DESIGNING AND ANALYSING A PETRI NET MODEL OF A SMALL EATERY 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:

SRISHTI PATEL , RIDHI BAKSHI

(2K19/MSCMAT/23) (2K19/MSCMAT/31)

Under the supervision of

PROF. SANGITA KANSAL

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2021

Contents

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We, Srishti Patel and Ridhi Bakshi, Roll No.s 2K19/MSCMAT/23, 2K19/MSCMAT/31 of Master in Science (Mathematics), hereby declare that the project dissertation titled DESIGNING AND ANALYSING A PETRI NET MODEL OF A SMALL EATERY 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.

Srishti

Place: Delhi
Date: May 25, 2021

Srishti Patel (2K19/MSCMAT/23)

Ridhi Bakshi

Ridhi Bakshi (2K19/MSCMAT/31)

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 EATERY IN TIMES OF COVID which has been submitted by Srishti Patel and Ridhi Bakshi , Roll No.s 2K19/MSCMAT/23 and 2K19/MSCMAT/31 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. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University.

Prof. Sangita Kansal
Supervisor

Place: Delhi
Date: May 25, 2021

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.

Srishti Patel (2k19/MSCMAT/23)

Ridhi Bakshi(2k19/MSCMAT/31)

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 eatery 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 two service tables (which can be occupied by new customers subsequently) and one service provider (waiter) and their respective activities, 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. The reachability tree has been made for drawing out more behavioral conclusions. Besides, inference of properties like cyclic/acyclic nature, conflict, concurrency, boundedness, conservativeness, safeness, liveness, and deadlock has been interpreted physically with the proposed model.

List of Figures

List of Tables

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 $\circ$ 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.***

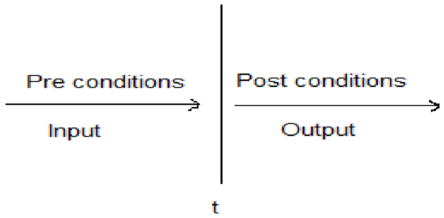


Figure 1.1: Conditions of the Petri Net

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.

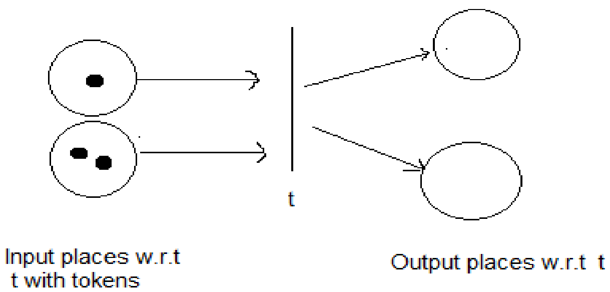


Figure 1.2: Tokens in a Petri Net

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* $\#(x, B_2) = \#(x, B_3) = 2$.

The preliminaries of Petri Nets $[5] - [7]$ *have been discussed in the following sections of this chapter.*

2.2 Structural Description

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

$$PN = (P, T, I, O), \quad \textit{where}$$

$P =$ *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,

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

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; $P \cap T = \phi$.

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

$$\begin{array}{ll} I(t_1) = \{p_1\} & O(t_1) = \{p_2, p_3\} \\ I(t_2) = \{p_3\} & O(t_2) = \{p_3, p_5, p_5\} \\ I(t_3) = \{p_2, p_3\} & O(t_3) = \{p_2, p_4\} \\ I(t_4) = \{p_4, p_5, p_5, p_5\} & O(t_4) = \{p_4\} \\ I(t_5) = \{p_2\} & O(t_5) = \{p_6\} \end{array}$$

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:

$$\begin{array}{ll} I(p_1) = \{\} & O(p_1) = \{t_1\} \\ I(p_2) = \{t_1, t_3\} & O(p_2) = \{p_3, p_5\} \\ I(p_3) = \{t_2, t_2\} & O(p_3) = \{p_2, p_3\} \\ I(p_4) = \{p_3, p_4\} & O(p_4) = \{t_4\} \\ I(p_5) = \{t_2\} & O(p_5) = \{t_4, t_4, t_4\} \\ I(p_6) = \{t_5\} & O(p_6) = \{\} \end{array}$$

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

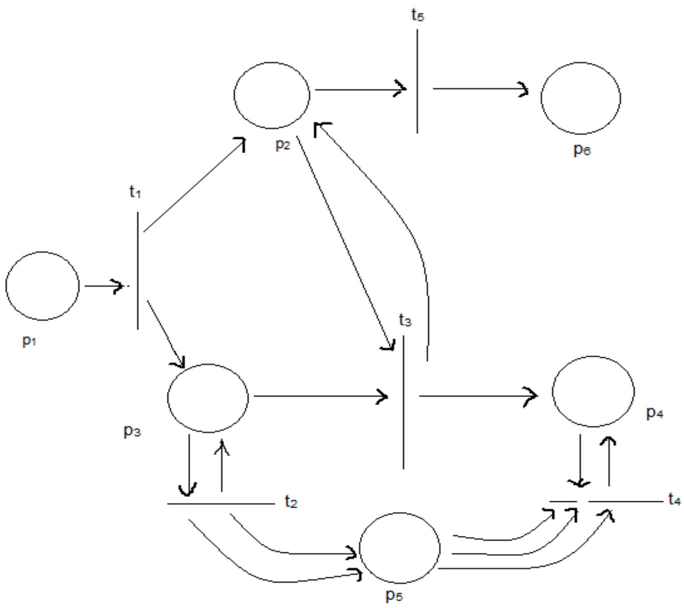


Figure 2.1: Structure of the Petri Net

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 $M_t = M_1, M_2, \dots, M_m$ where M_i gives the number of tokens that are available at the place p_i at a certain state t.

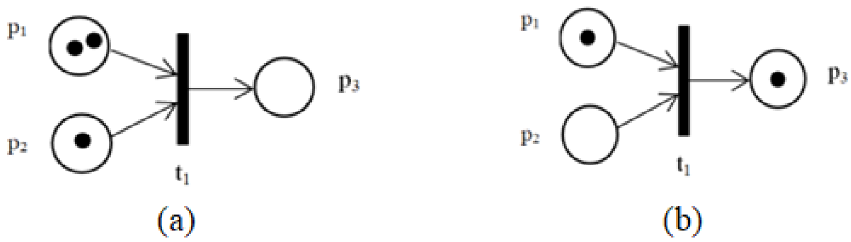
A marking M is a function defined from P, the set of all places to the non-negative integers i.e., $M : P \to Z^+$, where clearly, $M(p_i) = M_i$. The marking at initial state (at $t = 0$) is called the initial marking $M_0 : P \to Z^+$. A marked Petri Net PN w.r.t M_0 is a 5-tuple structure where $PN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

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

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



(a) Before firing of t_j. (b) After firing of t_j.

Figure 2.2: Transition enabling in a Petri Net

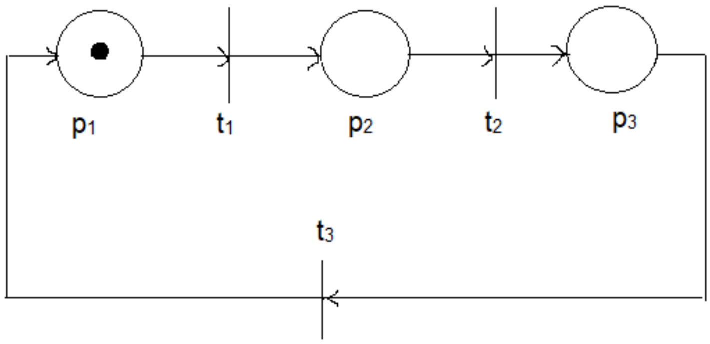


Figure 2.3: Transition enabling in a Petri Net

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

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

and

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

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

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

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***.

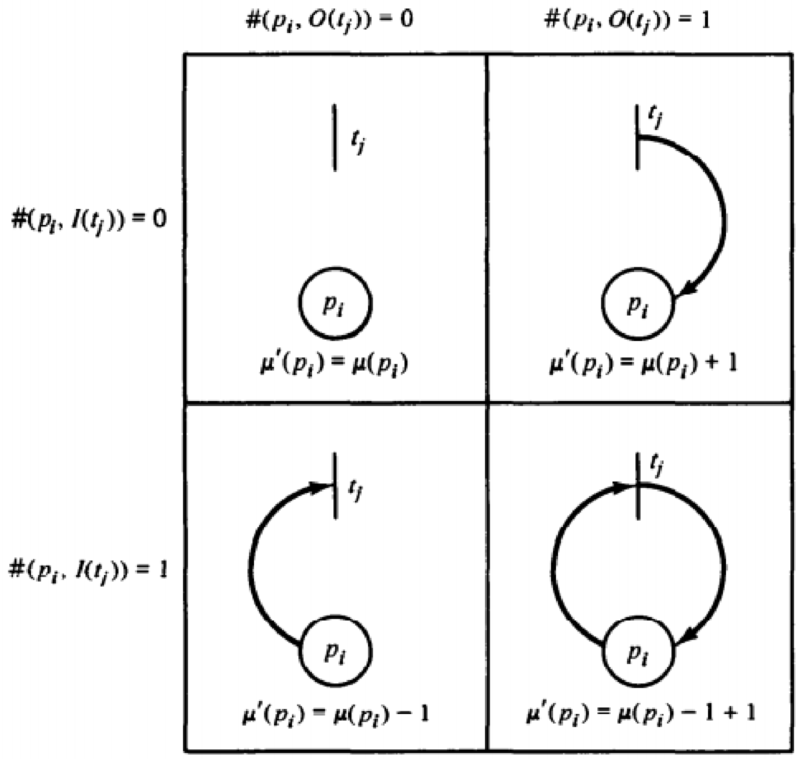


Figure 2.4: Illustration of the change in marking in a place when a transition fires.

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.

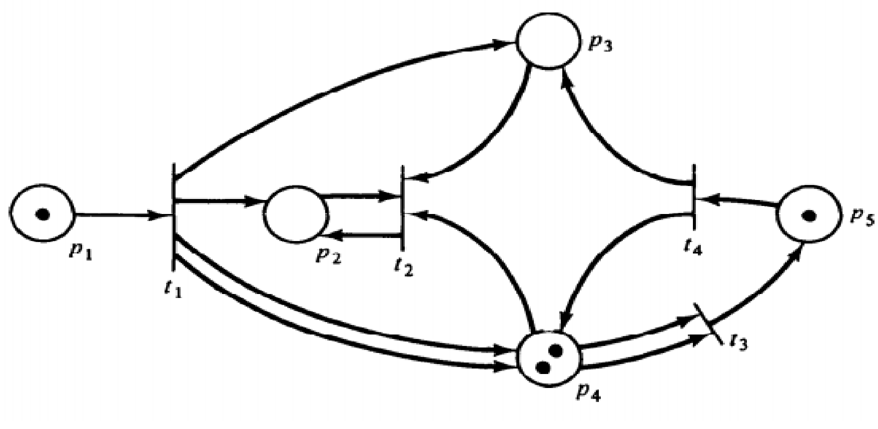


Figure 2.5: Transitions t_1, t_3, t_4 are enabled.

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.

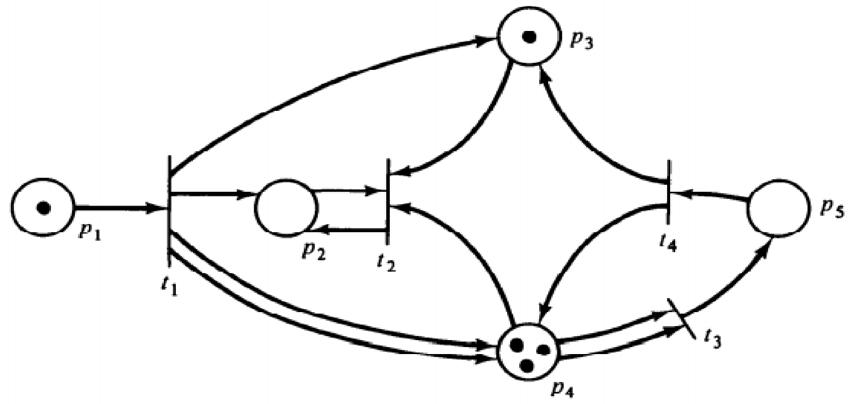


Figure 2.6: Transition t_4 fires

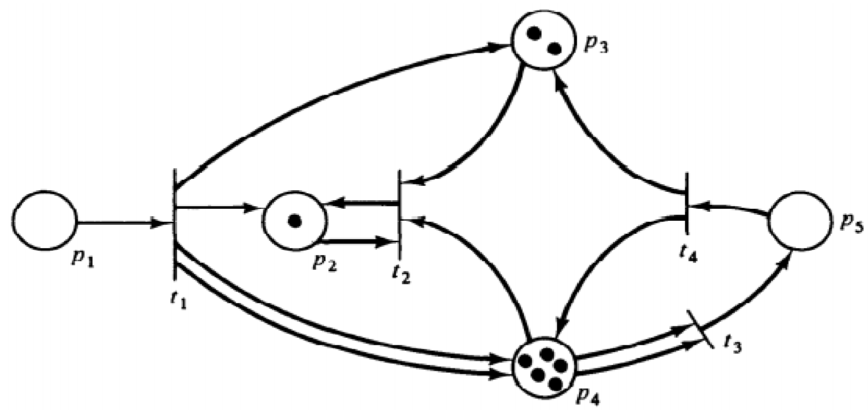


Figure 2.7: Transition t_1 fires

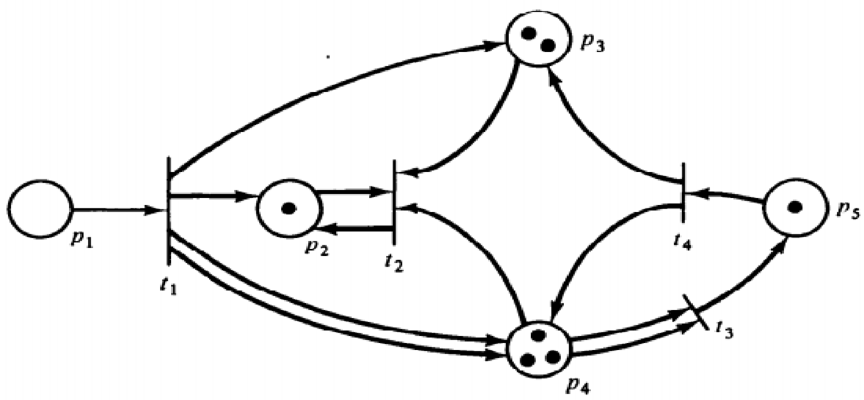


Figure 2.8: Transition t_3 fires

Chapter 3

Petri Net State Space and Components

3.1 Petri Net State Space

We define the ***state*** [6] 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.*** [6]

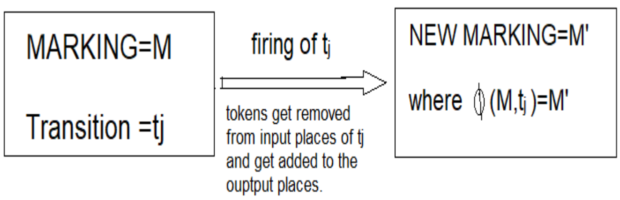


Figure 3.1: Petri Net marking.

