Basic Properties of Petri Nets

Gajendra Pratap Singh[1]

[1]Former Research Scholar, Department of Applied Mathematics Delhi Technological University Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India.

Email: gajendraresearch@gmail.com

Abstract: The concept of a Petri net was introduced by Carl Adam Petri, a tool for the
budy of certain discrete dynamical systems [7]. One of the most active fields of current research in mathematics is the subject of discrete dynamical system as Petri nets, whose structures form directed bipartite graphs [3], together with an initial marking. Petri nets are used for describing, designing and studying discrete event-driven systems that are characterized as being concurrent, asynchronous, distributed, parallel, and/or nondeterministic. As a graphical tool, Petri net can be used for planning and designing such a

system with given objectives, more effectively than flowcharts and block diagrams. As a mathematical tool, it enables one to set up state equations and algebraic equations and other mathematical models which govern the behavior of system dynamics. This paper is a small survey of basic concepts and application of Petri nets. Here we specially focus on some of its basic structural and dynamic properties.

1. INTRODUCTION

Petri net theory originated from his doctoral thesis "Communication with Automata" [7], submitted to the science Faculty of Darmstadt Technical University in July, 1961, and he defended his thesis in June 1962. The study of Petri nets has since developed in two distinct directions *viz.,* (*i*) 'Applied Petri net Theory', which is concerned mainly with applications of Petri nets to the modeling of systems and their analysis, giving insight into the functioning of the modeled systems. Successful work in this area requires a good knowledge of the application areas as also of corresponding Petri net techniques, and (*ii*) development of Pure Petri net theory itself, which is the study of the nature and properties of Petri nets so as to develop the basic tools, techniques and concepts needed for their applications. Diagrammatically the use of Petri nets for the modeling and analysis of system is shown in Figure 1.

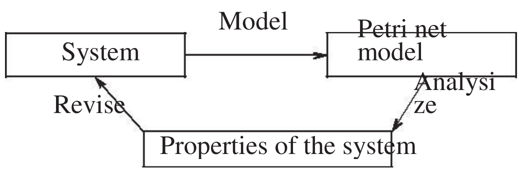

Figure 1

A Petri net is also used as a kind of diagram for describing processes that arise in understanding the functioning of systems with many components known as distributed systems. They are today a widely used model of concurrency in theoretical computer science and to describe chemical reactions in molecular biology, interactions between organisms, supply chains, and so on.

---

2 PRELIMINARIES

For standard terminology and notation on Petri net theory and graph theory, we refer the reader to Peterson [8] and Harary [3]. Throughout this paper, Jensen[4] definition of Petri nets has been used.

2.1 Matrix definition of a Petri net

A *Petri net* is a 5-tuple $C = (P, T, I^-, I^+, \mu^0)$, where

1. P is a nonempty set of 'places',
2. T is a nonempty set of 'transitions',
3. $P \cap T = \emptyset$,
4. $I^-, I^+ : P \times T \rightarrow \mathbf{N}$, where $\mathbf{N}$ is the set of nonnegative integers, are called the *negative* and the *positive* 'incidence functions' (or, 'flow functions') respectively,
5. $\forall\, p \in P, \exists\, t \in T : I^-(p,t) \neq 0$ *or* $I^+(p,t) \neq 0$ *and* $\forall\, t \in T, \exists\, p \in P : I^-(p,t) \neq 0$ *or* $I^+(p,t) \neq 0$
6. $\mu^0 : P \rightarrow \mathbf{N}$ is the *initial marking*.

In fact, $I^-(p, t)$ and $I^+(p, t)$ represent the number of arcs from p to t and t to p respectively. I^-, I^+ and μ^0 can be viewed as matrices of size $|P| \times |T|$, $|P| \times |T|$ and $|P| \times 1$, respectively.

2.2 Graphical Representation of a Petri net

As in many standard reference books (e.g., see [8, 9]), Petri net is a particular kind of directed graph [12], together with an initial marking μ^0. Petri nets have a well known graphical representation in which transitions are represented as boxes and places as circles with directed arcs interconnecting them, to represent the flow relation between them. The initial marking is represented by placing a token, shown as black dots (see Figure 2).

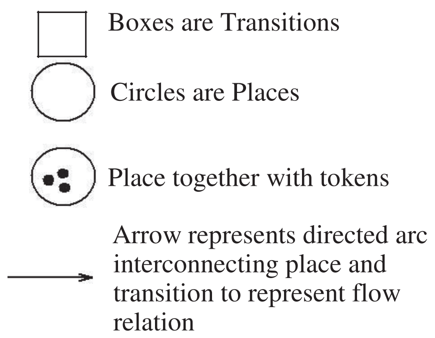

Figure 2: Elements of Graphical Representation of a Petri net

2.3 Petri net graph

The quadruple (P, T, Γ^-, Γ^+) in the matrix definition of the Petri net is called the *Petri net structure*. The *Petri net graph* is a representation of the Petri net structure, which is essentially a bipartite directed multigraph. Graphical representation of a Petri net structure is much useful for illustrating the concepts of Petri net theory.

