ON SIGNED PETRI NETS

A Thesis Submitted to

DELHI TECHNOLOGICAL UNIVERSITY

for the Award of the Degree of

DOCTOR OF PHILOSOPHY

In

MATHEMATICS

By

PAYAL

Under the Supervision of

Prof. Sangita Kansal

DEPARTMENT OF APPLIED MATHEMATICS

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

BAWANA ROAD, DELHI-110 042, INDIA.

September, 2023 **Enroll. No. : 2K16/Ph.D./AM/03**

DECLARATION

I declare that the research work reported in this thesis entitled "**On Signed Petri Nets**" for the award of the degree of *Doctor of Philosophy in Mathematics* has been carried out by me under the supervision of *Prof. Sangita Kansal*, Department of Applied Mathematics, Delhi Technological University, Delhi, India.

The research work embodied in this thesis, except where otherwise indicated, is my original research. This thesis has not been submitted by me earlier in part or full to any other University or Institute for the award of any degree or diploma. This thesis does not contain other person's data, graphs or other information, unless specifically acknowledged.

Date : **(Payal)**

CERTIFICATE

On the basis of declaration submitted by **Ms. Payal**, a student of Ph.D., I hereby certify that the thesis titled **"On Signed Petri Nets"** submitted to the Department of Applied Mathematics, Delhi Technological University, Delhi, India for the award of the degree of *Doctor of Philosophy in Mathematics*, is a record of bonafide research work carried out by her under my supervision.

I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy. To the best of my knowledge the work reported in this thesis is original and has not been submitted to any other Institution or University in any form for the award of any Degree or Diploma.

(Prof. Sangita Kansal)
Supervisor
Department of Applied Mathematics
Delhi Technological University
Delhi.

(Prof. S. Sivaprasad Kumar)
Head
Department of Applied Mathematics
Delhi Technological University
Delhi.

ACKNOWLEDGMENTS

I have received encouragement and support from lot of people during my Ph.D. work and I never would have been able to complete my Ph.D. without their guidance and support. The thesis is the most suitable way to express my gratitude to all of them.

First and foremost, I wish to express my sincere gratitude to my supervisor Prof. Sangita Kansal, Department of Applied Mathematics, Delhi Technological University (DTU), Delhi for her inspiring guidance and support during my research work, and for giving me the freedom to explore my own ideas. It is indeed a great pleasure for me to work under her supervision. I appreciate all her contributions of time and ideas to make my Ph.D. experience productive and stimulating. The joy and enthusiasm she has for her research was contagious and motivational for me. She has been truly a role model for me not in just research but, all the aspects of life due to the way she handles everything with such ease.

I sincerely thank Prof. S. Sivaprasad Kumar, Head, Department of Applied Mathematics, DTU, for providing me the necessary facilities, valuable suggestions and motivation during the progress of the work. I also extend my sincere thanks to all faculty members of the Department of Applied Mathematics, DTU for their constant support and encouragement.

I gratefully acknowledge the academic branch and administration of DTU for providing the environment and facilities to carry out my research work. I also express my thanks to office staff of Department of Applied Mathematics for their support.

I wish to record my profound gratitude to my late grandfather who always motivated me through his words of wisdom. I am indebted to my parents and in-laws who have provided me all kinds of support and help for my academic achievements, and for their constant love and care. I dedicate my work to my late father and my mother without whom my Ph.D. journey would never be possible. My father was my biggest motivator and I hope I have made him proud. I would especially like to thank my brother Anant who has been with me through all ups and downs and constantly supported me in every way. My husband Gaurav has been a pillar of support throughout the journey. Last but not the least, a big Thank you to Viona and Viraj who always brought a smile on my face by their childish antics whenever I felt stressed.

I want to express my thanks to all of them who have not been mentioned here but supported, encouraged and inspired me during my Ph.D. work. Last but not the least, I thank to almighty God for showing me the right path to complete this Ph.D. thesis.

Thank you all.

Date :

Place : Delhi, India **(PAYAL)**

Dedicated to My Parents

Late Sh. Hawa Singh

&

Smt. Santosh Dabas

Contents

Preface

In today's complex scenario of technological advancement, the role of event-driven discrete dynamical systems have an impact on man's ability to fast-forward the futuristic technologies that are likely to bring unimaginable progress in our time, and near, far futures. One such great innovation that inspired this thesis is the "Theory of Petri nets", a modeling tool for event-driven discrete dynamical systems.

The study of Petri nets and its various extensions that have developed over time is one of the most active and vibrant areas of research in current time, owing to its applications in the fields of engineering and sciences. The structure of Petri nets is a directed bipartite graph. They can be used as a graphical as well as a mathematical tool. As a graphical tool, they are easier to understand and interactive in nature while as a mathematical tool, they can be used to formulate state and algebraic equations for easier calculation and analysis.

The notion of Petri net was discovered by Carl Adam Petri at a mere age of 13 to describe chemical processes. More formally, it was described in his thesis "Communication with Automata" in 1962, submitted to the Science Faculty of Darmstadt Technical University.

Due to their dynamic nature, Petri nets soon became useful in modeling of asynchronous, distributed, concurrent, parallel, nondeterministic, and/or stochastic systems. Thus, various extensions to Petri nets have been introduced viz. Continuous Petri nets, Stochastic Petri nets, Timed Petri nets, Object Petri nets, Hybrid Petri nets, Workflow nets, Fuzzy Petri nets, Lending Petri nets, Multidimensional Petri nets etc., in order to better incorporate the characteristics of the system to be modeled. The Petri nets and their extensions have been widely used in various fields. Logic Petri nets (LPNs) have been defined as high level Petri nets, to describe batch processing functions and passing value indeterminacy in cooperative systems.

The thesis entitled **"On Signed Petri Nets"** contains seven chapters.

Chapter 1 titled *"General Introduction"* provides a brief review of Petri net theory. It provides the contributions of various researchers who have extended the theory of Petri nets after its introduction by Carl Adam Petri. A brief survey of the Petri nets research is given. The various extensions and applications of Petri nets in some of the fields have been discussed in brief. Thus, this chapter builds up a background and motivation behind the thesis work along with the tools required to achieve the goals.

Chapter 2 titled *"Signed Petri net"* describes the extension of Petri nets called Signed Petri net and the related terminology. The proposed concept is inspired from signed graphs and Petri nets and can be considered as an amalgamation of the two, utilizing the properties of both. The basic properties and terms associated with signed Petri nets are defined. The applications of signed Petri nets in message transmission system and production unit are discussed. Lastly, it is shown how a Logic Petri nets can be simulated using a signed Petri nets by merely changing the execution rules of a signed Petri nets. These modified signed Petri nets are called a Logic signed Petri nets. These concepts clearly demonstrate the advantages of the proposed approach of signed Petri nets. This chapter is published as a research paper "An introduction to Signed Petri net, *Journal of Mathematics*, (2021)".

In **Chapter 3** titled *"Analysis of Signed Petri nets"*, the work of previous chapter has been extended. Mere modeling of system is of no use unless the modeled system is interpreted, which led to the introduction of analysis techniques for analyzing signed Petri nets. Two techniques for analysis are provided: Reachability Tree and Matrix equations with main focus on matrix equations. An actual case scenario of a restaurant model is given and analyzed using the techniques given in the chapter. The benefits of using a signed Petri nets to model the restaurant system rather than using traditional Petri nets are also given. This chapter has been accepted with title "Analysis of Signed Petri net, *International Journal of Computing Science and Mathematics*, (2020)".

In **Chapter 4** titled *"Structural Matrices for Signed Petri nets"*, several matrices which show the relationship between transitions and places have been introduced. Three different matrices are defined by different products of the adjacency matrix of signed Petri nets with its transpose and with itself. In fact, if all these matrices are given, the signed Petri net structure can be obtained after analyzing them. Such matrices are useful because while creating algorithms for various procedures and results, it is not possible to extract data from a graph rather all the information can be provided in the form of matrices. The matrices are further utilized to find a directed cycle in signed Petri nets. Various subclasses of signed Petri nets along with their characterizations using the structural matrices have been introduced. This chapter is published with title "Structural Matrices for Signed Petri net, *AKCE International Journal of Graphs and Combinatorics*, (2022)".

In **Chapter 5** titled *"Structural and Dynamical Balanceness in Signed Petri net"*, the concept of structural and dynamical balanceness have been introduced. A structurally balanced signed Petri net has been defined and its characterization is given. It is shown how the dynamical balanceness approach is advantageous in analyzing social interactions, since all the signed graphs (directed or undirected) can be simulated by firing of different sequence of transitions in a single signed Petri net. Also, dynamics associated with a system can be easily represented using signed Petri nets rather than a signed graph. The equivalence between balanced signed graphs and dynamically balanced signed Petri nets is established. This chapter comprises the result from the research papers "An introduction to Signed Petri net, *Journal of Mathematics*, (2021) and Social Interactions through Signed Petri net, Communicated".

In **Chapter 6** titled *"Domination in Signed Petri nets"*, the concept of Domination has been introduced as such a concept doesn't exist for the dynamic systems. It can be seen with the help of the applications of finding the highest and lowest ranking officials in an institute based on a certain activity, producer- consumer problem, searching of food by bees and finding similarity in research papers, how the proposed concept is beneficial. The work in this chapter is in the research paper "Domination in Signed Petri net, *https://arxiv.org/abs/2001.04374*, (2020)".

Chapter 7 titled *"Conclusion and Future Scope"* concludes the thesis work and gives a future research plan. In the research work, an extension of Petri nets called as "Signed Petri nets" has been proposed. Various results have been obtained in our present work along with some real-life applications where the proposed research can be used. The authors are of the view that this extension has a great scope in the study of various dynamic systems as well as modeling real-life applications. In future, the thesis work will be extended by utilizing the signed Petri nets in modeling various scenarios and analyzing the modeled system.

Lastly, the bibliography is given to appreciate those who have made it possible to understand and use the vast theory of Petri nets. A list of author's publications is also given at the end of the thesis.

List of Figures

List of Tables

Chapter 1

General Introduction

This chapter provides a review of the developments that have occurred or are presently going on in the field of Petri net Theory. It deals with discovery of Petri nets and their various extensions. It explores the work of various authors in the literature who used Petri nets in numerous applications and also introduced different extensions of Petri nets that were developed over time in order to suit the requirements. The important extensions of Petri nets have been discussed briefly. The theory of signed graph has also been briefly discussed. Thus, this chapter builds up a background and motivation behind the thesis work along with the tools required to achieve the goals.

Figure 1.1: Carl Adam Petri

The theory of Petri net is one of the most sought after and vibrant research area owing to its applications in various branches of engineering and sciences [11, 13, 17, 22]. It has become an active research field in Mathematics as a discrete dynamical system due to its structure which is a directed bipartite graph. It is useful in the field of Mathematics as it is not just a modeling tool but a mathematical one too, and is far more effective than other modeling tools such as graphs, flow charts, block diagrams, etc. One can set up state and algebraic equations for Petri nets to govern the behavior of dynamical systems.

Carl Adam Petri introduced Petri net at the age of 13 to model chemical reactions. Formally, he introduced Petri nets in his Ph.D. dissertation submitted in 1962 [28]. Petri nets are an excellent network combining the attributes of a well-established theory of mathematics and representation of the dynamic behavior of systems in graphical form.

The modeling and analysis of the behavior of the modeled system is done using theoretical aspect of Petri nets while the changes in the modeled system are visualized via its graphical representation. Hence, Petri nets are one of the most prominently used tool for describing and analyzing the asynchronous, distributed and concurrent dynamical systems.

Petri nets invented by Carl Adam Petri do have limitations and therefore, its various extensions [1, 2, 4, 6, 9, 23, 24, 35, 36] have evolved over time in order to represent all the variables describing characteristics of the modeled system. Some of these extensions have been briefly mentioned here.

A desire for a modeling language that can be used for complex and large sized industrial projects led to the development of Coloured Petri nets (CP-nets or CPN) by Jensen [19]. Coloured Petri nets are used as graphical oriented language for design, specification, simulation and verification of systems. They amalgamates the characteristics of Petri nets in the form of description of the synchronization of concurrent processes and programming languages in the form of definition of data types and the manipulation of data values.

Explosion of the number of reachable states with the increase in number of tokens is one of the practical limitation to the use of traditional Petri nets. An approximation for discrete event systems can be achieved with the help of continuous models, a key principle leading to the introduction of Continuous Petri nets [4]. The transitions in continuous Petri nets can be fired a fractional number of times, and hence, places may contain a fractional number of tokens. In timed continuous Petri nets, each transition has a firing speed attached with it. Further, Hybrid Petri nets were introduced since the state of a machine (up or down) cannot be modeled by a real number and hence, hybrid Petri nets contains a discrete part as well as a continuous part.

Another innovative modeling technique is obtained by providing the structure of Petri nets to a token in a place, also termed as nets-within-nets. These nets, called Token nets or Object nets can be used to represent real world objects with a proper dynamical behaviour. Due to preservation of some basic properties while modeling, the key issues that arises are the partial ordering among constituent events and the resultant properties after the arrangement of state and transitions. Also, such representations occur when analysis of systems of practical use is based on reachability property or other property. However, the processes can not be depicted completely using such approach viz. when it is no longer a good approximation to assume that all transitions can fire instantaneously. Therefore, firing transitions are associated with a time delay in T-time Petri nets [34].

Various extensions of the classical Petri nets exist in the literature, some of which are mentioned above. To model attributes, color, time concepts, temporal behaviour of a system, various modifications have been incorporated in Petri nets. Another extension that unite all these extensions are Multi-dimensional Petri nets [2]. An advantage of using this extension is that any number of dimension may be identified, e.g. a colour dimension, a spatial dimension, a time dimension, etc. Such Petri nets can be analyzed using traditional techniques and can be projected onto a limited number of dimensions. The properties of multi-dimensional Petri nets can be deduced by analyzing the projected multi-dimensional Petri nets. Therefore, they are of interest from modeling point as well as analysis point of view. The places of Petri nets can represent the dimensions in multi -dimensional Petri nets. Each token has n dimensions in n- dimensional Petri nets.

Another extension called Logic Petri nets (LPNs) [12], can be used instead of inhibition Petri nets due to their simpler structure. Logic Petri nets can be used to describe and analyze batch processing functions and passing value indeterminacy in cooperative systems. In LPNs, the transitions are restricted by logic expressions. They have been applied efficiently to the modeling and analysis of electronic commerce, web services and cooperative systems.

Petri nets have spread even to the field of applied stochastic modeling in the form of another extension called as Stochastic Petri nets (SPNs)[23]. The main aim behind the development of SPNs was to find a tool that will allow the integration of formal description, proof of correctness, and performance evaluation. An extension of Petri nets where places may carry negative number of token is called a Lending Petri nets [6]. These Petri nets are used to model scenarios where a participant promises to give some of his/her resources under the guarantee that some other resources will eventually be obtained in exchange. For modeling knowledge representation and reasoning of rule-based expert systems, Fuzzy Petri nets (FPNs) [9, 22] are a great tool.

1.1 Petri net

The Petri net theory has been developed independently by many authors with different backgrounds and different set of goals, and at different times. Some of the early researchers gave an informal definition of Petri nets with all the relevant components viz. places, transitions, tokens and execution rules. Formally, Petri nets were first defined by Patil [26]. Patil defined it as a 4-tuple (P,T,F,μ^0), where P is the set of places, T is the set of transitions, $F \subseteq (P \times T) \cup (T \times P)$, is called the flow relation and $\mu^0 : P \rightarrow \{1,2,3,\ldots\}$ is the initial marking. But with Patil's definition of Petri nets, there exists a possibility of isolated places or transitions, due to the only structural constraint being $F \subseteq (P \times T) \cup (T \times P)$. Hence, a restriction i.e., $dom(F) \cup codom(F) = P \cup T$, was assumed in work that followed. This restriction ensures the connectivity of every transition to some place and every place to some transition i.e., no event can occur without a resource and resource is neither produced nor consumed by an event. It is not required to stress the fact that at least one place and one transition are required. Therefore, Petri net is redefined further to give a standard definition of Petri net as a 4-tuple (P,T,F,μ^0), where $P \cup T \neq \emptyset$, $P \cap T = \emptyset$ and $F \subseteq (P \times T) \cup (T \times P)$ with $dom(F) \cup codom(F) = P \cup T$. The definition of Petri nets was generalized by Winskel [37] by taking F as a multiset of $(P \times T) \cup (T \times P)$ i.e., $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$, where $\mathbb{N}$ stands for set of non-negative integers. Therefore, in Winskel's definition of Petri net multiple arcs can exist from a transition to a place and vice-versa.

Formal definition of Petri net

The standard definition of *Petri net* is taken as a 4-tuple $N = (P,T,F,\mu^0)$, where,

1. P is a finite set of *'places '*.

2. T is a finite set of *'transitions '*.

3. $P \cap T = \emptyset$, $P \cup T \neq \emptyset$.

4. $F \subseteq (P \times T) \cup (T \times P)$, called *'Flow Relation'* is chosen such that $dom(F) \cup codom(F) = P \cup T$, where
   $dom(F) = \{x \in P \cup T \mid \exists\, y \in P \cup T \text{ and } (x,y) \in F\}$
   $codom(F) = \{x \in P \cup T \mid \exists\, y \in P \cup T \text{ and } (y,x) \in F\}$

5. $\mu^0 : P \to \mathbb{N}$, called the *'initial marking '*, is a function that assigns to each place p_i a non-negative integer $\mu^0(p_i)$, often referred to as the number of tokens in place p_i.

Winskel generalized the standard definition as a 4-tuple $N = (P,T,F,\mu^0)$, where

1. P is a non-empty set of *'places '*.

2. T is a non-empty of *'transitions '*.

3. $P \cap T = \emptyset$.

4. F is a multiset of $(P \times T) \cup (T \times P)$, called the *'causal dependency relation '*.

5. μ^0 is a non-empty multiset of places, called the *'initial marking'* which satisfies the following conditions:

   (a) $\forall p \in P,\ \exists\, t \in T$ such that $F_{p,t} > 0$ and
   $\forall t \in T,\ \exists\, p \in P$ such that $F_{t,p} > 0$

   (b) $\forall p \in P,\ \mu_0(p) \neq 0$ or $(\exists\, t \in T$ such that, $F_{p,t} \neq 0)$ or
   $\exists(\ t \in T$ such that, $F_{t,p} \neq 0)$

1.1.1 Jensen's Definition of Petri net

The definition of Petri net that is adopted throughout the thesis is the one given by Jensen. According to him, a *Petri net* is defined as a 5-tuple $N = (P, T, I^-, I^+, \mu_0)$, where,

1. P is a finite, non-empty set of *'places '*.

2. T is a finite, non-empty set of *'transitions '*.

3. $P \cap T = \emptyset$.

4. $I^-,\ I^+ : (P \times T) \to \mathbb{N}$ where $\mathbb{N}$ is the set of non-negative integers, are called *'negative'* and *'positive incidence functions'* respectively.

5. $\forall p \in P,\ \exists\ t \in T$ such that $I^-(p,t) \neq 0\ or\ I^+(p,t) \neq 0$, and
   $\forall t \in T,\ \exists\ p \in P$ such that $I^-(p,t) \neq 0\ or\ I^+(p,t) \neq 0$

6. $\mu_0 : P \to \mathbb{N}$ is the *'initial marking'* which gives the initial distribution of tokens in places.

