Liveness

Resource allocation was the motivation to study conservation as a property in Petri Net.Another problem which may arise in resource allocation is deadlock.

Definition 4.5.1. Deadlock

A deadlock [5] in a Petri Net is a situation where a transition or a set of transitions cannot fire in that Petri Net.If there exists a transition, say t in T such that t can never be fired, then t is said to be dead.

Definition 4.5.2. Live

A Petri Net model is said to be live [5] w.r.t an initial marking if it is possible to fire all the transitions at least once using some firing sequence for all the markings in the reachability set.

A transition is live if it is not deadlocked. This does not mean that the transition is enabled,but the fact that it can be enabled in future.

Definition 4.5.3. Potentially Firable

A transition t_i of a Petri Net PN is potentially firable in a marking M_0 if there exists a marking M^0 $\in R(PN,M_0)$ and t_i is enabled in M^0.

Remark 4.5.4. Demonstration of Deadlock Property.

Consider the example of resource allocation for two processes and two resources below,

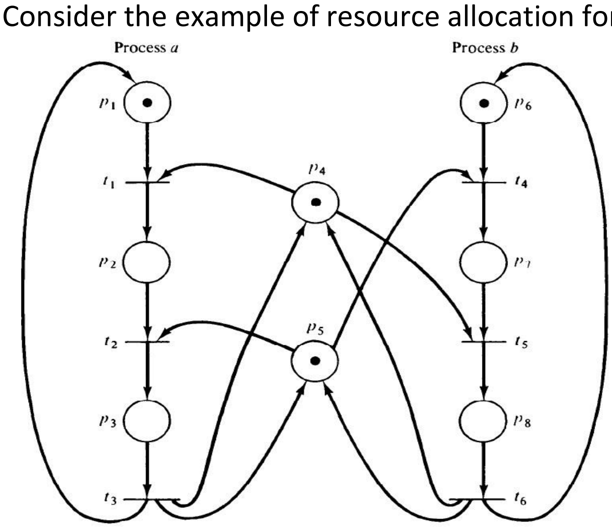


Figure 4.3: Demonstration of Deadlock

In this model illustrated in Fig 4.3 there are two processes , process a and process b , also there are two resources , r_1 in place p_4 and r_2 in place p_5 .The transition firing sequence $t_1t_2t_3t_4t_5t_6$ and $t_4t_5t_6t_1t_2t_3$ does not produce deadlock.

If both the processes need both the resources , then they would have to share the resources in such a way that each of the process asks for a resource and then later releases it so the other process can use it.

If we consider the transition firing sequence which starts from t_1t_4 , then process a would have the resources from p_4 and would want resources from p_5 and similarly process b would have resources from p_5 and would be needing resources from p_4.Thus a deadlock condition would be reached and neither of the two processes would be able to proceed further.

Chapter 5

Petri Nets : Firing rule and Formula

For the understanding of our model we will take a example of vending machine which is explain all the thing related to this model

5.1 A Cookie Vending Machine

For this introductory example, we describe a vending machine that sells cookies. The machine has a coin slot and a compartment into which the packets of cookies are dropped. In the initial state of the cookie vending machine, the coin slot contains a coin. The cookie compartment is empty.

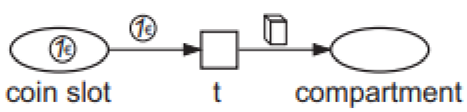


fig 5.1

Models this as a Petri net: coin slot and compartment, both depicted as ellipses, are the places of the Petri net. The coin slot contains a euro coin, which is a token of the net.

the transition t is enabled, because its incoming arc starts at a place containing a coin token, as required by the arc's label. Therefore, t can occur and thereby change the current marking.