For example, following is the Petri net structure in the quadruple (P, T, Γ^-, Γ^+) such that: $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3, t_4\}$, $\Gamma^-(p_1, t_1) = 1$, $\Gamma^-(p_1, t_4) = 3$, $\Gamma^-(p_2, t_2) = 1$, $\Gamma^-(p_3, t_1) = 1$, $\Gamma^-(p_3, t_3) = 1$, $\Gamma^+(p_1, t_1) = 1$, $\Gamma^+(p_2, t_1) = 2$, $\Gamma^+(p_2, t_3) = 1$, and $\Gamma^+(p_3, t_4) = 1$.

In Figure 3, Petri net graph of above Petri net structure has been shown.

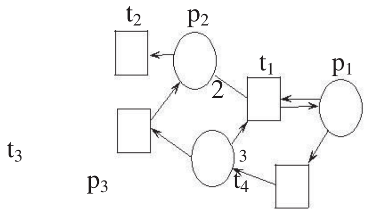

Figure 3: A Petri net graph

2.4 Marking of a Petri net

An arbitrary distribution of tokens in the places is called a *marking*. The initial distribution of tokens in the places is called the *initial marking* and it is denoted by μ^0.

In other words, the *initial marking* is represented by placing some tokens, in the circle representing a place p_i, $1 \le i \le n = |P|$. In general, a marking is a mapping $\mu: P \rightarrow \mathbf{N}$. A marking μ can hence be represented as a vector $\mu \in \mathbf{N}^n$: n=|P, such that the i^{th} component of μ is the value $\mu(p_i)$, viz., the number of tokens placed at p_i. The tokens are used to define the execution of a Petri net and the number and position of tokens may change during the execution of a Petri net at each interval of time of its operation.

In Figure 4 the initial marking $\mu^0 = (3, 0, 1)$, i.e., $\mu^0(p_1) = 3$, $\mu^0(p_2) = 0$ and $\mu^0(p_3) = 1$.

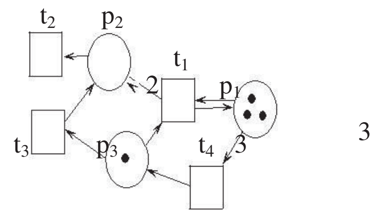

Figure 4: A Petri net graph having tokens at its places

2.5 Transition enabling and ring rule

By changing distribution of tokens on places, which may reflect the occurrence of events or execution of operations, for instance, one can study dynamic behavior of the modeled system. The following rule is used to govern the flow of tokens:

Let $C = (P, T, \Gamma^{-}, \Gamma^{+}, \mu\ )$ be a Petri net. A transition $t \in T$ is said to be *enabled* at if and only if $\Gamma^{-}(p;\ t) \leq \mu\ (p),\ \forall p\ \in P$. An enabled transition may or may not 'fire' (depending on whether or not the event actually takes place). After firing at , the new marking $'$ is given by the rule

$$\mu'(\mathrm{p}) = (p) - \Gamma^{-}(p, t) + \Gamma^{+}(p, t);\ \forall p \in P$$

Note that since only enabled transitions can fire, the number of tokens in each place always remains non-negative when a transition is fired. Firing transition can never remove a token that is not there at any place.

2.6 Firing sequences and reachability

A transition t in a Petri net $C = (P, T, \Gamma^{-}, \Gamma^{+}, \mu)$ fi*res at* μ to yield μ' (or, that *t fi res* μ to μ'), and we write $\mu \rightarrow (\mathrm{t})\ \mu'$, whence μ' is said to be *directly reachable* from . Hence, it is clear, what is meant by a sequence like $\mu^0 \rightarrow (\mathrm{t}_1)\ \mu^1 \rightarrow (\mathrm{t}_2)\ \mu^2 \rightarrow \ldots \rightarrow (\mathrm{t}_k)\ \mu^k$ which simply represents the fact that the transitions $\mathrm{t}_1,\ \mathrm{t}_2,\ \mathrm{t}_3, \ldots\ \mathrm{t}_k,$ have been successively fired to transform the *initial marking* μ^0 into the *terminal* marking μ^k and called firing sequence.

A marking is said to be *reachable from* μ^0, if there exists a firing sequence of transitions which successively fire to reach the state from μ from μ^0.

In the following Figure 5, consider the simple Petri net with the initial marking, $\mu^0 = (1, 2, 1, 0)$ and by a simple Petri net we mean multiplicity of arcs are not allowed.

