A Review of Petri Net Modeling of Dynamical Systems

Arundhati Lenka
S.O.A University,Bhubaneswar
l_arundhati@yahoo.co.in
Contact-+91-9861058591

Dr.Chakradhar Das
S.I.E.T College,Dhenkanal
dashchakradhar@gmail.com

Abstract

Petri nets are graphical and mathematical modeling tools which are gaining popularity in recent years. It is a tool for the representation of complex logical systems, such as synchronization, sequentially, concurrency and conflict. The present introduction highlights some of the formalism of Petri nets with special emphasis on reachability and coverability analysis. We have also extended our topic by explaining place & transition invariance with examples.

1. Introduction

The concept of Petri net (PN) was introduced by C.A. Petri in 1962[4]. It provides an elegant and useful mathematical tool for modeling dynamical systems and their behaviors. Being graphical tool, it helps in describing the flow of activities in a complex system. Petri nets are particularly suitable to represent in a natural way logical interaction among parts or activities in a system. The dynamical systems that can be modeled by PN are synchronization, sequential concurrence, mutual exclusion and conflict. The properties characterize the discrete event dynamical systems whose examples include industrial automation, communication systems and computer based systems, making Petri net a promising tool and technology for various applications to industrial automation[14].

The most successful application areas of Petri nets have been the modeling and analysis of manufacturing systems [5]. In this context Petri Nets were used to represent simple production lines with buffers, machine shops configurations, automatic production systems, flexible manufacturing systems, automated assembly lines, resource sharing systems, just-in-time systems
[8].

Petri nets as a mathematical tool, use in performance evaluation of both deterministic and stochastic systems. The performance evaluation can be conducted using either analytical technique based on solving underlying (semi) Markov process or discrete event simulation [7].The use of models which incorporate time functions having probabilistic distributions allows one to obtain production rates for the manufacturing system models, throughputs, delays, capacity for communication and microprocessor system models, as well as critical resource utilization and reliability measures for underlying systems [7]. These classes of Petri nets are now a days extensively used to model and study performance of multiprocessor systems, multiprocessor system buses [8], DSP communication channels [9], parallel computer architecture [3] and parallel and distributed algorithms [10].

The main objective of the present paper is to introduce the fundamentals of Petri net concepts to the researchers and practitioners who are actively engaged in the working areas of modeling analysis of industry oriented systems.In this present chapter we focus on ordinary Petri nets.

The paper consists of five sections.Section 1 represents the introduction to PN. In section 2, symbols notations and description of Petri nets are given. The reachability and coverability is given in section 3, section 4, represents the structural analysis of Petri nets. In section 5, we represent, Petri net invariant problems. The conclusion is given in section 6.

2 . Symbols, Notations and Descriptions of Petri nets

A marked PN is a tuple (P, T, F, W, I_0, M_0), where

- $P= \{p_1, p_2, p_3 \ldots\ldots\ldots p_n\}$ is the set of n- places exhibited graphically by circles.
- $T= \{t_1, t_2, \ldots t_m\}$ is the set of m- transitions drawn as bar pictorially. $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$

- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs (flow relation)
- $W: F \rightarrow N = \{1,2,\ldots\}$ is the weight function
- $I : P \times T \rightarrow N$ is an input function that defines directed arcs for place to transitions where N is the set of non-negative integers.
- $O: T \times P \rightarrow N$ is the output function which defines the directed arcs from transition to places
- $Mo: P \rightarrow N$ is the initial marking.

If $I(p,t) = k$, $O(p,t) = k$ then there exist k directed (parallel) arcs connecting place p to transition t and (transition t to place p). If $I(p, t) = O(t, p) = 0$ then, there exist no directed arcs connecting p to t (t to p) . Frequently, in the graphical representation, parallel arcs connecting a place to transition and transition to place are represented by a single directed arc labeled with its multiplicity or weight k.

Graphically a PN is a bipartite directed graph with two kinds of nodes (places and transitions) and the edges (arcs) connecting places to transition or vice versa. A place is an input place to a transition if there exist a directed arc from place to transition. A place is an output place if there exists a directed arc from transition to place. This elementary set may be used to represent various aspects of modeled systems. For instance, input output places may represent preconditions, (post conditions) and the transitions represent events. In the simplest form, a PN may be represented by a transition together with input and output places which may represent the availability of resources.The transitions represents their utilization, and the output places represent the release of the resource. See the figure below.

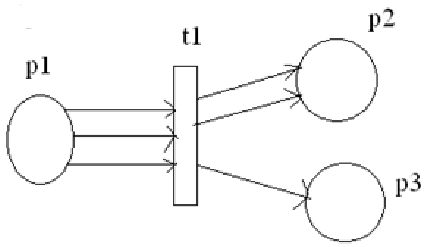


Figure -1 Multiple arcs

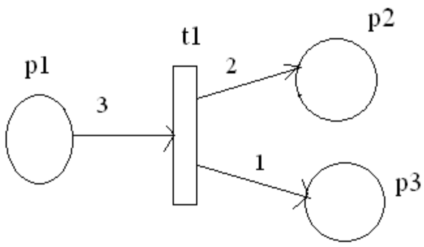


Figure -2 Compact form of Multiple arcs

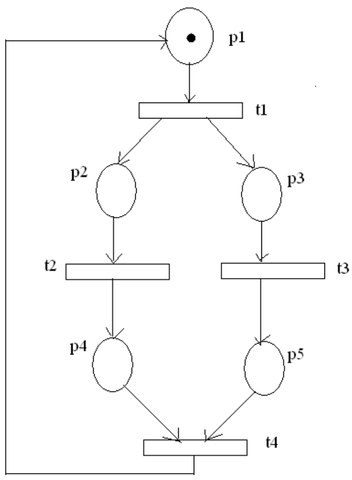


Figure - 3

Example:

The set in Fig- 3 consist of five places and four transitions, depicted by circles and bars respectively. p_1 is input place of transition t_1 and p_2, p_3 are output places of transition t_1.