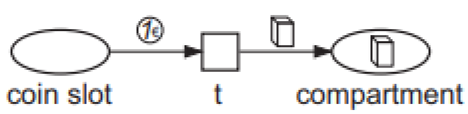


fig 5.2

shows the new marking: The place coin slot does not contain any tokens

5.2 Inside the the model

If we look inside the machine, we will find several components that store coins and cookie packets and handle the sale. Figure 5.3 in turn models the interior of the vending machine as a Petri net: there is a storage filled with five cookie packets and one – initially empty – cash box

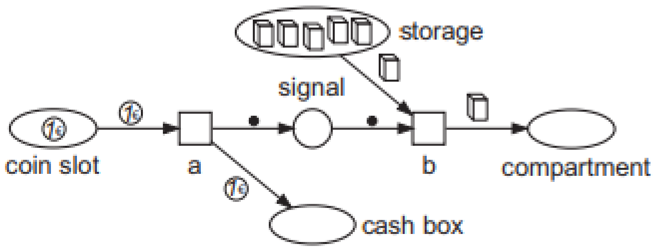


fig 5.3

In the marking shown, transition a can occur. Its effect can be deduced from the arrows starting or ending in a: the coin disappears from the coin slot and appears in the cash box. Simultaneously, a signal for transition b is generated, depicted by a black dot. Figure 5.4 shows the marking after the occurrence of a. Now, transition b can occur, because the arrows ending in b are labeled with objects that are actually present in the respective places: a black dot in signal and a cookie packet (even several) in the storage. After the occurrence of b, the marking shown in Fig. 5.5 is reached. It corresponds to the marking in Fig. 5.2. No more transitions can occur now.

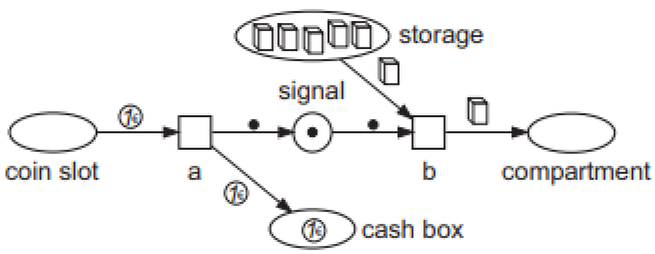


fig 5.4

After occurrence of a

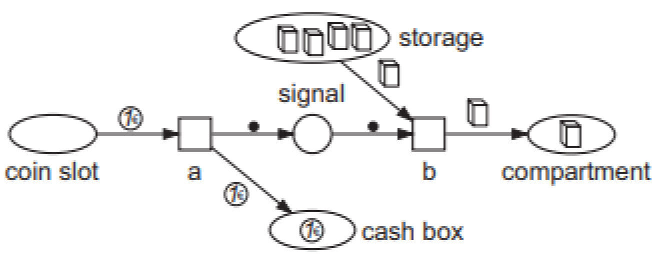
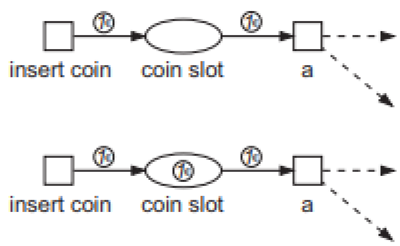


fig 5.5

After occurrence of b

5.3 Interfaces

So far, we have modeled the vending machine as a closed system: coins and cookie packets are distributed across places and the occurrence of transitions redistributes them. What is missing are actions of the environment: someone inserts a coin, for example, or takes out a cookie packet. How do we model this? As Fig. 5.6 shows, a transition insert coin sits in front of the – empty – coin slot. insert coin does not have any preconditions, so it can occur anytime. In the real world, of course, this action, in fact, does have further preconditions. Most im?portantly, the environment has to provide a coin. Likewise, the take packet models the taking of a packet out of the compartment. The two transitions insert coin and take packet model the vending machine's interface. Both are enabled in the marking of Fig. 5.6. Connected with this is the label "ε", which we will deal with next