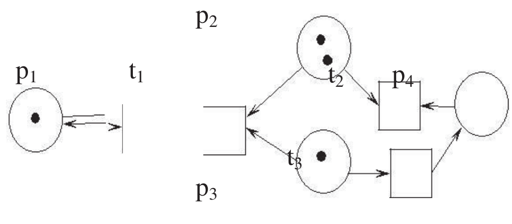

Figure 5: A simple Petri net

Under the initial marking, $\mu^0 = (1, 2,,1,0)$, only t_1 and t_3 are enabled.

After, firing of transition t_1, the new token distribution, i.e., new marking $\mu^1 = (1, 1,0, 0)$ is given by the firing rule.

$$\mu^1(p_1) = \mu^0(p_1) - \Gamma^{-}(p_1, t_1) + \Gamma^{+}(p_1, t_1) = 1 - 1 + 1 = 1;$$

$\mu^1(p_2) = \mu^0(p_2) - \Gamma^-(p_2, t_1) + \Gamma^+(p_2, t_1) = 2 - 1 + 0 = 1;$
$\mu^1(p_3) = \mu^0(p_3) - \Gamma^-(p_3, t_1) + \Gamma^+(p_3, t_1) = 1 - 1 + 0 = 0;$
and $\mu^1(p_4) = \mu^0(p_4) - \Gamma^-(p_4, t_1) + \Gamma^+(p_4, t_1) = 0 - 0 + 0 = 0;$
and is shown in Figure 6.

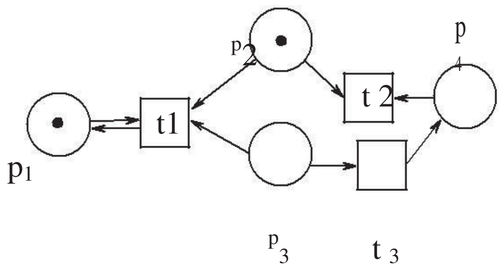

Figure 6: Firing of transition t_1 in Petri net as shown in Figure 5

After, firing of transition t_3, the new token distribution of the Petri net as shown in Figure 5, i.e., new marking $\mu^2 = (1,2,\ 0,\ 1)$ is given by the firing rule.
$\mu^2(p_1) = \mu^0(p_1) - \Gamma^-(p_1, t_3) + \Gamma^+(p_1, t_3) = 1 - 0 + 0 = 1;$
$\mu^2(p_2) = \mu^0(p_2) - \Gamma^-(p_2, t_3) + \Gamma^+(p_2, t_3) = 2 - 0 + 0 = 2;$
$\mu^2(p_3) = \mu^0(p_3) - \Gamma^-(p_3, t_3) + \Gamma^+(p_3,t_3) = 1 - 1 + 0 = 0;$
and $\mu^2(p_4) = \mu^0(p_4) - \Gamma^-(p_4, t_3) + \Gamma^+(p_4, t_3) = 0 - 0 + 1 = 1$
is shown in Figure 7.

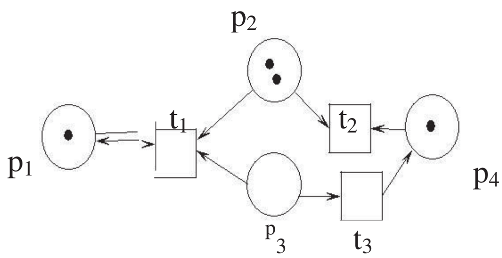

Figure 7: Firing of transition $t3$ in Petri net as shown in Figure 5

In the second stage of firing of Petri net shown in Figure 5, at the marking $\mu^1 = (1,1, 0, 0)$, none of the transitions is enabled. But at the marking $\mu^2 = (1, 2, 0, 1)$ only the transition t_2 is enabled and fires. After firing, it give the marking $\mu^3 = (1, 1, 0, 0)$ as shown in Figure 8. For the third stage of firing at $\mu^3 = (1, 1, 0, 0)$, none of the transitions is enabled and hence the system halts.

Note that the markings $\mu^1 = (1, 1, 0, 0)$ and $\mu^2 = (1, 2, 0, 1)$ are directly reachable from the initial marking $\mu^0=(1,2,1,0)$ and can be written as $\mu^0 \rightarrow (t_1)\ \mu^1$, $\mu^0 \rightarrow (t_3)\ \mu^2$.

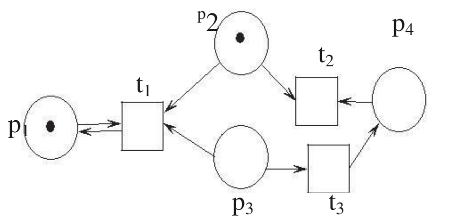

Figure 8: Firing of transition t_2 at $\mu^3 = (1, 1, 0, 0)$ of Petri net as shown in Figure 5

2.7 Self-loop

A pair of a place p and a transition t is called a *self-loop* if p is both an input and output place of t. A Petri net is said to be *pure* if it has no self loops [6]. In Figure 5, the pair of symmetric arcs (p_1, t_1) and (t_1, p_1) forms a self-loop.