In order to study the dynamic behavior of a PN, in terms of its status and their changes, each place may be assigned either none or a positive number of tokens, marked by black dots, as shown in figure- 3 in place p_1. The presence or absence of a token in a place can indicate whether conditions associated with the place are true or false. For a place representing the availability of resources, the number of tokens in the place indicates the number of available resources. At any given time instance, the distribution of tokens on places called **Petri Net marking,** defining the current state of the modeled system. A marking of a PN with places is represented by (nx1) vector. M, which is denoted by M (p) are non-negative integers representing the number of tokens in the corresponding places. A PN containing tokens is called a marked Petri Net. For example, PN exhibited in Fig-3, $M_o = (1, 0, 0, 0, 0)^T$

By changing distribution of tokens on places which may reflect the occurrence of events or execution of operations, thus exhibiting the dynamic behavior of PN. The following rules are used to govern the flow of tokens on places.

- A transition is enabled if all its input places carry at least one token.
- An enabled transition fires by removing one token per arc from the input place and adding one token per arc to each output place.

Given an initial marking M_0, the reachability set of all markings resulted by repeated firing of transitions. Moreover, we can say a transition t_k is enabled in marking M_o if:

For any $p_i \in I(t_k), \quad M_i \geq 1.$

The marking M' obtained from M_0 by firing t_k is said to be immediately reachable from M_0 and the firing operation is denoted by $M_0 \longrightarrow M'$ The token counts in M' is pictorially represented by Fig- 4 and given by as follows.

$$M^I(P_i) = \begin{cases} M(P_i)+1, \text{if } P_i \in 0(t_k), p_i \notin I(t_k) \\ M(P_i)-1, \text{if } P_i \in 0(t_k), p_i \notin 0(t_k) \\ M(P_i), \text{otherwise} \end{cases}$$

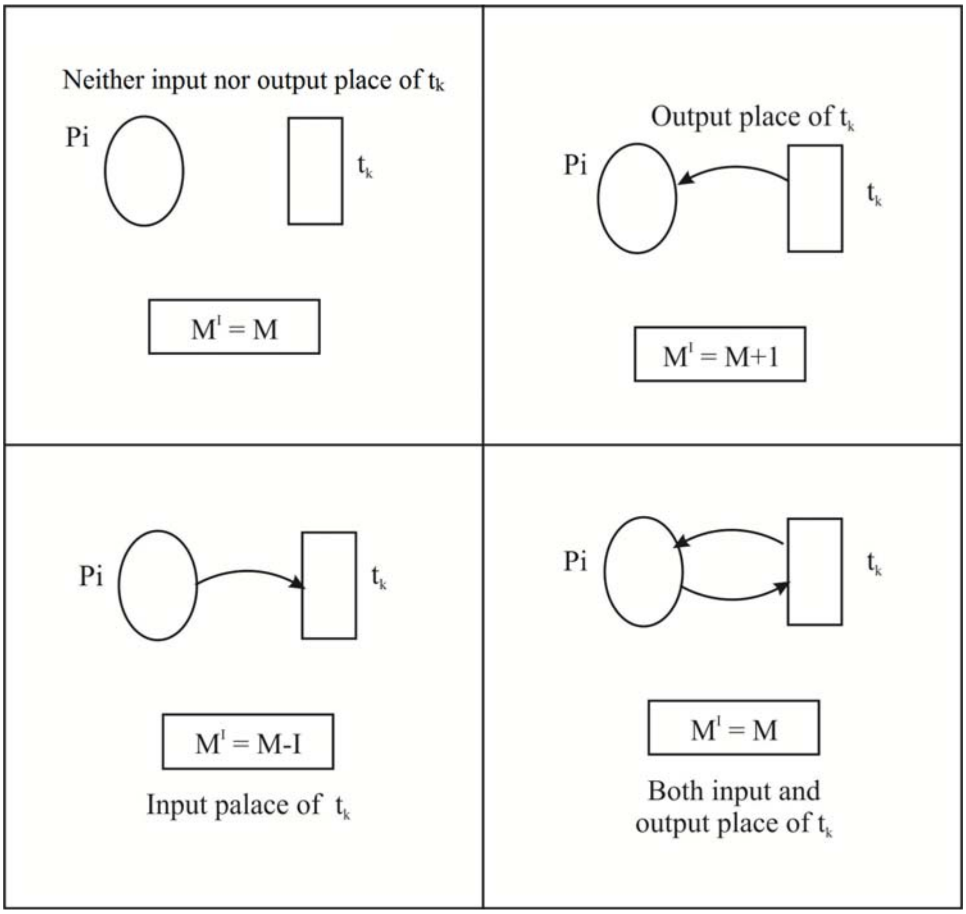
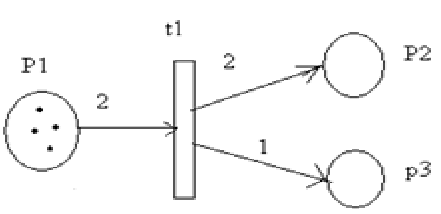


Figure - 4



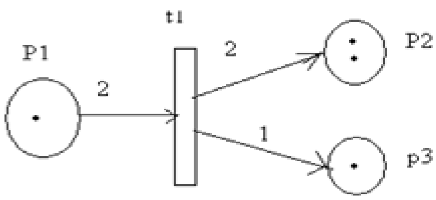


Figure - 5

Figure - 6

Before firing t_1

After firing t_1

The modeling power of PN can be increased by adding zero testing ability i.e. the ability to test whether a place has no token. This is achieved by introducing an inhibitor arc. The inhibitor arc connects an input place to a transition and pictorially represented by an arc terminated with a small circle.

A PN with inhibitor arc shown in Figure- 7

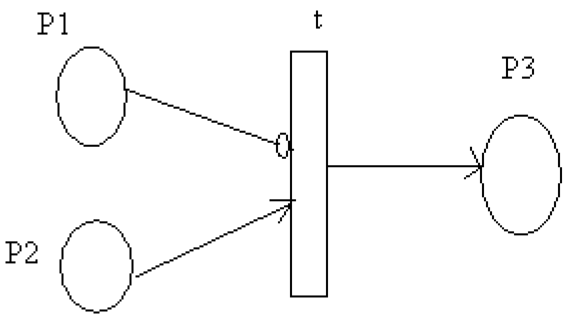


Figure - 7 PN with inhibitor arc

The presence of an inhibitor arc connects to input place to transition, changes the transition firing conditions. In the presence of inhibitor arc a transition is regarded as enabled if each input place, connected to the transition by a normal arc (an arc terminated with an arrow) contains the number of tokens equal to the weight of the arc and no tokens are present on each input place connected to transition by the inhibitor arc. The transition firing rules are same as for the normally connected places. The firing however, does not change the marking in inhibitor arcs connected to places. A transition without any input place is called a source transition, and one without any output place is called a sink transition. A source transition is unconditionally enabled, and the firing of sink transition consumes tokens, but does not produce any.