An arbitrary distribution of tokens in the places is called a *'marking'* given by

$$\mu : P \to \mathbb{N}$$

A marking can hence be represented as a $(1 \times n)$ vector, where $n = |P|$ such that its i^{th} component gives the value $\mu(p_i)$, the number of token in the place p_i.
$I^-(p,t)$ represents the number of arcs from place p to transition t and $I^+(p,t)$ represents the number of arcs from transition t to place p. The number of arcs from a place p (or transition t) to a transition t (or place

p) can also be considered as the *'weight'* of the arc (p,t) (or (t,p)).

Incidence Matrix

Let $N = (P,T,I^-,I^+,\mu_0)$ be a Petri net with $|P| = n$ and $|T| = m$, then the *'incidence matrix'*, I is $m \times n$ matrix of integers and its entries are given by $(a_{ij}^+ - a_{ij}^-)$ where, $a_{ij}^+ = I^+(p_j,t_i)$ is the number of arcs from transition t_i to place p_j, and the corresponding matrix is known as *'Positive incidence matrix'* and $a_{ij}^- = I^-(p_j,t_i)$ is the number of arcs from place p_j to transition t_i, and the corresponding matrix is known as *'Negative incidence matrix'*.

Pre-Set and Post-Set of Place/Transition

The set of all input places of t , i.e., ${}^\bullet t = \{p \in P \mid I^-(p,t) > 0\}$ is called the *'pre-set'* of a transition t. The set of all input transitions of p , i.e., ${}^\bullet p = \{t \in T \mid I^+(p,t) > 0\}$ is called the *'pre-set'* of a place p. The set of all output places of t , i.e., $t^\bullet = \{p \in P \mid I^+(p,t) > 0\}$ is called the *'post-set'* of a transition t. The set of all output transitions of p , i.e., $p^\bullet = \{t \in T \mid I^-(p,t) > 0\}$ is called the *'post-set'* of a place p. A *'source transition'* (place) is a transition (place) without any input place (transition), i.e., ${}^\bullet t = \emptyset$ $({}^\bullet p = \emptyset)$. A *'sink transition'* (place) is a transition (place) without any output place (transition), i.e., $t^\bullet = \emptyset$ $(p^\bullet = \emptyset)$.

A Petri net without its initial marking is called a *'Petri net structure'* and a Petri net with all arcs of weight one is called an *'Ordinary Petri net'*. A pair of place p and transition t is called a *'self-loop'* if p is both an input and an output place of t. A Petri net without any self-loop is called a *'Pure Petri net'*.

A Petri net $N' = (P',T',I_1^-,I_1^+,\mu_0')$ is called a *sub-Petri net* of a Petri net $N = (P,T,I^-,I^+,\mu_0)$ if

1. $P' \subseteq P$

2. $T' \subseteq T$

3. $I_1^+(p,t) \neq 0$ if $I^+(p,t) \neq 0$ for $(p,t) \in (P' \times T')$

4. $I_1^-(p,t) \neq 0$ if $I^-(p,t) \neq 0$ for $(p,t) \in (P' \times T')$

5. $\mu_0'(p) = \mu_0(p)\ \forall\ p \in P'$

1.1.2 Graphical Representation of Petri nets

Petri nets can be represented graphically by representing its transitions by a rectangle, the places using circles, the tokens in places by filled circles and the directed arcs with the help of arrows. Elements of graphical representation of a Petri net is shown in **Figure 1.2**.

The set of vertices, $V = P \cup T$ in a Petri net is divided into two disjoint

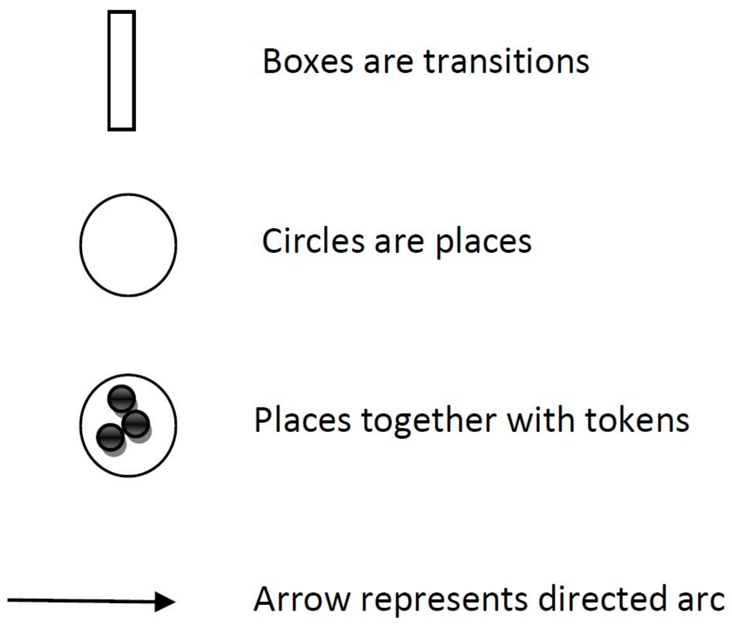


Figure 1.2: Elements of Graphical Representation of a Petri net

subsets, P and T, such that an arc joins vertices of one subset to another and not the vertices of the same subset. Thus, structurally a Petri net is a **bipartite digraph**.

An example of a Petri net is given in **Figure 1.3** and its various components are discussed.

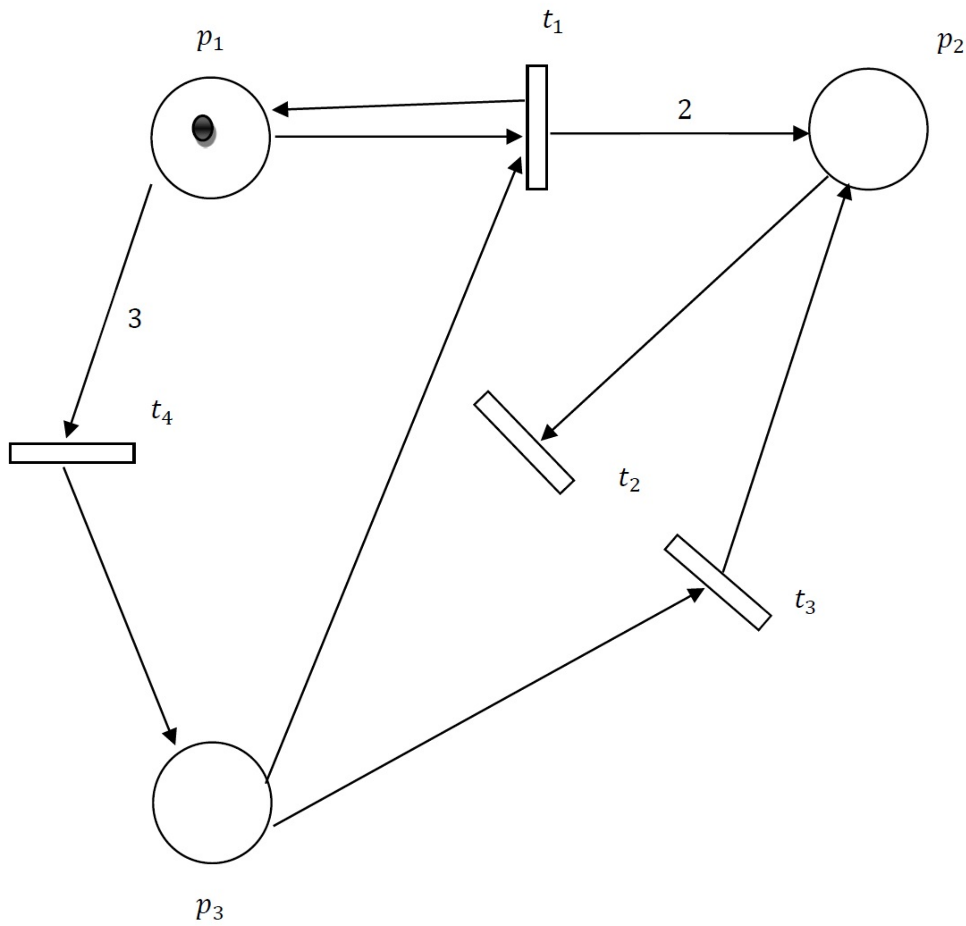


Figure 1.3: An example of a Petri net

In the example, the set $P = \{p_1, p_2, p_3\}$ and $T = \{t_1, t_2, t_3, t_4\}$.

$I^-(p_1, t_1) = 1$

$I^-(p_1, t_4) = 3$

$I^-(p_3, t_1) = 1$

$I^-(p_2, t_2) = 1$

$I^-(p_3, t_3) = 1$

$I^+(p_1, t_1) = 1$

$I^+(p_2,t_1) = 2$

$I^+(p_2,t_3) = 1$

$I^+(p_3,t_4) = 1$

Also, the pre-set and post-set of transitions and places are as below:

${}^\bullet t_1 = \{p_1, p_3\}$	${}^\bullet t_2 = \{p_2\}$	${}^\bullet t_3 = \{p_3\}$	${}^\bullet t_4 = \{p_1\}$
$t_1^\bullet = \{p_1, p_2\}$	$t_2^\bullet = \emptyset$	$t_3^\bullet = \{p_2\}$	$t_4^\bullet = \{p_3\}$
${}^\bullet p_1 = \{t_1\}$	${}^\bullet p_2 = \{t_1, t_3\}$	${}^\bullet p_3 = \{t_4\}$	
$p_1^\bullet = \{t_1, t_4\}$	$p_2^\bullet = \{t_2\}$	$p_3^\bullet = \{t_1, t_3\}$	

The initial marking of the Petri net is $(1,0,0)$.

1.1.3 Execution Rules for Petri nets

Once the conditions (represented by places) are met, firing of a transition can be used to represent the occurrence of an event. To check whether the conditions are fulfilled or not, an enabling rule is used and if the conditions are met, firing of transition takes place using the rules mentioned below.

Enabling

A transition t in a Petri net N is *enabled* at a marking μ if

$$I^-(p,t) \leq \mu(p) \; \forall p \in P$$

Firing

An enabled transition t may *fire* at μ to yield a new marking μ_1 given by the rule:

$$\mu_1(p) = \mu(p) - I^-(p,t) + I^+(p,t) \; \forall p \in P$$

and then, μ_1 is said to be directly reachable from μ and we write $\mu \overset{t}{\rightarrow} \mu_1$

Firing sequence and Reachability

If there exists a firing sequence $\eta = t_1, t_2, \ldots, t_k$ that transforms μ to μ' and is written $\mu \overset{\eta}{\rightarrow} \mu'$, then, a marking μ' is said to be reachable from μ.

A *firing or occurence sequence* is a sequence of transitions $\eta = t_1 t_2 \ldots t_k$ such that

$$\mu \overset{t_1}{\rightarrow} \mu_1 \overset{t_2}{\rightarrow} \mu_2 \overset{t_3}{\rightarrow} \mu_3 \ldots \overset{t_k}{\rightarrow} \mu'$$

Note that a transition $t_j, 1 \leq j \leq k$ can occur more than once in the firing sequence η.

The set of all possible firing sequences from μ_0 of a Petri net N with initial marking μ_0 is denoted by $L(N, \mu_0)$.

Consider a real life example of change of seasons to see how the execution rules of a Petri net work [30].

There are four seasons: Spring, Summer, Autumn and Winter represented by places of a Petri net. The transitions are used to represent the change from one season to another as shown in **Figure 1.4**. Assuming that at present the spring season is going on, a token is put in place labeled 'Spring' to represent this situation (see **Figure 1.5**). By the enabling condition as mentioned above, it can be seen that the transition labeled 'start of Summer' is enabled and can fire. After firing, the place labeled 'Summer' will receive one token and all other places will have no tokens (see **Figure 1.6**) which shows the transition of season from Spring to Summer.

By using enabling and firing rules, transition from one season to another takes place. In this manner, the scenario of change of seasons can be modeled using a Petri net dynamically, making it easier to visualize the changes and also comment on the present state of the model.

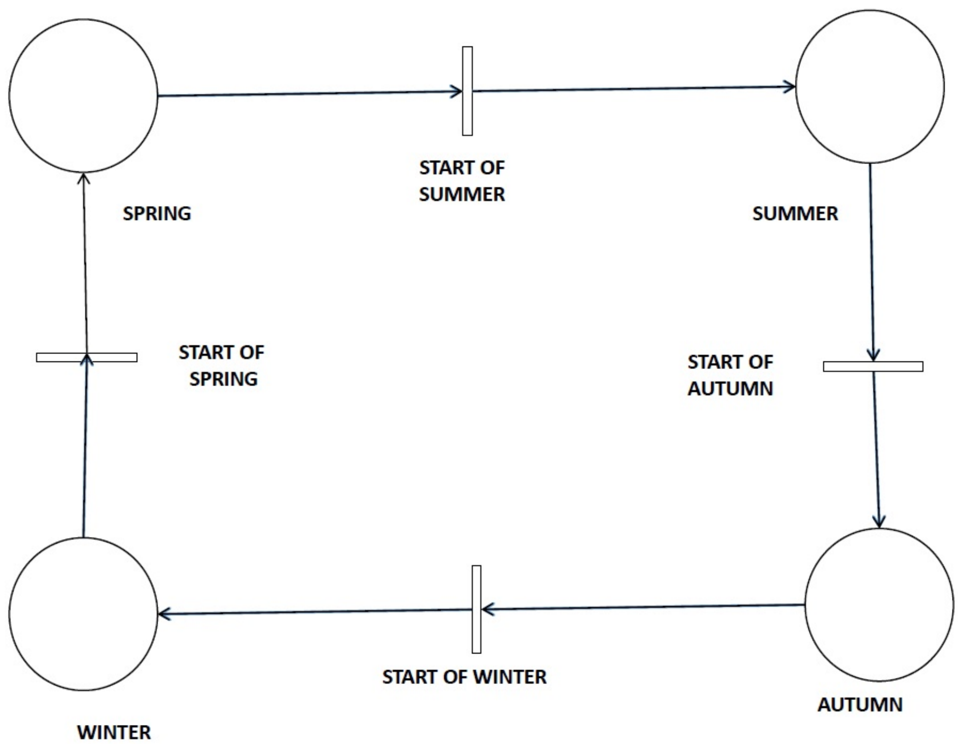


Figure 1.4: Change of Seasons

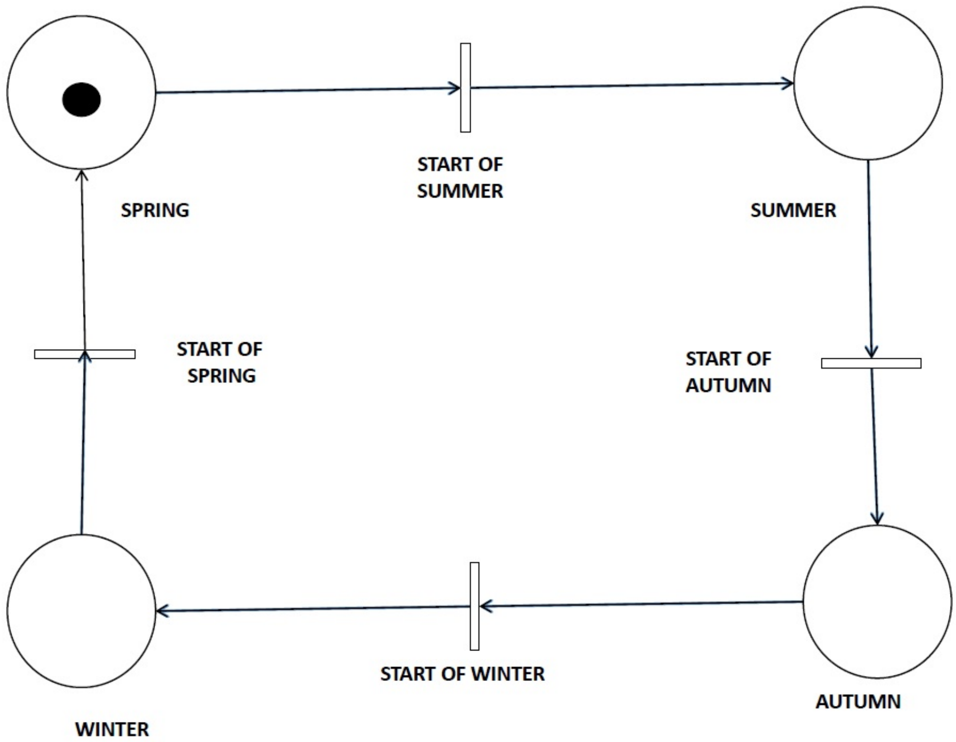


Figure 1.5: Spring Season

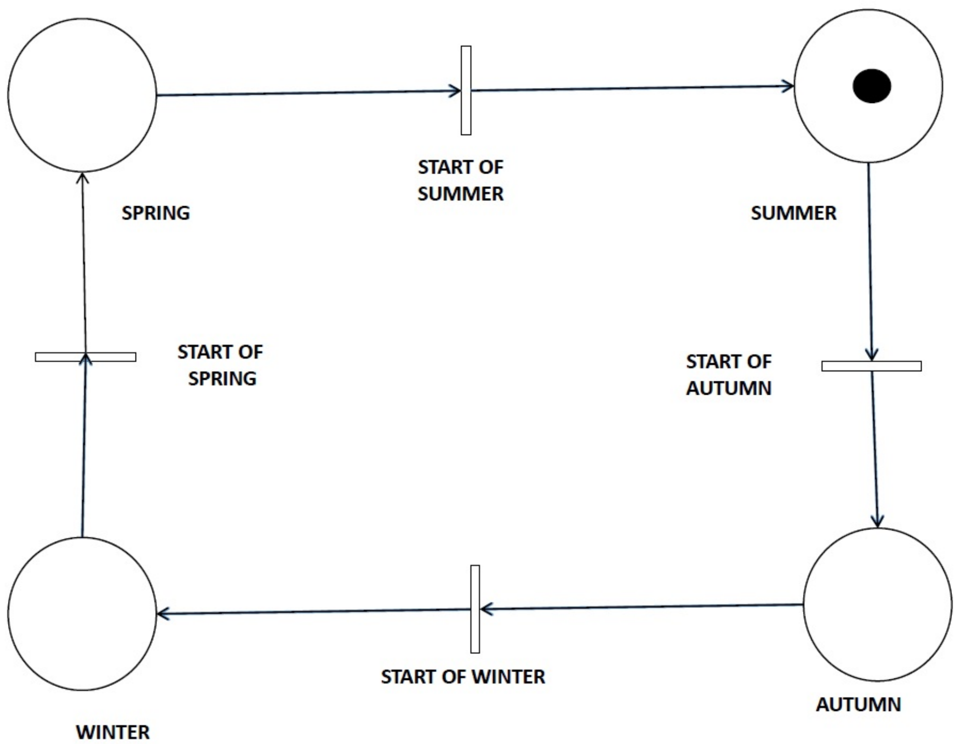


Figure 1.6: Summer Season

Reachability Set and Reachability Tree

The *'Reachability Set '*, $R(N,\mu)$ of a Petri net N is the set of all markings of N reachable from μ.