Formally, if we define the ***next-state function***, $\phi : N^m \times T \to N^m$ 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 $\phi(M, t_j)$ is defined, then $\phi(M, t_j) = M'$, 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:** $(t_{j_0}, t_{j_1}, \cdots)$

These two above mentioned sequences are related as:

$$\phi(M_k, t_{j_k}) = M_{k+1}, \quad k = 0, 1, 2, \cdots$$

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 $M'' = \phi(M', t_j)$ 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	-

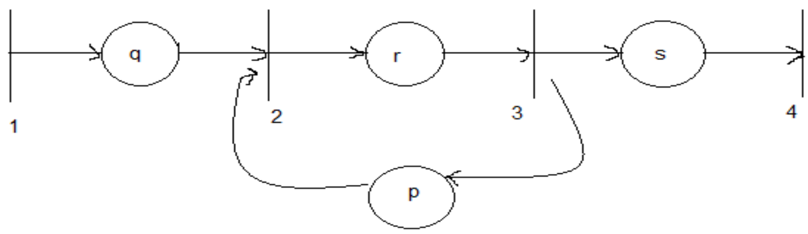


Figure 3.2: Events and conditions

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).

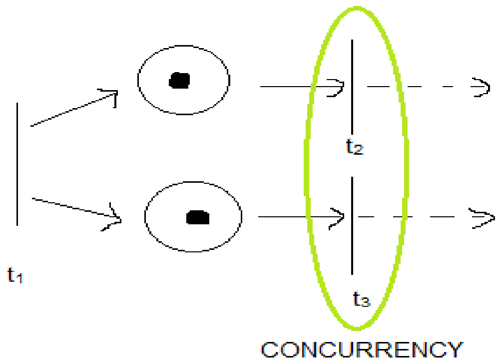


Figure 3.3: Transitions t_2 and t_3 executing concurrency.

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.

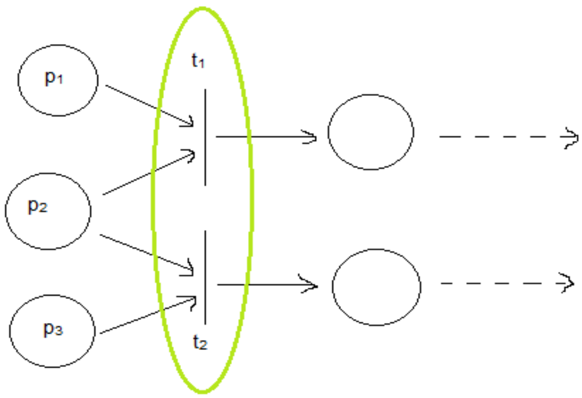


Figure 3.4: Conflict w.r.t p_2

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.

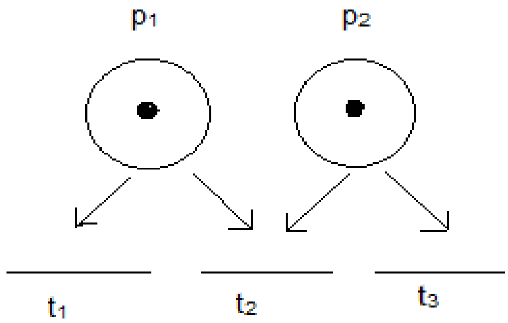


Figure 3.5: Symmetric Confusion

(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.

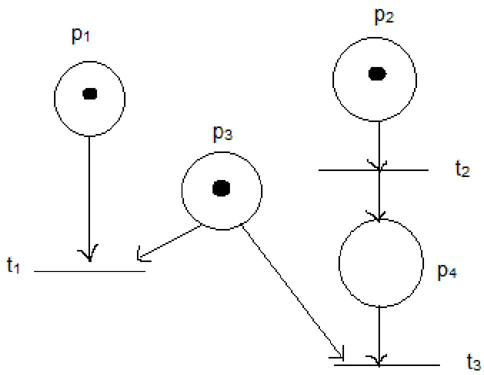


Figure 3.6: Asymmetric Confusion

4. ***Sequential Action***

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

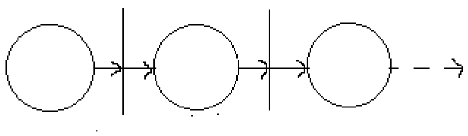


Figure 3.7: Straightforward sequence

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.

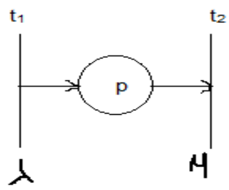


Figure 3.8: M/M/1 model

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.

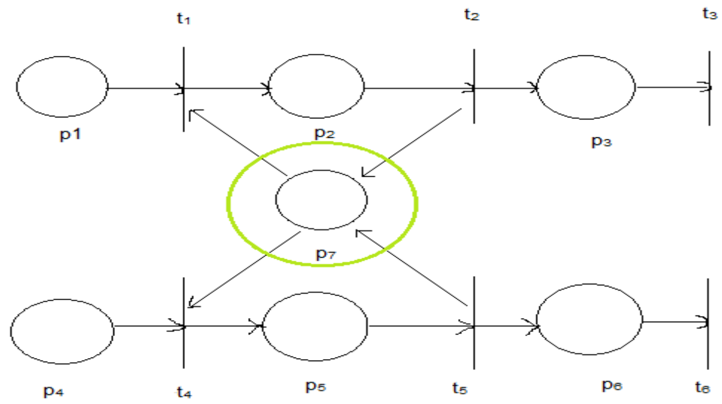


Figure 3.9: Resource Sharing

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.

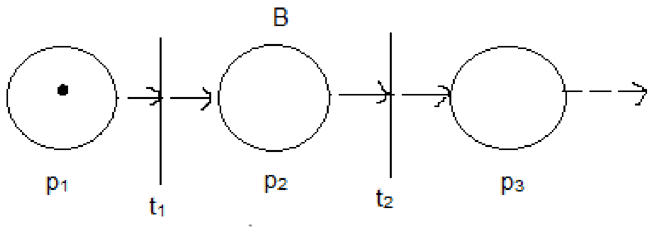


Figure 3.10: Buffering

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 $p_2 + p_4 \leq B$.

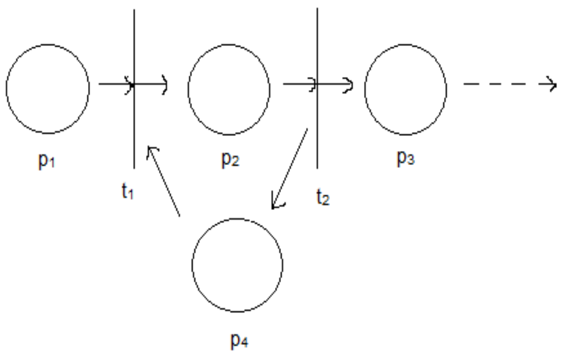


Figure 3.11: Adding Conditions in Buffering

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.

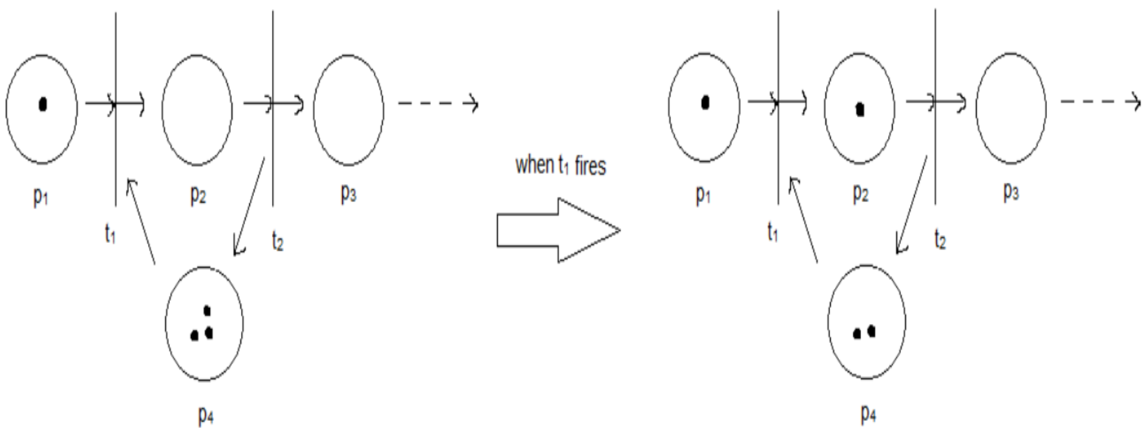


Figure 3.12: Petri Net representation

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 $M' \in R(PN, M_0)$, $M'(p_i) \leq 1$. 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

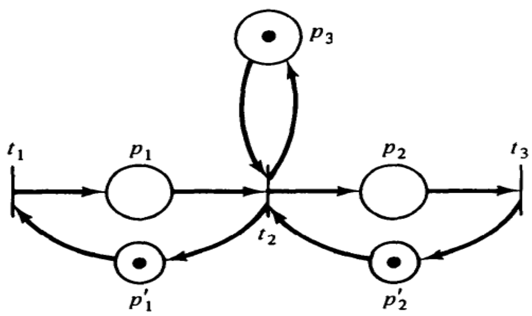


Figure 4.1: Safe-Petri Net

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 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** [6] 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 ,

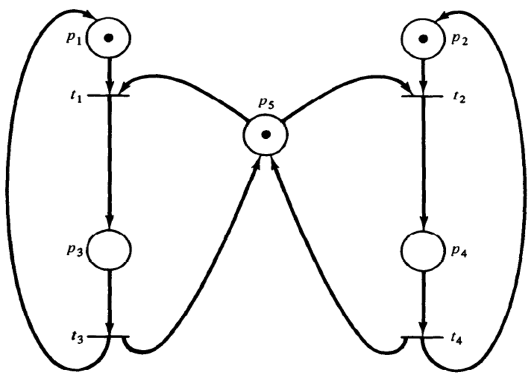


Figure 4.2: Not Strictly Conservative.

Here the enabled transitions are t_1 and t_2 .

$$\begin{aligned}
M_0 &= (1,1,0,0,1) \ \ (initial\ \ marking) \\
M_1 &= (0,1,1,0,0) \ \ (On\ firing\ t_1) \\
M_2 &= (1,1,0,0,1) \ \ (On\ firing\ t_3) \\
M_3 &= (1,0,0,1,0) \ \ (On\ firing\ t_2) \\
M_4 &= (1,1,0,0,1) \ \ (On\ firing\ t_4)
\end{aligned} \tag{4.3.1}$$

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 $M^{'} \in R(PN, M_0)$,

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

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

$$\begin{aligned} (1,1,0,0,1).(1,1,2,2,1) &= (1,1,0,0,1) \\ (0,1,1,0,0).(1,1,2,2,1) &= (0,1,2,0,0) \\ (1,0,0,1,0).(1,1,2,2,1) &= (1,0,0,2,0) \end{aligned} \tag{4.4.1}$$

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 $M' \in R(PN, M_0)$ and t_i is enabled in M'.

Remark 4.5.4. Demonstration of Deadlock Property.
Consider the example of resource allocation for two processes and two resources below,

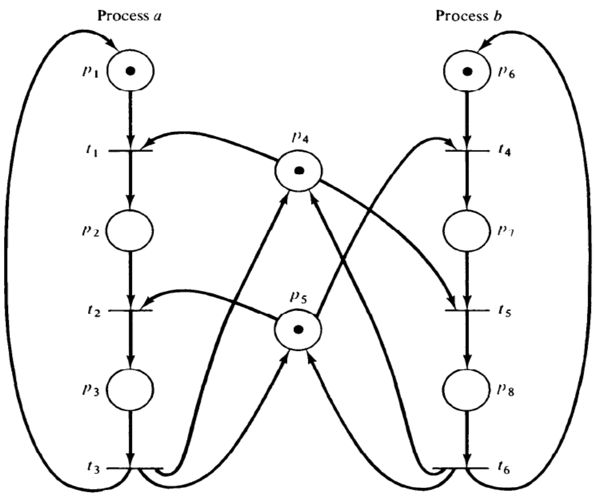


Figure 4.3: Demonstration of Deadlock

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.