A pair (p, t) is called a self loop if p is both an input and output place of it. A Petri Net is pure if it has no self loop. A PN is ordinary if all its arc weights are 1'[s].

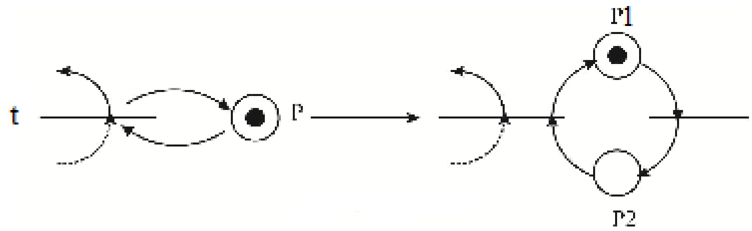


Figure - 8 Self loop removals

3. Basic Properties of Petri nets

3.1.- Reachability Graph Analysis

Petri nets being mathematical tools posses a number of properties in the context of modeled systems. Two types of properties such as behavioral and structural are important. The behavioral properties are marking dependant of the Petri nets .The structural properties, on the other hand are independent of initial marking of Petri nets. In behavioral properties, we intend to discuss reachability, boundedness, conservativeness, liveness, reversibility and home state.

Reachability is a fundamental basis for verifying the dynamic properties of PN. A sequence of firings will result in sequence of markings. A marking M_n is reachable from a marking M_0 if there exist a sequence of firings that transforms M_o to M_n.The firing sequence is denoted by $\sigma = M_0 t_0\ M_1 t_1\ M_2\ t_2$……. $M_0\ t_n$ or simply by σ and we write M_0 [σ >Mn]. The set of all possible marking reachable from M_o is a set denoted by $R(M_o)$.The reachability problem is decidable. But it takes at least exponential space (time) in general case. The equality problem, i.e. L (N', Mo) = L (N', Mo) is undecidable for two Petri Nets N and N'.Undecidability means there is no algorithm for determining if L (N, Mo) = L (N', M_o).

Reachability is an important issue in designing distributed systems. The question is whether the system modeled with Petri net, exhibits all derivable properties, as specified in the requirements specification, and no undecidable one. Firing enabled transitions sequentially, the system changes from one into another state. The sequence of firings of transitions which would result in transforming a making M_o to M_i .where M_i represents the specific state and the sequence of firings represents the required functional behavior.The firing sequence is denoted by $\sigma = M_o t1\ M_1 t2$……. $M_n tn$ or simply by σ and we write M_0 [6>Mn]. The set of all possible marking reachable from M_0 is a set denoted by R (M_0). The set of all firing sequence is denoted by L (M_0).The reachability problem is decidable, but it takes at least exponential space (time).In general . The equality problem L (N', M_o) = L (N, M_o) is undecidable for two Petri nets N and N'.
Example –2 Consider the following Petri net and the corresponding firing sequence producing the reachability tree in fig-9.

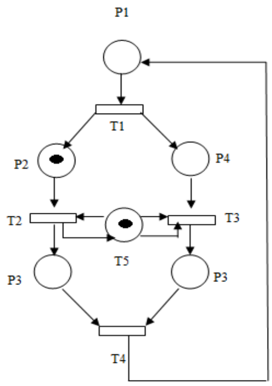


Figure - 9

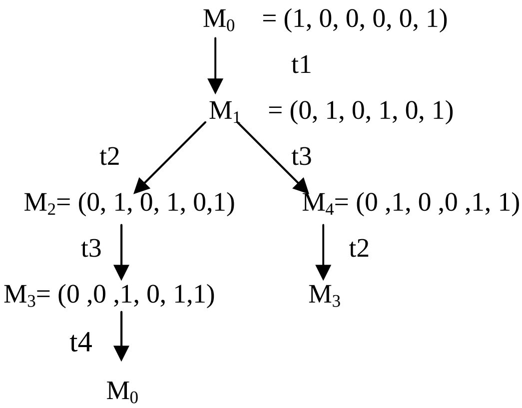


Figure - 10 Rechability Tree

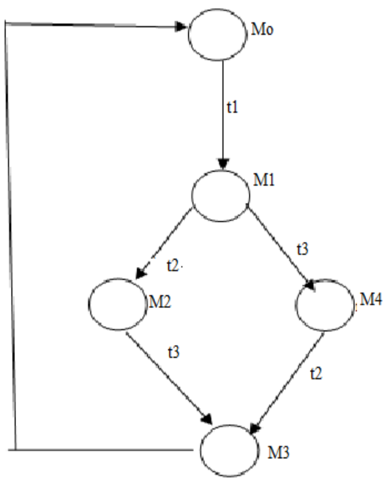


Figure -11

3.1 Coverability tree

Given a PN (N, M_0) with initial marking M_0, we can obtain many "new" markings by firing the number of enabled transitions. From each marking we can again reach more markings. The process gives a tree representation. Nodes represent marking generated from M_0 (root) and its successors arc represent the transition firing which transforms one marking to another. The tree will go infinitely large if the set is unbounded. To keep tree finite a special symbol ω is introduced which can be thought of "infinity". Having the property that ω > n for any integer n, ω+n = n+ ω = ω, ω - n = ω.