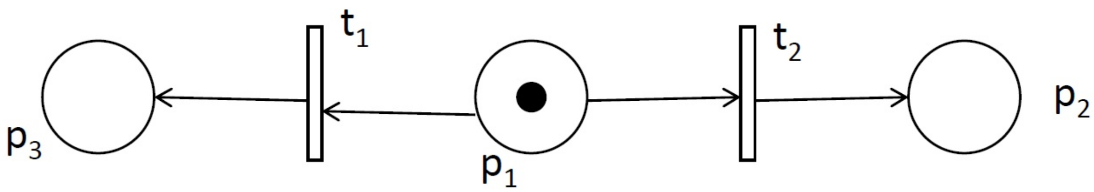


Figure 1.7: A Petri net with initial marking $\mu_0 = (1,0,0)$

An example of the Petri net is given in **Figure 1.7** with initial marking $(1,0,0)$. In this Petri net, both the transitions t_1 and t_2 are enabled. On firing of transitions t_1 and t_2, the markings $(0,0,1)$ and $(0,1,0)$ are obtained respectively. After that no transition is enabled. Hence, the reachability set of the given Petri net is

$$R(N,(1,0,0)) = \{(1,0,0),(0,0,1),(0,1,0)\}$$

A *'Reachability Tree'* represents the reachability set of a given Petri net. The reachability tree of Petri net in **Figure 1.8** is given in the **Figure 1.9**. The nodes of the tree are the markings reachable from the initial marking (root node of the tree). An arrow is labeled by a transition to show which transition is fired to move to the next marking from the present marking.

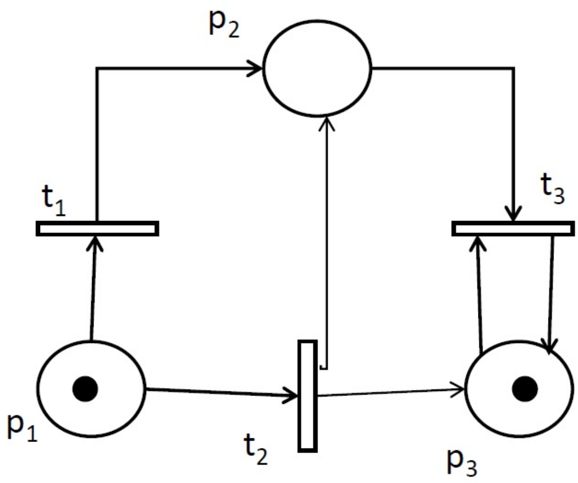


Figure 1.8: Petri net with initial marking $\mu_0 = (1,0,1)$

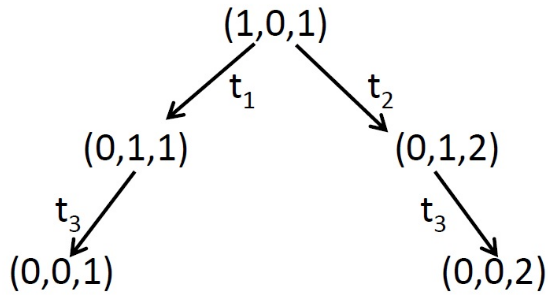


Figure 1.9: Reachability Tree corresponding to Petri net in **Figure 1.8**

1.1.4 Behavioral Properties of Petri nets

A modeled system is useless unless it is analyzed for various problems and properties associated with the system. Two types of properties can be studied for Petri nets : Marking dependent (which is dependent on the initial marking) and marking independent (independent of the initial marking). Here, only basic behavioral properties and their analysis are considered.

1. **Boundedness**

   A place p_i in a Petri net N is said to be *'k-bounded'* if, for all $\mu \in R(N, \mu_0)$,

   $$\mu(p_i) \leq k$$

   A Petri net is *'k-bounded'* if all of its places are k-bounded. A Petri net which is 1-bounded is called a *'Safe Petri net'*.
   In various applications, the tokens in places of a Petri net are used to represent resources, buffers or registers, etc. On verifying the boundedness of the Petri net, it can be made sure that no excess resources are used, the buffer or registers are not overflowed.

2. **Strict Conservation**

   In a Petri net N, if for all $\mu \in R(N, \mu_0)$,

   $$\sum_i \mu_0(p_i) = \sum_i \mu(p_i)$$

   then, N is *'strictly conserved'*.

   Conservation with respect to a weighting vector

   In a Petri net N, a weighting vector w is a $n \times 1$ vector ($n = |P|$) which gives the weight associated with tokens in a place. If for all $\mu \in R(N, \mu_0)$,

   $$\sum_i \mu_0(p_i).w(p_i) = \sum_i \mu(p_i).w(p_i)$$

$$\text{or} \quad \mu_0.w = \mu.w$$

then, N is *'conserved with respect to w'*.

Remark 1.1. *A Petri net N which is conservative with respect to a weighting vector, $w = (1,1,\ldots,1,1)$ is strictly conservative.*

To model resource allocation systems, resources should neither be created nor destroyed. Hence, the tokens used to represent them should be conserved.

3. **Liveness**

The absence of deadlocks in operating systems and the notion of liveness are closely related. A Petri net N with initial marking μ_0 is said to be live (or equivalently μ_0, is said to be a live marking for N) if, no matter what marking has been reached from μ_0, it is possible to ultimately fire any transition of the net by progressing through some further firing sequence. Thus a live Petri net guarantees deadlock-free operation, no matter what firing sequence is chosen. Liveness is an ideal property for many systems. But verification of this property becomes costly in some systems such as the operating system of a large computer, and hence, this property is relaxed by defining different levels of liveness [10, 20]. A transition t in a Petri net N with initial marking μ_0 is said to be:

(a) **L0-Live or Dead** if t can never be fired in any firing sequence in $L(N,\mu_0)$.

(b) **L1-Live (potentially firable)** if t can be fired at least once in some firing sequence in $L(N, \mu_0)$.

(c) **L2-Live** if, given any positive integer k, t can be fired at least k times in some firing sequence in $L(N, \mu_0)$.

(d) **L3-Live** if t appears infinitely, often in some firing sequence in $L(N, \mu_0)$.

(e) **L4-live or Live** if t is L1-live for every marking μ in $R(N, \mu_0)$.

A Petri net N with initial marking μ_0 is said to be $Lk-live$ if every transition in the net is $Lk-live$, $k = 0, 1, 2, 3, 4$.
It is easy to see the following implications:
$L4-liveness \implies L3-liveness \implies L2-liveness \implies L1-liveness.$
It is said that that a transition is strictly $Lk-live$ if it is $Lk-live$ but not $L(k+1)live$, $k = 1, 2, 3$.

1.1.5 Analysis Techniques

The analysis techniques provide solution mechanisms for analyzing the problems mentioned in the previous section. There are two main techniques for analyzing the modeled system.

1. Reachability Tree

2. Matrix Equations

Reachability Tree Approach

The behavioral properties of Petri nets can be analyzed using its reachability tree as follows:

- Boundedness: By observing values at components of the markings in the reachability tree, one can find a bound on places of the Petri net and hence the Petri net.

- Conservation: By observing the sum of components of the markings in the reachability tree, it can be checked whether net is strictly conserved or not.

- Liveness: A liveness of a Petri net ($Lk-$live, $k = 0, 1, 2, 3, 4$) can be analysed by checking the number of times a transition occurs in the reachability tree as the arc labels.

Consider the Petri net given in **Figure 1.8** and its corresponding reachability tree in **Figure 1.9**, the following observations are made:

1. Boundedness: The Petri net is bounded.

2. Conservation: The Petri net is not conserved.

3. Liveness: The Petri net is $L1-live$ (Since, transitions t_1, t_2, t_3 are $L1-live$, $L1-live$ and $L4-live$ respectively).

Matrix Equations Approach

The flow relation in Petri nets is represented by using an $m \times n$ incidence matrix I, where $m = |T|$ and $n = |P|$.

$$I = [a_{ij}^{+}] - [a_{ij}^{-}] = I^{+} - I^{-}$$

where, a_{ij}^{+} = Number of arcs from transition t_i to place p_j

a_{ij}^{-} = Number of arcs from place p_j to transition t_i

Execution Rules for Petri nets in Matrix Form

The execution rules, i.e., enabling and firing conditions mentioned in section 1.1.3 can also be represented using the matrix form.

A transition t_i is *enabled* at a marking μ if

$$\mu(p_j) \geq a_{ij}^- \quad \forall \, j$$

$$\mu \geq e[i].I^-$$

where $I^- = [a_{ij}^-]$ and $e[i] = (0,0,\ldots,0,1,0,\ldots,0,0)$ is a $(1 \times m)$ vector with 1 at the i^{th} position.

Firing an enabled transition t_i at marking μ yields a new marking μ_1 given by:

$$\mu_1 = \mu - e[i].I^- + e[i].I^+$$

$$\mu_1 = \mu + e[i].I$$

Firing a sequence of transitions, $\eta = t_{j_1}t_{j_2}\ldots t_{j_k}$ results in

$$\mu_1 = \mu + (e[j_1] + e[j_2] + \ldots + e[j_k]).I$$
$$= \mu + f(\eta).I$$

The vector $f(\eta)$ is called the *firing vector* of the sequence η whose i^{th} element $f(\eta)_i$ is the number of times the transition t_i fires in the sequence η.

By considering the conservation problem, it can be seen how the matrix approach for analysis is useful.

Theorem 1.1. *[27] A Petri net N is conservative if and only if there exists a positive vector w such that $I.w = 0$.*

The above theorem follows from the firing rules mentioned in section 1.1.3 and the definition of conservation of Petri net w.r.t. weighing vector given in section 1.1.4.

1.1.6 Modeling of Systems using Petri nets

In modeling a system using Petri nets, the conditions associated with the system are represented using places and the events that occur in the system are represented using transitions. The present state of the system is represented via a token in the places.

Here, two systems are taken and modeled using Petri nets.

Chemical Reactions

A system in which three chemical reactions take place is considered. The product of the previous reaction is used as a reactant in the next reaction. Thus, the given example is actually a sequential execution in Petri nets.

$$C + O_2 \rightarrow CO_2 \tag{1.1.1}$$

$$CO_2 + NaOH \rightarrow NaHCO_3 \tag{1.1.2}$$

$$NaHCO_3 + HCl \rightarrow H_2O + NaCl + CO_2 \tag{1.1.3}$$

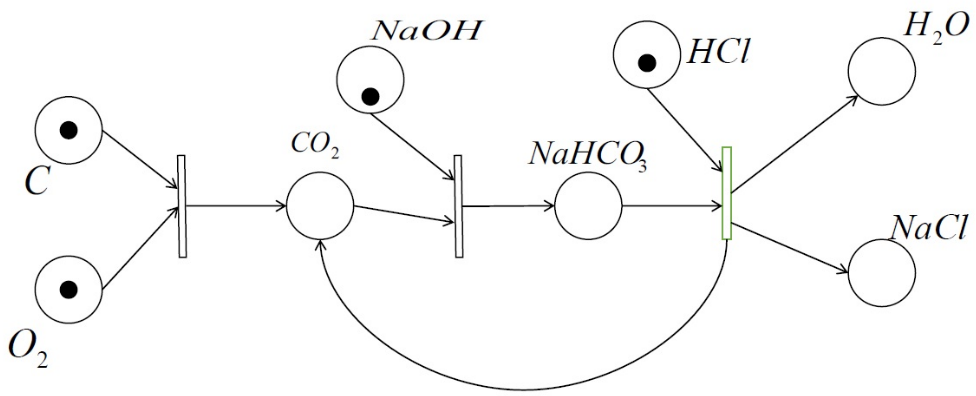


Figure 1.10: Chemical reaction with token in the places showing the reactants available initially

In the first reaction (1.1.1), carbon and oxygen combine to give carbon dioxide, which in turn is used in the second reaction (1.1.2) to give the product sodium bicarbonate after reaction with sodium hydroxide. Finally, the product from second reaction is combined with hydrochloric acid to give the final products. This system of reactions is modeled using a Petri net shown in **Figures 1.10** to **1.13**.

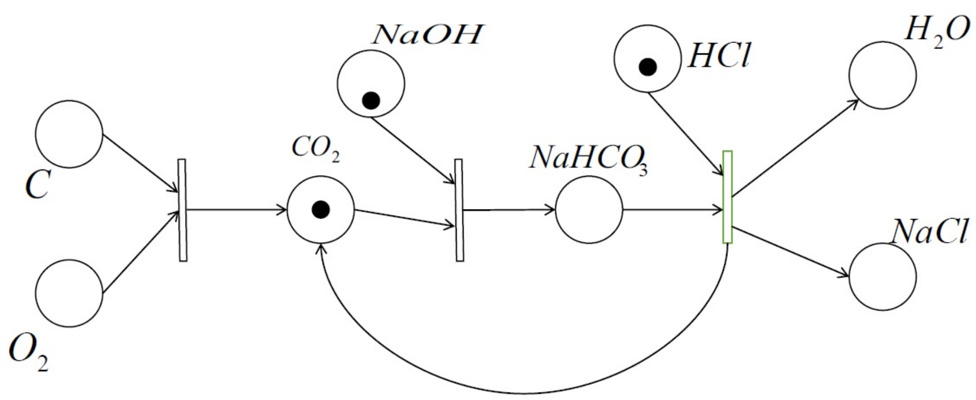


Figure 1.11: First chemical reaction takes place, represented by equation (1.1.1)

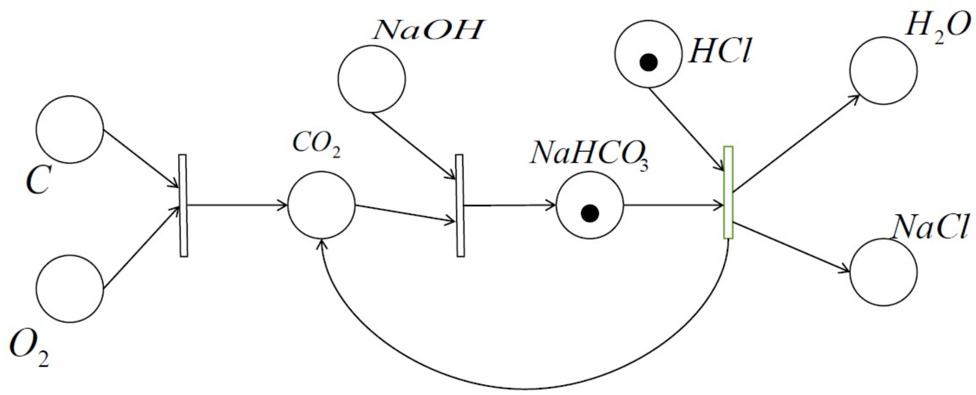


Figure 1.12: Second chemical reaction takes place, represented by equation (1.1.2)

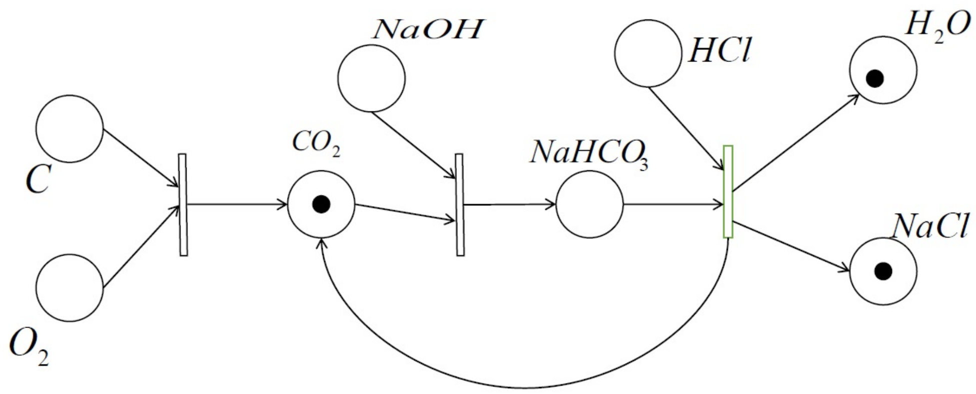


Figure 1.13: Third chemical reaction takes place, represented by equation (1.1.3)

Puzzle

Next, a well known puzzle is modeled using Petri nets. In the puzzle, there is a man who is traveling with a wolf, a goat, and a cabbage. They come across a river which need to be crossed. But the available boat can carry the man and at most one other object. Also, if the man is not around, the wolf may eat the goat and the goat may eat the cabbage.

The given puzzle is modeled via a Petri net and solved. The notations used to represent various objects and events in the puzzle

Table 1.1: Notations used in Application of Puzzle

Notation	Object/Event
M	Man
W	Wolf
G	Goat
C	Cabbage
L	Left
R	Right
MGLR	Man and goat from left to right
MCRL	Man and cabbage from right to left
MGRL	Man and goat from right to left
MWLR	Man and wolf from left to right
MRL	Man from right to left

are given in **Table 1.1**. The solution to the puzzle is shown with the help of Petri net in the **Figures 1.14** to **1.28**.