4.6 Reachability

Definition 4.6.1. Reachability Problem
The reachablity problem considers a Marked Petri Net PN with initial marking M_0 and a marking $M^{'}$ .Then it aims at answering if $M^{'}$ is reachable from M_0 , that is if $M^{'} \in R(PN, M_0)$?
This is an important property to analyse ,consider the previous example in Fig 4.3. In this example we can see that for the marking $M^{'} = (0, 1, 0, 0, 0, 0, 1, 0)$ a deadlock will appear, so we would want to know whether from the initial marking is $M^{'}$ reachable.

4.7 Coverability

Definition 4.7.1. The coverablity Problem [5]-[7]
Consider a Petri Net PN with an initial marking M_0 and a marking $M^{'}$, then is there a reachable marking that is , $M^{''} \in R(PN, M_0)$ such that $M^{''} \geq M^{'}$?

Remark 4.7.2. This property is important if we want to consider the scenario where we want to ignore the contents of same places and would want to focus on covering the contents or items of only few relevant places.

Chapter 5

Analysis Technique-Reachability Tree

5.1 Introduction

The reachability tree/graph is an analytic technique for representing the reachability set of the Petri Net.
If we want to construct the reachability tree for a given Petri Net , then we need to consider the marking of that Petri Net at that time as a node and an arc would represent firing of transition from the initial marking to the subsequent new set of markings.
Consider the Petri Net in the Fig 5.1 given below,

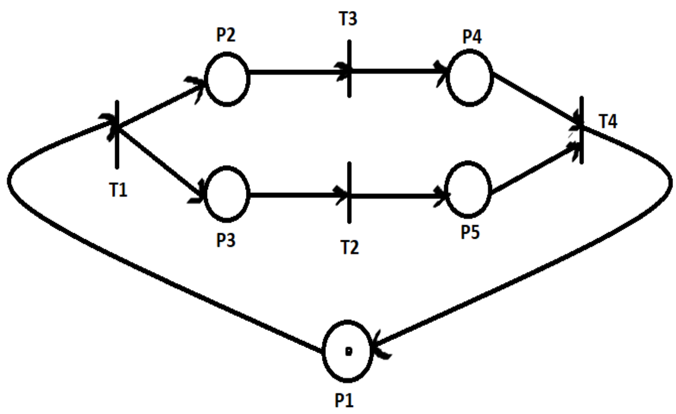


Figure 5.1: A Petri Net Model

If we want to draw the ***reachability graph*** of the Petri Net, then consider the initial marking $M_0 = (1, 0, 0, 0, 0)$, fire the enabled transition T_1 from here, and the new markings are obtained. Fig 5.2 show how the reachability graph gets constructed following this procedure for the Petri Net model of Fig 5.1.

Many a times we want to analyse the Petri Net in such a way that the direct relations in the markings are visible and the digraph has no cycles, thus we draw it as a **reachability tree**, where the nodes might get repeated , thus deleting the existing cycles. The reachability tree for the previous example of Fig 5.1 is shown in Fig 5.3.

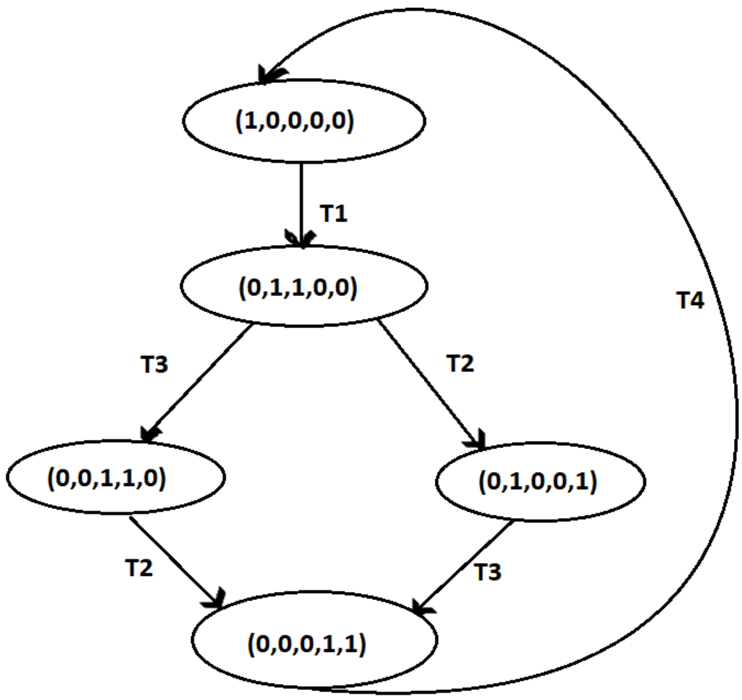


Figure 5.2: Reachability graph of Petri Net model.

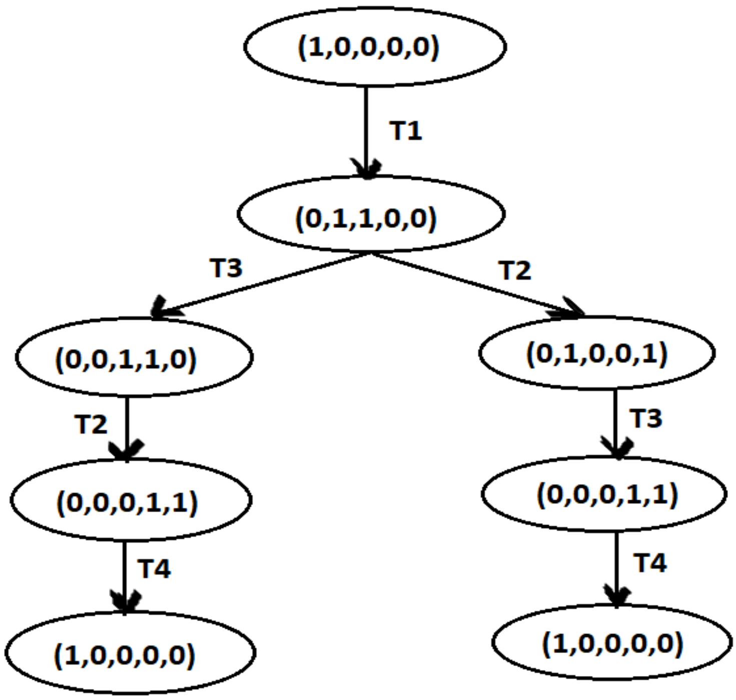


Figure 5.3: Reachability tree with terminaing nodes.

5.2 Scenarios

Here forward we shall be discussing two possible scenarios which are possible for constructing the reachability set and the reachability tree of a given Petri Net.

Consider the following Petri Net in Fig 5.4,

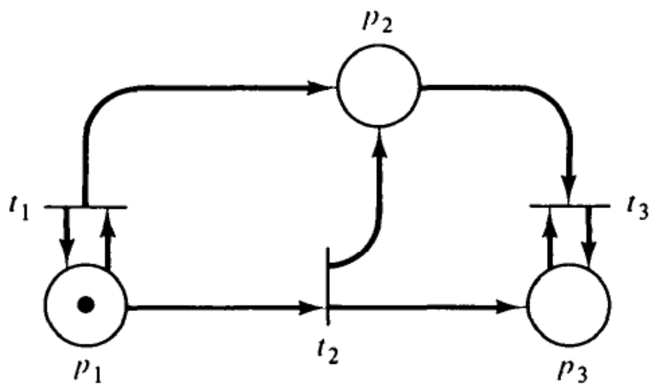


Figure 5.4: A Petri Net

The reachability tree for the above Petri Net is given by ,

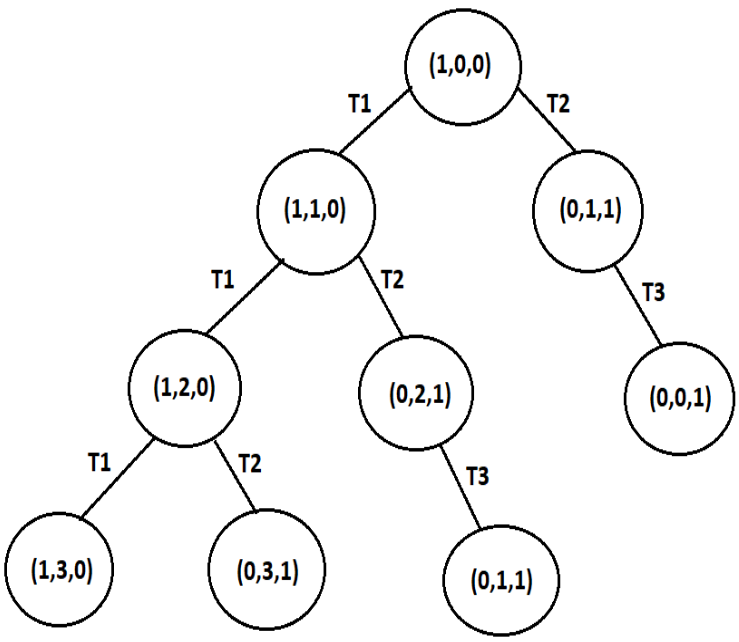


Figure 5.5: Reachability Tree

On repeating this process again and again in order to proceed, at every stage new marking will be produced, and we will get an **Infinite Reachability set** and thus **Infinite Reachability tree** .
In order to perform analysis for the Petri Net model we need to limit the size of the tree to a finite one.

5.3 Scenario-2

Another possibility that can exist is when the finite reachability set can also produce an infinite reachability tree.
Consider the following example in Fig 5.6,

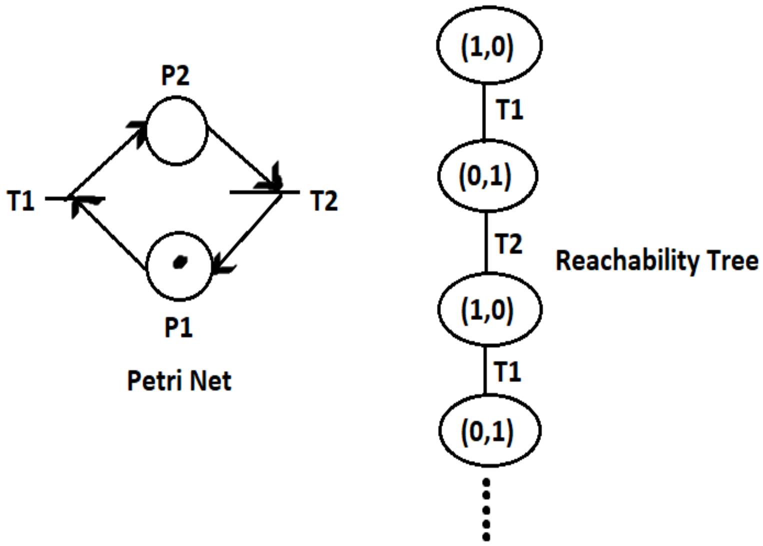


Figure 5.6: Finite reachability set and infinite reachability tree

In this particular example the reachability set is $\{(1,0),(0,1)\}$ which is finite , but the reachability tree here is alternative in nature and is infinite.

5.4 Finite Representation of Infinite Reachability Tree

In order to represent the infinite reachability tree in a finite way, the limitation of new marking is done at each step.The nodes in the infinite tree are categorized in such a way that they limit themselves in a finite manner.

Definition 5.4.1. (Terminal Node)
They are the nodes that are represented by dead markings.
Dead Markings: Markings where no transitions are enabled.
In this way extension of a marking is not pursued once we reach the terminal node since no transitions are enabled and thus no transition can be fired.

Definition 5.4.2. (Duplicate Nodes)
They are represented by duplicate markings.
Duplicate Marking: Set of markings that have previously appeared in the tree.
The markings are not extended further once duplicate nodes are detected since successors of these markings have already been produced in the first occurrence.

5.5 ω Representation

Consider the sequence of transition firing , α which starts from M_0 and ends at the marking M' such that $M' > M_0$.

$$\text{i.e. } M_0 \rightarrow \alpha \rightarrow M'$$

The marking M' is same as M_0 except that it has some extra tokens in some of the places.

$$\text{i.e. } M' = M_0 + (M' - M_0) \rightarrow \ extra\ tokens\ (> 0)$$

If α is fired again ,this time from M' , then again $M' - M_0$ tokens will be added to the marking M'.

$$\text{i.e. } M' \rightarrow \alpha \rightarrow M''$$
$$M'' = M' + (M' - M_0) = M_0 + 2(M' - M_0)$$

In general if we fire α sequence of transitions n times then we obtain the marking $\mathbf{M_0 + n(M' - M_0)}$ .Thus for the places which have gained tokens from the sequence α , arbitrarily large number of tokens can be accumulated just by reiterating the sequence of transitions again and again.
This arbitrarily large number of tokens are represented by ω.