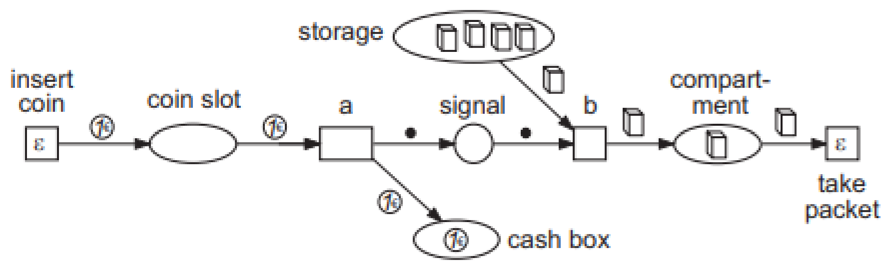


fig 5.6

So this is our example where we understand the basic terminology and other things related to our work.

Now come to the firing formula

5.4 Working/firing related terminology

Components of a Net

Places

A Petri net is a structure with two kinds of elements. One kind of element is places. Graphically, a place is represented by a circle or ellipse represented as

Transitions

The second kind of elements of a Petri net are transitions. Graphically, a transition is represented by a square or rectangle represented as

Arcs

Places and transitions are connected to each other by directed arcs. Graphically, an arc is represented by an arrow. An arc never models a system component, but an abstract, sometimes only notional relation between components such as logical connections, access rights, spatial proximities or immediate linkings.

The arc represented as (→)

An arc never connects two places or two transitions. An arc rather runs from a place to a transition or vice versa from a transition to a place.

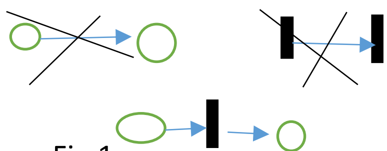


Fig.1

Net Structure

It is customary to denote the sets of places, transitions and arcs with P, T and F, respectively, and to regard arcs as pairs, that is, F as a relation $F \subseteq (P \times T) \cup (T \times P)$. Then

$N = (P, T, F)$

is a net structure. The places and transitions are the elements of N. F is the flow relation of N.

Definition (Petri net, Syntax). A Petri net is 4-tuple $N = (P, T, f, M_0)$,

Where

– P and T are finite, non-empty and disjoint sets. P is the set of places (in the figures represented by circles). T is the set of transitions (in the figures represented by rectangles).

– $f : ((P \times T) \cup (T \times P)) \rightarrow IN_0$ defines the set of directed arcs, weighted by nonnegative integer values.

– $M_0 : P \rightarrow IN_0$ gives the initial marking

Labeling of Arcs

The labeling of an arc (p, t) or (t, p)

Definition (Firing rule). Let $N = (P,T,f,m_0)$ be a Petri net.

– A transition t is enabled in a marking m, written as m(t), if

$\forall p \in t : m(p) \geq f(p,t)$, else disabled.

– A transition t, which is enabled in m, may fire.

– When t in m fires, a new marking m’ is reached, written as m[(t)m’

, with ∀p ∈P : m’(p) = m(p) − f(p,t) + f(t, p).

– The firing happens atomically and does not consume any time.

What is the effect of firing transition t ?

Suppose we have place p then

$$\boldsymbol{m'(a) = m(a) - f(a,t) + f(t,a)}$$

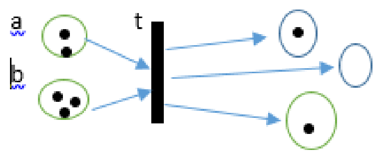


Fig.2

$$m'(a) = m(a) - w(a,t) + w(t,a)$$

$$m'(a) = 2 - 1 + 0 = 1$$

Chapter 6

Model for small hospital

6.1 Petri net model on small hospital

Given the spread of the COVID-19 virus, social distancing is the need of the hour. In this paper, designed and proposed is a small hospital model that is supposed to provide treatment to its patient while practicing social distancing and allowing a minimum number of people together at any possible stage. Thus, this proposed model of a small hospital that originally had three bed has the seating plan to provide service at three beds while taking all necessary precautions against the virus and ensuring social distancing. Where distance between each bed 2 meter Further, for the arrival of any patient at the hospital, three separate waiting places have been marked, which shall eventually pave the way to the respective table (either first , second and third). The model has been designed such that the new patient cannot occupy the bed unless the previous patient who are seated on the bed get treatment.