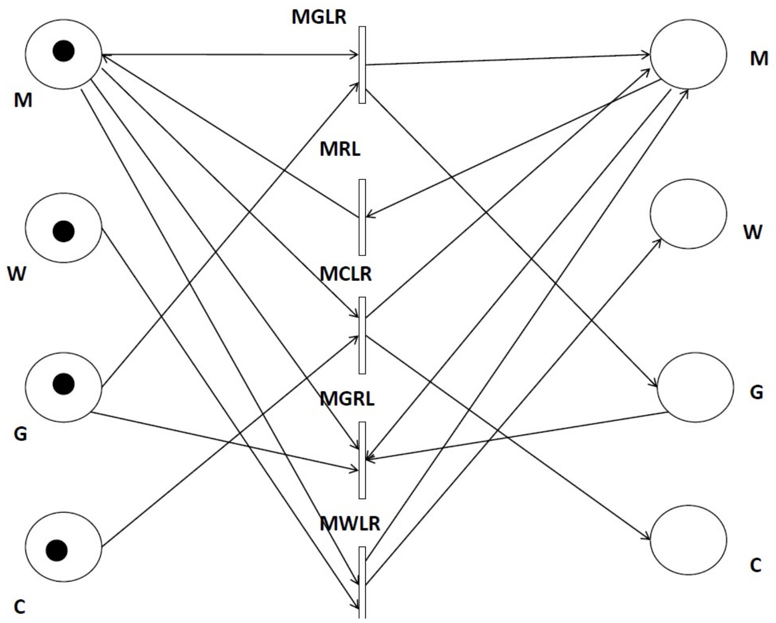


Figure 1.14: Petri net with tokens in places M,W,G and C showing that man, wolf, goat, and cabbage are present on left side of the river, which need to be crossed

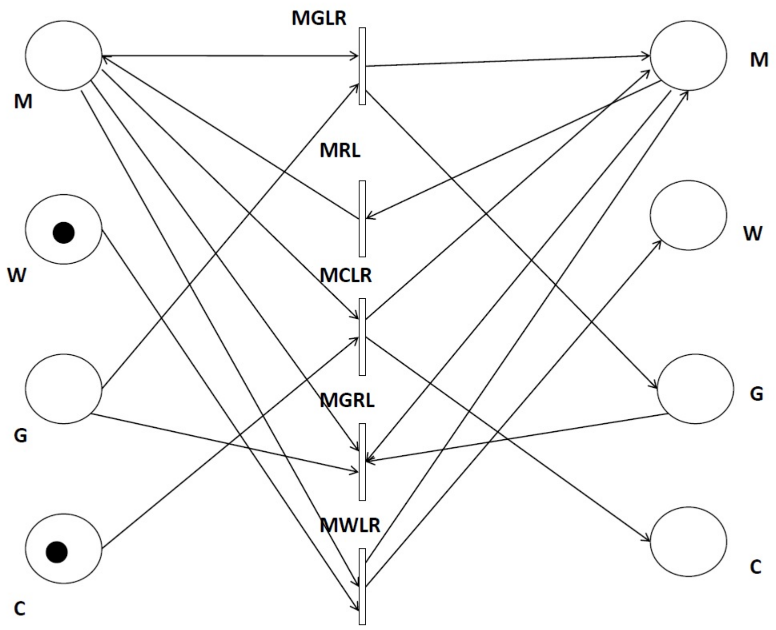


Figure 1.15: Petri net showing that Man and goat are crossing the river on firing of transition MGLR

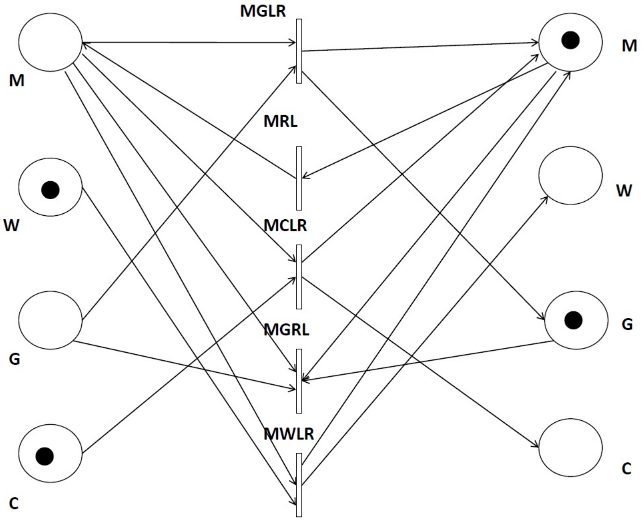


Figure 1.16: Petri net showing that Man and goat crosses and reaches right side of the river with token in the corresponding places on right hand side

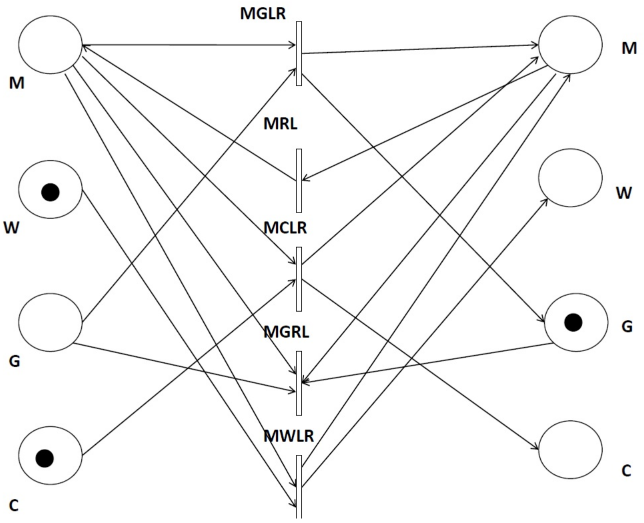


Figure 1.17: Petri net showing that Man is crossing the river after firing of transition MRL

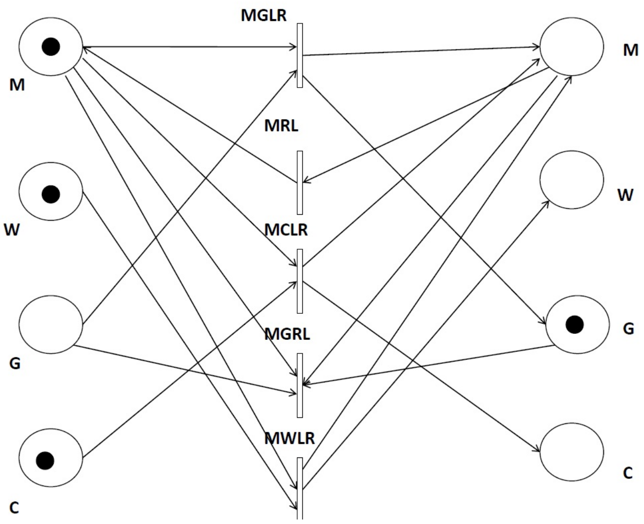


Figure 1.18: Petri net showing that Man reaches the left side of the river

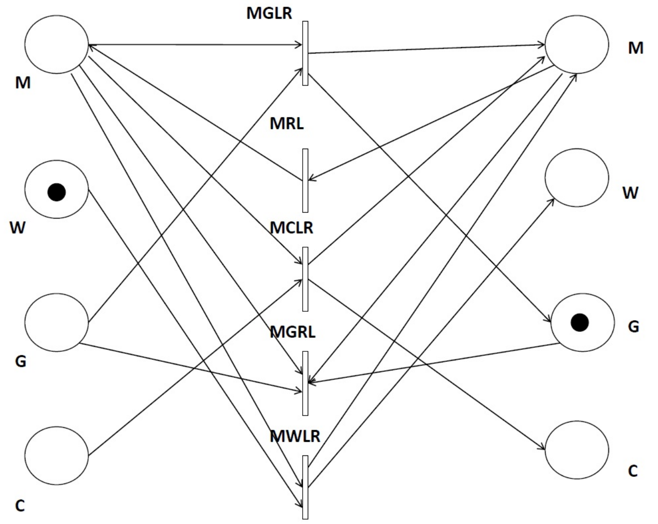


Figure 1.19: Petri net showing that Man takes cabbage and is crossing the river on firing of transition MCLR

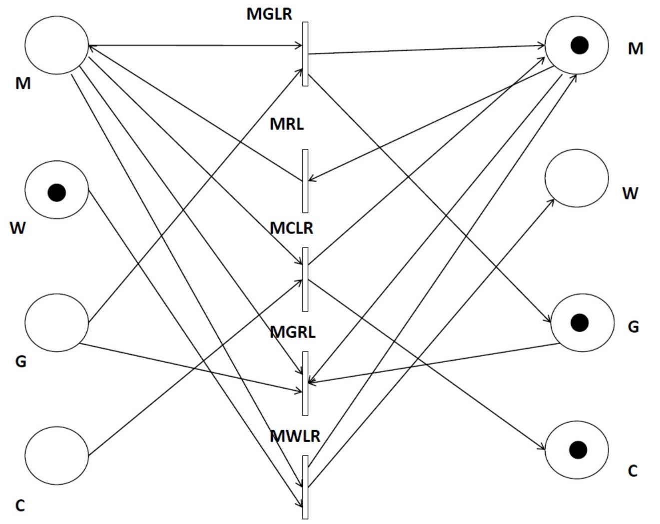


Figure 1.20: Petri net showing that Man and cabbage reach right side of the river with token in the corresponding places on right hand side

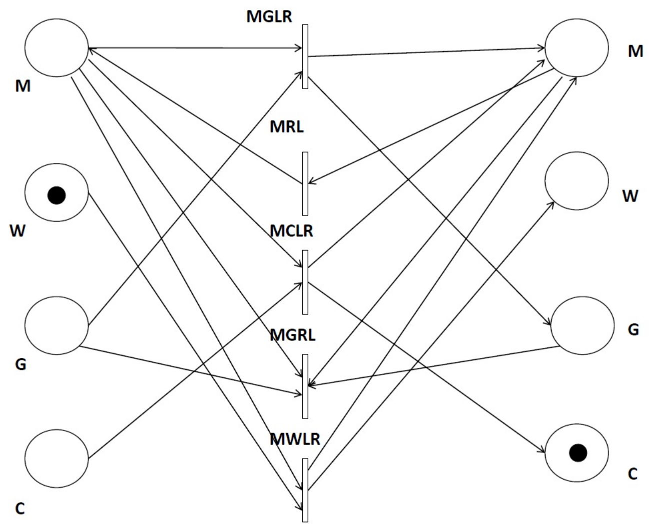


Figure 1.21: Petri net showing that Man and goat are crossing the river (from right to left) after firing of transition MGRL

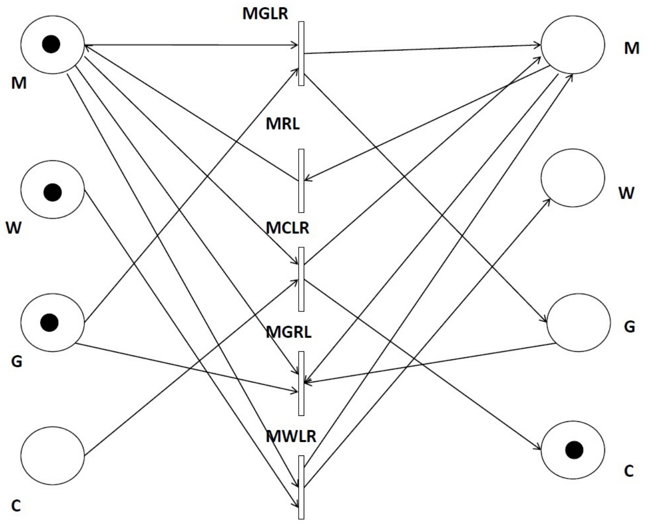


Figure 1.22: Petri net showing that Man and goat reaches the left side of the river

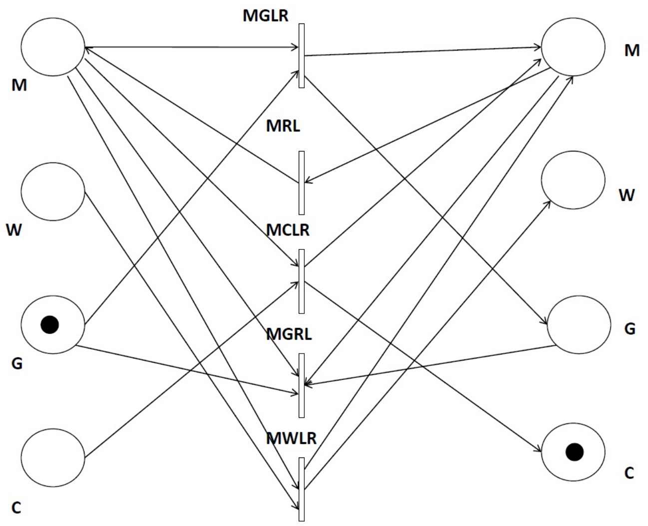


Figure 1.23: Petri net showing that Man and wolf are crossing the river after firing of transition MWLR

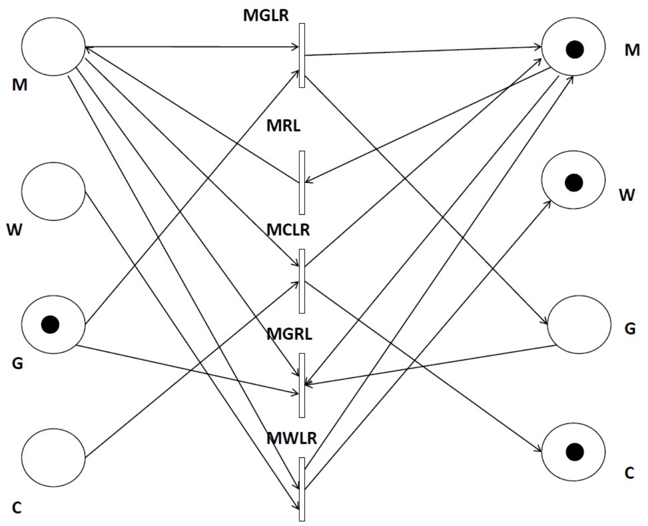


Figure 1.24: Petri net showing that Man and wolf reach right side of the river with token in the corresponding places on right hand side

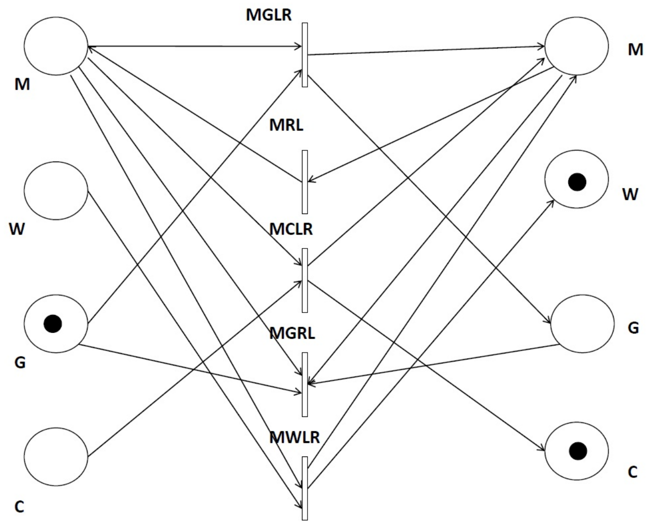


Figure 1.25: Petri net showing that Man is crossing the river after firing of transition MRL

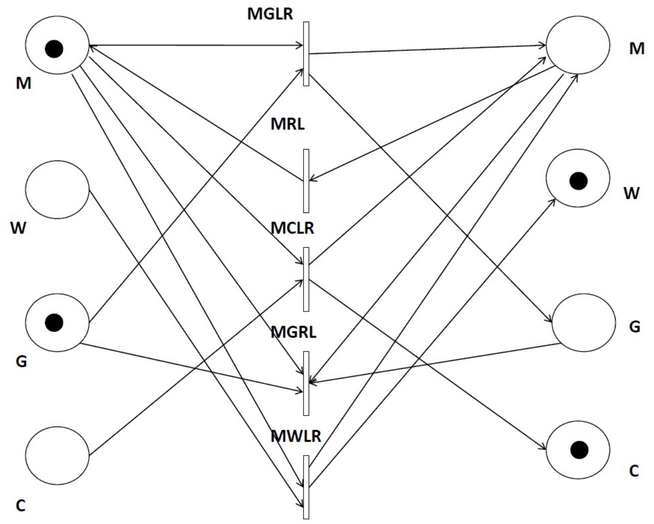


Figure 1.26: Petri net showing that Man reaches the left side of the river

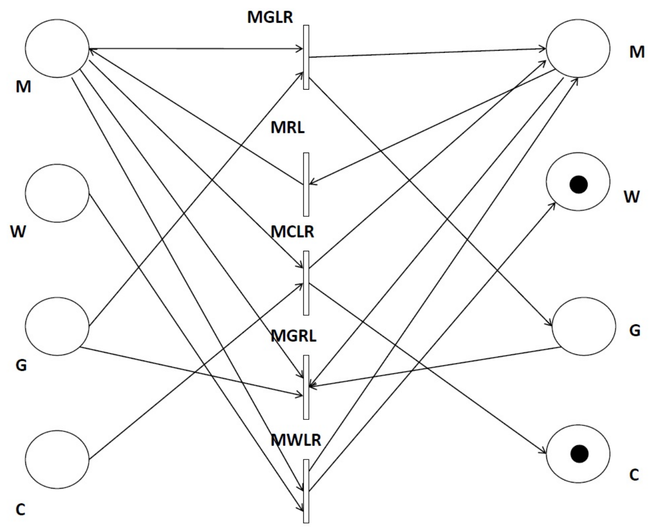


Figure 1.27: Petri net showing that Man and goat are crossing the river after firing of transition MGRL

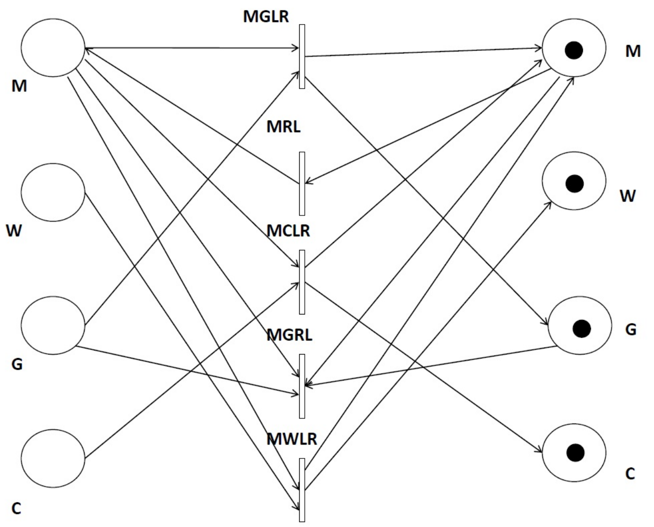


Figure 1.28: Petri net showing that Man and goat reach right side of the river with token in the corresponding places on right hand side

Thus, finally man, goat, wolf and cabbage safely reach the right side of the river.

1.2 Signed Graph

In the field of graph theory, Harary [15] introduced a signed graph and gave the characterization for a balanced signed graph. A signed graph depicts situations or structures in which both, a relation and its opposite may occur, e.g., ‘like’ and ‘dislike’ in social relations, a simple reply of either ‘Yes ’or ‘No’ in communication, etc. A signed graph has been used extensively to represent the interpersonal relationship among the group of individuals, study international relations between nations and to study the stability of traffic control problem at an intersection. These graphs were developed to model social interactions involving disliking, indifference, and liking. This concept of signed graph has been used for predicting positive and negative links in online social networks [21] and for determining the stability of the social model using the degree of balanceness [8, 38]. Intuitively, in a weighted graph, an edge with a positive and negative weight can be used to denote similarity or proximity of its endpoints and dissimilarity or distance respectively.

Formal Definition of Signed Graph [15]

A *Signed graph (Sigraph)* is an ordered pair $S = (G, \sigma)$ where $G = (V, E)$ is called the underlying graph of S and $\sigma : E \to \{+, -\}$ is a function called *signature (or sign)* of S, which assigns either a positive or negative sign to each edge.
Further,

$$E^+(S) = \{e \in E(S) : \sigma(e) = +\}$$

$$E^-(S) = \{e \in E(S) : \sigma(e) = -\}$$

The sets $E^+(S)$ and $E^-(S)$ are called the set of *'positive '* and *'negative edges'* of S respectively. An example of a signed graph is given in **Figure 1.29**, where solid lines represent the edges which are assigned positive sign while dotted lines represent the edges that are assigned negative sign. A signed graph in which all the edges are positive is called an *'all-positive signed graph'(all-negative signed graph* is defined similarly).
A signed graph is said to be *'homogeneous'* if it is either all-positive or all-negative and *'heterogeneous'* otherwise. Thus, a graph can be viewed as a signed graph where each edge is positive in sign. A maximal connected subgraph of S consisting of only positive (negative) edges of S is called a positive (or negative) section in a signed graph S. This natural generalization of graphs occurs while modeling of

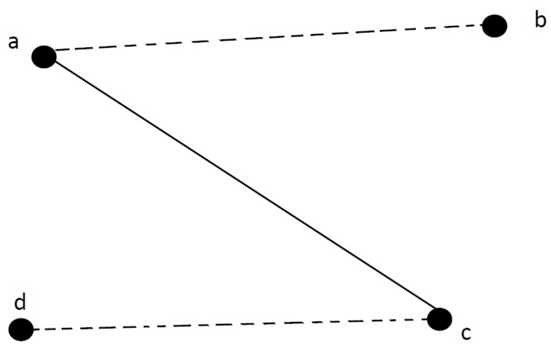


Figure 1.29: A Signed Graph

cognitive interpersonal relationship in a social group, where vertices represent the individuals in the group and edges are used to represent the interpersonal relationship between any two individuals, say A and B. A negative edge AB is interpreted as a negative relationship (such as enmity, dislike etc.) while a positive edge AB is interpreted as a positive relationship (such as friendly, like, etc.) between A and B.