Hence for each marking the number of tokens in a place are either non negative integer or ω .Terminal nodes , duplicate nodes along with ω representation restrict the infinite tree to a finite one. In the previous example of Fig 5.5 , where we were obtaining an infinite reachability tree , after again applying the finite representation techniques , the new tree that we get is shown in Fig 5.7

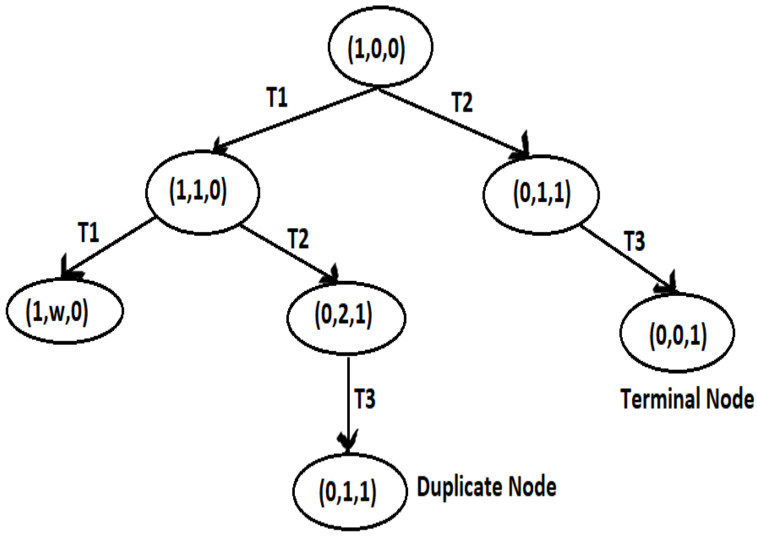


Figure 5.7: Finite reachability tree

Firing t_1 as many times , arbitrary number of tokens can be built in P_2.

5.6 Analyzing properties of Petri Nets through Reachability Tree.

- A Petri Net is **bounded** iff the symbol ω never appears in the reachability tree. Hence if the Petri Net is bounded , it represents the finite state system.Thus to determine the bound for a particular place , we need to draw the reachability tree and examine the tree for the largest value of the tokens in the markings corresponding to that place.

  Thus the reachability tree helps in determining the **boundedness** or **safeness** property for individual places in the Petri Net , or the entire Net .

- Since the reachability tree is finite , **conservation** can be easily tested by computing the weighted sum of tokens in places for each marking , if the weighted sum is constant and same for all subsequent markings , then the Petri Net is strictly conservative .

- If the symbol ω appears for any place in a marking and the corresponding element of the weighing vector for that place is 0 , then there can be a scope for the Petri Net to be conservative , but if the weight is positive of the element of the weighing vector for any place then the Net will not be conservative under any circumstances.
  Since now the symbol ω tells us that the number of tokens for some place can be arbitrarily increased, thus clearly the Petri Net won't be conservative.

- **Coverability Problem** can also be solved through reachability tree by inspecting and scanning the reachability tree since all we wish is to determine for a given marking , M' is, if the marking $M'' \geq M'$ is reachable or not .

- A Petri Net is **deadlock free** iff the reachability graph of the Petri Net has no node(represented by a marking) without an outgoing arc.

  Consider the example in Fig 5.4, here the reachability graph of the petri net has no node which does not have an outgoing arc hence it is deadlock free.

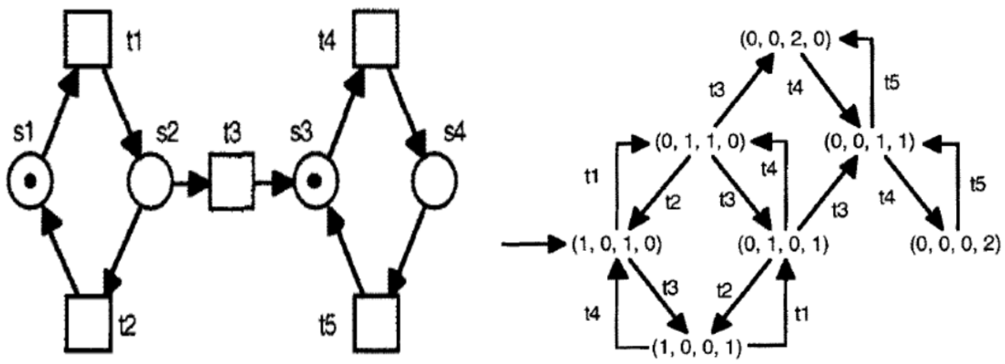


Figure 5.8: Petri Net and its reachability graph.

- A Petri Net is **live** iff for each node of the reachability graph of the corresponding Petri Net, there exists a path

$$\text{i.e. } M_0 \to t_1 \to M_1 \to t_2 \to M_2 \cdots M_{i-1} \to t_i \to M_i$$

  Here the sequence of path $t_1, t_2, \cdots t_i$ is such that it contains all the transitions of the Petri Net.

Chapter 6

Analysis Technique - Matrix Equations

6.1 Introduction

Another technique for analysis of Petri Nets is Linear Algebra methods known as **Matrix Equations**. A Petri Net $PN = (P, T, I, O)$ can be defined as the tupple $PN = (P, T, A^-, A^+)$ where A^- is the matrix for input function and A^+ is the matrix for output function of the Petri Net.
The matrix A^- and A^+ have transitions as the rows $(t_1, t_2, ..., t_m)$ and the places of the Petri Net as columns of the matrix $(p_1, p_2, p_3, ..., p_n)$ .Thus both input and output matrix are matrices of order $m \times n$.

$$\mathbf{A^-[j, i] = \#(p_i, I(t_j))}$$

Here the quantity represents inputs to transition t_j from place p_i , where $j = (1, 2, ..., m)$ and $i = (1, 2, .., n)$.

$$\mathbf{A^+[j, i] = \#(p_i, O(t_j))}$$

here the quantity represents outputs from transition t_j to place p_i.
Let $e[j]$ be the m unit vector which is zero everywhere except the jth element in the tupple .The transition t_j is expressed through this unit m vector $e[j]$.
$e[j] = (0, 0, 0, ...1, 0, 0, 0....0)_{1\times m}$.The unit entry is at the jth position in the m tupple.

Now a transition t_j is enabled in a marking M if ,

$$\mathbf{M_{(1\times n)} \geq e[j]_{(1\times m)}.A^-_{(m\times n)}}$$

If t_j is fired in the marking M then the next state function is given by,

$$\phi(\mathbf{M, t_j}) = \mathbf{M - e[j].A^- + e[j].A^+}$$

Since earlier we saw that $M^{'}(p_i) = M(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$.
Hence $\phi(M, t_j) = M + e[j].(A^+ - A^-) = M + e[j].A$
where $A = A^+ - A^-$ is the **Composite Change Matrix** [6].

Now consider a sequence of transition firings $\alpha = t_{j_1}t_{j_2}.....t_{j_k}$ .Then

$$\begin{aligned}\phi(M,\alpha) &= \phi(M, t_{j_1}t_{j_2}.....t_{j_k}) \\ &= M + e[j_1].A + e[j_2].A + ... + e[j_k].A \\ &= M + (e[j_1] + e[j_2] + ... + e[j_k]).A\end{aligned} \tag{6.1.1}$$

Let $g(\alpha) = e[j_1] + e[j_2] + ... + e[j_k]$ be **the firing vector** of sequence $t_{j_1}t_{j_2}.....t_{j_k}$ .The ith element of $g(\alpha)$ would give the number of times the transition t_j has been fired in sequence $t_{j_1}t_{j_2}.....t_{j_k}$ .Thus $g(\alpha)$ is the vector of non negative integers.

6.2 Reachability Problem through Matrix Equations

If M' is reachable from a marking M then there exist a sequence of transition firings which will lead M to M' , i.e. for $M' = M + x.A$ .Here x is a solution vector of non-negative integers. Consider the Petri Net,

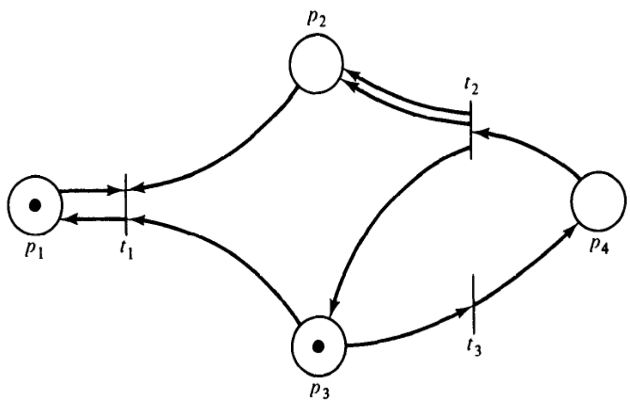


Figure 6.1: A Petri Net Model for Matrix Analysis

The initial marking in the above Petri Net of Fig 6.1 is $M = (1, 0, 1, 0)$. The input and output matrices are given by ,

$$A^- = \begin{bmatrix}1 & 1 & 1 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix} \quad \text{and} \quad A^+ = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 2 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix}$$

The composite matrix is given by

$$A = A^+ - A^- = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 2 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} - \begin{bmatrix}1 & 1 & 1 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix} = \begin{bmatrix}0 & -1 & -1 & 0\\0 & 2 & 1 & -1\\0 & 0 & -1 & 1\end{bmatrix}$$

From the figure of the Petri Net model we can see that if t_3 is fired then the new marking would be $M' = (1, 0, 0, 1)$. i.e. $M \to t_3 \to M'$.

Remark 6.2.1. Now we obtain this result from matrix analysis.

$$\begin{aligned} M^{'} &= (1,0,1,0) + (0,0,1).\begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\ &= (1,0,1,0) + (0,0,-1,1) \\ &= (1,0,0,1) \end{aligned} \tag{6.2.1}$$

Thus we we have validated the above result through matrix analysis .

Remark 6.2.2. If the sequence of firing transitions is given

i.e. $M \to \alpha \to M^{'}$ and $\alpha = t_3 t_2 t_3 t_2 t_1$

$$\begin{aligned} g(\alpha) &= e[t_3] + e[t_2] + e[t_3] + e[t_2] + e[t_1] \\ &= e[t_1] + 2e[t_2] + 2e[t_3] \\ &= (1,2,2) \end{aligned} \tag{6.2.2}$$

This $g(\alpha) = (1,2,2)$ is the **Firing Vector**. Now to obtain the new marking after applying this sequence of transition firing ,

$$\begin{aligned} M^{'} &= (1,0,1,0) + (1,2,2).\begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\ &= (1,0,1,0) + (0,3,-1,0) \\ &= (1,3,0,0) \end{aligned} \tag{6.2.3}$$

Hence after firing the sequence of transitions α to the initial marking M , the resultant marking obtained is $M^{'} = (1,3,0,0)$.

Remark 6.2.3. It can be shown that the marking (1,7,0,1) is not reachable from (1,0,1,0)

$$\begin{aligned} (1,7,0,1) &= (1,0,1,0) + x.\begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\ (0,7,-1,1) &= (0,0,0,0) + x.\begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \end{aligned} \tag{6.2.4}$$

For the above equation x has no solution , and since x was the firing vector .Thus we know that no such sequence of transitions exist such that (1,7,0,1) is reachable from (1,0,1,0).

6.3 Matrix Approach for Conservation Problem.

For given Marked Petri Net, if we want to know whether the Petri Net is conservative or not , then matrix approach can be applied. For conservation, we must find a non zero weighing vector u for which the weighted sum of tokens over all reachable markings is constant.

If u is a $n \times 1$ vector and M_0 is the initial marking and $M^{'}$ is the arbitrary reachable marking from M_0 then for conservation ,

$$M_0.u = M^{'}.u$$

Since $M^{'}$ is reachable from M_0 then after a sequence of transitions firing α we get,

$$\begin{aligned} M^{'} &= \phi(M_0, \alpha) \\ &= M_0 + g(\alpha).A \end{aligned} \tag{6.3.1}$$

Now since for conservation , $M_0.u = M^{'}.u$, therefore ,

$$\begin{aligned} M_0.u &= (M_0 + g(\alpha).A).u \\ M_0.u &= M_0.u + g(\alpha).A.u \\ \Rightarrow g(\alpha).A.u &= 0 \ \ (true\ for\ all\ g(\alpha)) \\ A.u &= 0 \end{aligned} \tag{6.3.2}$$

Hence we can formulaise **A test for conservation of Petri Net.**

Definition 6.3.1. A Petri Net is said to be **conservative** iff there exists a vector u of positive integers such that A.u=0 , where A is the composite change matrix (or the incidence matrix) of the given Petri Net.

Definition 6.3.2. A Petri Net is said to be **partially conservative** iff there exists a vector u of non negative integers such that A.u=0 , where A is the composite change matrix (or the incidence matrix) of the given Petri Net.

Thus through matrix approach we can directly obtain tests for conservation (partial conservation) of Petri Net.

6.4 Issues in Matrix Analysis Approach

6.4.1 Lack of sequencing information in Firing Vector

Consider the following Petri Net in Fig 6.2 ,
Here the initial marking is $M_0 = (1, 0, 0, 0, 0)$. If we want to know that if (0,0,0,0,1) is reachable from (1,0,0,0,0) then ,

$$(0, 0, 0, 0, 1) = (1, 0, 0, 0, 0) + x. \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 1 \end{bmatrix} \tag{6.4.1}$$

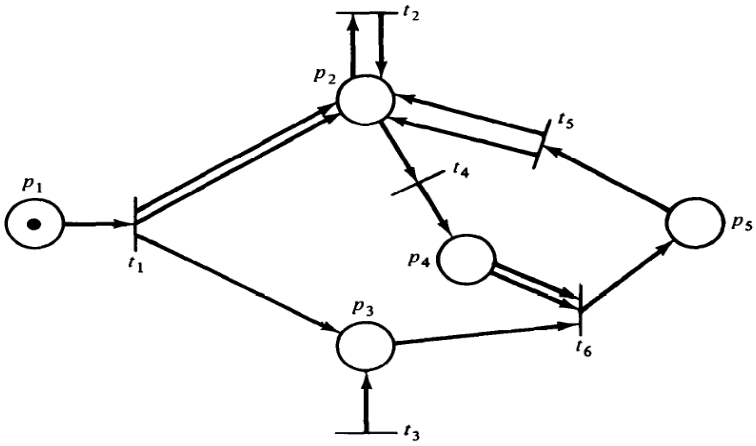


Figure 6.2: Example of Petri Net Model

This equation , does not have a unique solution.