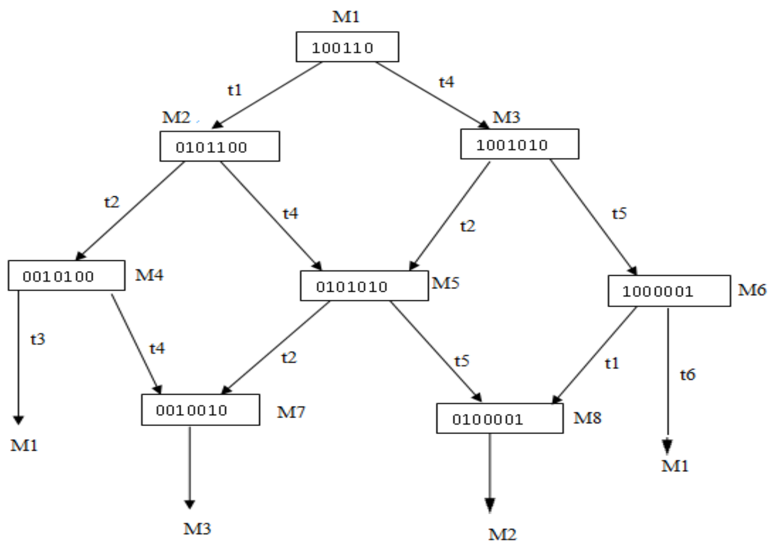


Figure-12

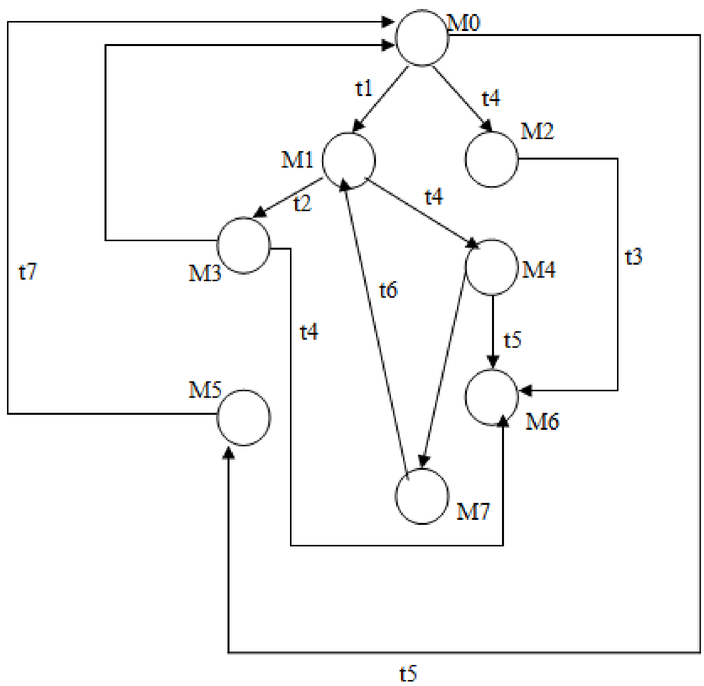


Figure-13

THE ALGORITHM FOR COVERABILITY TREE

Step 1: Consider the initial marking M_0 as the root and tag it 'new'.
Step 2: If the new marking from M_0 exist do the following:

a) If M is identical on the path from M_0 to M, then tag M as 'old' and perform another new marking by firing other enabled transition.

b) If no transition is enabled at M tag M as 'dead end'.

c) For enabled transition at M do the following for each enabled transition t at M.

   - Obtain a marking M' that results from firing t at M.
   - On the path from M_0 to M if there exists a marking M' such that M' (p) $\geq$ M'' (p) for each place p and M' $\neq$ M'', i. e M''is coverable by M' that replace M'(p) by ω for each p such thatM' (p) $\geq$ M'' (p) "may be M_0 also.
   - Introduce M' as a node and draw as arc with label t from M to M' and tag M' as "new".