It can be proved that if the place belonging to the loop cannot be found in the input set of any other transition and the transition has no other input place then once the transition becomes enabled, it remains enabled throughout the execution of the net. The self-loop can refer to a continuous operation like a catalyst of any device because after firing of transition, token remains at the same place.

2.8 Preset and postset of a place/transition

The *preset* of a transition t is the set of all input places to t, i.e., $^{\bullet}t=\{p \in P : \Gamma^-(p; t) > 0\}$. The *postset* of t is the set of all output places from t, i.e., $t^{\bullet}=\{p \in P : \Gamma^+(p; t) > 0\}$. Similarly, $p's$ preset and postset are $^{\bullet}p=\{t \in T : \Gamma^+(p; t) > 0\}$ and $p^{\bullet}=\{t \in T : \Gamma^-(p; t) > 0\}$, respectively.

In Figure 4,
$^{\bullet}t_1 = \{p_1, p_3\}$, $^{\bullet}t_2 = \{p_2\}$, $^{\bullet}t_3 = \{p_3\}$, $^{\bullet}t_4 = \{p_1\}$, $t^{\bullet}_1 = \{p_1; p_2\}$, $t^{\bullet}_2 = \emptyset$, $t^{\bullet}_3 = \{p_2\}$, $t^{\bullet}_4 = \{p_3\}$,
$^{\bullet}p_1 = \{t_1\}$, $^{\bullet}p_2 = \{t_1, t_3\}$, $^{\bullet}p_3 = \{t_4\}$, $p^{\bullet}_1 = \{t_1, t_4\}$, $p^{\bullet}_2 = \{t_2\}$, and $p^{\bullet}_3 = \{t_1, t_3\}$.

2.9 Incidence matrix

Let $C = (P, T, \Gamma^-, \Gamma^+, \mu^0)$ be a Petri net with $|P|=n$ and $|T|=m$, the incidence matrix $I=[a_{ij}]$ is an nxm matrix of integers and its entries are given by $a_{ij}=a_{ij}^+-a_{ij}^-$, where $a_{ij}^+ = \Gamma^+(p_i,tj)$ is the numberof arcs from transition t_j to its output place p_i, known as positive incidence matrix and $a_{ij}^- = \Gamma^-(p_i,t_j)$ is the number of arcs from place p_i to its output transition t_j, known as negative incidence matrix. In other words, $I= \Gamma^+ - \Gamma^-$.

2.10 Source/sink transition

A transition without any input place is called a *source transition*, i.e., ${}^{\bullet}t = \emptyset$. A source transition is unconditionally enabled that represents an event or operation which can occur in every step.

A transition without any output place is called a *sink transition*, i.e., $t^{\bullet} = \emptyset$. The firing of a sink transition consume tokens but does not generate new token in the net.

In Figure 9, transition t_3 is only a source transition as ${}^{\bullet}t_3 = \emptyset$ and the transition t_2 is only a sink transition such that $t^{\bullet}_2 = \emptyset$.

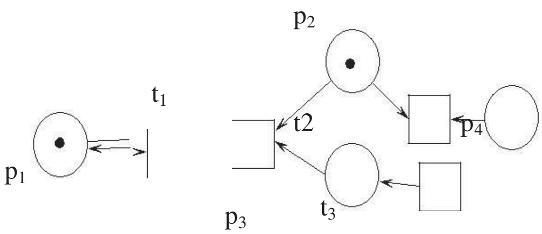

Figure 9: A Petri net having a source transition and a sink transition

3. PETRI NET PRIMITIVES TO REPRESENT SYSTEM FEATURES

The typical characteristics exhibited by the activities in a dynamic event-driven system, such as concurrency, decision making, synchronization and priorities, can be modeled effectively by Petri nets.

3.1 Sequential execution

Given the initial marking, in *sequential* execution only transition t_1 may occur. Then, transition t_2 may only fire after the firing of transition t_1 and transition t_3 may only fire after the firing of transition t_2 (see Figure 10).

3.2 Con flict

Figure 11 models the *conflict* among transitions. Given the actual marking, all the transitions t_1, t_2, and t_3 are enabled. However, if any of such transitions fires, then the others are disabled. We also say that these transitions are in *choice* or *decision* among them.