The solution for x is $(1, x_2, x_6 - 1, 2x_6, x_6 - 1, x_6)$. If $x_6 = 1, x_2 = 1$ and now solving for x we will get the firing vector $\mathbf{g}(\alpha) = (\mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{2}, \mathbf{0}, \mathbf{1})$.
Hence this implies that $t_1t_2t_4t_4t_6$ and $t_1t_4t_2t_4t_6$ correspond to the same vector.Hence we just know about transition firings and nothing about the order of transitions firings.

6.4.2 Solution to Matrix Equation is necessary for reachability but not sufficient

Solving for x in $M' = M + x.A$ in reachability problem is necessary but not sufficient condition . Consider the Petri Net in Fig 6.3,

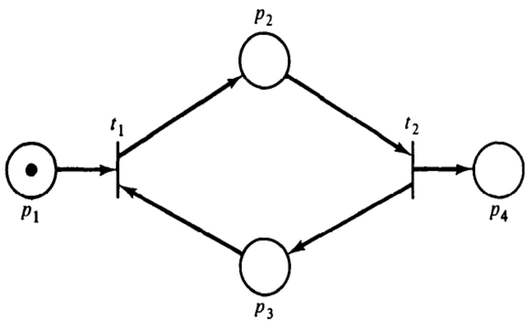


Figure 6.3: A Petri Net

Here the initial marking is (1,0,0,0). If we want to know that (0,0,0,1) is reachable from (1,0,0,0) ,

$$(0,0,0,1) = (1,0,0,0) + g(\alpha).\begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} \tag{6.4.2}$$

$\Rightarrow g(\alpha) = (1, 1)$ corresponds to the sequence t_1t_2 or t_2t_1. But if we observe the Petri Net we see that neither of these two transitions are possible since t_1 and t_2 are not enabled in the initial marking (1,0,0,0).

Thus a solution to the matrix equation , $M^{'} = M_0 + x.A$ is not sufficient to prove reachability of the marking.

Remark 6.4.1. The incidence matrix as used in linear algebra techniques helps in understanding the dynamic behavior of a system.

1. It provides test for conservative property.
2. It can tell if a marking is reachable from an initial marking or not.
3. The solution to the matrix equations is limited since the rank of the composite matrix may not always be full.Thus the structural conclusions can not be drawn in this case. To counter this , adding counter places in the composite matrix was used but was discarded as a more sophisticated concept of transitive matrix was introduced.
4. Nothing much about the behavioral properties (except reachability) can be derived through incidence matrix.

The limitations of the matrix analysis techniques paved way for an improved analysis using the labelled place (or transition) transitive matrices, which shall be discussed in the subsequent chapter.

Chapter 7

Transitive Matrix

7.1 Petri Nets conversion into Directed Graphs

A graph $G = (V, E)$ is a directed graph for a Petri Net PN, where the vertex set $V = P \cup T$ and $E \subset (P \times T) \cup (T \times P)$ is the set of directed arcs.
Another graph $G_1 = (V_p, E_p)$ (or $G_2 = (V_T, E_T)$) is known as a ***place (transition) transitive graph***, where $V_p = P$ (or $V_T = T$) is the set of all vertices and $E_p =$ Bag(T) (or $E_T =$ Bag(P)) is the bag of directed arcs,

Further, $t_k \in E_p(or p_k \in E_T)$ is the input and output transition (or place) of p_i(or t_i) and p_j (or t_j) respectively.
Below is an example to illustrate such a conversion.

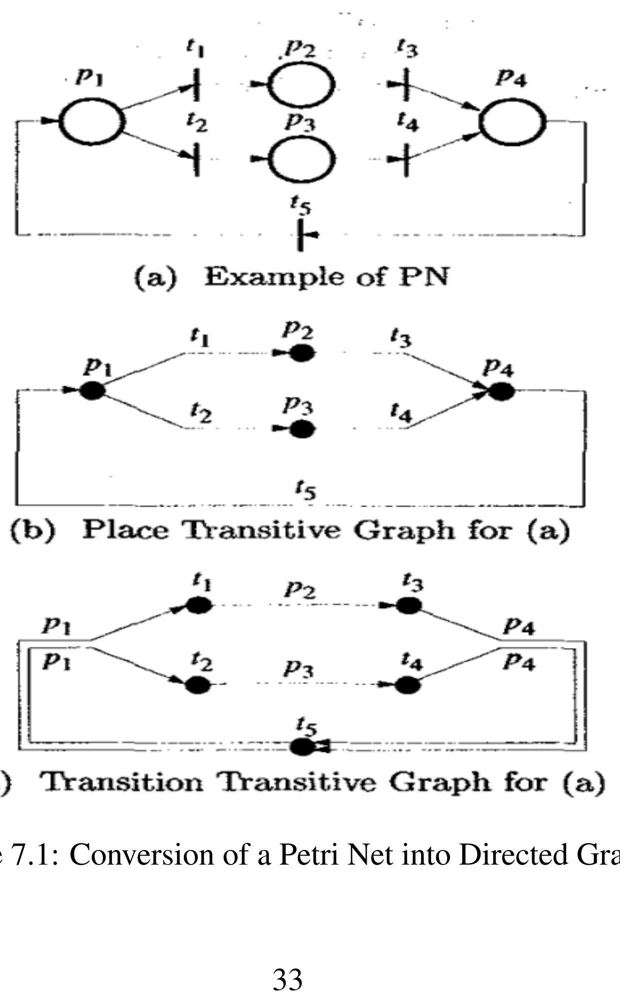


Figure 7.1: Conversion of a Petri Net into Directed Graph

7.2 Concept of Transitive Matrices

The ideology of transitive matrices [4] were introduced in order to understand and analyse the properties of a Petri Net.

Before moving forward, we define the following-

Definition 7.2.1. (Adjacent Matrix)

$$AM = \begin{bmatrix} 0 & (B^+)^T \\ B^- & 0 \end{bmatrix} \tag{7.2.1}$$

where $B^-[i,j] = \#(p_i, T(t_j)) \quad ; \quad B^+[i,j] = \#(p_i, O(t_j))$ are the $m \times n$ matrices. The incidence matrix B is defined by $B = B^+ - B^-$.

Definition 7.2.2. (Transitive Matrix)

$$TM = AM.AM = \begin{bmatrix} (B^+)^T B^- & 0 \\ 0 & B^-(B^+)^T \end{bmatrix} \tag{7.2.2}$$

where $TM_p = (B^+)^T B^-$ and $TM_t = B^-(B^+)^T$ are defined to be the ***place transitive and transition transitive matrices*** respectively. Further defined are the ***Labelled place transitive and transition transitive matrices*** as,

$$L_{BP} = (B^+)^T D_t B^- \tag{7.2.3}$$

$$L_{BT} = (B^-) D_P (B^+)^T \tag{7.2.4}$$

Here, $D_t = \text{diag}(t_1, t_2, ..., tn)$ and $D_p = \text{diag}(p_l, p_2, ..., p_m)$ and t_i and (or) $p_j = 1$ if the transition can fire and (or) the place has a token respectively, else $t_i, p_j = 0$.

Remark 7.2.3. Results drawn in [4] discuss well the use of the Labelled place transitive and transition transitive matrices to analyse and draw inferences from various Petri Nets.

7.3 Issues in Transitive matrix approach

It was observed that the Transitive matrix approach failed for Petri Nets which had a source and (or) sink transition(s).The calculated L_{BP} could not include all the possibly appearing transitions of such a Petri Net.

This can be inferred as, say if same transition, say $t_k \in T$ of a Petri Net PN is a source transition which has no input place and $p_m \in O(t_k)$, then the corresponding m^{th} column entries of $L_{BP} = 0$. Similarly, for a sink transition $t_k \in T$ which has no output place and $p_m \in I(t_k)$, then the corresponding m^{th} row entries of $L_{BP} = 0$.

Hence such a labeled place-transition transitive matrix is insufficient to describe such a Petri Net. To support the same, consider the following Petri Nets in Fig 7.2.

Using equations (7.2.1), (7.2.2), (7.2.3), we can calculate the Labeled Place transitive matrix for the above three considered Petri Nets. It is evaluated to be,

$$L_{BP} = (B^+)^T D_t B^- = \begin{bmatrix} 0 & 0 & t_1 \\ 0 & 0 & t_2 \\ 0 & 0 & 0 \end{bmatrix}.$$

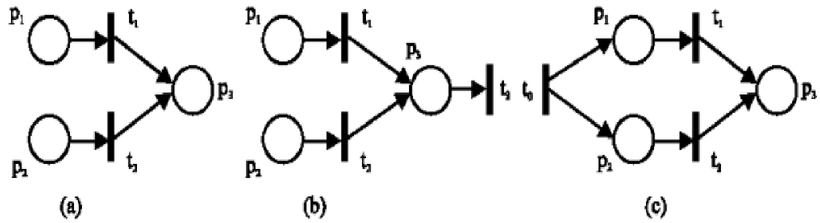


Figure 7.2: (a) PN without source/sink transition. (b) With source transition. (c) With sink transition.

$$\text{For 7.2(a),} \quad B^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } B^- = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$
$$\text{For 7.2(b),} \quad B^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } B^- = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$
$$\text{For 7.2(c),} \quad B^+ = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } B^- = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

The same value of L_{BP} for three different Petri Nets as shown above depicts the vagueness and uncertainty while predicting the behaviour of a particular type of a Petri Net uniquely.

Remark 7.3.1. The labeled place (transition)-transitive matrix is defined as $L_{BP} = I^t.D_t.O$, where D_t is the diagonal-transition matrix for $T = \{t_1, t_2, \ldots, t_n\}$ and I, O are the input and output matrices respectively. The entry in the i^{th} row and j^{th} column of labeled place transitive matrix ($L_{BP}[i, j] = t_k$) denotes a transitive relation from the input place p_i to the output place p_j by firing the transition t_k. L_{BP} fails to include all source/sink transitions and thus, limits the analysis to Petri Nets without source/sink transitions.

Chapter 8

Transition Vectors

As discussed in the previous chapter, the concept of ***transition vectors*** [2] was introduced for more preciseness and uniqueness while analysing a Petri Net model. This proves to be a useful tool and allows to encounter all types of Petri Nets, with or without source and (or) sink transitions.

8.1 What are Transition Vectors?

Definition 8.1.1. **(Transition Vectors)** For a Petri Net PN with m-places and n-transitions with L_{BP} as its labeled place transitive matrix, the transition vectors are defined in two fashions.

1. The row m-vector of transitions where $T_R : P \to I(p_i), I : P \to T$ is input function is defined as,

$$T_R = \left[\sum_{i=1}^{m} L_{BP}[i,1] \quad \sum_{i=1}^{m} L_{BP}[i,2] \quad \cdots \sum_{i=1}^{m} L_{BP}[i,m] \right]$$

2. The column n-vector of transitions where $T_C : P \to O(p_i), O : P \to T$ is output function is defined as,

$$T_C = \left[\sum_{j=1}^{m} L_{BP}[1,j] \quad \sum_{j=1}^{m} L_{BP}[2,j] \quad \cdots \sum_{j=1}^{m} L_{BP}[m,j] \right]^t$$

To understand this literally, what we mean to say is that, in general, the k^{th} component of T_R is nothing, but the set of all input transitions for the place p_k denoted by $T_R(p_k)$.

Similarly, in general, the k^{th} component of T_C is nothing, but the set of all output transitions for the place p_k denoted by $T_C(p_k)$.

These components of both T_R and T_C are finite linear combinations of transitions with positive coefficients, which indeed represent the number of incoming arcs and output arcs in T_R and T_C respectively.

8.2 Analysis of various properties of Petri Nets

A Petri Net can be analysed and inferred with respect to various properties [1], [2],[4]. These properties can be broadly classified into-

1. Structural Properties : Static in nature, these are dependent on nodes and arcs and independent of tokens availability.
   - Structural conservation (partial)
   - Cyclic/ acyclic
   - Structural Conflict(or free)
   - State Machine
   - Self-loop (or free)
   - Structural Concurrency
2. Behavioral Properties : Dynamic in nature, these are dependent on the initial marking / availability of tokens.
   - Behavioral conservation
   - Safeness and Boundedness
   - Behavioral Conflict
   - Reachability
   - Deadlock and Liveness
   - Behavioral Concurrency

8.2.1 Cyclic / Acyclic Nature

A Petri Net is acyclic if T_R and (or) T_C has at least one zero component. Also, a Petri Net is acyclic if and only if it has at least one source and (or) one sink place. [2]

8.2.1.1 Algorithm to find directed cycle

The transition vectors (row and column) are computed as input.

1. Consider the ith element (place) of the column transition vector.
2. If the considered place is same as previous place then step -6, else next step.
3. Select the transition of the selected(previous) place in the column transition vector.
4. Locate the same transition in the row transition vector.
5. Go to step 2.
6. Directed cycle is obtained.
7. END.