The positive (negative) degree of a vertex is the number of positive (negative) edges incident on the vertex. The degree of a vertex, $v \in V(S)$, is denoted by $d(v)$ and is defined as: $d(v) = d^+(v) + d^-(v)$, where $d^+(v)(d^-(v))$ denotes the positive (negative) degree of vertex v. A vertex v of odd (even) degree is called odd (even) vertex. The edge degree of an edge vw, denoted by $d_e(vw)$ is the total number of edges adjacent to vw. Clearly, $d_e(vw) = d(v) + d(w) - 2$.

A cycle in a signed graph is said to be positive (negative) if the product of the signs of its edges is positive (negative), that is, it contains and even (odd) number of negative edges. Harary [15] introduced the concept of balanceness and derived the structural criteria called partition criterion for balance in signed graphs.

Balanced signed graph

A signed graph is said to be balanced (or stable), if all the cycles of the signed graph are positive, where the sign of the cycle is defined as the product of sign of edges lying on the cycle.

A signed graph is unbalanced if there exists at least one cycle which is negative.

Theorem 1.2. *[15] A signed graph S is balanced iff its vertex set $V(S)$ can be partitioned into two disjoint subsets V_1 and V_2; one of which may be empty such that all edges between vertices of the same subset are positive and all edges between vertices of different subsets are negative.*

A signed graph is said to be partitionable or clusterable if it is possible to partition vertices of a signed graph, so that every edge that connects vertices belonging to the same cluster is positive and every edge that connects two vertices belonging to different clusters is negative. The signed graphs where vertices can be partitioned into two clusters are called balanced.

Theorem 1.3. *[15] A signed graph S is balanced iff for each pair of distinct vertices u and v in S, all the paths joining u and v have the same sign.*

Marking of signed graph

A marking of a signed graph S is a function, $\mu : V(S) \to \{+,-\}$.
The idea of marking the vertices with signs derived from the edge sign was introduced by Sampathkumar [31]. The marking denoted by

$\mu_\sigma : V(S) \to \{+,-\}$ is given as

$$\mu_\sigma(v) = \prod_{vw \in E(S)} \sigma(vw)$$

, where $\sigma(vw)$ is the sign of the edge vw. This marking is called *'canonical marking '*.

The following characterization of balanced signed graphs given by Sampathkumar is well known.

Theorem 1.4. *[29] A signed graph S is balanced iff there exists a marking μ of S such that every edge uv of S satisfies $\sigma(uv) = \mu(u)\mu(v)$.*

Domination in signed graph

The concept of domination can be traced back to the chess problem of finding the minimum number of queens required such that all the squares are either occupied or can be attacked by a queen [18]. The applications of theory of domination includes communication network problems, facility location problem, routings, etc. [14, 33]. The domination in graphs and signed graphs have been well studied by various authors in different forms viz. roman domination, double domination, total domination, signed domination, signed total domination etc. [3, 5, 7, 16, 25, 32, 39].

Dominating Set

Let $S = (G,\sigma)$ be a signed graph. A subset $D \subseteq V$of vertices of S is a *dominating set* of S, if there exists a marking $\mu : V \to \{+1,-1\}$ of S such that every vertex u of S is either in D or whenever $u \notin D$, $N(u) \cap D \neq \emptyset$

and $\sigma(uv) = \mu(u)\mu(v)$ for every $v \in N(u) \cap D$, where $N(u)$ is the set of vertices in the neighborhood of the vertex u.

1.3 Motivation

Various developments in the field of Graph Theory and Petri net Theory in the form of balanceness and negative tokens respectively motivated us to bridge the gap between Petri nets and signed graphs and introduce a new concept of ***Signed Petri net***. A signed Petri net is a bipartite signed digraph with dynamics associated with it in the form of tokens in places. There are various concepts in signed graph that help in modeling of social interactions and checking the stability of the system so modeled. But due to non dynamic nature of signed graph, the system needs to be modeled using multiple signed graphs. Hence, it seemed interesting to introduce an extension of Petri nets called signed Petri net which can represent all possible interactions between a group of individuals via single signed Petri net as opposed to multiple signed graphs. The signed Petri nets can be used to represent various configurations of a signed graph by a change in the marking of signed Petri net due to firing of transitions. This shows the advantage of the proposed research over a signed graph.

The tokens existing in a place of a Petri net can not be differentiated. In order to overcome this limitation, two types of tokens (positive and negative) exist in signed Petri nets making it easier to represent the resources/processes common to a place via different types of tokens. The positive and negative arcs which exist in signed Petri nets allow to extend balanceness notion of a signed graph to a Petri net which can be used to model social networks and study them using a single signed Petri net rather than multiple signed graphs. Thus, a signed Petri net is an extension of Petri nets which adopt the characteristics of both signed graphs and Petri nets and has advantages over the both.

The behavioral properties for signed Petri nets are introduced and further used to analyze a restaurant model using signed Petri nets. Using signed Petri nets, it is shown how any changes, if required in the restaurant model can be easily incorporated. Next, some structural matrices are provided for signed Petri nets, which can be further used to define its structure. Subclasses of signed Petri nets and their characterizations are given using the structural matrices defined.

The concepts of structural and dynamical balanceness are introduced which are the main motivation for developing the concept of signed Petri net. Also, the concept of domination for signed Petri nets is provided as such concept does not exist for dynamic systems.

References

[1] Aalst van der, W., Stahl, C., and Westergaard, M. (2013). *Strategies for Modeling Complex Processes Using Colored Petri nets*. Transactions on Petri Nets and Other Models of Concurrency VII, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 7480:6–55.

[2] Aalst van der, W. M. P. (1993). *Multi-dimensional Petri nets*. Computing science notes Eindhoven:Technische Universiteit Eindhoven, 9326.

[3] Acharya, B. D. (2013). *Domination and absorbance in signed graphs and digraphs I: Foundations*. Journal of Combinatorial Mathematics and Combinatorial Computing, 84:5–20.

[4] Alla, H. and David, R. (1998). *Continuous and Hybrid Petri Nets*. Journal of Circuits,Systems and Computers, 8(1):159–188.

[5] Ashraf, P. K. and Germina, K. A. (2015). *On minimal dominating sets for signed graphs*. Advances and Applications in Discrete Mathematics, 15:101–112.

[6] Bartoletti, M., Cimoli, T., and Pinna, G. (2015). *Lending Petri Nets*. Science of Computer Programming, 112:75–101.

[7] Blidia, M., Chellali, M., Haynes, T. W., and Henning, M. (2006). *Independent and double domination in trees*. Utilitas Mathematica, 70:159–173.

[8] Cartwright, D. and Harary, F. (1956). *Structural Balance:A Generalization of Heider's Theory*. Psychological Review, 63(5):277–293.

[9] Chen, S.-M., Ke, J.-S., and Chang, J.-F. (1990). *Knowledge representation using fuzzy Petri nets*. IEEE Transactions on Knowledge and Data Engineering, 2(3):311–319.

[10] Commoner, F. G. (1972). **Deadlocks in Petri-nets**. *Massachusetts Computer Assoc., Incorporated*.

[11] Cong, X., Fanti, M. P., Mangini, A. M., and Li, Z. (2017). *Decentralized Fault Diagnosis by Petri nets and Integer Linear Programming*. IFAC-PapersOnLine, 50(1):13624–13629.

[12] Du, Y. Y. and Guo, B. (2009). *Logic Petri Nets and Equivalency*. Information Technology Journal, 8:95–100.

[13] Febbraro, A. D., Giglio, D., and Sacco, N. (2017). *On analyzing the vulnerabilities of a railway network with Petri nets*. Transportation Research Procedia, 27:553–560.

[14] Gupta, P. (2013). *Domination in Graph with Application*. Indian Journal of Research, Paripex, 2(3):115–117.

[15] Harary, F. (1953). *On the notion of balance of a signed graph*. Michigan Math. Journal, 2:143–146.

[16] Haynes, T., Hedetneimi, S. T., and Slater, P. J. (1998). **Fundamentals of domination in graphs**. *1st edition, CRC Press*.

[17] Helouet, L. and Kecir, K. (2018). *Realizability of schedules by stochastic time Petri nets with blocking semantics*. Science of Computer Programming, 157:71–102.

[18] Jaenisch, C. D. (1862). **Trait des Applications de l'Analyse Mathematique au Jeu des Echecs**. Petrograd.

[19] Jensen, K. (1987). *Coloured Petri nets*. Coloured Petri nets. In: Brauer, W., Reisig, W., Rozenberg, G. (eds) Petri nets: Central Models and Their Properties. ACPN 1986. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. , 254:248–299.

[20] Lautenbach, K. (1975). **Liveness in Petri nets**. *Selbstverl. GMD*.

[21] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). *Predicting Positive and Negative Links in Online Social Networks*. Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, WWW '10:641–650.

[22] Liu, H.-C., You, J.-X., Li, Z., and Tian, G. (2017). *Fuzzy Petri nets for knowledge representation and reasoning: A literature review*. Engineering Applications of Artificial Intelligence, 60:45–56.

[23] Marsan, M. (1990). *Stochastic Petri nets: An Elementary Introduction*. Stochastic Petri nets: An elementary introduction. In: Rozenberg, G. (eds) Advances in Petri nets 1989. APN 1988. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 424:1–29.

[24] Murata, T. (1989). *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, 77(4):541–580.

[25] Palakkutty, A., Germina, A., and Lishan, L. (2016). *Double domination in signed graphs*. Cogent Mathematics, 3(1):1186135.

[26] Patil, S. (1970). **Coordination of asynchronous events**. Ph.D. dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[27] Peterson, J. (1981). **Petri net theory and modeling of systems**. *Prentice Hall-Inc*.

[28] Petri, C. (1962). **Kommunikation mit automaten**. Schriften des Institutes fur Instrumentelle Mathematik, Bonn.

[29] Reddy, P., Sampathkumar, E., and Subramanya, M. (2010). *Common-edge signed graph of a signed graph*. Journal of the Indonesian Mathematical Society , 16:105–113.

[30] Reisig, W. (2013). *The Synthesis Problem*. Transactions on Petri Nets and Other Models of Concurrency VII, Lecture Notes in Computer Science, Springer, Heidelberg, 7480:303–313.

[31] Sampathkumar, E. (1984). *Point-signed and line-signed graphs*. National Academy Science Letters, 7(3):91–93.

[32] Sampathkumar, E. (1998). *Domination Parameters of A Graph*. Domination in Graphs: Advanced Topics, 2:271–300.

[33] Sasireka, A. and Nandhu Kishore, A. H. (2014). *Applications of Dominating Set of Graph in Computer Networks*. International Journal of Engineering Sciences and Research Technology, 10(9):2801–2810.

[34] Silva, J. and Foyo, P. M. G. D. (2012). *Timed Petri nets*. Petri nets - Manufacturing and Computer Science, InTech:359–378.

[35] Valk, R. (2004). *Object Petri nets*. Lectures on Concurrency and Petri nets: Advances in Petri nets, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3098:819–848.

[36] Wang, J. (1998). *Time Petri nets. In: Timed Petri nets*. The Kluwer International Series on Discrete Event Dynamic Systems, Springer, Boston MA, 9:63–123.

[37] Winskel, G. (1985). *Categories of models for concurrency*. Seminar on Concurrency, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 197:246–267.

[38] Zaslavsky, T. (1982). *Signed Graphs*. Discrete Applied Mathematics, 4(1):47–74.

[39] Zelinka, B. (2001). *Signed total domination number of a graph*. Czechoslovak Mathematical Journal, 51(2):225–229.

Chapter 2

Signed Petri Net

In this chapter, the notion of a signed graph and Petri net are combined to introduce a new concept of ***Signed Petri net***. The basic properties and terms associated with signed Petri nets are defined. An illustrative example is provided which utilizes the newly introduced concept of assignment of sign to places of signed Petri nets, to determine whether an online transaction initiated by a bank customer is denied or approved by the bank. Lastly, applications of signed Petri nets in message transmission system and production unit are discussed. These concepts clearly demonstrate the advantages of the proposed approach of signed Petri nets.

2.1 Introduction

Petri nets were first introduced in Carl Adam Petri's dissertation submitted in 1962 [14]. However, it is difficult to model complex processes using the classical definition of Petri nets given by him and thus, many extensions and restrictions of Petri nets have been proposed [6, 7, 8, 10, 11, 13, 15, 16]. Various applications of Petri nets have been given in the literature [1, 3, 11]. A modified Petri net model with negative tokens for automated reasoning and to represent time constraint in batch activity has been introduced in [2, 12].

In the field of graph theory, Harary [5] introduced a signed graph. This concept of signed graph has been used for predicting positive and negative links in online social networks [9]. There are various concepts in signed graph that help in modeling of social interactions and checking the stability of the system so modeled. But due to non dynamic nature of signed graph, the system needs to be modeled using multiple such graphs.

These developments in the field of Graph Theory and Petri net Theory motivated us to bridge the gap between Petri nets and signed graphs. Hence, a new extension of Petri nets called as *'signed Petri net'* has been formulated, which can represent all possible interactions between a group of individuals via a single signed Petri net as opposed to multiple signed graphs. By a change in the marking of signed Petri net due to firing of a transition or a sequence of transitions, a signed Petri net can be used to represent various configurations of a signed graph. This shows the advantage of the proposed research over a signed graph. Also, the tokens existing in a place of a Petri net can not be differentiated. In order to overcome this limitation, two types of tokens (positive and negative) exist in signed Petri nets making it easier to represent the resources/processes common to a place via different types of tokens.

Thus, a signed Petri net is an extension of Petri nets which adopt the characteristics of both signed graphs and Petri nets and has advantages over the both.

2.2 Signed Petri Net

Let $N = (P, T, I^-, I^+, \mu_0)$ be a Petri net. The arc set of the Petri net N is defined as:

$$E = \{(p,t) : I^-(p,t) > 0\} \cup \{(t,p) : I^+(p,t) > 0\}$$

A ***Signed Petri net*** [1] (SiPN) is defined as a 3-tuple $N^* = (N', \sigma, \mu_0)$, where

1. $N' = (P, T, I^-, I^+)$ is a *'Petri net structure '*.

2. $\sigma : E \to \{+,-\}$, where E is the arc set of N'. An arc is called a *'positive '* or *'negative arc '* respectively according to the sign $+$ or $-$ assigned to it using the function σ.

3. $\mu_0 = (\mu_0^+, \mu_0^-)$ is the *'initial marking '* of signed Petri net, where

   (a) $\mu_0^+ : P \to \mathbb{N}$ gives the initial distribution of positive tokens in the places, called *'positive marking '* of signed Petri net.

[1] Throughout the thesis, signed Petri net is abbreviated as SiPN.

(b) $\mu_0^- : P \to \mathbb{N}$ gives the initial distribution of negative tokens in the places, called *'negative marking '* of signed Petri net.

Thus, a *'marking '* of signed Petri net can be represented as a vector $\mu = (\mu^+, \mu^-)$ with μ^+, $\mu^- \in \mathbb{N}^n$, $n = |P|$, such that $\mu(p_i) = (\mu^+(p_i), \mu^-(p_i)) \; \forall \; p_i \in P$.

In the graphical representation of a signed Petri net, positive and negative arcs are represented by solid and dotted lines respectively. A positive token is represented by a filled circle and a negative token by an open circle as shown in **Figure 2.1**. A signed Petri net is said to be

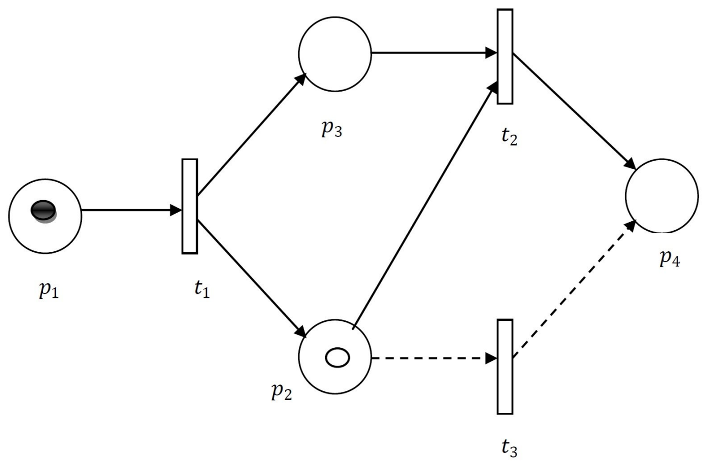


Figure 2.1: A signed Petri net with initial marking ((1,0,0,0),(0,1,0,0))

'negative ' if all of its arcs are negative in sign.

A *'Complete signed Petri net '* is a signed Petri net where every place-transition pair is connected by a bi-directional arc (either positive or negative).

A signed Petri net $N_1^* = (N_1', \sigma_1, \mu_0')$, where $N_1' = (P_1, T_1, I_1^-, I_1^+)$, is called a *'Sub-signed Petri net '* of a signed Petri net $N^* = (N', \sigma, \mu_0)$, if

1. $P_1 \subseteq P$

2. $T_1 \subseteq T$

3. $I_1^+(p,t) \neq 0$ if $I^+(p,t) \neq 0$ for $(p,t) \in (P_1 \times T_1)$

4. $I_1^-(p,t) \neq 0$ if $I^-(p,t) \neq 0$ for $(p,t) \in (P_1 \times T_1)$

5. $\sigma_1 = \sigma \; \forall (p,t)$ such that $I_1^+(p,t) \neq 0 \; or \; I_1^-(p,t) \neq 0$

6. $\mu_0'(p) = \mu_0(p) \; \forall \; p \in P_1$

2.2.1 Execution Rules for signed Petri nets

The execution of a Petri net depends on the distribution of tokens in its places. The execution takes place by firing of a transition. A transition may fire if it is enabled.

A transition t in a signed Petri net N^* is said to be *'enabled '* at a marking $\mu = (\mu^+, \mu^-)$ if

$$I^-(p,t) \leq \mu^+(p) \; \forall p \in {}^{\bullet}t \text{ for which } \sigma(p,t) = +$$

and

$$I^-(p,t) \leq \mu^-(p) \; \forall p \in {}^{\bullet}t \text{ for which } \sigma(p,t) = -$$

An enabled transition t may *'fire '* at $\mu = (\mu^+, \mu^-)$ provided $\exists \; p_k \in t^{\bullet}$ such that:

$$\sigma(t,p_k) = \begin{cases} + & \text{if } \sigma(p,t) = + \; \forall p \in {}^{\bullet}t \\ - & \text{if } \sigma(p,t) = - \; \forall p \in {}^{\bullet}t \\ + \text{ or } - & \text{if } \sigma(p,t) = + \text{ for some } p \in {}^{\bullet}t \\ & \text{and } - \text{ for some } p \in {}^{\bullet}t \end{cases}$$

After firing, it yields a new marking $\mu_1 = (\mu_1^+, \mu_1^-)$ given by the rule:

$$\mu_1^+(p) = \mu^+(p) - I^-(p,t) + I^+(p,t)\ \forall p \in P$$

where (p,t) & (t,p) are positive arcs, if exist

$$\mu_1^-(p) = \mu^-(p) - I^-(p,t) + I^+(p,t)\ \forall p \in P$$

where (p,t) & (t,p) are negative arcs, if exist

The marking μ_1 is said to be directly reachable from a marking μ and written as $\mu \xrightarrow{t} \mu_1$. A restriction is taken to allow the movement of positive (negative) tokens to positive (negative) arcs only.