The above algorithm is illustrated for the following PN of Fig - 14

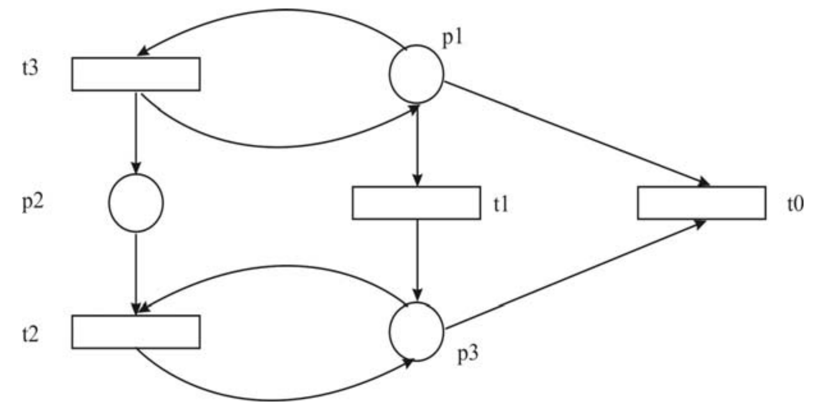


Figure-14

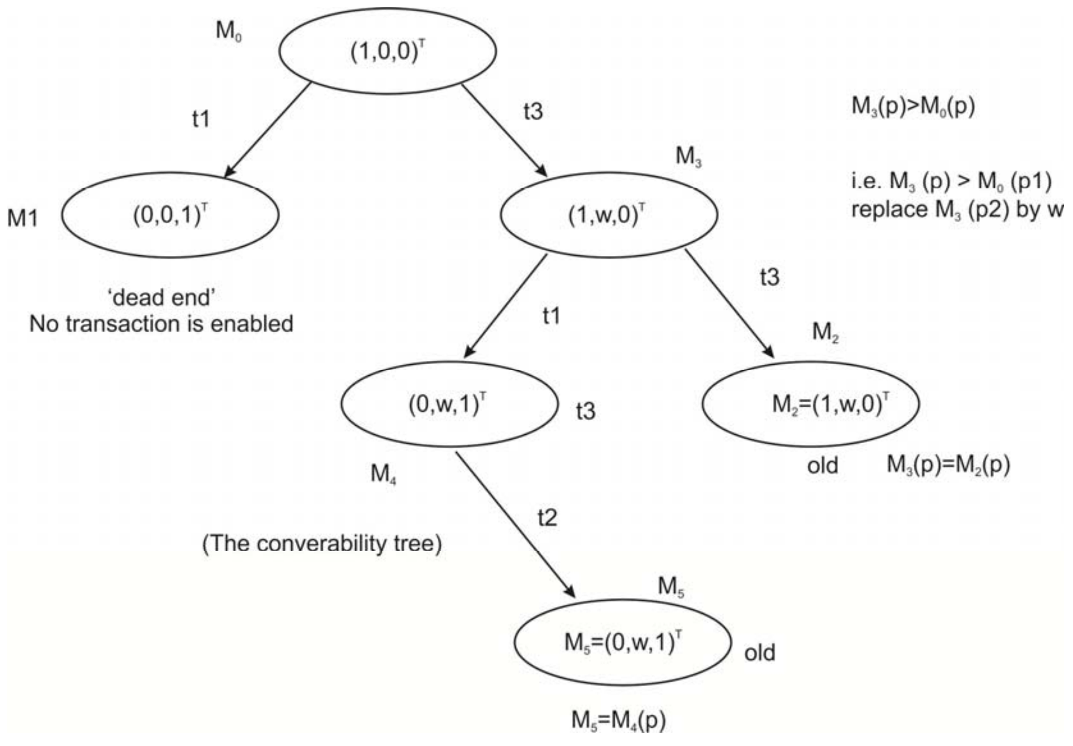
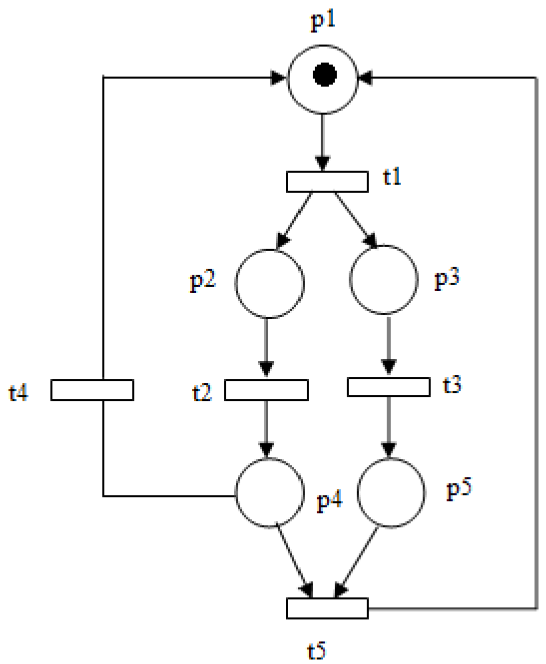


Figure-15

P = {p1, p2, p3)
T = {t1, t2, t3, t4}

$M_0 = (1, 0, 0)$

Example



$P = \{p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5\}$

$T = \{t_1\ \ t_2\ t_3\ t_4\ \ t_5\}$

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

$I\ (t_2) = \{\ p_2\}$ $\quad O\ (t_2) = \{\ p_4\}$

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

$I\ (t_4) = \{\ p_4\}$ $\quad O\ (t_4) = \{\ p_2\}$

$I\ (t_5) = \{\ p_4\ \ p_5\}$ $\quad O\ (t_5) = \{\ p_1\}$

$M_1 = (1, 0, 0, 0, 0)$

Figure-16 A PN graph with input and output functions

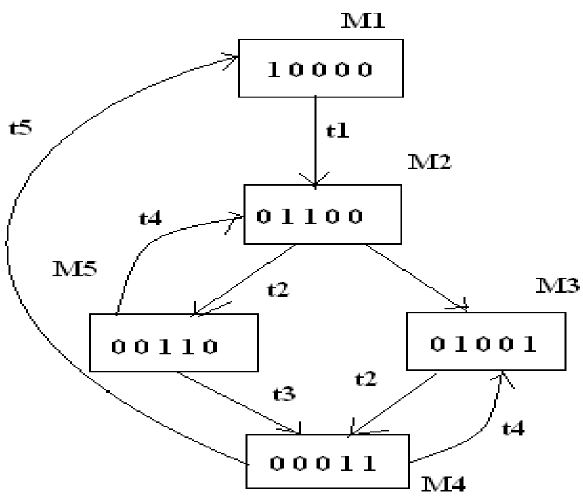


Figure-17 Reachability graph

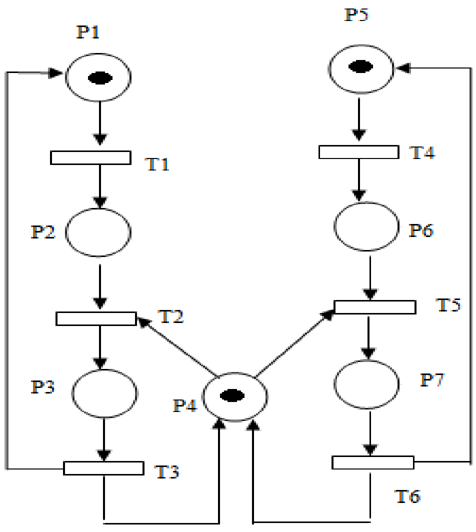


Figure- 18 Mutual exclusion problem

The initial marking $M_0 = (1, 0, 0, 1, 1, 0, 0)$

3.2 Boundedness and Safe ness

A Petri net is said to be k-bounded if the number of tokens in any place p Є P is always less or equal to k (k being a non negative integer) for every marking M reachable from initial marking M_0, M Є R (M_0). A Petri net Shown in fig-19 is safe.

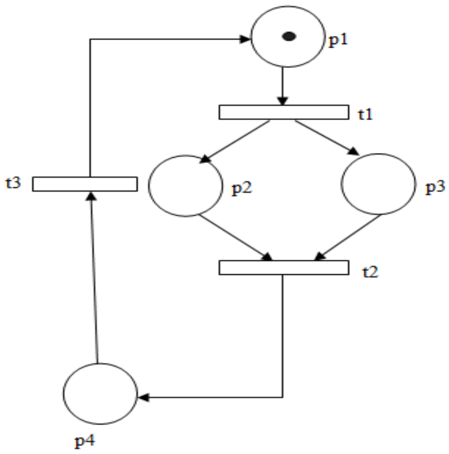


Figure- 19 Petri net is safe

An example of Perti net which is unbounded is shown in fig-20

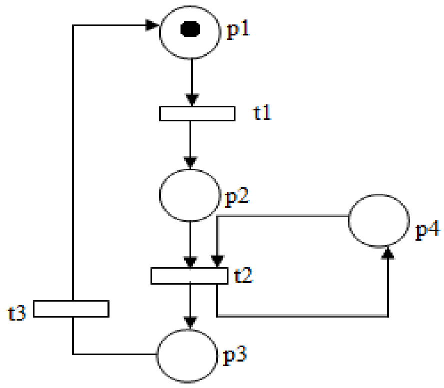


Figure- 20 Unbounded Petri net

3.3 Conservativeness

A Petri net is conservative it the number of tokens is preserved in each marking when transition fires. Its weights are associated with the place, then a Petri net is conservative if there exists a vector w, w= (w_1, w2…. w_m), where m is number of places, and w(p) > 0 for each p Є P such that the weighted sum of the tokens remains constant. That is M_0 T_w = M^T W for any marking reachable for M_0. Consider the Petri net below.