8.2.2 Conflict (or free)

A Petri Net is conflict-free if and only if every component of T_C has exactly one transition, or, in case any component has more than one transition, the same transitions set must appear in the corresponding component of T_R. [2]

8.2.3 State Machine

A Petri Net is said to be state machine if and only if all the components of T_R and T_C are respectively distinct. [2]

8.2.4 Self loop (or free)

A Petri Net is said to be free of a self loop i.e. pure if and only if the corresponding components of both T_R and T_C have no transitions that are same or identical at a certain place p_k. [2]

8.2.5 Structural Concurrency

A Petri Net is said to be structurally concurrent if and only if at least one entry of the column transition vector T_C has coefficient 2 or greater than 2. [2]

8.2.6 Demonstration of transition vector results through an example.

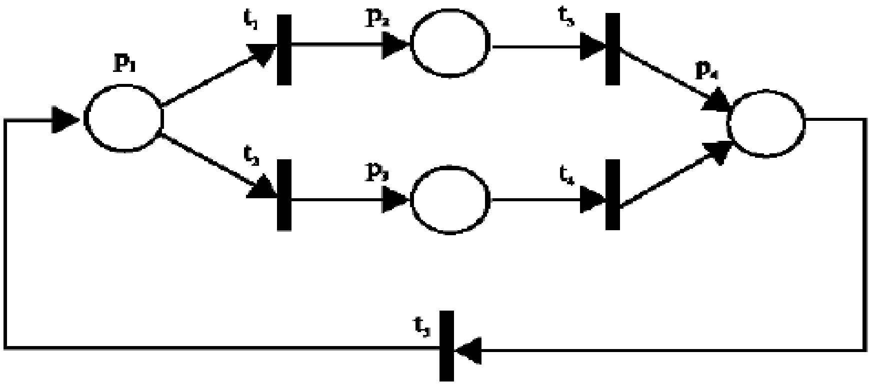


Figure 8.1: Example of a Petri Net.

The labelled place transitive matrix for the above example is given as,

$$L_{BP} = \begin{bmatrix} 0 & t_1 & t_2 & 0 \\ 0 & 0 & 0 & t_3 \\ 0 & 0 & 0 & t_4 \\ t_5 & 0 & 0 & 0 \end{bmatrix}$$

The transition vectors T_R and T_C for the above example in Fig. 8.1 is given by ,

$$T_R = [t_5 \quad t_1 \quad t_2 \quad t_3 + t_4]$$

and

$$T_C = [t_1 + t_2 \quad t_3 \quad t_4 \quad t_5]^t$$

- Since the transition vectors don't have any zero component thus, the PN in the given example is cyclic.
- The Petri Net graph is self loop free since the corresponding distinct components of row transition vector and column transition vector are different.
- The elements of row transition vectors and column transition vectors are different thus, it can be implied that they have only one (input place) and one (output place) and hence it is a state machine.
- The first element of T_C is $t_1 + t_2$ and hence t_1 and t_2 are in conflict and have choice for firing when the token is available in p_1. The firing sequence $(t_1.t_3)$ or $(t_2.t_4)$ can take place.
- Using the algorithm for directed cycle , we get directed cycles as,

$$p_1 \to t_1 \to p_2 \to t_3 \to p_4 \to t_5 \to p_1$$

and

$$p_1 \to t_2 \to p_3 \to t_4 \to p_4 \to t_5 \to p_1$$

Remark 8.2.1. The transition vectors, T_R and T_C, and results mentioned based on it, as suggested in [2], have been used to overcome the ambiguous situation as determined in transitive transition matrices. . The ith components of T_R and T_C give the set of input and output transitions of p_i, respectively. These components of the transition vectors are a finite linear combination of $t_k \in T(k = 1, 2, \ldots, n)$ with positive integer coefficients.

Chapter 9

A Petri Net model of a Small Eatery

Given the spread of the COVID-19 virus, social distancing is the need of the hour. Here, designed and proposed is a small eatery model that is supposed to provide service to its customers while practicing social distancing and allowing a minimum number of people together at any possible stage.

Thus, this proposed model of a small eatery that originally had four tables has revised the seating plan to provide service at two alternate tables while taking all necessary precautions against the virus and ensuring social distancing.

Further, for the arrival of any new customer at the eatery, two separate waiting places p_9, p_{10} have been marked, which shall eventually pave the way to the respective table (either the first or the second respectively). The model has been designed such that the new customers cannot occupy the tables unless the previous customers who are seated on the tables eat and vacate.

Also, the seated customer shall place the order once. It is assumed and supposed that the customer at a certain waiting place cannot shift to the other waiting place w.r.t the other table.

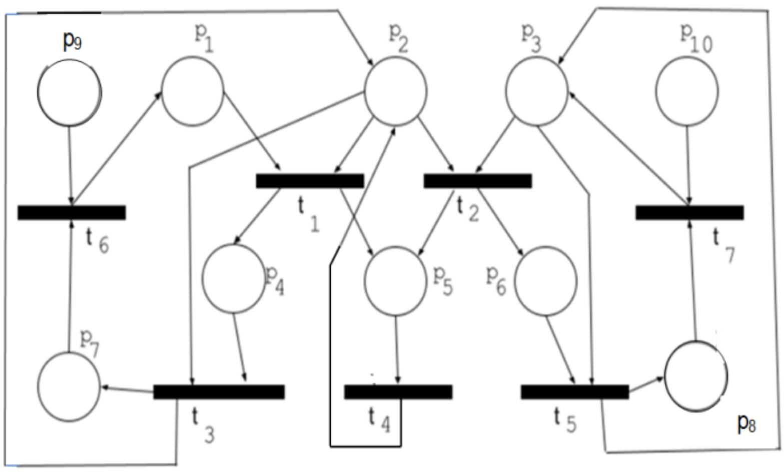


Figure 9.1: Petri Net graph of the proposed model.

The above figures shows the Petri Net graph of the proposed model of the small eatery where $P = \{p_1 p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$ and $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ are the places (conditions) and transitions (events) respectively.

The presence of a token at any 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).

PLACES	CONDITIONS
p_1	Customer is seated at sanitized first table and is then ready to place the order.
p_2	The service provider (waiter) is free and can provide service at either table.
p_3	Customer is seated at sanitized second table and is then ready to place the order.
p_4	The customer at first table is waiting for the food to be served.
p_5	The order is finalized by the service provider (waiter).
p_6	The customer at second table is waiting for the food to be served.
p_7	The customer at first table eats and vacates the place.
p_8	The customer at second table eats and vacates the place.
p_9	A new customer has arrived to occupy the first table.
p_{10}	A new customer has arrived to occupy the second table.

Table 9.1: Depiction of places (conditions) of the proposed model.

TRANSITIONS	ACTIVITIES
t_1	The service provider (waiter) takes orders from the first table.
t_2	The service provider (waiter) takes orders from the second table.
t_3	The service provider (waiter) serves the order at the first table.
t_4	The service provider (waiter) reports the order in the kitchen.
t_5	The service provider (waiter) serves the order at the second table.
t_6	First Table is sanitized and a menu card is placed.
t_7	Second Table is sanitized and a menu card is placed.

Table 9.2: Depiction of transitions (activities) of the proposed model.

A prior illustrative assumption for this hypothesis is that when the first two customers arrive, a token gets created at the places p_1 and p_3 respectively, and another token is created at the place p_2 signifying the availability of the service provider (waiter) to provide service.

Thereafter, whenever the new customer arrives, it is denoted in the model by the creation of a token at p_9 and p_{10} for the first and second table respectively.

9.1 Analysis and Interpretation of the proposed model

9.1.1 Structural Analysis

1. ***Conservation Property***

With an initial marking M_0, we call a Petri Net $PN = (P, T, I, O)$ conservative if for all $M^+ \in R(PN, M_0)$,

$$\sum_{(p_i \in P)} M^+(p_i) = \sum_{(p_i \in P)} M_0(p_i) \tag{9.1.1}$$

where, the reachability set $R(PN, M_0)$ for a Petri Net PN with the initial marking M_0 is the smallest set of markings defined as:

- $M^+ \in R(PN, M_0)$
- If $M^+ \in R(PN, M_0)$ and M^{++} is immediately reachable from M^+, then $M^{++} \in R(PN, M_0)$.

Thus by conservation, we mean conservation of the tokens in place i.e., the token count for all the places at any state remains constant. A Petri Net is structurally conservative (partially) [5] if there exists a positive (non-negative) vector w such that

$$D.w = 0 \quad \text{, where} \quad w \geq 0.$$

In order to find the incidence matrix D, we calculate the input matrix I, and the output matrix O for the proposed model as follows,

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

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

The incidence matrix $D = O - I$ is given by,

$$D = O - I = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix}$$

The above matrix can be expressed as the system of linear equations when substituted in $D.w = 0$. The following is the set of linear equations.

$$\begin{aligned} -w_1 - w_2 + w_4 + w_5 &= 0 \\ -w_2 - w_3 + w_5 + w_6 &= 0 \\ -w_4 + w_7 &= 0 \\ w_2 - w_5 &= 0 \\ -w_6 + w_8 &= 0 \\ w_1 - w_7 - w_9 &= 0 \\ w_3 - w_8 - w_{10} &= 0 \end{aligned}$$

Solving these equations, we get a solution as below;

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \\ 1 \\ 3 \\ 0 \\ 0 \end{bmatrix}$$

The existence of a non-negative vector w such that $D.w = 0$ implies that the Petri Net model is partially conservative.

This suggests that the Petri Net might be conservative w.r.t some initial marking and might not be conservative w.r.t some other initial marking. Hence the token count may vary after certain stages. This is further elaborated in the proposed model in subsequent section.

For studying more properties like conflict, concurrency, self-loop, state machine, cyclic/acyclic nature [3], [5]-[7], we compute the transitive matrix [4] and transition vectors [2] for the proposed model.

The Labeled Place-Transitive matrix, $L_{BP} = I^t.D_t.O$ is given by,

$$L_{bp} = \begin{bmatrix} 0 & 0 & 0 & t_1 & t_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_3+t_5 & 0 & t_1 & t_1+t_2 & t_2 & t_3 & t_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2 & t_2 & 0 & 0 & 0 & 0 \\ 0 & t_3 & 0 & 0 & 0 & 0 & t_3 & 0 & 0 & 0 \\ 0 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_5 & 0 & 0 & 0 & 0 & 0 & t_5 & 0 & 0 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The row-transition vector corresponding to the above labeled Place-Transitive matrix is given by,

$$T_R = [2t_6 \quad 2t_3+t_4+2t_5 \quad 2t_7 \quad 2t_1 \quad 2t_1+2t_2 \quad 2t_2 \quad 2t_3 \quad 2t_5 \quad 0 \quad 0]$$

The column-transition vector corresponding to the above labeled Place-Transitive matrix is given by,

$$T_C = [2t_1 \quad 2t_1+2t_2+2t_3+2t_5 \quad 2t_2 \quad 2t_3 \quad t_4 \quad 2t_5 \quad t_6 \quad t_7 \quad t_6 \quad t_7]^t$$

2. ***Cyclic/acyclic nature***

With the concept as mentioned in section (**8.2.1**), the transition vector corresponding to our proposed model T_R has zero components at the ninth and tenth place; thus, the Petri Net model is acyclic.

Also, a Petri Net is acyclic if and only if it has at least one source and (or) one sink place.

To understand the physical interpretation of the Petri Net's acyclic nature, let us suppose the Petri Net is not acyclic. Then there must exist a directed path $x_i \rightarrow x_j$, $\forall x_i, x_j \in P \cup T$. However, for $x_i = p_9$, there is no incoming arc; hence the directed path is not possible for p_9. Further, if p_9 and p_{10} would not have been source places, there would have been a cycle that would have repeated endlessly; hence the dynamics between the customer and the waiter would go on, without the possibility of the arrival of a new customer.

Moreover, it is inferred from [2] that the acyclic nature of Petri Net does not mean that it cannot contain a cycle. With the implementation of section (**8.2.1.1**), we can find a directed cycle $p_1 t_1 p_4 t_3 p_7 t_6 p_1$, in the proposed model. The same can be interpreted as the arrival of a customer at the first table; the service provider(waiter) taking his order; the customer waiting for food; the service provider(waiter) serving him; and finally, the customer eating and vacating the place, followed by the arrival of a new customer.

3. ***Self-loop free***

With the concept as mentioned in section (**8.2.4**), the transition vectors that correspond to our proposed model show that the PN is self-loop free.

From Fig. 9.1, it can be inferred that under any condition that happens to be accurate, it would not continue to occur an indefinite number of times. To support this, on the contradictory, let if possible, a self-loop exists at the place p_4 w.r.t the transition t_3. This self-loop can be interpreted as the customer waiting for the food, followed by being served; waiting for the food again; and the cycle repeats. Undoubtedly, this turns out to be an irrelevant situation making our assumption wrong. Thus, the proposed Petri Net model is self-loop free, and no condition/event shall occur indefinite number of times.

4. ***State Machine***

With the concept as mentioned in section (**8.2.3**), in the transition vector, T_R of the proposed model, the ninth and tenth component is identical, equal to zero.

Hence the model is not a state machine, which implies that it necessarily does not have one input and one output place for every transition i.e., more than one conditions can simultaneously hold true for an activity to occur.

For e.g., from Fig.9.2 it is evident that both p_1 and p_2 lead to the happening of t_1. Therefore, this shall give rise to either concurrency, or conflict, or both.