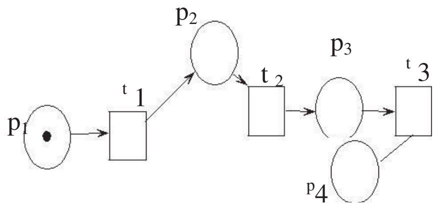

Figure 10: Sequential firing of transitions

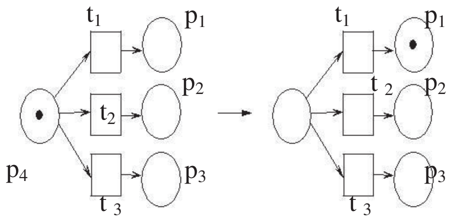

Figure 11: Conflict

3.3 Concurrency

Concurrency is an important attribute of system interactions. Note that a necessary condition for transitions to be concurrent is the existence of a forking transition that deposits a token in two or more output places. Figure 12 models concurrency. After the firing of the only enabled transition t_1 at the initial marking, all the transitions t_2, t_3 and t_4 are independently enabled and may fire in any order, even simultaneously.

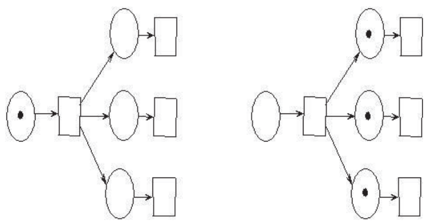

Figure 12: Concurrency

3.4 Synchronization

It is quite normal in a dynamic system that a transition requires multiple resources. The resulting *synchronization* of resources can be captured by transitions of the type shown in Figure 13. Here, the transition t_4 is enabled only when each of p_1, p_2 and p_3 receives a token. The arrival of a token into each of the three places could be the result of a possibly complex sequence of operations elsewhere in the rest of the Petri net model. Essentially, transition t_4 models the joining operation.

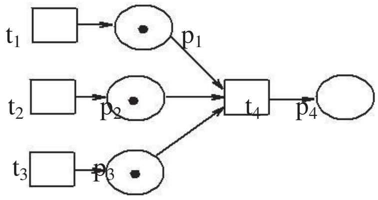

Figure 13: Synchronization

3.5 Mutually exclusive processes

Two processes are *mutually exclusive* if they cannot be performed at the same time due to constraints on the usage of shared resources. Figure 14 shows this structure. For example, a robot may be shared by two machines for loading and unloading. Two such structures are parallel mutual exclusion and sequential mutual exclusion.

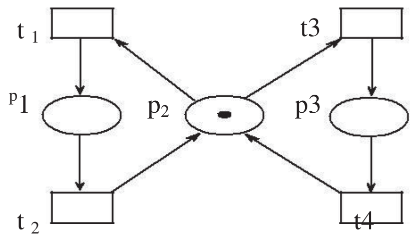

Figure 14: Mutually Exclusive processes

3.6 Priority

The classical Petri nets discussed so far have no mechanism to represent priorities. Such a modeling power can be achieved by introducing an inhibitor arc. The inhibitor arc connects an input place to a transition, and is pictorially represented by an arc terminated with a small circle. The presence of an inhibitor arc connecting an input place to a transition changes the transition enabling conditions. In the presence of the 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 at least the number of tokens equal to the weight of the arc, and no tokens are present on each input place connected to the transition by the inhibitor arc. The transition firing rule is the same for normally connected places. The firing, however, does not change the marking in the inhibitor arc connected places. A Petri net with an inhibitor arc is shown in Figure 15. t_1 is enabled if p_1 contains a token, while t_2 is enabled if p_2 contains a token and p_1 has no token. This gives priority to t_1 over t_2.

4 PROPERTIES OF PETRI NETS

As a mathematical tool, Petri nets possess a number of properties. These properties, when inter-preted in the context of the modeled system, allow the system designer to identify the presence or absence of the application domain specific functional properties of the system under design. Two types of properties can be distinguished, behavioral and structural ones. The behavioral properties

10

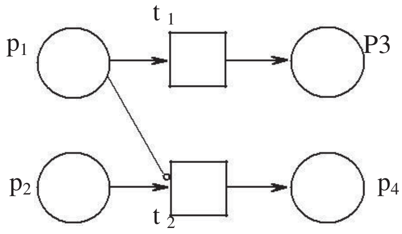

Figure 15: Priority

are those which depend on the initial state or initial marking of a Petri net. The structural prop-erties, on the other hand, do not depend on the initial marking of a Petri net. They depend on the topology, or net structure, of a Petri net. Here we provide an overview of some of the most important, from the practical point of view, behavioral properties as well as structural properties.

4.1 Behavioral properties of Petri nets

The reachability, boundedness, safeness, liveness, terminating marking, deadlock, conservation, reversibility, home state, coverability, persistence and fairness are oftenly referred behavioral proper-ties.

4.1.1 Reachability

An important issue in designing event-driven systems is whether a system can reach a specific state, or exhibit a particular functional behavior. Therefore, it is necessary to know if there is a firing sequence that can bring the net to a new pre-designated marking. This problem is referred to as the reachability problem.