A *firing or occurrence sequence* is a sequence of transitions $\eta = t_1 t_2 \ldots t_k$ such that

$$\mu \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \mu_2 \xrightarrow{t_3} \mu_3 \ldots \xrightarrow{t_k} \mu'$$

μ' is said to be reachable from μ. Note that a transition $t_j,\ 1 \le j \le k$ can occur more than once in the firing sequence η.

Remark 2.1. *A source transition is always enabled while a sink transition never fires in a signed Petri net. Irrespective of the marking at any time, the condition of firing depends only on the sign of the incoming and outgoing arcs. So, without loss of generality, any sink transition and a transition with incoming arcs of only one sign and outgoing arcs of only the other sign can be assumed to be absent.*

In **Figure 2.1**, transitions t_1, t_3 are enabled at marking

$\mu = ((1,0,0,0),(0,1,0,0))$ and can fire. Look at the execution of a signed Petri net with the help of some more examples.

In **Figure 2.2**, t_1 and t_2 both are enabled at μ_0. Firing of t_1 at μ_0 yields a new marking $\mu_1 = ((0,1,1,0),(1,0,1,0))$ while firing of t_2 at μ_0 yields the marking $\mu_2 = ((1,0,2,0),(0,1,0,1))$. In **Figure 2.3**, t_1 is enabled, while t_2 is not. t_1 can fire at μ_0 to give a new marking $\mu_1 = ((0,0,1,0),(0,0,0,1))$.

In **Figure 2.4**, t_1 and t_2 both are enabled. Firing of transition t_1 at μ_0 gives a new marking $\mu_1 = ((0,0,0,1),(0,0,0,0))$, while firing of transition t_2 at μ_0 gives a new marking $\mu_2 = ((0,1,0,0),(1,0,0,1))$.

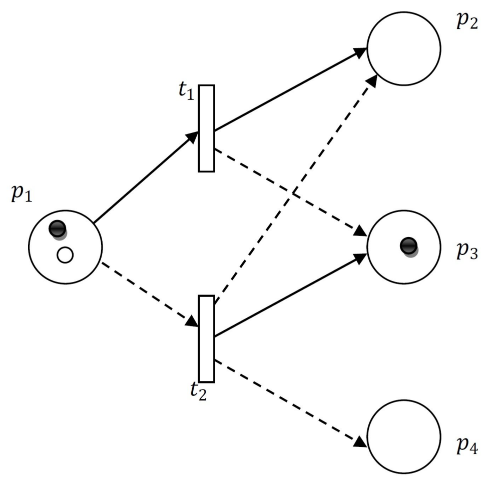


Figure 2.2: A SiPN with $\mu_0 = ((1,0,1,0),(1,0,0,0))$

2.2.2 Assignment of sign to vertices of a signed Petri net

The vertices in a signed Petri net can also be assigned sign. A transition is assigned sign by the product of sign of arcs (incoming and outgoing) incident on it. In **Figure 2.1**, all transitions are positive in sign. Places can be assigned sign in one of the two ways:

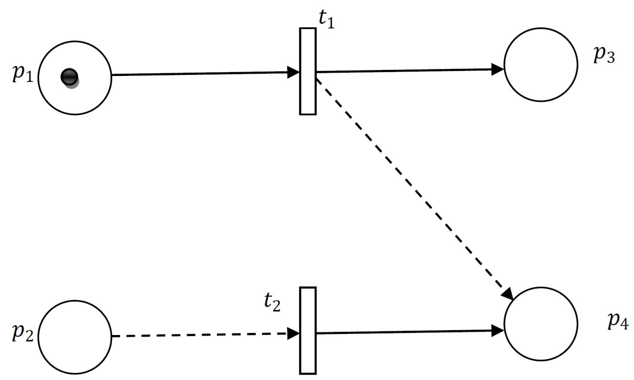


Figure 2.3: A SiPN with $\mu_0 = ((1,0,0,0),(0,0,0,0))$

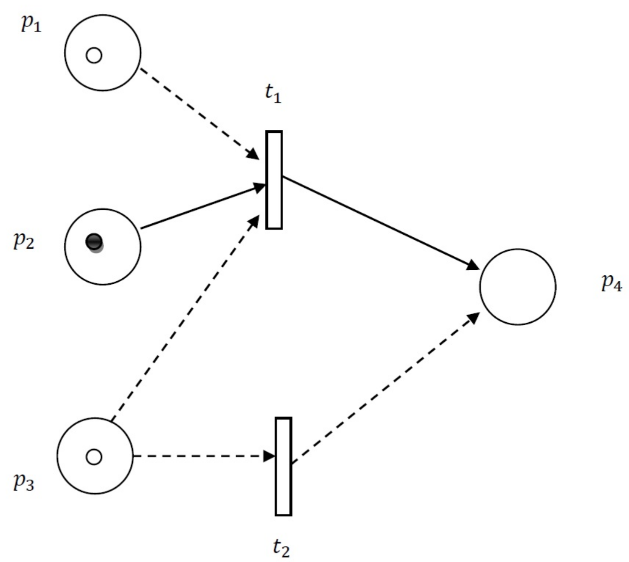


Figure 2.4: A SiPN with $\mu_0 = ((0,1,0,0),(1,0,1,0))$

1. **With respect to arcs –** Sign is assigned to a place by taking the product of incident arcs (incoming and outgoing) on that place. In **Figure 2.1**, places p_1 and p_3 are positive in sign while p_2 and p_4 are negative in sign.

2. **With respect to marking –** Sign is assigned to a place by taking the product of sign of tokens in that place in the given marking. A place without a token is considered to be positive. In **Figure 2.1**, places $p_1,\ p_3,\ p_4$ are positive in sign while p_2 is negatively signed

with respect to μ_0.

Remark 2.2. *Assigning sign to places with respect to arcs doesn't utilize the most important characteristic of Petri nets which is its dynamic behaviour. Hence, assigning a sign to places with respect to marking has been used throughout the thesis.*

An example is given which utilizes the concept of sign of places to determine whether an online transaction initiated by a bank customer is approved or denied by the bank. This transaction is based on the verification of One Time Password (OTP) sent by the bank to the registered mobile number of customer. This situation is modeled by a signed Petri net as shown in **Figure 2.5**. When a customer enters an

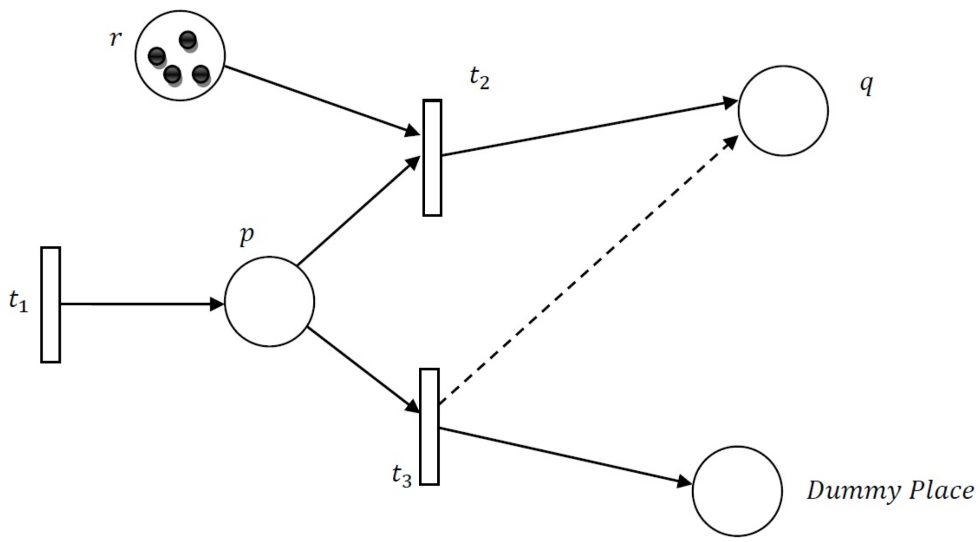


Figure 2.5: A SiPN Model for OTP verification

OTP, firing of transition t_1 takes place. The number of times that t_1 fires is equal to the number of digits in the OTP. On firing of t_1, a positive token is generated in place p which represents the digit entered by the customer. This entered digit is then verified, by comparing it with the corresponding digit of the OTP (sent by the bank), which exists in the bank system and is represented by positive tokens in place r. If these two digits match then, transition t_2 fires, else transition t_3 fires. After the firing of either t_2 or t_3, the sign of place q is checked. If the sign of place q with respect to this marking is negative, then the transaction is denied by the bank. On the other hand, if it is positive, the next digit of the OTP is checked in a similar manner until all the digits are exhausted.

Thus, it can be concluded whether a transaction is denied or approved by the bank based on the sign of place q. If at any marking, the sign of place q is negative, bank denies transaction to the customer. However, if for all the markings with $\mu(q) \neq 0$, the sign of place q is positive, then the transaction is approved by the bank. Another way to check the same is by utilizing the marking of the SiPN. If at any marking, $\mu(DummyPlace) \neq 0$, then the transaction is denied by the bank.

2.2.3 Reachability Tree of signed Petri nets

The *'Reachability Set'*, $R(N^*, \mu)$ of a signed Petri net N^* is the set of all markings of N^* reachable from μ.

A *'Reachability Tree'* represents the reachability set of a given signed Petri net. The reachability tree of the SiPN in **Figure 2.1** is given in **Figure 2.6**.

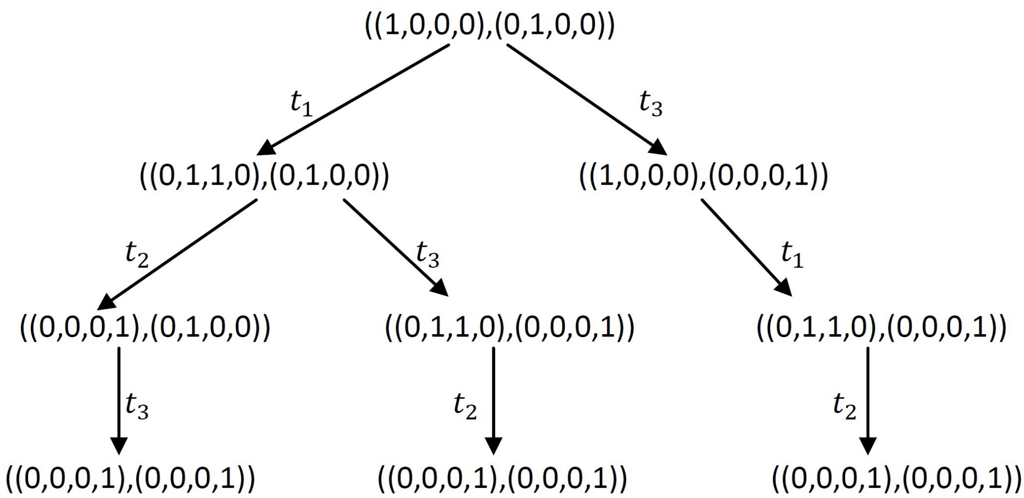


Figure 2.6: Reachability tree $R(N^*, \mu_0)$ of SiPN in **Figure 2.1** with $\mu_0 = ((1,0,0,0),(0,1,0,0))$

2.3 Applications of signed Petri nets

2.3.1 Message Transmission System

When two parties need to communicate securely i.e., they do not want a third party to listen in, they require a channel for secure communication. Secure communication is a way to transfer message from source to receiver so that it is not susceptible to eavesdropping or interception. The concept of signed Petri net is used to model a message transmission system given in **Figure 2.7** which is secure.

Consider a message which needs to be transferred from source to a destination securely. In order to make the transmission secure, the message is divided into two parts– the first half of the message is represented by a positive token and the other half via a negative one. The places p_1 and p_6 are source and destination for the message respectively. All other places are buffer for the message, which hold the message until it is ready to be transmitted, by firing of corresponding transitions. The transitions are events which transfer the message from one place to another. Note that $|{}^{\bullet}t| = |t^{\bullet}| = 1 \ \forall \ t \in T$.

In a signed Petri net, positive tokens move on positive arcs and negative tokens on negative arcs only. Therefore, there is a fixed path for the movement of both the tokens and no part of the message can move on the path of the other part of the message. Also, all transitions are restricted to have only arc of one sign either positive or negative i.e., for $p_i \in t^{\bullet}$

$$\sigma((t,p_i)) = \begin{cases} + & \text{if} \quad \sigma((p,t)) = + \text{ for } p \in {}^{\bullet}t \\ - & \text{if} \quad \sigma((p,t)) = - \text{ for } p \in {}^{\bullet}t \end{cases}$$

This restriction makes it impossible for any intermediate place/transition to have access to both parts of the message, thus avoiding leakage. In

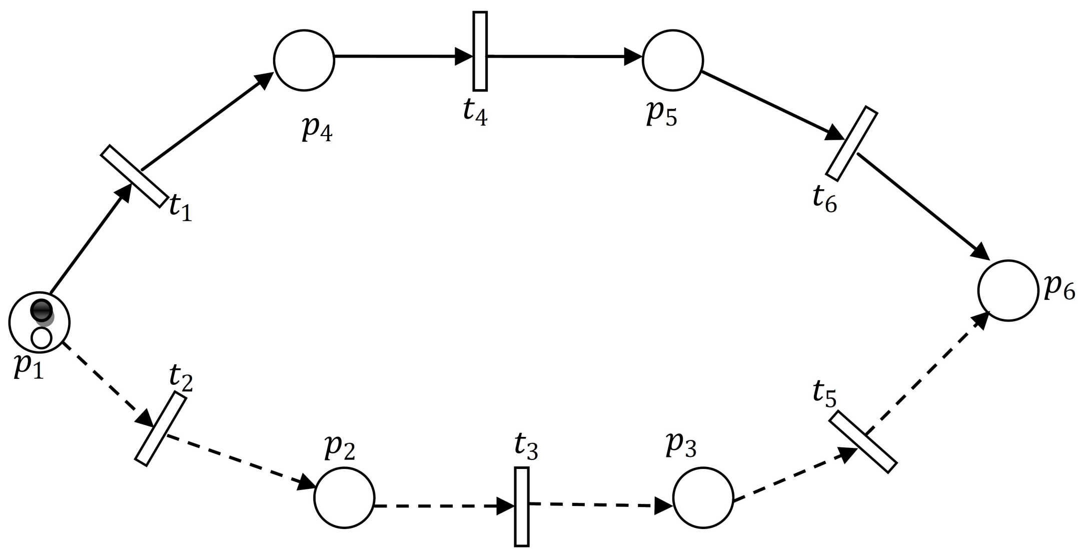


Figure 2.7: A message transmission system with message divided into two parts.

Figure 2.7, a message is divided into two parts at place p_1 which is the source of the message. The positive token (which is the first half of the message) enables transition t_1 and the negative token (the other half of the message) enables transition t_2. When the transitions t_1 and t_2 fire, the message transmission begins. To reach the destination p_6, the positive token moves on path $t_1p_4t_4p_5t_6$ and the negative token on path $t_2p_2t_3p_3t_5$. These two parts of the message can then be combined to get the complete message at place p_6.

In this way, the message is transmitted from the source to destination securely. The intermediate places between the source and destination, help in the transmission of the message on the given path. Thus, the division of message into two parts and transfer of these parts via different paths reduce the chances of leakage of the message.

Extension of the Message Transmission System

In the extended model, places and transitions have usual meanings as in the previous model, except for the transitions t_1 and t_8, which represent a process of dividing the incoming part of the message further into two parts.

In **Figure 2.8**, a message is divided into two parts at place p_1 which is the source of the message. The positive token (which is the first half of the message) enables transition t_8 and the negative token (the other half of the message) enables transition t_1. When the transitions t_1 and t_8 fire, the parts of the message are further divided into two parts each. Thus, the original message is now divided into four parts. WLOG, assume that a positive token always represents the first half of the message or the first half of the part of the message. As the message transmission continues, the positive token (i.e., first half of the first part of the message) moves on the path $p_8t_9p_9t_{10}$ and finally reaches the destination which is place p_{10}. Similarly, negative token (second half of the first part of the message) moves on the path $p_6t_6p_7t_7$ and reaches the destination. In a similar fashion, other half of the original message which is further divided into two parts by transition t_1 reaches the destination p_{10}.

In this way, the message is transmitted by dividing it into four parts and transmitting these parts through different routes. This makes it difficult for anyone to access the whole message simultaneously, thereby making it secure. The model can be modified to divide the message into more parts so as to increase the level of secureness by dividing the message further, at the transitions which follow the transition t_1 and t_8.

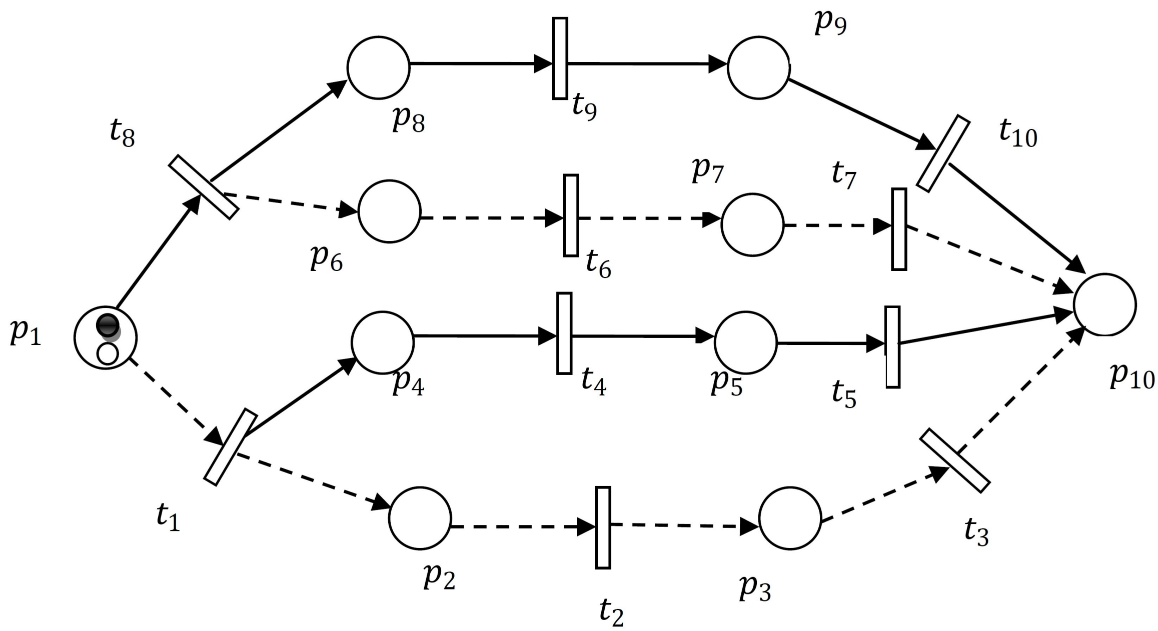


Figure 2.8: A message transmission system with message divided into four parts.