5. ***Conflict***

As it has been seen that the proposed Petri Net model is not a state machine, thus conflict may or may not arise. With the concept as mentioned in section (**8.2.2**), in the proposed model, the column transition vector T_C has more than one transitions in the first component, and the corresponding entry of T_R is different, implying that there is a conflict between t_1 and t_2 .Thus, the Petri Net model is not conflict-free.

All possible conflicts that can arise in the Petri Net model are between the transitions t_1, t_2, t_3, t_5. The preciseness of the conflicting transitions is mentioned in the behavioral aspect of conflict.

6. ***Concurrency***

With the concept as mentioned in section (**8.2.5**), since the column transition vector T_C of the proposed model has coefficient 2 in the entries of its first, second, third, fourth and sixth components, the transitions in these components along with the remaining transitions form structurally possible concurrent transitions.

All the possible concurrent transitions have been mentioned in Table 9.3 .

First Transition	Second Transition	Structural Concurrency
t_1	t_4	Yes
t_1	t_6	Yes
t_1	t_7	Yes
t_2	t_4	Yes
t_2	t_6	Yes
t_2	t_7	Yes
t_3	t_4	Yes
t_3	t_6	Yes
t_3	t_7	Yes
t_5	t_4	Yes
t_5	t_6	Yes
t_4	t_7	Yes

Table 9.3: List of all possible structural concurrent transitions.

9.1.2 Behavioral Analysis

There can be many possible scenarios for this Petri Net model to work, depending upon the customers (those being provided the service inside and those waiting outside) and the service provider at the eatery. We throw light on two possible scenarios for the proposed model.

9.1.2.1 Scenario 1

As per the first scenario, after the eatery is open to provide service at the initial stage, it has two customers seated on the sanitized first and second tables respectively, and the service provider (waiter) is free to serve the two customers. In this scenario, it is assumed that no new customer has arrived at the eatery The initial marking that depicts this scenario is $M_0 = \{1, 1, 1, 0, 0, 0, 0, 0, 0, 0\}$.

Fig. 9.2 depicts the Petri Net model for the above-depicted scenario with the initial marking M_0 and the set of all reachable markings are mentioned in Table 9.4 .

9.1.2.2 Scenario 2

The second possible scenario says that, after the eatery is open to provide service, it has two customers at the initial stage, seated on the sanitized first and second table

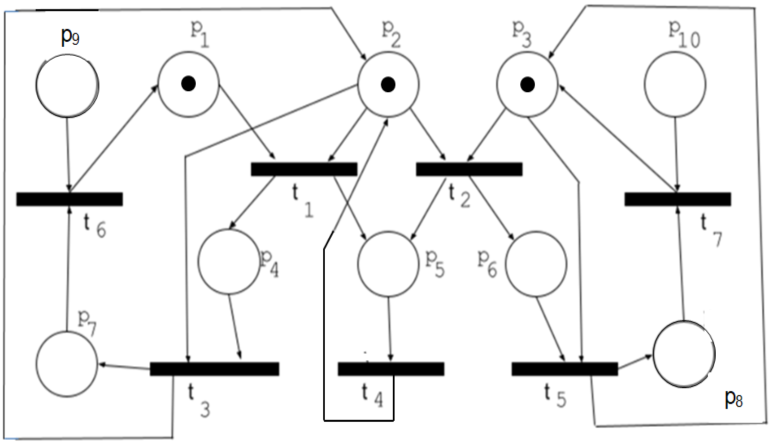


Figure 9.2: Petri Net graph of the proposed model depicting scenario 1 with the initial marking M_0.

Marking	Value of Marking
M_0	$\{1,1,1,0,0,0,0,0,0,0\}$
M_1	$\{0,0,1,1,1,0,0,0,0,0\}$
M_2	$\{1,0,0,0,1,1,0,0,0,0\}$
M_3	$\{0,1,1,1,0,0,0,0,0,0\}$
M_4	$\{1,1,0,0,0,1,0,0,0,0\}$
M_5	$\{0,1,1,0,0,0,1,0,0,0\}$
M_6	$\{0,0,0,1,1,1,0,0,0,0\}$
M_7	$\{1,1,0,0,0,0,1,0,0,0\}$
M_8	$\{0,0,0,1,1,0,0,1,0,0\}$
M_9	$\{0,1,0,1,0,0,1,0,0,0\}$
M_{10}	$\{0,1,0,0,0,0,1,1,0,0\}$
M_{11}	$\{0,0,0,0,1,1,1,0,0,0\}$
M_{12}	$\{0,1,0,0,0,1,1,0,0,0\}$
M_{13}	$\{0,1,0,1,0,1,0,0,0,0\}$

Table 9.4: Markings of the proposed model as per Scenario 1.

respectively, and the service provider (waiter) is free to serve the two customers. In this scenario, the arrival of new customer(s) is possible at the eatery; and it is assumed that new customer(s) is/are in the separate waiting places (p_9 and p_{10}) to occupy the first and second table respectively, after the previously seated customers vacate the tables. The initial marking that depicts this scenario is $G_0 = \{1,1,1,0,0,0,0,0,1,1\}$. Fig. 9.3 depicts the Petri Net model for the above-depicted scenario with the initial marking G_0 .

We shall now discuss these scenarios and further analyze the proposed Petri net model for the same.

One possibility in this first scenario is that the service provider (waiter) initially visits the first table and takes the order, where after the order is reported to the kitchen, and the customer waits until the order gets prepared for serving. Then, he visits the second table and takes the order, which is further reported to the kitchen. Now, orders for both the tables have been placed, and the service provider (waiter) is idle until the

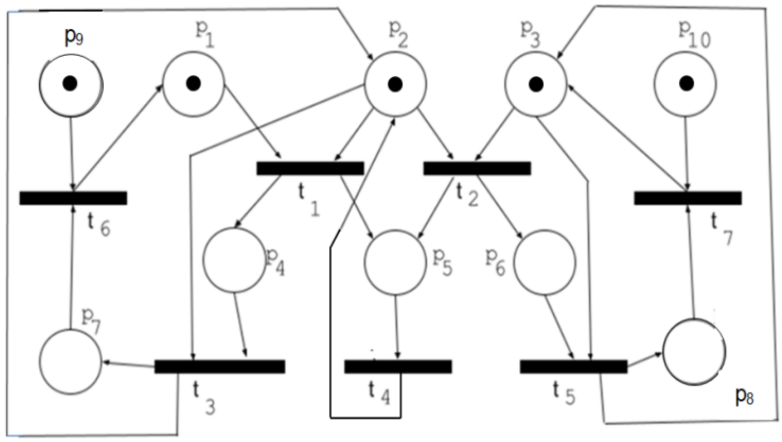


Figure 9.3: Petri Net graph of the proposed model depicting scenario 2 with the initial marking G_0.

food gets prepared. When the service has been provided at both the tables, the customers eat and vacate the place. Continuing further, if Scenario 2 is taken into account, then the arrival of a new customer is possible for the first table and (or) the second table, and the process can repeat.

1. ***Reachability***

From [6],[8], the reachability problem considers a marked Petri Net PN with an initial marking M_0 and a marking M_1 and aims at answering if M_1 is reachable from M_0, that is; whether M_1 belongs to the set $R(PN, M_0)$. The reachability tree or graph is an analytic tool for representing the reachability set of the Petri Net.

Suppose we construct a reachability tree for the given Petri Net; in that case, we need to consider the marking of the Petri Net at that state as a node, and an arc would represent firing of transition from the initial marking to the subsequent set of new markings. Fig. 9.4 shows the Reachability tree of the Petri Net model shown in Fig. 9.2.

Additionally, refer to Fig. 10.1 [1] for the reachability tree of the Petri Net model as shown in Fig. 9.3.

From Fig. 9.4 below, it can be perceived that several paths can be chosen for the functioning of the proposed model yielding dissimilar results.

In total, six different paths exist from the initial marking M_0 to the final marking M_{10}, which infers that both the customers have been served and have vacated the eatery, and no new customer has arrived (as per scenario 1).

Table 9.5 highlights all the likely paths from the reachability tree in Fig. 9.4. The six paths exhibit the possible order of activities w.r.t the occurrence of the required pre-conditions.

2. ***Safeness***

A place p_i 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, $M'(p_i) \leq 1$ i.e. the total number of tokens at any state in a place after firing of a transition is either 1 or 0.

[1] In Appendix on page number 54

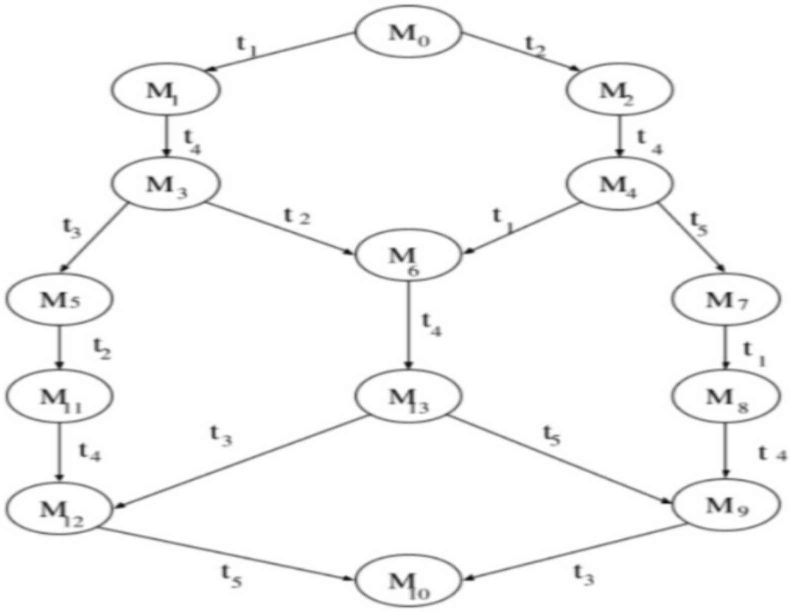


Figure 9.4: Reachability Tree for the Petri Net Structure of the proposed model depicting scenario 1 and all possible markings due to various transitions firing.

1.	M_0	M_1	M_3	M_5	M_{11}	M_{12}	M_{10}
2.	M_0	M_1	M_3	M_6	M_{13}	M_{12}	M_{10}
3.	M_0	M_1	M_3	M_6	M_{13}	M_9	M_{10}
4.	M_0	M_2	M_4	M_6	M_{13}	M_9	M_{10}
5.	M_0	M_2	M_4	M_6	M_{13}	M_{12}	M_{10}
6.	M_0	M_2	M_4	M_7	M_8	M_9	M_{10}

Table 9.5: List of all possible paths.

A Petri Net is said to be safe if all the places in that Petri Net are safe.

Here, it can be inferred from the reachability tree in Fig. 9.4 and Fig. 9.5[2], that the proposed Petri Net model is safe for any marking reachable from M_0 (in case of the first scenario) and G_0 (in case of the second scenario).

Since the tokens' presence at a place in this model justifies the happening of that particular activity, the number of tokens at any place should be either 0 or 1. An infinite number of tokens at any place, or number of tokens other than 0 and 1 at any place, does not have any physical meaning with respect to the proposed model.

Therefore, the conditional nature of the proposed Petri Net model is validated through the safeness property.

3. ***Behavioral Conservation***

We have already shown in section (**9.1.1**) in (1.) that the model is partially conservative.

For scenario-1: It can be inferred from the Table 9.4 that the model is conservative, having an initial marking M_0.

To relate this property with the proposed model, let us consider a state where we have two customers seated at the first and second table respectively, and a service

[2]In Appendix on page number 53

provider (waiter). At any given instance, the possible states of the customers are either of the following-

1) Arrival at the table and ready to place the order.
2) Waiting for the food after placing the order.
3) Eating the served food and eventually leaving.

Similarly, the possible states of the service provider (waiter) are either of the following

1) Free to attend the customers.
2) Confirm the order for serving the food.

It can be seen that the state of customers and service provider (waiter) cannot mingle within itself; for example, a customer cannot be waiting for food and leaving at the same instance. Hence the shuffling of tokens is happening between the three states of the customer; and two states of the service provider. This physical nature of the model is validated by the conservative nature of the model with respect to the initial marking M_0.

For scenario-2: It can be inferred from the Table 10.1[3] that the model is not conservative, as the token count remains constant for few markings but eventually varies from being five for some markings; to four; and to three, at various stages. The physical nature indicates that the tokens are shuffling between the states of two customers at both the tables respectively; the service provider; and the two customers at the waiting place (p_9, p_{10}).

Up to this activity (before firing of t_6 and t_7), the token count is constant, but as soon as the customers of the first and second table vacate, and the waiting customers get seated, the token count represents shuffling between those two new customers and the service provider; instead of four customers (since another pair of new customers has not arrived), and the service provider.

3. ***Boundedness***

A place p_i 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, $M'(p_i) \leq 1$ 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.

Further, 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. The reachability trees [4] in Fig. 9.4, 10.1 (in Appendix), and Tables 9.4, 10.1 imply that the Petri Net is not unbounded. Thus, the proposed Petri Net model, which exhibits safeness, is 1-bounded i .e., behaviorally bounded for both the scenarios.

4. ***Deadlock***

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.

[3] In Appendix on page number 53

[4] A Tree is a connected digraph without a directed cycle.

For the first scenario, from Fig. 9.2, it can be seen that the reachability tree in Fig. 9.4 has one vertex i.e., M_{10}, which has no outgoing arc [1]. Hence, deadlock appears at M_{10} as no transition from $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ can fire. This suggests that the customers at both tables have been served, and the customers have vacated the eatery. Then, the service provider (waiter) becomes idle with no arrival of a new customer.