A marking μ is said to be *reachable from* μ^0, if there exists a firing sequence of transitions which successively fire to reach the state from μ^0. The set of all markings of a Petri net *C* reachable from the marking μ^0 is denoted by M(*C*, μ^0) and, together with the arcs of the form $\mu^i \rightarrow (t_r)\ \mu^j$ represents what in standard terminology is called the *reachability graph* of the Petri net *N*, denoted by *R*(*C*, μ^0). It can be shown that the reachability graph of any Petri net is weakly connected, in the sense ~~that a~~ny two of its nodes are connected by a semipath (cf.: [12]). In particular, if the reachability graph has no semicycle then it is called the *reachability tree* of the Petri net.

The reachability set of Petri net shown in Figure 4 with the initial marking $\mu^0 = (3, 0, 1)$ is given by *{*(3,0,0), (3,2,0), (3,1,0), (0,0,2), (0,2,1), (0,1,1), (0,3,0), (0,0,1), (0,2,0), (0,1,0), (0,0,0), (3,0,1)*}*

4.1.2 Boundedness and Safeness

In a Petri net, places are often used to represent information storage areas in communication and computer systems, product and tool storage areas in manufacturing systems, etc. It is important to be able to determine whether proposed control strategies prevent from the overflows of these storage areas. The Petri net property which helps to identify the existence of overflows in the modeled system is the concept of boundedness.

A place $p_i \in P$ of a Petri net C = (P, T, I^-, I^+, μ^0) is *k*-bounded or *k*-safe if every $\mu' \in M(C; {}^0)$ is such that $\mu'(p_i) \leq k$. A Petri net is *k*-bounded if all the places in it are *k*-bounded.

A place in a Petri net is 1-*safe* if the number of tokens in that place never exceeds one. A Petri net is 1-*safe* if all its places are 1-safe. It is also called 1-bounded because it is the particular case of *k*-bounded Petri nets.

The interpretation of safeness and boundedness depends on the system to be modeled. The presence of more than one token in a place may mean many different things. For example, there can be two pumps in the system, which are represented by two tokens in a place. The boundedness property is especially useful when we want to define the size of the system or to reveal some design errors, i.e., boundedness represents the absence of overflow in Petri nets. The concept of safe and bounded nets is independent of the firing sequences. It should be noticed that boundedness indicates the finiteness of reachability space. Places in a Petri net are often used to represent buffers and registers for storing intermediate data. By verifying that the net is bounded or safe, it is guaranteed that there will be no overflows in the buffers or registers, no matter what firing sequence is taken.

Petri net shown in Figure 4 is 3-bounded. Petri net in Figure 9 is unbounded because the source transition t_3 is unconditionally enabled and may fire and, as a result, token count will keep on increasing infinitely.

The following Petri net model for seasons and their evolutions [9] as shown in Figure 16 is the example of 1-safe Petri net, i.e., number of tokens in each dynamic state cannot exceed one.

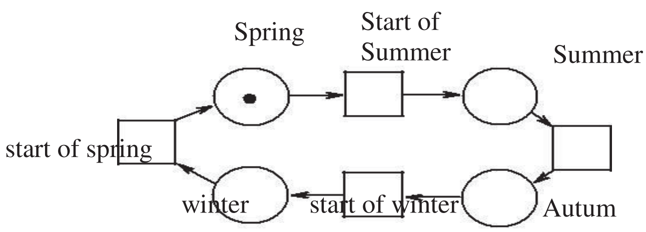

Figure 16: Petri net model for seasons and their evolutions

4.1.3 Terminating marking

A Petri net is *terminating* if there is no infinite firing sequence. The Petri net of Figure 4 has terminating marking.

4.1.4 Deadlock

A transition is *deadlocked* if it can never fire.

4.1.5 Dead

A transition is *dead* if and only if it does not appear in the reachability tree as an arc. In Figure 5, the transition t_2 is in dead situation.

4.1.6 Liveness

Liveness implies the lack of deadlocks, but not vice-versa. A Petri net is *live* if and only if from any reachable state and for any transition it is possible to reach a state from which the transition is firable. Note that the Petri net in Figure 16 is live.

Liveness addresses the equation whether it is always possible to activate a specific transition or the system can reach a state where this transition is ”dead”. As a generalization of this problem can investigate whether the system can reach a state where there is no enabled transition at all. This state is called a dead-lock. A system can get into a dead-lock when the operating procedure is over but it could also happen that the process stops before the final state. A dead-lock is very dangerous in the latter case because it refers to a state where the operator has no possibility to intervene, that is the system is out of control.

4.1.7 Conservation

The conservation property is related to the changes in the sum of tokens in Petri net during execu-tion. A Petri net is *strictly conservative* if the number of tokens is the same in all markings starting from an initial marking. Petri net in Figure 16 is conservative because for all the marking the sum of tokens is one.