2.3.2 Production Unit

Consider a production unit which produces and packages a product. The product is produced by first creating its parts, which are six in number separately and then assembling them together to get the final product in the assembling unit. In **Figure 2.9**, parts of the product are produced at place p_1. Assume that the parts which are heavier than a prescribed weight of W units are represented by a positive token and the remaining parts via a negative one. This is done so that the heavier parts move on the path $p_1t_2p_3t_4p_4$, which is suitable for transporting heavy parts, rather than on the path $p_1t_1p_2t_3p_4$ through which lighter parts are transported. All the transitions in the signed Petri net represent an event of the transfer of token (part) from one place to another. The places p_2 and p_3 are the test units which check the working of a part. Whenever a part of a product comes, it is tested for the required characteristics (depending on the product), and based on it, the part is segregated after being classifed as "Tested Okay" or "not". Thus, the transitions t_4 and t_7 can fire accordingly. If the part is working correctly, it passes the test and moves to the assembling unit via firing of transition t_3 or t_4. On the other hand, if the part turns out to be defective it is rejected and moved to place p_6 by firing of transitions t_6 or t_7. Whenever a part moves to the assembling unit or it is rejected, production of such a part should start again. Hence, arcs $(t_3, p_1), (t_4, p_1); (t_6, p_1), (t_7, p_1)$ are added.

If none of the six parts are rejected, the final product is assembled at place p_4. The final product is represented by a token whose sign is equal to the product of sign of all the tokens representing parts of the product (This sign is positive here for the given marking). This final product is then moved to place p_5 for packaging by firing of transition t_5. It should be noted that the sign of arcs (p_4, t_5) and (t_5, p_5) are same as the sign of token representing the final product.

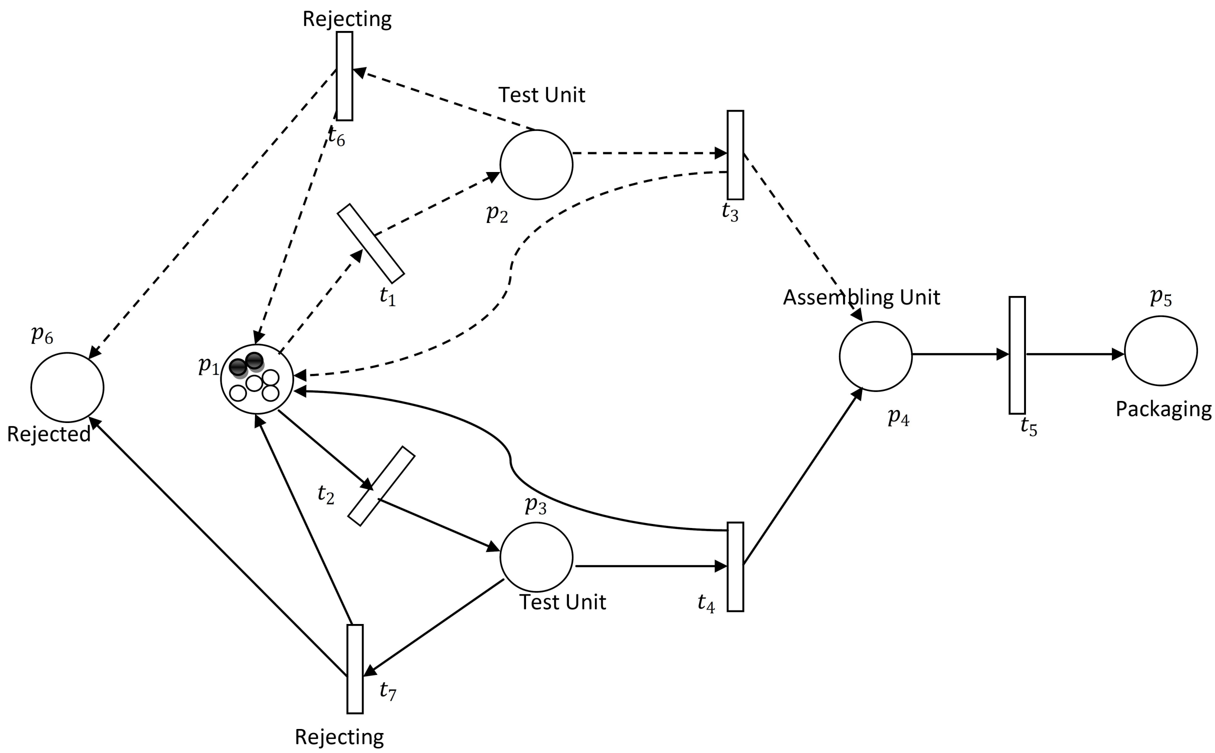


Figure 2.9: A production unit

2.3.3 Modifications in signed Petri nets to simulate effects of Logic Petri net

To describe batch processing functions and passing value indeterminacy in cooperative systems, Logic Petri nets have been defined as high level Petri nets [4]. In this section, versatility of the signed Petri nets have been established by showing an equivalent Logic Petri net can be constructed using a signed Petri net, which the authors call a *'Logic signed Petri net'*.

Definition 2.1. ***Logic Petri net (LPN)*** *[4] LPN, $N_l = (P,T,I^-,I^+,I,O,\mu_0)$ is a logic Petri net if and only if*

1. *(P,T,I^-,I^+,μ_0) is a Petri net.*

2. *T includes three subsets of transitions, i.e.,$T = T_D \cup T_I \cup T_O$, $\forall\ t \in T_I \cup T_O : {}^\bullet t \cap t^\bullet = \emptyset$ where*

    - *T_D denotes a set of traditional transitions.*
    - *T_I denotes a set of logic input transitions.*
    - *T_O denotes a set of logic output transitions.*

3. *I is a mapping from logic input transitions to a logic input expression, i.e., $\forall\ t \in T_I,\ I(t) = f_I(t)$, where $f_I(t)$ is a logic input expression associated with the transition t.*

4. *O is a mapping from logic output transitions to a logic output expression, i.e., $\forall\ t \in T_O,\ O(t) = f_O(t)$, where $f_O(t)$ is a logic output expression associated with the transition t.*

5. Transition Firing Rules

    - *$\forall t \in T_D$, the firing rules are same as in Petri nets.*
    - *$\forall t \in T_I$, t is enabled at μ if the input expression $f_I(t)$ is true at μ. After firing it yields a new marking μ' given by the following rule:*
      $\forall p \in {}^\bullet t$, if $\mu(p) = 1$, then $\mu'(p) = \mu(p) - 1$;
      $\forall p \in t^\bullet,\ \mu'(p) = \mu(p) + 1$
      $\forall p \notin {}^\bullet t \cup t^\bullet,\ \mu'(p) = \mu(p)$
    - *$\forall t \in T_O$, t is enabled if $\forall p \in {}^\bullet t : \mu(p) = 1$. Firing t will generate a new marking μ', given by: $\forall p \in S : \mu'(p) = \mu(p) + 1$;*

$\forall p \in {}^{\bullet}t,\ \mu'(p) = \mu(p) - 1$

$\forall p \notin {}^{\bullet}t \cup t^{\bullet}$ *or* $\forall p \in t^{\bullet} \backslash S : \mu'(p) = \mu(p)$ *where* $S \subseteq t^{\bullet}$*, such that output expression* $f_O(t)$ *is true at* μ'*.*

Remark 2.3. *In the definition of logic Petri nets, a logic input expression is attached to a logic input transition. It means that that there is a logic expression involving the input places of the given transition.*

Similarly, a logic output expression is attached to a logic output transition. It means that there is a logic expression involving the output places of the given transition.

Logic signed Petri net (LSiPN)

The *'Logic signed Petri net '* (LSiPN) has been obtained by modification in the execution rules of signed Petri nets.

Definition 2.2. ***Logic signed Petri net***

A logic signed Petri net is defined by $N_L = (P, T, A^-, A^+, B^-, B^+, \mu_0)$ *where*

1. *P is a finite, non empty set of places.*

2. *T includes three subsets of transitions, i.e.,* $T = T_D \cup T_I \cup T_O$, $\forall\, t \in T_I \cup T_O : {}^{\bullet}t \cap t^{\bullet} = \emptyset$ *where*

   - T_D *denotes a set of traditional transitions.*
   - T_I *denotes a set of logic input transitions.*
   - T_O *denotes a set of logic output transitions.*

   If $t \in T_I$ *then* $\forall\ p\ \in {}^{\bullet}t$ *either* $\sigma(p,t) = +$ *or* $\sigma(p,t) = +$ *as well as* $-$ *and* $\forall\ p\ \in t^{\bullet}\ \sigma(t,p) = +$*.*

   Similarly, if $t \in T_O$ *then* $\forall\ p\ \in t^{\bullet}$ *either* $\sigma(t,p) = +$ *or* $\sigma(t,p) = +$ *as* *well as* $-$ *and* $\forall\ p\ \in {}^{\bullet}t\ \sigma(p,t) = +$, *where* σ *denotes the same function by which arcs are assigned a sign in the definition of SiPN.*

3. $A^- = [a^-_{ij}]$ *where* a^-_{ij} *gives the number of positive arcs from* p_j *to* t_i.

4. $A^+ = [a^+_{ij}]$ *where* a^+_{ij} *gives the number of positive arcs from* t_i *to* p_j.

5. $B^- = [b^-_{ij}]$ *where* b^-_{ij} *gives the number of negative arcs from* p_j *to* t_i.

6. $B^+ = [b^+_{ij}]$ *where* b^+_{ij} *gives the number of negative arcs from* t_i *to* p_j.

7. *An initial marking in an LSiPN can be represented as a vector* $\mu_0 = (\mu_0^+, \mu_0^-)$ *with* $\mu_0^+, \mu_0^- \in \mathbb{N}^n$, $n = |P|$ *such that* $\mu_0(p_i) = (\mu_0^+(p_i), \mu_0^-(p_i))\ \forall\ p_i \in P$.

A marking in an LSiPN can be represented as a vector $\mu = (\mu^+, \mu^-)$ *with* $\mu^+, \mu^- \in \mathbb{N}^n$, $n = |P|$ *such that* $\mu(p_i) = (\mu^+(p_i), \mu^-(p_i))\ \forall\ p_i \in P$.

Execution rules for logic signed Petri nets

1. *If* $t_i \in T_D$, *then the execution rules are same as for signed Petri nets.*

2. *For* $t_i \in T_I$

   - Enabling Condition- *A transition* $t_i \in T_I$ *is enabled at* μ *provided*

     (a) $\mu^+(p_j) = 1\ \forall\ p_j\ \in\ S_1 = \{p_j \in {}^{\bullet}t_i \mid a^-_{ij} = 1, b^-_{ij} = 0\}$ *and*

     (b) $\forall p\ \in\ S_2 = \{p_j \in {}^{\bullet}t_i \mid a^-_{ij} = 1, b^-_{ij} = 1\}$, $\exists p_k \in S_2$ *with* $\mu^+(p_k) = 1$ *and* $\forall p_j \in S_2 \backslash \{p_k\}$, *either* $\mu^+(p_j) = 1$ *or* $\mu^-(p_j) = 1$.

- Firing Condition- *An enabled transition $t_i \in T_I$ may fire at a marking $\mu = (\mu^+, \mu^-)$ to yield a new marking μ_1 given by :-*

  – $\forall\ p_j\ \in\ S_1,$

  $$\mu_1^+(p_j) = \mu^+(p_j) - a_{ij}^- + a_{ij}^+$$

  – $\forall\ p_j\ \in\ S_2,$ *for which* $\mu^+(p_j) = 0\ \&\ \mu^-(p_j) = 1$

  $$\mu_1^+(p_j) = \mu^+(p_j)$$

  $$\mu_1^-(p_j) = \mu^-(p_j) - b_{ij}^- + b_{ij}^+$$

  – $\forall\ p_j\ \in\ S_2,$ *for which* $\mu^+(p_j) = 1\ \&\ \mu^-(p_j) = 0$

  $$\mu_1^-(p_j) = \mu^-(p_j)$$

  $$\mu_1^+(p_j) = \mu^+(p_j) - a_{ij}^- + a_{ij}^+$$

  – *For all* $p_j \in P \backslash {}^\bullet t_i$

  $$\mu_1^+(p_j) = \mu_0^+(p_j) - a_{ij}^- + a_{ij}^+$$

  $$\mu_1^-(p_j) = \mu_0^-(p_j) - b_{ij}^- + b_{ij}^+$$

*An LSiPN with a logic input transition is shown in **Figure 2.10**. In this LSiPN, transition t is enabled at $\mu = ((1,0,1,0),(0,1,0,0))$. t fires to yield a new marking $\mu_1 = ((0,0,0,1),(0,0,0,0))$.*

3. *For $t_i \in T_O$*

- Enabling Condition- *A transition $t_i\ \in T_O$ is enabled if*

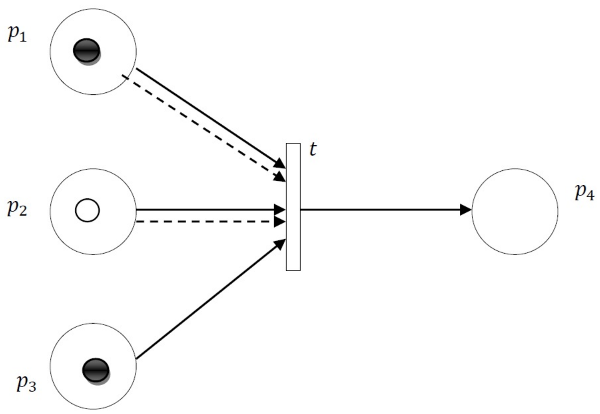


Figure 2.10: An LSiPN with a logic input transition.

$\forall p_j \in S_1, \; \mu^+(p_j) = 1.$

- Firing Condition- *An enabled transition $t_i \in T_O$ may fire at a marking $\mu = (\mu^+, \mu^-)$ to yield a new marking μ_1 given by :-*
  $\forall p_j \in t_i^\bullet$
  - $\mu_1^+(p_j) = 1$ *whenever* $a_{ij}^+ = 1, b_{ij}^+ = 0$
  - *If* $a_{ij}^+ = 1, b_{ij}^+ = 1$, *then,* $\exists p_k \in t_i^\bullet$ *with* $\mu_1^+(p_k) = 1$ *and* $\forall p_l \in t_i^\bullet \setminus \{p_k\}$ *either* $\mu_1^+(p_l) = 1$ *or* $\mu_1^-(p_l) = 1$

$\forall p_j \in P \setminus t_i^\bullet$

$$\mu_1^+(p_j) = \mu_0^+(p_j) - a_{ij}^- + a_{ij}^+$$

$$\mu_1^-(p_j) = \mu_0^-(p_j) - b_{ij}^- + b_{ij}^+$$

An LSiPN with a logic output transition is shown in **Figure 2.11**. In this LSiPN, transition t is enabled at $\mu = ((0,0,0,1),(0,0,0,0))$. t fires to yield a new marking $\mu_1 \in \{((1,1,0,0),(0,0,1,0)),((1,0,1,0),(0,1,0,0)), ((1,1,1,0),(0,0,0,0))\}$.

While modeling a workflow process using an LSiPN, a positive token in a data place represents that the data has arrived from the organization

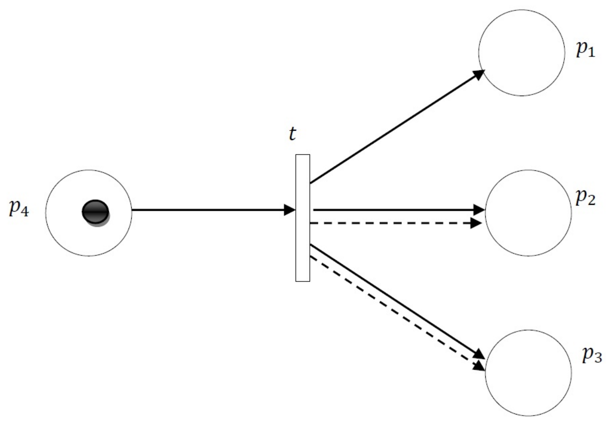


Figure 2.11: An LSiPN with a logic output transition.

representing this place, while a negative token represents that the data has not arrived from the organization. So, in a given workflow cycle, all the arrived data is processed while the data which arrives late is processed in the next workflow cycle.

Equivalence of Logic Petri net and Logic Signed Petri net

The definitions of isomorphism and equivalence given below are used to prove the equivalence between a logic Petri net and a Logic signed Petri net .

Definition 2.3. ***Isomorphism***

Let $N_l = (P,T,I^-,I^+,I,O,\mu_0)$ *be an LPN and* $N_L = (P',T',A^-,A^+,B^-,B^+,\mu_0')$ *be an LSiPN. Let* $R(N_l,\mu_0)$ *be the reachability set of* N_l *and* $R^+(N_L,\mu_0')$ *be the reachability set of positive markings of* N_L*.*

Then, logic Petri net N_l *and logic signed Petri net* N_L *are said to be equivalent if there is a bijective function* f *such that for any initial marking* μ_0 *of the logic Petri net, the logic signed Petri net* N' *with initial marking*

$f(\mu_0)$ satisfies the condition:

$$\forall \mu_1, \mu_2 \in R(N_l, \mu_0),\ t \in T; \mu_1 \xrightarrow{t} \mu_2 \iff \exists t' \in T'; f(\mu_1) \xrightarrow{t'} f(\mu_2)$$

Definition 2.4. ***Equivalence***
Let $N_l = (P, T, I^-, I^+, I, O, \mu_0)$ be an LPN and $N_L = (P', T', A^-, A^+, B^-, B^+, \mu_0')$ be an LSiPN. Then, N_l & N_L are equivalent iff $R(N_l, \mu_0)$ & $R^+(N_L, \mu_0')$ are isomorphic to each other.

Theorem 2.1. *For any Logic Petri net, there exists an equivalent Logic signed Petri net.*

Proof. Given an LPN $N_l = (P, T, I^-, I^+, I, O, \mu_0)$, an LSiPN $N_L = (P', T', A^-, A^+, B^-, B^+, \mu_1)$ can be constructed as follows.