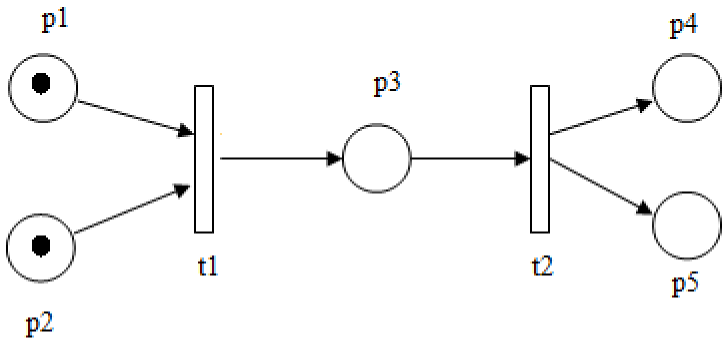


Figure- 21 Petri net is conservative with respect to ω [1, 1, 12, 1,1]

M_0 = (1, 1, 0, 0, 0) ^{T}
By firing transitions t_1 and t_2 successively
We obtain M_1=(1,1,0,0,0)^T and M_2=(0,0,0,1,1) ^{T} respectively. It is obvious that
M_0 T_w = M_1 T_w = M_2T_w = 2
If a Petri net is conservative with respect to a vector with all elements equal to one i,e (sum vector) then the net is strictly conservative. Consider the example in Fig-22.

Example

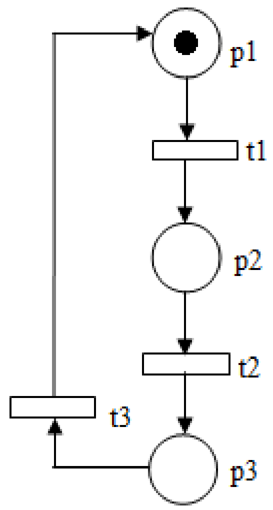


Figure -22 Petri net, strictly conservative.

Here w = [1,1,1]

$M_0 = (1,0,0)^T \xrightarrow{t1} M_1 = (1,0,0)^T \xrightarrow{t2} M_2 = (1,0,0)^T \xrightarrow{t3} M_3 = (1,0,0)^T = M_0$

$M_0^T w = M_1^T w = M_2^T w = 1$

3.4 Liveness

The concept of live ness is closely related to the dead lock situation which has been studied extensively in the context of operating systems. A Petri net is said to be live (or equivalently m_0 is said to be live marking). No matter what marking has been reached from M_0, if it is possible to ultimately fire any transition of the net by progressing through some further firing sequence. Alive Petri net guarantees dead lock free operation.

Example

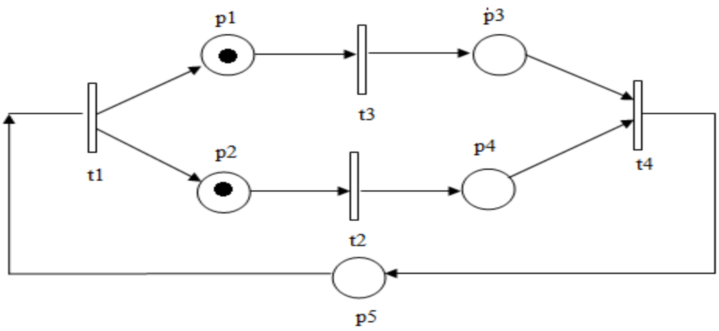


Figure - 23 Live Petri net

Liveness is an ideal property of many systems, it is too costly to verify this strong property for some complex systems such as operating system of a large computer for this reason, different levels of live ness for transition to and marking Mo were introduced.

A transition t in a Petri net is said to be:

Lo-live (or dead) if there is no firing sequence in L (M_0) in which t can be fired.

L1-Live (Potentially friable) if t can be fired at least once in some firing sequence in L (M_0).

L2-live if t can be fired at least k times in some firing sequence in L(M_0) given any positives integer k.

L3-Live if t can be fired infinitely often is some fixing sequence in L (M_0) and

L4-Live (or live) if t is L1-live (potentially friable) in every marking in R (M_0)

Example:

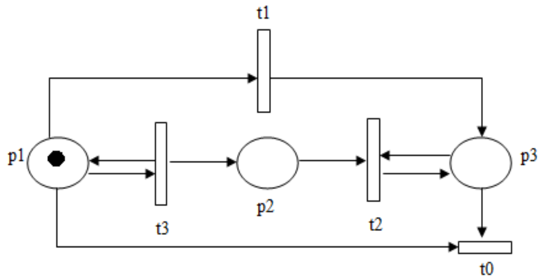
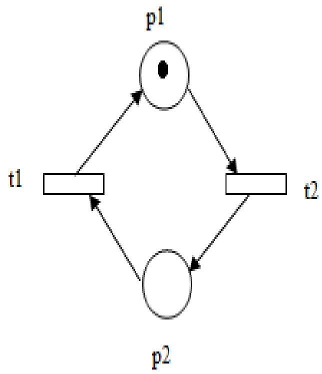


Figure – 24

A Petri net with different levels of live ness of transitions.
Transitions t_0, t_1, t_2, and t_3 and L_0, L_1, L_2 and L_3 live respectively.

