1 Introduction

The Petri Nets have been originally proposed by German mathematician Carl Adam Petri in 1962. Since then huge amount of papers and books have been published in this field. Design and analysis of Petri Nets is based on strict and definite mathematical theory. Software tools developed have made Petri Nets a powerful mechanism for modelling and analysis particular applications.

The theory of Petri Nets provides a well-defined theoretical mechanism for modeling of the discrete event systems and analysis of their characteristics. Petri Nets have been successfully used for learning behavior of various problems arising in certain fields of science, engineering and industry including serial and parallel processes, problem solving in computer science and computer engineering. The use and study of Petri Nets has spread widely in the last years. Computers and their components, computer networks, computer programs, operating systems, discrete industrial systems such as factory pipelines are just few areas where discrete dynamic systems can be
successfully implemented. Various mathematical models and methods have been used for investigating behavior of a discrete system. Petri Nets represent one of the mathematical techniques for such systems. Implementing Petri Nets to analysis of discrete systems allows us to get sure about whether or not these discrete systems can efficiently work.

2 Basic concepts

2.1. Formal Definitions

Petri nets can be described by tuple $PN = <P, T, I, O>$ satisfying the following requirements:

- $P$ is a finite set of places denoted $P = \{p_1, \ldots, p_m\}$.
- $T$ is a finite set of transitions denoted $T = \{t_1, \ldots, t_n\}$.
- $I$ is a finite set of at input arcs from $P$ to $T$.
- $O$ is a finite set of output arcs from $T$ to $P$.

A Petri Net can be represented as directed biparted graph. The following symbols are usually used to denote the components of a Petri Net.
A simple Petri Net is shown in Fig.2.2. In this figure the transition $t_1$ has one input place $p_1$, and two output places $p_2$ and $p_3$. In this Petri Net, $PN = \{p_1, p_2, p_3, p_4\}$, $T = \{t_1, t_2\}$, and input and output arcs are defined as follows:

\[
I(t_1) = p_1, I(t_2) = p_2, I(t_2) = p_3,
O(p_1) = t_1, O(p_2) = t_2, O(p_3) = t_2.
\]

A Petri Net is an object that is characterized by its states. Each state is defined by the number of tokens in the places. A token can move from place to another place if
certain conditions are satisfied. The basic rules which regulates the work of a Petri Net are discussed below.

Enable transition. A transition is said to be **enable transition** if each input place has at least one token. The number of tokens in places of enable transition must be equal to or greater than the number of arcs connecting those places with the transition. The transition $t_1$ of the Petri Net in Fig.2.3 is enable transition.

![Fig.2.3 Petri Net with enable transition $t_1$.](image)

Disabled Transition. We say that a transition is **disable transition** if the number of tokens in its input places is
less than the number of its input arcs. That is, the number of tokens in at least one of the input places is less than the number of input arcs from that place to the transition. For instance, \( t_1 \) in Fig.2.4 is disable transition.

![Diagram](image)

**Fig.2.5.** Petri Net's two states: initial, after \( t_1 \) fires.

**Firing Action.** An enable transition can *fire* (or *occur*). When transition fires its input places lose some of their tokens. The number of tokens removed from source places must be equal to the number of incoming arcs. Then output places of a transition receive tokens. The number of tokens received equal to the number of outgoing arcs. This is illustrated in the Fig.2.5.

![Diagram](image)

**State 1:**

\[ \downarrow \text{Firing} \]

**State 2:**

![Diagram](image)

**Fig.2.6** Two states of another Petri Net.
The initial state of Petri Net with $P = \{p_1, p_2\}$ and $T = \{t_1\}$ in Fig.2.6 is $(1,0)$. After $t_1$ fires it changes to $(0,1)$. The input arcs and outputs of this Petri Net are defined as follows:

<table>
<thead>
<tr>
<th>Input functions</th>
<th>Output functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I(t_1) = {p_1}$</td>
<td>$O(t_1) = {p_2}$</td>
</tr>
<tr>
<td>$I(p_1) = {}$</td>
<td>$O(p_1) = {t_1}$</td>
</tr>
<tr>
<td>$I(p_2) = {t_1}$</td>
<td>$O(p_2) = {}$</td>
</tr>
</tbody>
</table>

![Fig.2.7 A Petri Net.](image)

![Fig.2.8 A Petri Net after $t_1$ fires.](image)
Fig. 2.9 A Petri Net after $t_2$ fires.

Fig. 2.10 A Petri Net after $t_3$ fires.

Fig. 2.11 Petri net with initial setting $M_0 = (1,1,1,0)$. 
Each state of a Petri Net is described by allocation of tokens in its places. Consider the Petri Net in Fig.2.7. Initially, this Petri Net is enable. Its initial state is described by \( M_0 = (1, 0, 1, 0, 0, 0) \). Occurrence of \( t_1 \) changes the state of the Petri Net from \( (1, 0, 1, 0, 0, 0) \) to \( (0, 1, 1, 0, 0, 0) \). Then \( t_2 \) occurs. This cause the change of the Petri Net state from \( (0, 1, 1, 0, 0, 0) \) to \( (0, 0, 1, 1, 0, 0) \). Finally, firing of \( t_2 \) sets Petri Net to \( (0, 0, 1, 0, 1, 0) \). Then Petri Net becomes disable.

A state \( M_1 \) of the Petri Net is said to be \textit{reachable} from \( M_0 \) if there exists a sequence of occurrences of the transitions changing state of the Petri Net from \( M_0 \) to \( M_1 \). A state \( M_1 \) is \textit{directly reachable} from \( M_0 \) if there exists a single occurrence at state \( M_0 \) setting the Petri Net to \( M_1 \).

Sometimes multiple transitions are enable in the given state of a Petri Net. This means that each of them may occur. These transitions are called transitions \textit{concurrently enable} in the given state. In Fig.2.11, there are two pairs of trantions enabled in \( M_0 \). These are \( t_1 \) and \( t_2 \), and \( t_1 \) and \( t_3 \). This transitions may occur “at the same time” or “in parallel”. We also say each of the steps \( S_1 = \{ t_1, t_2 \} \) and \( S_2 = \{ t_1, t_3 \} \) is enable in \( M_0 \). A transition may even occur \textit{concurrently to itself}. If so, a step contains multiple copies of the transitions. In general, each step is a multi-set over the set of all transactions. We usually
denote a multiset by sums where each element has a coefficient telling how many times it appers. For instance in step $S_3 = \{ 2t_1, t_2, 2t_3 \}$, which is enable in $M_0$, transitions $t_1$, $t_2$ and $t_3$ occur 2, 1, and 2 times, respectively. This step sets the Petri Net to $\{3, 0, 2, 2\}$.

In general, a place can get infitely many tokens. If the number of tokens in a place is large enough we denote them by numbers rather than put dots in a place. For example, a Petri Net in Fig.2.12 is characterized by the initial state $\{47, 13, 7, 42\}$.

![Fig.2.12. A Petri Net.](image-url)