Similarly, for scenario-2 from Fig. 9.3, it can be seen that the reachability tree in Fig. 10.1[5] shows deadlock at the node M_{10}. This deadlock is the stage when the initial customers seated at both the tables have eaten and vacated; also, the customers who were waiting to be seated at these tables have eaten and vacated while the service provider (waiter) has served all four customers and is currently idle as no new customer has arrived after this. The deadlock will thus persist until the new customer arrives.

5. ***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.

For scenario-1 in the proposed model, t_6 and t_7 act as the dead transitions because it is assumed that no new customer arrives in this scenario. Hence, the proposed model does not exhibit liveness for the initial marking M_0.

Now considering scenario -2, we have a situation where every transition can be fired, as there are new customers who can occupy the first and the second table. Thus, there is no transition that is dead, and hence, the Petri Net model with respect to the initial marking G_0 is live. The physical interpretation of liveness is that all the activities that are mentioned in the model will happen at least once.

6. ***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.

From section (8.2.2), the conflicts can possibly exist between the transitions t_1, t_2, t_3 and t_5. The design of the model (the token placing) is such that t_1 and t_3 cannot be enabled simultaneously since tokens can only be at p_4 after firing of t_1 (similarly, t_2 and t_5 cannot be enabled simultaneously).

Hence the only conflicts that would occur in the Petri Net model for both scenarios would be between t_1 and t_2, t_1 and t_5, t_2 and t_3, and t_3 and t_5.

The physical interpretation of the conflict between t_1 and t_2 implies that when both customers at first and second table are ready to place the order, the service-provider (waiter) can only serve/attend one table at a time.

7. ***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.

Table 9.6 gives the behavioral concurrency between the transitions.

For understanding concurrency between two transitions, say t_1 and t_7, tokens are available for places p_1, p_2, p_8, and p_{10} i.e., a customer has arrived at the first table and

[5] In Appendix on page number 54

First Transition	Second Transition	Behavioral Concurrency
t_1	t_4	No
t_1	t_6	No
t_1	t_7	Yes
t_2	t_4	No
t_2	t_6	Yes
t_2	t_7	No
t_3	t_4	No
t_3	t_6	No
t_3	t_7	Yes
t_5	t_4	No
t_5	t_6	Yes
t_4	t_7	No

Table 9.6: List of all possible behaviorally concurrent transitions.

is ready to place the order. Also, the service provider (waiter) is available to provide service to the customer, and a customer at the second table has already been served, after which he leaves, allowing the arrival of the next customer.

Henceforth, the service provider (waiter) who is serving at the first table; and sanitization of the second table, along with the placing of a menu card on it are two independent activities that exhibit concurrency.

For understanding concurrency between t_1 and t_4 (not behaviorally concurrent), the tokens must be available at places p_1, p_2, and p_5, which is not possible as per the Petri Net's design (token placing). Also, if true, it would imply that the waiter is idle and is finalizing the order for some customer at the same instance, which is absurd. Thus t_1 and t_4 not being behaviorally concurrent, validate the physical interpretation.

Chapter 10

Conclusion

In this paper, we have discussed and interpreted the structural and behavioral properties of the proposed model. The proposed small eatery model has been construed so that it can be suitably related to the actual restaurant scenarios. This paper can be a motivation towards designing and analyzing larger restaurant/eatery models.

Appendix

Marking	Value of Marking
G_0	$\{1, 1, 1, 0, 0, 0, 0, 0, 1, 1\}$
G_1	$\{0, 0, 1, 1, 1, 0, 0, 0, 1, 1\}$
G_2	$\{0, 1, 1, 1, 0, 0, 0, 0, 1, 1\}$
G_3	$\{0, 1, 1, 0, 0, 0, 1, 0, 1, 1\}$
G_4	$\{1, 1, 1, 0, 0, 0, 0, 0, 0, 1\}$
G_5	$\{0, 0, 1, 1, 1, 0, 0, 0, 0, 1\}$
G_6	$\{0, 1, 1, 1, 0, 0, 0, 0, 0, 1\}$
G_7	$\{0, 0, 0, 1, 1, 1, 0, 0, 0, 1\}$
G_8	$\{0, 1, 1, 0, 0, 0, 1, 0, 0, 1\}$
G_9	$\{0, 0, 0, 0, 1, 1, 1, 0, 0, 1\}$
G_{10}	$\{0, 1, 0, 0, 0, 1, 1, 0, 0, 1\}$
G_{11}	$\{0, 1, 0, 0, 0, 0, 1, 1, 0, 1\}$
G_{12}	$\{0, 1, 0, 1, 0, 1, 0, 0, 0, 1\}$
G_{13}	$\{0, 1, 0, 1, 0, 0, 0, 1, 0, 1\}$
G_{14}	$\{1, 0, 0, 0, 1, 1, 0, 0, 0, 1\}$
G_{15}	$\{1, 1, 0, 0, 0, 1, 0, 0, 0, 1\}$
G_{16}	$\{1, 1, 0, 0, 0, 0, 0, 1, 0, 1\}$
G_{17}	$\{0, 0, 0, 1, 1, 1, 0, 0, 1, 1\}$
G_{18}	$\{0, 0, 0, 1, 1, 1, 0, 0, 1, 1\}$
G_{19}	$\{0, 1, 0, 1, 0, 1, 0, 0, 1, 1\}$
G_{20}	$\{0, 1, 0, 0, 0, 1, 1, 0, 1, 1\}$
G_{21}	$\{0, 1, 0, 1, 0, 0, 0, 1, 1, 1\}$
G_{22}	$\{0, 1, 0, 0, 0, 0, 1, 1, 1, 1\}$
G_{23}	$\{0, 1, 1, 0, 0, 0, 1, 0, 1, 0\}$
G_{24}	$\{0, 0, 0, 0, 1, 1, 1, 0, 1, 0\}$
G_{25}	$\{0, 1, 0, 0, 0, 1, 1, 0, 1, 0\}$
G_{26}	$\{0, 1, 0, 0, 0, 0, 1, 1, 1, 0\}$
G_{27}	$\{0, 1, 1, 1, 0, 0, 0, 0, 1, 0\}$
G_{28}	$\{0, 0, 0, 1, 1, 1, 0, 0, 1, 0\}$
G_{29}	$\{0, 1, 0, 1, 0, 1, 0, 0, 1, 0\}$
G_{30}	$\{0, 1, 0, 1, 0, 0, 0, 1, 1, 0\}$
G_{31}	$\{0, 0, 0, 0, 1, 1, 1, 0, 1, 1\}$
G_{32}	$\{1, 0, 0, 0, 1, 1, 0, 0, 1, 1\}$
G_{33}	$\{1, 1, 0, 0, 0, 1, 0, 0, 1, 1\}$
G_{34}	$\{1, 1, 0, 0, 0, 0, 0, 1, 1, 1\}$
G_{35}	$\{0, 0, 0, 1, 1, 0, 0, 1, 1, 1\}$
G_{36}	$\{1, 1, 1, 0, 0, 0, 0, 0, 1, 0\}$
G_{37}	$\{0, 0, 1, 1, 1, 0, 0, 0, 1, 0\}$
G_{38}	$\{1, 0, 0, 0, 1, 1, 0, 0, 1, 0\}$
G_{39}	$\{1, 1, 0, 0, 0, 1, 0, 0, 1, 0\}$
G_{40}	$\{1, 1, 0, 0, 0, 0, 0, 1, 1, 0\}$
G_{41}	$\{0, 0, 0, 1, 1, 0, 0, 1, 1, 0\}$

Table 10.1: Markings of the proposed model as per Scenario 2.

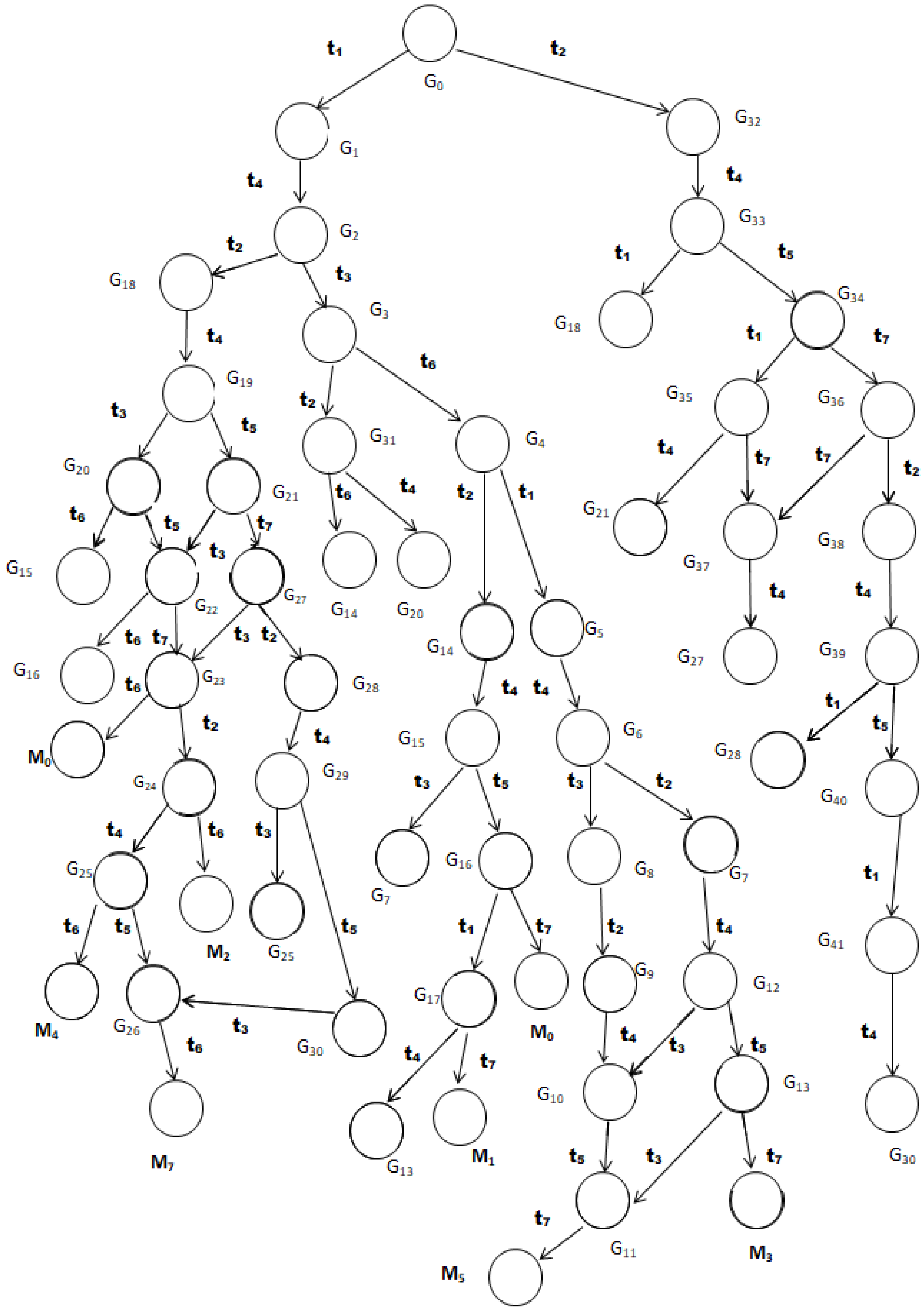


Figure 10.1: Reachability Tree for the Petri Net Structure of the proposed model depicting scenario 2 and all possible markings due to various transitions firing.

References

[1] Desel J. and Reisig W, *Place/Transition Petri Nets. Lectures on Petri Nets I* , Basic Models, Vol. 1491; pp. 122-173, 1998.

[2] Farooq Ahmad, Huang Hejiao and Wang Xiaolong, *Analysis of Structure Properties of Petri Nets Using Transition Vectors*, Information Technology Journal Vol. 7, No. 2; pp. 285-291, 2008.

[3] Heiner M., Gilbert D. and Donaldson R., *Petri Nets for Systems and Synthetic Biology* ,in Formal Methods for Computational Systems Biology, Springer, Vol. 5016 of LCNS; pp. 215-264, 2008.

[4] Itoh Y., Miyazawa I., Liu JH. and Sekiguchi T.,*On Analysis of Petri Net Properties based on Transitive Matrix*, Proc. of IEEE on IECON, pp: 973-978, 1999.

[5] Murata T., *Petri Nets: Properties, Analysis and Applications*, Proc. of IEEE, Vol. 77, No. 4; pp. 541-580, April 1989.

[6] Peterson JL., *Petri Net Theory and Modeling of Systems*, Prentice-Hall, 1981.

[7] Peterson JL., *Petri Nets, Computing Surveys*, Vol. 9, No. 3; pp. 223–252, September 1977.

[8] Rongming Zhu, Ya Zhang and Lan Yang., *A class of generalized Petri Nets and its State Equation*, Advances in Mechanical Engineerig, Vol. 9, No. 8; pp. 1–17, 2017.

Details of Candidate's Publication

The Paper titled "**Designing and Analysing a Petri Net Model of a Small Eatery in Times of Covid**" was accepted and presented in **Scopus Indexed** conference titled *International Conference on Recent challenges in Engineering Science and Technology*, Eluru, Andhra Pradesh, held on $9^{th} - 10^{th}$ April, 2021.

The same has been published in the conference proceedings bearing ISBN -978-93-90214-20-4.