3.5 Reversibility and Home State.

Reversibility and Home State are important issue for error recovery in manufacturing systems. These systems are required to return from the failure state to the preceding correct state. A Petri net with initial marking Mo is said to be reversible if for each marking M in R (M0), Mo is reachable form M. A Petri net with marking M_i is said to be a home state if for each marking M in R (M_0), M_i is reachable from M. The Petri net show in Fig-25 is reversible and the Petri net shown in Fig- 26 is non reversible.



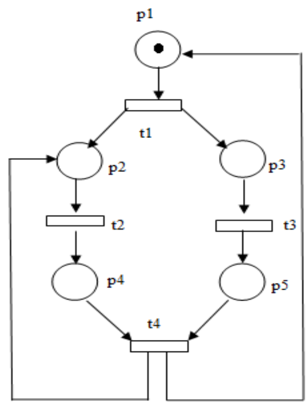


Figure-25 Reversible

Figure – 26 Non reversible

4. State Space Analysis

A The incident matrix notation is an ultimate way of expressing input and out put function .Let A denote incidence matrix of dimension n x m.

(Where n is the number of transitions and m is the number of places). The elements of the incidence matrix are defined as $a_{ij}^+ = a_{ij}^+ - a_{ij}^-$ where a_{ij}^+ - i,e the number of arcs connecting transition t_i to output place p_j, a_{ij}^-.
$a_{ij}^+ = 0$ (t_i, p_j), and a_{ij}^- is equal to the number of arcs connecting transition t_i to input place p_j ($a_{ij}^- = I(p_j, t_i)$) .When transition t_i fires, a_{ij}^+ represents the number of tokens deposited on out put place p_j, a_{ij}^- represents the number of tokens remove from the input place p_j, a_{ij} represents the change in the number of tokens in place p_j. So, transition t_i is said to be enabled in marking M iff

$a_{ij}^- \le M(p_j)$, i= 1,2,3,4. . . . . .m. Eq (1)

For Petri nets with self loop a_{ij}=0 for place p_j and transition t_i. We assume that the net is pure (no self loops).
The state equation for a state Petri net represents a change in the distribution of tokens on places (markings) as a result of transition firing. The equation is defined as follows:

$M_k = M_{k-1} + A^T uk$, k = 1, 2, 3, 4 Eq (2)

M_k is mx1 column vector representing marking M_k immediately reachable from a marking M_{k-1} after firing transition t_i .The k^{th} firing vector uk , an nx1 column vector has only one non zero entry 1 in the i^{th} position represents a transition t_i firing in k^{th} firing of the sequence of firing starting with initial marking M_0 .This entry corresponds to i^{th} row of the incidence matrix A, which represents a change of a matrix as a result of firing transition t_i .The matrix equation is useful in studying the reach ability problem .We illustrate the concept by the following example.

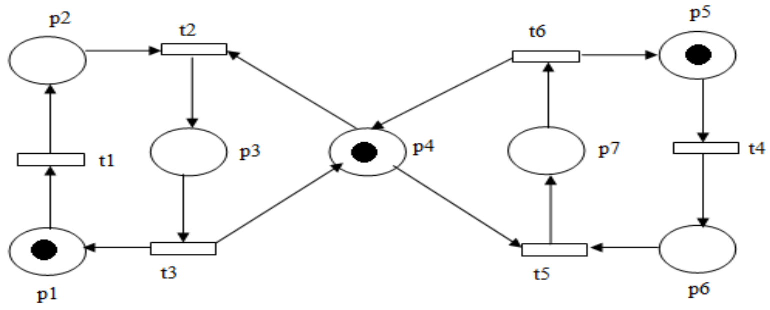


Figure - 27

Initial marking M_0= (1, 0, 0, 1, 1, 0, 0) ^{T}
The firing sequence $\sigma = t_1t_2t_4$, each transition fires once. Thus x = (1, 1, 0, 1, 0, 0) ^{T}
The state equation:
$M_0 + A^Tx = M_1$ Eq (3)

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

M_1 = (0, 0, 1, 0, 0, 1, 0)
x = (1, 1, 0, 1, 0, 0) ^{T}
The equation $M_0 + A^Tx = M_1$ holds good
Thus M_1 is reachable from M_0
Consider the example in fig- 28
Example-

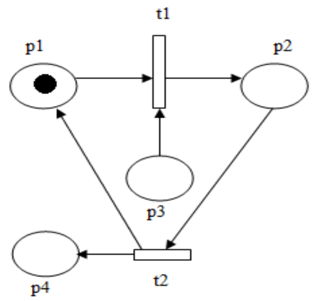


Figure -28

M_0 = (1, 0, 0, 0) ^{T} no transition can be fired.

The incidence matrix

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

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

$M^0 + A^T x = M1$ Eq (4)

Taking $X = (1, 1)^T$ $M_1 = (1, 0, 0\ 1)^T$, we have

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

But none of the sequences corresponding to $(1, 1)^T$ i.e. t_1t_2 or t_2t_1 can happen

Example, consider the following net and the marking $M = (1, 1)^T$

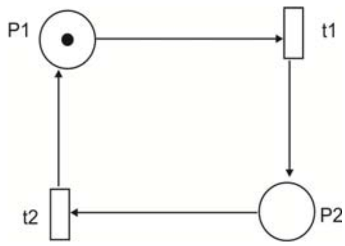


Figure - 29

The state equation

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Has no solution, so M is not reachable.

5. Invariant problems

The concepts related to the incidence matrix are particularly useful in studying properties of pertinent models are T- invariant and P-invariant.

5.1 Transition-invariant (T-invariant)

An integer solution x of $A^T x$ is called T-invariant. The non zero entries in a T-invariant represent the firing counts of the corresponding transitions which belong to a firing sequence transforming a marking M_0 back to M_0

An integer solution y of Ay=0 is called a P-invariant. The non zero entries in a

P- Invariant represents weights associated with the corresponding places so that the weighted sum of tokens on these places is constant for all markings reachable from an initial marking.

The subset of places (transitions) corresponding to the non zero entries of a

T-invariant (P-invariant) is called the support of an invariant.