Strict conservation is a very strong property. It can be useful in the case of modeling resource allocation systems where tokens may represent the resources. It is a very natural requirement for these systems that tokens are neither created nor destroyed there. It is possible to assign conservation weight to places and check the sum based on the linear combination of tokens computed using the place weights. In this case the weighted sum for all reachable markings should be constant for a conservative Petri net.

4.1.8 Reversibility and home state

A Petri net is said to be *reversible* if, for each marking $\mu \in M(C, \mu^0)$, μ^0 is reachable from . In many applications, it is not necessary to get back to the initial state as long as one gets back to some (home) state. Therefore, we relax the reversibility condition and define a home state.

A marking $'$ is said to be a *home state* if, for each marking $\mu \in M(C,\mu^0)$, μ' is reachable from μ.

4.2 Structural properties of Petri nets

The most important structural question in the analysis of Petri nets is the determination of *place and transition invariants*. A place invariant is a set of places, in which the sum of tokens remains constant, independently of which transition fires. Tokens of this set of places are neither generated nor consumed, only ”moved” between places.

A transition invariant is a set of transitions. When these transitions fire starting from an initial state then the system returns to the same initial state. Transition invariants correspond to the cyclic behaviors of the modeled system.

There are two major classes of Petri net analysis techniques: the construction and analysis of the reachability graph and use of matrix equations.

The aim of constructing a reachability graph is to answer the initial state dependent questions. It involves the determination of all possible markings that belong to a given initial state. On the another hand matrix equations are used for determining structural properties. Both techniques can be implemented on a computer. The use of computers is very important during the analysis because apart from some simpler cases the analysis of the above mentioned properties is very difficult without software support.

5 SOME APPLICATIONS OF PETRI NETS

Chemical Systems [11]

A chemical process can be modeled by Petri nets. Chemical reactions are modeled by transitions such that every transition has a finite set of input states and output states, places (also called states) are the chemical components and tokens represent molecules of a given kind.

For example, the following chemical reactions may be represented as a Petri net as shown in Figure 17 and Figure 18.

$C + O_2 \rightarrow CO_2$
$CO_2 + NaOH \rightarrow NaHCO_3$
$NaHCO_3 + HCl \rightarrow H_2O + NaCl + CO_2$

Petri net models shown in Figure 17 and Figure 18 describe a situation when one atom of C, one molecule of O_2, one molecule of $NaOH$ and one molecule of HCl are present. No molecule of CO_2, $NaHCO_3$, H_2O or HCl are present at the initial time.

We can then describe the process that occurs with the passage of time by moving the tokens around. If C reacts with O_2, then token movement after firing is shown as in Figure 17.

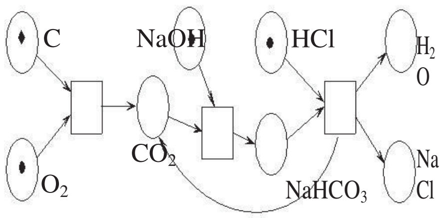

Figure 17

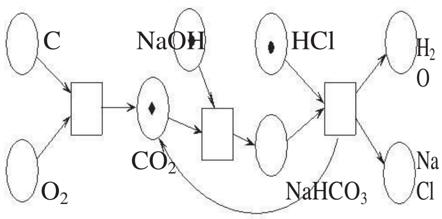

Figure 18:

If CO_2 combines with $NaOH$ to form $NaHCO_3$, then after firing, token moves to $NaHCO_3$ then $NaHCO_3$ and HCl react and the token moves to the places H_2O and $NaCl$. In this way, the chemical process can be visualized by using Petri nets.

Petri net Model for Hair Dresser's Saloon

A hair dresser's saloon can be modeled by a Petri net. Activities e.g., start, finish etc. modeled by transitions, conditions like waiting, busy, free, ready etc. can be modeled by places and token represents the number of client waiting, hair dresser ready to begin etc. In Figure 19, when the condition is met, i.e, sufficient number of clients and hair dressers are ready to begin, the transition 'start' will be enabled and fires. After firing, token moves to the place 'busy'. After that the transition, 'finish' is enabled and fires, and token moves to the places 'free' and 'ready'.

Then, after completing the process, either the customer will leave the saloon's and hair dresser will be ready to begin again. In this way, this process will continue.

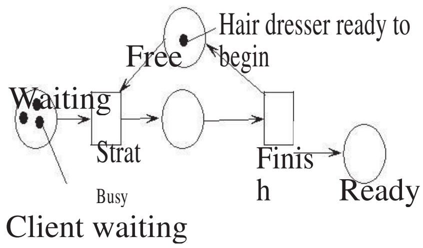

Figure 19:

Petri net model for traffic lights Traffic light signals can be modeled by a Petri net. for instance, Here we are explaining the modeling of single traffic light by the use of a Petri net. Here, places are the lights red, green and yellow and transitions indicate the change of lights like red to green, green to yellow and yellow to red. Token in a place indicates that particular light is on. In Figure 20, token in the place 'red' means red light is on and when the condition is met to change red to green light then transition 'red to green' is enabled and fires. After firing, red light token moves to the place 'green', meaning that red light is off and green light is on. By the same logic one can give interpretations for the occurrence of other color lights as well.