Fig shows the Petri net structure of the proposed model where P= {P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14,P15,P16,P17}and T= {T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13 } are the places (conditions) and transitions (events) respectively.

The presence of a token at a place denotes the condition to be true; while the absence of a token means that the condition corresponding to a place is not true (or is not happening).

A prior assumption for this hypothesis is that when the first three patient arrive, a token is created at the places P1 ,P3,P11 respectively, and another token is created at the place p2 signifying the availability of the staff.

Places	Condition
P1	Patient seats at fresh and sanitized first bed and is ready to get the treatment
P2	The staff is free and can provide the care to the patient at either bed
P3	Patient seat at fresh and sanitized second bed and is ready to get the treatment
P4	The patient at first bed is waiting for treatment to be done
P5	The patients of the beds are confirmed by staff
P6	The patient at second bed is waiting for treatment
P7	The patent is cured/dead and the first bed is vacant
P8	The patent is cured/dead and thesecond bed is vacant
P9	New patient arrives to occupy the first bed
P10	New patient arrives to occupy the second bed
P11	Patient seat at fresh and sanitized third bed and is ready to get the treatment
P12	The patient at third bed is waiting for treatment
P13	The patent is cured/dead then third bed is vacant
P14	New patient arrives to occupy the third bed
P15	Doctor checks the patient of the first bed
P16	Doctor checks the patient of the second bed
P17	Doctor checks the patient of the third bed

Transitions	Activities
T1	The staff is taking care of the patient on first bed
T2	The staff is taking care of the patient on second bed
T3	Doctor provides the treatment on the first bed
T4	The staff informs the doctor that the patient has arrived
T5	Doctor provide the treatment on second bed
T6	The bedsheet of the first bed is changed and sanitized
T7	The bedsheet of the second bed is changed and sanitized
T8	The bedsheet of the third bed is changed and sanitized
T9	The staff is taking care of patient on the third bed
T10	Doctor provide the treatment on third bed

T11	Doctor visits first bed
T12	Doctor visits second bed
T13	Doctor visits third bed