Example:

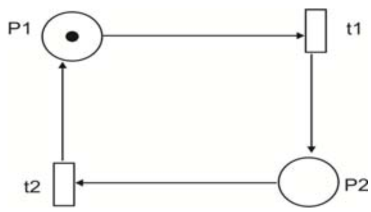


Figure-30 Transition invariant

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} x1 \\ x2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The solution vector x = $(x1, x2)^T = (1,1)^T$ is T – invariant. A T-invariant indicates a possible loop in the net, i.e., a sequence of transitions whose net effect is null, i.e., which leads back to the marking it starts in.
Example : Consider the PN in Fig-31

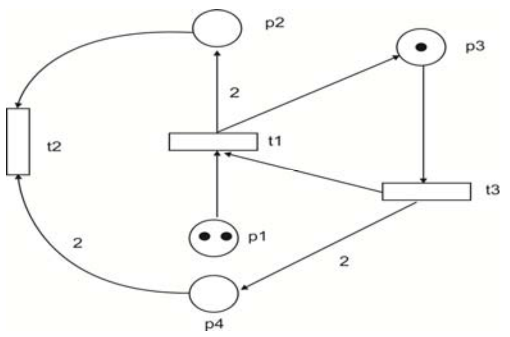


Figure-31

Now $A^T x = 0 \Rightarrow$

$$\begin{bmatrix} \mathbf{-2} & \mathbf{1} & \mathbf{1} \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{x1} \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \end{bmatrix}$$

$X = (1,1,1)^T$

The following matrix has solution hence the vector $[0\ 0\ 0]^T$ is a T invariant

5.2 Place – Invariant (p- Invariant)

For an incident matrix A n × m, where n is the number of transitions and m is the number of places, an m vector y is called p-invariant if Ay = 0.It is also true that an m-vector y is an p-invariant iff $M^{TY} = M_0^{TY}$ for any initial marking Mo and any M ϵ R (M_0). An invariant vector Y is said to be minimal if there is no other invariant y1 such that $y_1(p) \leq y(p)$ for all p.
Given a minimal support of an invariant, there is a unique minimal invariant corresponding to the minimal support .We call such an invariant a minimal support invariant.

Example.

Consider the Petri net in Fig-32 .

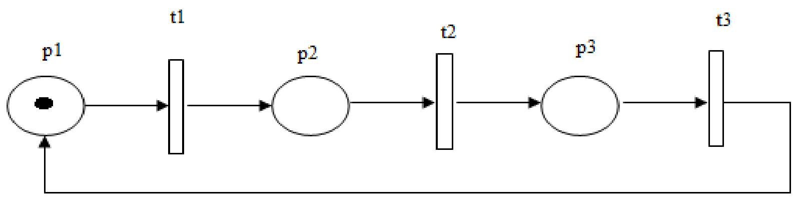


Figure - 32

$Ay = 0$

$$\text{Where } A = \begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \end{array} \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The minimum p- invariant = (1, 1, 1)

6. Conclusion

In the present paper, we have presented a brief review and fundamentals of PN .The emphasis is given for reachability analysis in a simplified manner. Various properties of PN are illustrated with the help of simple examples. The review is basically meant for researchers to have a glimpse of the fundamentals of PN theory. We haven't touched more comprehensive part of the new emerging field of the PN.

REFERENCES

[1] A.E. Kostin. A reachability algorithm for general Petri nets based on transition invariants. In R. Kralotic and P. Urtyctyn, editors, math Found computer Science, Vol. 4162.
[2] A.E. Kostin. Reachability analysis in T-invariant less Petri, Nets IEEE Trans Autom contr., 48 (6); 1019-1024 June 2003.
[3] A. Pagnoni and G. rotten berg. Ed. Advances in Petri Nets 1985 Lecturer Nets, in computer Science, Vol. 222 New York : springer-verlag, 1986
[4] C.A. Petri, Kommunikation mit Automation, Bonn Institute fur Instrmentelle mathematik, schritten des lim NV.3 (1962) English translation, Communication with Automata.
[5] C.V. Ramamoorthy and G.S. HO. Performance evaluation of asynchronous concurrent systems using Petri nets. IEEE trans. Software Eng., Vol. SE-6, No. 5, PP 440-449, Sept. (1950).
[6] Hack, M the recursive equivalence of the rechability problem and the live ness problem for Petri nets and vector addition systems FOCS PP. 156-164 (1974).

[7] Ichikawa A. and Hiraishi, K. Analysis and control of discrete event systems represented by Petri nets, LNCIS 103, PP. 115-134, (1987).

[8] j.Magott. Performance evolution of concurrent systems using Petri nets, inform processing Lett., Vol. 18, No. 1, PP 7-13, (1984).

[9] K. Garg, An approach to performance specification of communication protocols using timed Petri nets, IEEE Trans, software Eng, Vol. 5E-11, No. 10, PP. 1216-1225, Oct (1985).

[10] Mayer, E. An algorithm for the general Petri net reachability problem, SIAM. J. Comput, 13 : PP 441460 (1984).

[11] Merlin, P. A study of the recoverability of computing systems, Ph. D. thesis Dept. of Information and Computer Science, Univ. of Calfornia at Irvine, (1974).

[12] M.K. Molloy, performance analysis, using stochstic Petri nets, IEEE Trans, computers, Vol. C-31, No.9, PP. 913-197, Sept. (1982).

[13] M. Silva and S. Veloilla Error detection and correction on Petri net models of discrete events control systems, in pros. 7985 IEEE Int. sym position circuits and system, PP 921-924, (1985).

[14] Murata, T. Petri Nets: properties, analysis and applications Proc. Of the IEEE, 77 (a) : 541-580, (1989).

[15] Patil, S. Coordination of asynchronous events Ph.D. thesis, Sept. of Elec. Eng, MIT Cambridge Mass, May (1970)

[16] Rockoff, C. The covering and bounded ness problems for vector addition system, Theoretical computer science 6. 223-231 (1978).

[17] Ed. Advance in Petri Nets 1987, Lecturer Nutes in computer science, Vol. 266 New York: Springer-Verlag 1987.

[18] T. Murtata, state equation, controllability, and maximal matchings of Petri nets, IEEE Trans, Automat contr., Vol., AC-22, No. 3 PP 412-416 Jun. 1977.

[19] W. Reisig. Petri Nets An Introduction EATCS Monographs on Theoretical Computer Science, Springer – Verlag, 1995.