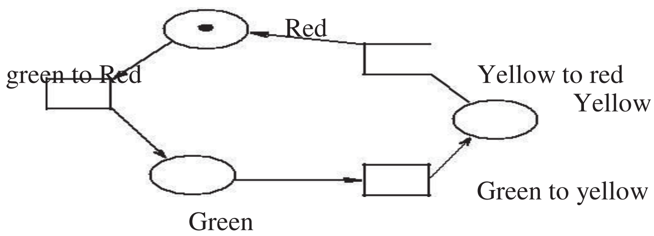

Figure 20

6 SCOPE FOR RESEARCH

Which Petri nets generate all the binary n-vectors as their marking vectors? This has been a hotly pursued research problems. Kansal *et al.* [5] proposed a 1- safe *star Petri net* S_n, with $|P| = n$ and $|T| = n + 1$, having a central transition, that generates all the binary n-vectors, as its marking vectors. The structure of a 1-*safe star Petri net* S_n is obtained by *subdividing* every edge of the graph $K_{1;n}$; $n \geq 1$, so that every subdividing vertex is a place node and the original vertices of $K_{1;n}$; $n \geq 1$, are the $(n + 1)$ transition nodes, $(n + 1)^{th}$ being the central node which is a transition. Further, every arc incident to the central node is directed toward it, and every arc incident to a pendent node is directed towards the pendent node, which is shown as in Figure 21.

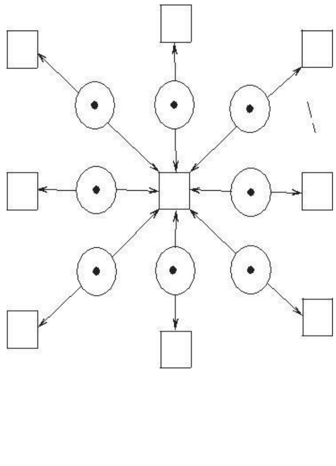

Figure 21: 1-safe star Petri net S_n

In the design of generalized cyclic multi-switches[1] such as those used to control automatic washing machines, suppose that we have a cyclic sequence of $2^n - 1$ terminals each of which can be at either a low-voltage (denoted by zero ‘0′) or a high-voltage (denoted by unity, ‘1’). It is required to arrange them so that all but one of the 2^n sequences of n bits[11], corresponding to the 2^n binary (or ‘0-1’) n-tuples can appear on the terminals around the cycle C_2n_{-1} so that the Hamming distance between adjacent pairs of terminals are all distinct. Can one design a 1-safe Petri net that automatically generates such an assignment of voltages?. Therefore, a computationally good characterization of such 1-safe Petri nets generating all the 2^n binary n-vectors is needed.

Acknowledgement:

The author deeply acknowledges and thankful to the my respected teachers Dr. B.D. Acharya, Dr. Sangita Kansal and Dr. Mukti Acharya for the encouragement to do the work in Petri net Theory.

References

[1] Acharya, B.D., *Set-indexers of a graph and set-graceful graphs*, Bull. Allahabad Math. Soc. 16, 2001, 1-23.

[2] Best, E., and Thiagarajan, P.S., Some classes of Live and Safe Petri nets, Concurrency and Nets: Advances in Petri nets, Springer-Verlag New York, 1987, 71-94.

[3] Harary, F., *Graph theory*, Addison-Wesley, Massachusettes, Reading, 1969.

[4] Jensen, K., *Coloured Petri nets*, Lecture notes in computer science, 254, Springer-Verlag, Berlin, 1986, 248-299.

[5] Kansal, S., Singh, G. P. and Acharya, M., *On Petri nets generating all the binary n-vectors*, Scientiae Mathematicae Japonicae, 71(2), 2010), 209-216.

[6] Murata, T., *Petri nets: Properties, analysis and applications*, Proc. IEEE, 77(4), 1989, 541-580.

[7] Petri, C.A., *Kommunikation mit automaten*, Schriften des Institutes fur Instrumentelle Math-ematik, Bonn 1962.

[8] Peterson, J.L., Petri net theory and the modeling of systems, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981.

[9] Reisig, W., Petri nets, Springer-Verleg, New York, 1985.

[10]www.azimuthproject.org.

[11]Birkhoff, G. and Bartee, T.C., Modern Applied Algebra, CBS Publishers & Distributors, New Delhi, India, New Delhi, 1999, ISBN 978-81-85015-89-7.

[12]Harary, F., Norman Z. R. and Cartwright, D., Structural Models: An introduction to the theory of directed graphs, John Wiley & Sons, Inc., Boston, New York, 1965.