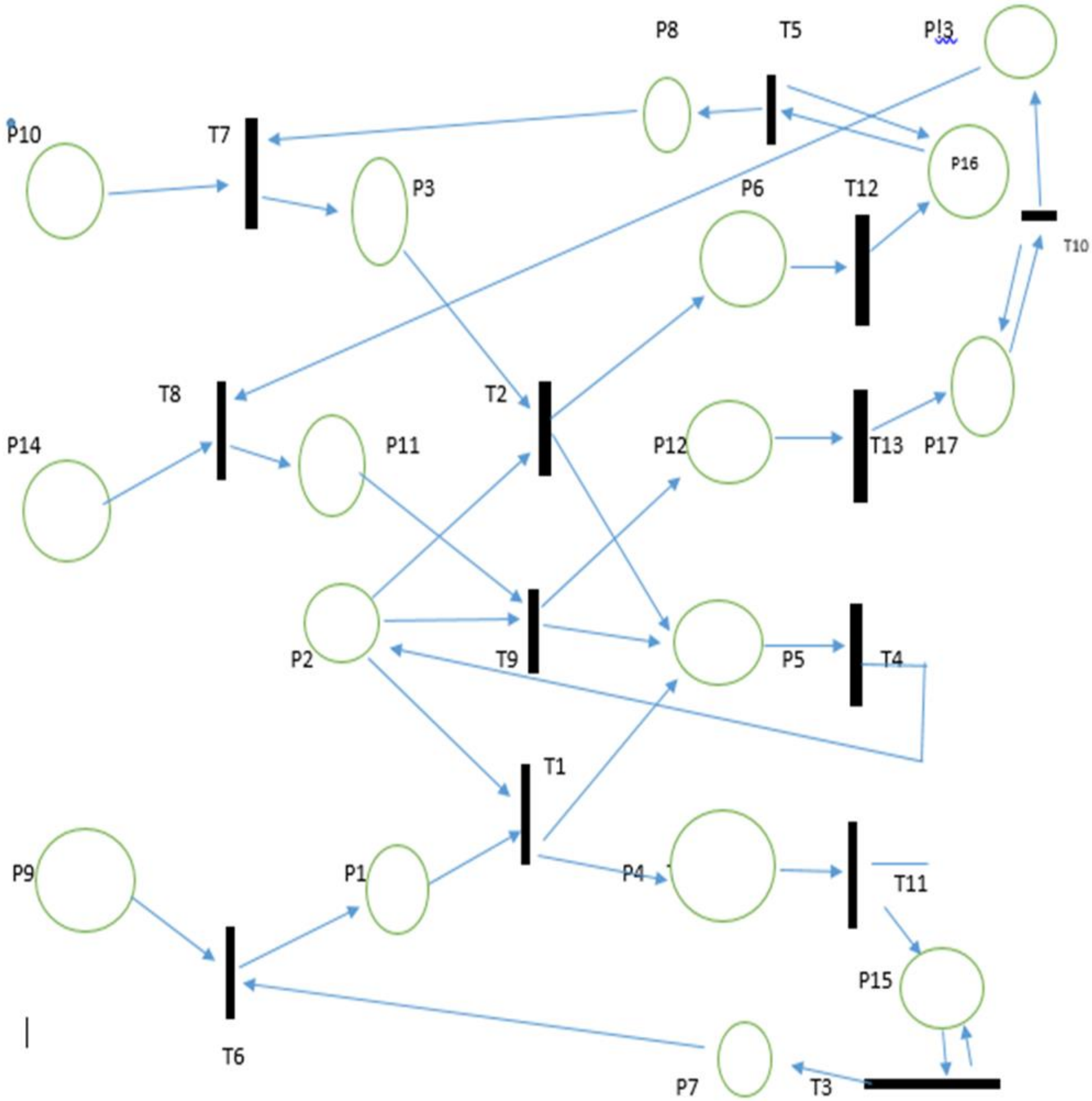


Fig 6.1

Now our first scenario

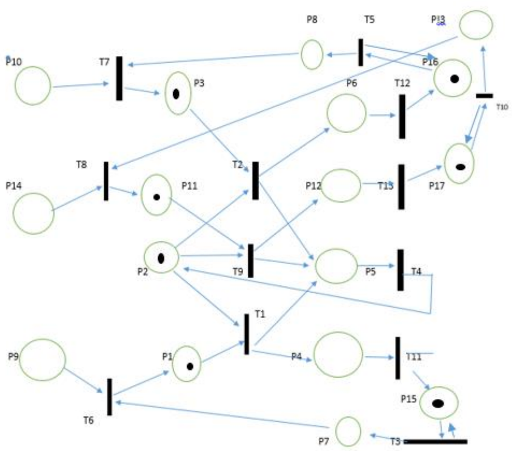


Fig.6.2

Where 3 beds occupy by patient where P3,P11 and P1 form of token

As per the first scenario, It has three patient seated on the sanitized and fresh beds on first , second and third beds respectively, and staff is free to serve the two customers. In this scenario, it is assumed that no new patient has arrived at the hospital.

First we will cheak the transitions are enable or not

$m(p) \geq f(p,t)$,

After cheaking the enabling condition now further process

Now use the formula

$$\boldsymbol{m'(p) = m(p) - f(p,t) + f(t,p)}$$

For P1

$m'(p1)=1-1+0=0$

for P3

$m'(p3)=1-1+0=0$

for P11

$m'(p11)=1-1+0=0$

Now after transition completed the token is gone further which are

$m'(p4)=0-0+1=1$

$m'(p6)=0-0+1=1$

m'(p12)=0-0+1=1

Similarly the in the P5 the after occurring transition T1,T9,T2 the token are m'(P5)=3

Now we will consider only **first bed** (same for second and third bed) so hear P4 have one token now P4 enable and fire and token will go to P15 where the docter visits at first bed after the firing of T11 the token in P15 is 2,

m'(p15)=1-0+1=2

and lastly T3 enable and fire and token is go to P7 and the process is occur again and again for each patient.

In the second scenario

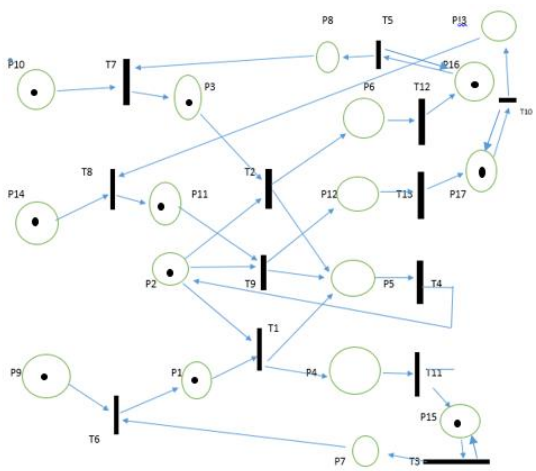


Fig.6.3

In this scenario the patient at waiting place P9 ,P14,P10 when bed are vacant then provide the a new bed and the process are similar.

6.2 ANALYSIS AND INTERPRETATION OF THE MODEL

1) A petri net model is structurally conservative if there exist a positive(non-negative) vector w such that

$$Dw=0; w\geq 0.$$

In order to find the incidence matrix D, we calculate the input matrix I, and output matrix of the model-

INPUT MATRIX

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17
t1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
t4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
t6	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
t9	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
t11	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
t13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

OUTPUT MATRIX

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17
t1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
t4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
t6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
t9	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
t11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
t12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
t13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

D= OUTPUT MATRIX – INPUT MATRIX

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17
t1	−1	−1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	−1	−1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
t4	0	1	0	0	−1	0	0	0	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
t6	1	0	0	0	0	0	−1	0	−1	0	0	0	0	0	0	0	0
t7	0	0	1	0	0	0	0	−1	0	−1	0	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	1	0	−1	−1	0	0	0
t9	0	−1	0	0	1	0	0	0	0	0	−1	−1	0	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
t11	0	0	−1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
t12	0	0	0	0	0	−1	0	0	0	0	0	0	0	0	0	1	0
t13	0	0	0	0	0	0	0	0	0	0	0	−1	0	0	0	0	1

Solving system of equation we get-

$$-w_1 - w_2 + w_4 + w_5 = 0$$

$$-w_2 - w_3 + w_5 + w_6 = 0$$

$$w_7 = 0$$

$$w_2 - w_5 = 0$$

$$w_8 = 0$$

$$w_1 - w_7 + w_9 = 0$$

$$w_3 - w_8 - w_{10} = 0$$

$$w_{11} - w_{13} - w_{14} = 0$$

$$-w_2 + w_5 - w_{11} - w_{12} = 0$$

$$w_{13} = 0$$

$$-w_4 + w_{15} = 0$$

$$-w_6 + w_{16} = 0$$

$$-w_{12} + w_{17} = 0$$

We get-

$$
w = \begin{bmatrix} w1 \\ w2 \\ w3 \\ w4 \\ w5 \\ w6 \\ w7 \\ w8 \\ w9 \\ w10 \\ w11 \\ w12 \\ w13 \\ w14 \\ w15 \\ w16 \\ w17 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

In the above matrix the petri net not fully conservative, because 2 place the value negative.

Safeness: A place pi in a Petri Net structure with an initial marking M_0 is safe if for all markings M' that belong to the Petri Net's reachability set, M'(pi) ≤1 i.e. the total number of tokens at any state in a place after firing of a transition is either 1 or 0. A Petri Net is said to be safe if all the places in that Petri Net are safe.

In our project the scenario one calculation, in this all the places we have a token means all the places M'(pi) have certain value , means our model is fully fallows the above definition, and so all the places are safe , so petri net are safe.

Boundedness: It is apparent that conservativeness is a particular case of boundedness.

Hence, this model is structurally as well as behaviorally bounded. Further, the number of tokens at any place can be either 0 or 1.but model are not bounded because the particular case of conservative are not fallows the model.

Deadlock: Deadlock in a Petri Net is a situation where a transition or a set of transitions is unable to fire.

In our first and second scenario hear the the deadlock is when the seats of hospital are vacant, mean the our model is deadlock at this point of time.

In this the model is deadlock is when then patient is cured or dead the vacant seat is position of deadlock. And the places is P7,P8,P13 is the deadlock position.

Liveness: A Petri net model is said to be live w.r.t an initial marking if it is possible to fire all the transitions at least once using some firing sequence for all the markings in the reachability set. If there exists a transition, say t in T such that t can never be fired, then t is dead.

In first scenario the P9,P10,P14 are assume as dead transition and means hear assume as new patient arrives at hospital.

In second scenario we have a situation where every transition can be fired, as there are new patient who can occupy the first ,second and third beds. Thus, there is no transition that is dead, and hence, the Petri Net model with respect to the initial marking is live. The physical interpretation of liveness is that all the activities that are mentioned in the model will happen at least once

Behavioral Conflict: Two transitions are said to show conflict when the activities are in parallel i.e., either of the possible transition can occur/fire, but both cannot simultaneously. Two transitions have a common input place and exhibit non-determinism.

The conflicts can possibly exist between the transitions T1,T2,T9 and T11,T13,T12 The design of the model (the token placing) is such that T1, and T11cannot be enabled simultaneously since tokens can only be at after firing of (similarly, and cannot be enabled simultaneously). Hence the only conflicts that would occur in the Petri Net model for both scenarios . The physical interpretation of the conflict between and implies that when both patient at first and second and beds are ready for treatment can only attend one beds at a time.

Behavioral Concurrency: Two transitions are concurrent if they are independent to each other i.e., where one transition occurs independently of the other, either it can fire before, after, or in parallel to another enabled transition.

In the our case the structurally the transition and activity are actually assumed . but in case of behaviorally there are some concurrency mean the staff and doctor are one person whose takes responsibility but when one patient are arrived and other activity also happen the conflict are happen in the behavior.

Chapter 7

Conclusion

In this paper, we have discussed and interpreted the structural and behavioral properties of the proposed model. The proposed small hospital model has been interpreted so it is apply on some big hospital.

And also extend our work on this model with the help of others tools of petri nets.

References

[1] M. Dotoli, M.P. Fanti, A.M. Moretti, W. Ukovich Modeling and Management of a Hospital Department via Petri Nets (2010)

[2] Wolfgang Reisig, Petri Nets Modeling Techniques, Analysis Methods, Case Studies February 22, 2013

[3] W.M.P. van der Aalst, The Application of Petri Nets to Workflow Management

[4] Andrea BOBBIO SYSTEM MODELLING WITH PETRI NETS (1990)

[5] James L. Peterson, Petri Net Theory and Modeling of Systems, Prentice-Hall (1981).

[6] T. Murata, Petri Nets: Properties, Analysis and Applications, Proc. of IEEE, 77-4, (1989), 541.