1. The set of places of N_L is same as N_l, i.e., $P' = P$.

2. Construction of set of transitions T' of N_L:

   (a) All the traditional transitions of N_l will be the traditional transitions of N_L, i.e., if $t_i \in T_D = T \backslash (T_I \cup T_O)$. Then $t_i \in T_D' \subseteq T'$. This traditional transition t_i in N_L is connected with a place in N_L given by the rule below:

   $$\forall p_j \in P, \text{ if } I^-(p_j, t_i) \neq 0 \text{ then, } a_{ij}^- = 1, b_{ij}^- = 0$$

   $$\forall p_k \in P, \text{ if } I^+(p_k, t_i) \neq 0 \text{ then, } a_{ik}^+ = 1, b_{ik}^+ = 0$$

   (b) All the logic input transitions of N_l are logic input transitions of N_L, i.e., if $t_i \in T_I = T \backslash (T_D \cup T_O)$. Then $t_i \in T_I' \subseteq T'$ and this transition $t_i \in T_I'$ is connected with a place in N_L by the rule below:

Let ${}^{\bullet}t_i = \{p_1, p_2, p_3, ..., p_k\}$ in N_l and $f_I(t_i)$ be the logic input expression associated with t_i, convert $f_I(t_i)$ into its disjunctive normal form (DNF), which is unique. If any p_j occurs in both the forms i.e., p_j and $\neg p_j$ in the DNF then, there exist two arcs in N_L from p_j to t_i, one of positive sign and other of negative sign. Therefore, for such $p_j \in {}^{\bullet}t_i$, $a^-_{ij} = 1, b^-_{ij} = 1$.

On the other hand, if p_j occurs only in positive form, i.e., as p_j and not as $\neg p_j$ in the DNF, then a positive arc is formed from p_j to t_i. Therefore, for such $p_j \in {}^{\bullet}t_i$, $a^-_{ij} = 1, b^-_{ij} = 0$.

$\forall p_k \in t_i^{\bullet}$, $a^+_{ik} = 1, b^+_{ik} = 0$.

(c) All the logic output transitions of N_l are logic output transitions of N_L, i.e., if $t_i \in T_O = T \backslash (T_D \cup T_I)$. Then $t_i \in T'_O \subseteq T'$ and this transition $t_i \in T'_O$ is connected with a place in N_L by the rule below:

Let $t_i^{\bullet} = \{p_1, p_2, p_3, ..., p_k\}$ in N_l and $f_O(t_i)$ be the logic output expression associated with t_i, convert $f_O(t_i)$ into its disjunctive normal form (DNF) which is unique. If any p_j occurs in both the forms i.e., p_j and $\neg p_j$ in the DNF then, there exist two arcs in N_L from t_i to p_j, one of positive sign and other of negative sign. Therefore, for such $p_j \in t_i^{\bullet}$, $a^+_{ij} = 1, b^+_{ij} = 1$.

On the other hand, if p_j occurs only in positive form, i.e., as p_j and not as $\neg p_j$ in the DNF, then a positive arc is formed from t_i to p_j. Therefore, for such $p_j \in t_i^{\bullet}$, $a^+_{ik} = 1, b^+_{ik} = 0$.

$\forall p_k \in {}^{\bullet}t_i$, $a^-_{ik} = 1, b^-_{ik} = 0$.

3. Assignment of tokens

(a) For all p_j satisfying $(a^-_{ij} = 1$ and $b^-_{ij} = 1)$ or $(a^+_{ij} = 1$ and $b^+_{ij} = 1)$

$\mu_1^+(p_j) = 1$ & $\mu_1^-(p_j) = 0$ in N_L if $\mu_0(p_j) = 1$ in N_l, and

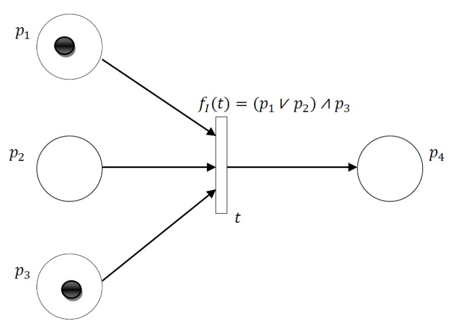
$\mu_1^+(p_j) = 0$ & $\mu_1^-(p_j) = 1$ in N_L if $\mu_0(p_j) = 0$ in N_l

(b) For all p_j which has not been assigned a token in step (a) and satisfying either $(a_{ij}^- = 1, b_{ij}^- = 0)$ or $(a_{ij}^+ = 1, b_{ij}^+ = 0)$, $\mu_1^+(p_j) = 1$ & $\mu_1^-(p_j) = 0$ in N_L if $\mu_0(p_j) = 1$ in N_l , and

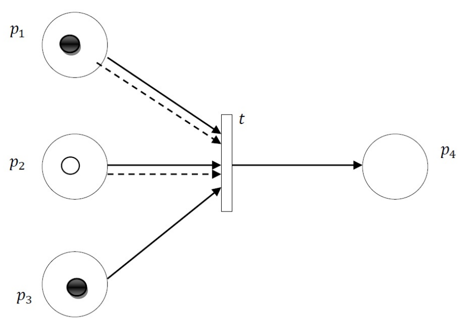
$\mu_1^+(p_j) = 0$ & $\mu_1^-(p_j) = 0$ in N_L if $\mu_0(p_j) = 0$ in N_l

The equivalence between N_l and N_L can be easily proved, because each marking of N_l corresponds to a positive marking in N_L with the same number of positive tokens in the corresponding places in LPN and LSiPN (By Construction). That is, in N_l $\forall$ $\mu_1, \mu_2 \in R(N_l, \mu_0)$, $t \in T$; $\mu_1 \xrightarrow{t} \mu_2 \implies \exists\, t' \in T'; f(\mu_1) \xrightarrow{t'} f(\mu_2)$

This means that N_l and N_L have same behaviour characteristics. Moreover, the structure of N_L is unique since the DNF is unique. So, f is a bijective function and $R(N_l, \mu_0)$ and $R^+(N_L, \mu_1)$ are isomorphic. Consequently, N_l and N_L are equivalent. $\square$



(a) An LPN with a logic input transition

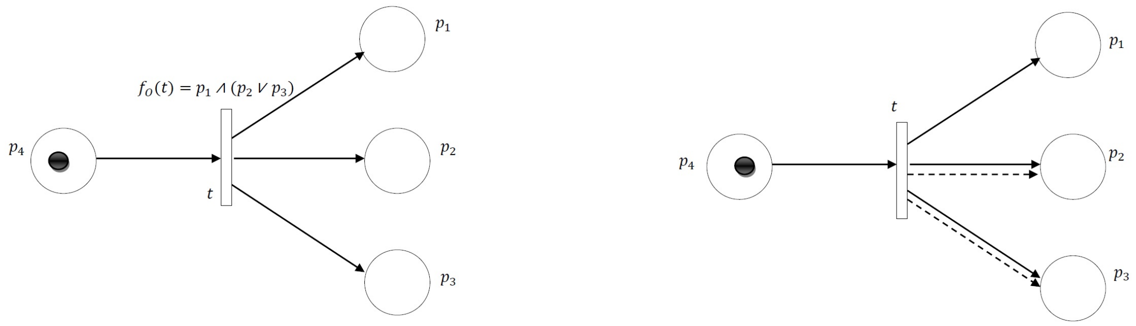


(b) Equivalent LSiPN for LPN (a)

Figure 2.12: An LPN and Equivalent LSiPN

In the **Figure 2.12**, an LPN with a logic input transition is converted to an equivalent LSiPN using the procedure in the Theorem 2.1. In the **Figure 2.13**, an LPN with a logic output transition is converted to an equivalent LSiPN using the procedure in the Theorem 2.1.

In the example in **Figure 2.14**, an LSiPN is constructed from a given



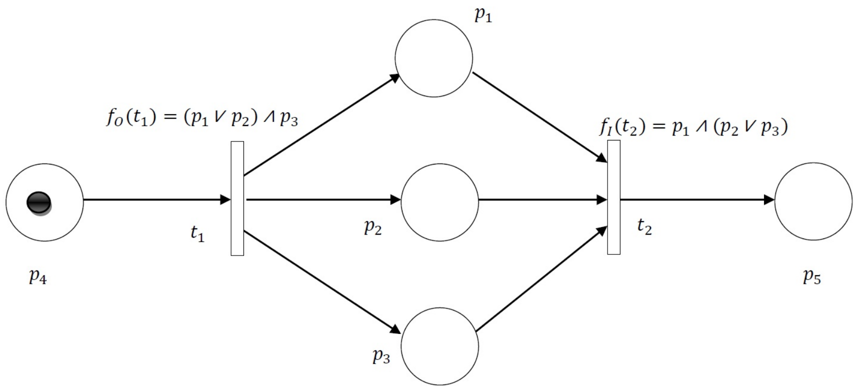
(a) An LPN with a logic output transition (b) Equivalent LSiPN for LPN (a)

Figure 2.13: An LPN and Equivalent LSiPN

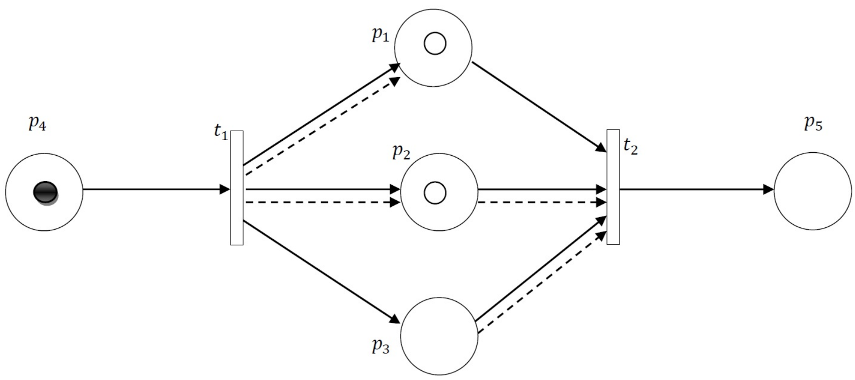
LPN using the procedure in the Theorem 2.1. The reachability tree of the LPN and the LSiPN are also given in **Figure 2.15** and it can be clearly seen that corresponding to every marking in LPN there exists a positive marking in LSiPN. Thus, the reachability set of LPN and the set of positive markings of LSiPN are isomorphic which implies that the two Petri nets are equivalent.

In concluding remark, a signed Petri net has been introduced and its associated terms and concepts are defined in this chapter. An example which utilizes the newly introduced concept of assignment of sign to places of signed Petri nets is given which demonstrates the advantages of using signed Petri nets. The applications of signed Petri nets in message transmission system and production unit have been discussed. An LSiPN is formulated using SiPNs by merely changing its firing rules to simulate an LPN. This shows the versatile nature of the SiPNs, introduced in the chapter.

The introduced concept of SiPNs will make it possible to define the concepts of domination, structural and dynamical balanceness for

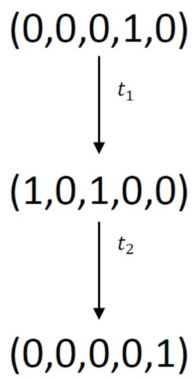


(a) AN LPN

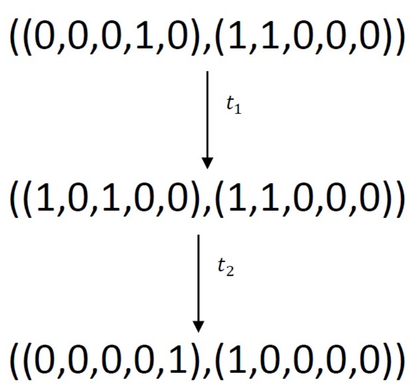


(b) Equivalent LSiPN

Figure 2.14: Example



(a) Reachability Tree for LPN



(b) Reachability Tree for LSiPN

Figure 2.15: Reachability Trees for LSiPN Example in **Figure 2.14**

dynamic systems. These concepts do not exist for previously existing Petri nets models and show how the proposed extension of signed Petri nets is advantageous.

References

[1] Aalst van der, W. M. P. (1998). *The application of Petri nets to workflow management*. Journal of Circuits, Systems and Computers, 8(1):21–66.

[2] Brunsch, T., Hardouin, L., and Raisch, J. (2010). *Control of cyclically operated High-Throughput Screening Systems*. IFAC Proceedings Volumes, 43(12):167–172.

[3] Brunsch, T., Raisch, J., Hardouin, L., and Boutin, O. (2013). *Discrete- Event Systems in a Dioid Framework:Modeling and Analysis*. Control of Discrete-Event Systems: Automata and Petri net Perspectives. Lecture Notes in Control and Information Sciences, Springer, London, 433:431–450.

[4] Du, Y. Y. and Guo, B. (2009). *Logic Petri Nets and Equivalency*. Information Technology Journal, 8:95–100.

[5] Harary, F. (1953). *On the notion of balance of a signed graph*. Michigan Math. Journal, 2:143–146.

[6] Jensen, K. (1987). *Coloured Petri nets*. Coloured Petri nets. In: Brauer, W., Reisig, W., Rozenberg, G. (eds) Petri nets: Central Models and Their Properties. ACPN 1986. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. , 254:248–299.

[7] Kansal, S., Singh, G. P., and Acharya, M. (2010). *On the Petri nets generating all the binary n-vectors*. Scientiae Mathematicae Japonicae Online, 71(2):113–120.

[8] Kansal, S., Singh, G. P., and Acharya, M. (2011). *1-safe Petri nets generating every binary n-vector exactly once*. Scientiae Mathematicae Japonicae Online, 74:127–134.

[9] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). *Predicting Positive and Negative Links in Online Social Networks*. Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, WWW '10:641–650.

[10] Marsan, M. (1990). *Stochastic Petri nets: An Elementary Introduction*. Stochastic Petri nets: An elementary introduction. In: Rozenberg, G. (eds) Advances in Petri nets 1989. APN 1988. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 424:1–29.

[11] Murata, T. (1989). *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, 77(4):541–580.

[12] Murata, T. and Yamaguchi, H. (1990). *A Petri net with Negative tokens and its application to Automated Reasoning*. Proceedings of the 33rd Midwest Symposium on Circuits and Systems, 2:762–765.

[13] Peterson, J. (1981). **Petri net theory and modeling of systems**. *Prentice Hall-Inc*.

[14] Petri, C. (1962). **Kommunikation mit automaten**. Schriften des Institutes fur Instrumentelle Mathematik, Bonn.

[15] Reisig, W. (1991). *Petri nets and algebraic specifications*. Theoretical Computer Science, 80(1):1–34.

[16] Valk, R. (2004). *Object Petri nets*. Lectures on Concurrency

and Petri nets: Advances in Petri nets, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3098:819–848.

Chapter 3

Analysis of Signed Petri Nets

In this chapter, the study of previous chapter is extended as mere modeling of system is of no use, unless the modeled system is interpreted. The behavioral properties of signed Petri nets and analysis techniques for analyzing it are given in this chapter. Two techniques for analysis are provided: Reachability tree and Matrix equations, with focus on Matrix equations. An actual case scenario of a restaurant model is given and analyzed using the techniques given in the chapter. The benefits of using SiPNs to model the restaurant system rather than using traditional Petri nets are also given.

3.1 Introduction

Due to their graphical representation and dynamic nature, Petri nets soon became useful in modeling of systems which are characterized as being distributed, parallel, concurrent, asynchronous, nondeterministic, and/or stochastic [1, 2, 3, 4, 5, 6, 7]. But mere modeling of system is of no use unless the modeled system is interpreted. This led to the introduction of analysis techniques for analyzing such system [3, 4]. The analysis of the system to be modeled is an important aspect to be considered while modeling as it provides insights into the behavior of the system so modeled. In this chapter, the focus is on frequently used behavioral properties as mentioned in [4] while many other properties are given in [3].

The signed Petri nets introduced in the previous chapter are an extension of Petri nets. The behavioral properties for SiPNs can be defined in order to include the effect of positive and negative tokens both. Some important properties of SiPNs which are to be analyzed have been presented in this chapter. Two analysis techniques are given with focus on the technique of analysis by matrix equations. Lastly, an actual case scenario of a restaurant system modeled by SiPNs is given and analyzed using these techniques.

3.2 Behavioral Properties of Signed Petri nets

The behavioral properties of SiPNs differs from that of Petri nets as they include the effect of both positive and negative tokens in it. The properties are given below.

3.2.1 Boundedness

A place p_i in a SiPN N^* is said to be *'positively k_1-bounded'* if for all $\mu \in R(N^*, \mu_0)$,

$$\mu^+(p_i) \leq k_1$$

A SiPN is *'positively k_1-bounded'* if all of its places are positively k_1-bounded.

A place p_i in a SiPN N^* is said to be *'negatively k_2-bounded'* if for all $\mu \in R(N^*, \mu_0)$,

$$\mu^-(p_i) \leq k_2$$

A SiPN is *'negatively k_2-bounded'* if all of its places are negatively k_2-bounded.

A place p_i in a SiPN N^* is said to be *'k-bounded'* if for all $\mu \in R(N^*, \mu_0)$,

$$\mu^+(p_i) + \mu^-(p_i) \leq k$$

A SiPN is *'k-bounded'* if all of its places are k-bounded.

Let $k' = max\{k_1^i |\ 1 \leq i \leq n\}$ and $k'' = max\{k_2^i |\ 1 \leq i \leq n\}$ where k_1^i, k_2^i are bounds for positive and negative tokens in place p_i respectively and $n = |P|$. Clearly, the value of k for a $k-$bounded SiPN can be determined by using k' and k''. If at least one place has both type (positive and negative) of tokens then, $k = k' + k''$. However, if all places have only one type of token then k is given by $max\{k', k''\}$.

A SiPN which is 1-bounded is called a *'Safe SiPN'*.

3.2.2 Conservation

Resources should neither be created nor destroyed while modeling resource allocation systems. Hence, the tokens used to represent them should be conserved.

Strict Conservation

In a SiPN N^* with initial marking μ_0. If for all $\mu \in R(N^*, \mu_0)$,

1.
$$\sum_i \mu_0^+(p_i) = \sum_i \mu^+(p_i)$$
Then, N^* is *'strictly positively conserved'*.

2.
$$\sum_i \mu_0^-(p_i) = \sum_i \mu^-(p_i)$$
Then, N^* is *'strictly negatively conserved'*.

A place p_i is said to be *'strictly conservative'* in N^* if for all $\mu \in R(N^*, \mu_0)$,

$$\sum_i (\mu_0^+(p_i) + \mu_0^-(p_i)) = \sum_i (\mu^+(p_i) + \mu^-(p_i)) \tag{3.2.1}$$

Conservation with respect to a weighing vector

In a SiPN N^*, a weighing vector is given by $w = (w^+, w^-)$, where w^+, w^- are $n \times 1$ positive vectors ($n = |P|$) and gives the weight associated with positive and negative tokens respectively in a place.

If for all $\mu \in R(N^*, \mu_0)$,

1.
$$\sum_i (\mu_0^+(p_i).w^+(p_i)) = \sum_i (\mu^+(p_i).w^+(p_i))$$
$$\text{or} \quad \mu_0^+.w^+ = \mu^+.w^+$$

Then, N^* is *'positively conserved'* with respect to w^+.

2.
$$\sum_i (\mu_0^-(p_i).w^-(p_i)) = \sum_i (\mu^-(p_i).w^-(p_i))$$
$$\text{or} \quad \mu_0^-.w^- = \mu^-.w^-$$

Then, N^* is *'negatively conserved'* with respect to w^-.

The above two equations can be combined to give:

$$\mu_0^+.w^+ + \mu_0^-.w^- = \mu^+.w^+ + \mu^-.w^- \tag{3.2.2}$$

which implies that the SiPN N^* is conserved with respect to weighing vector $w = (w^+, w^-)$.

Remark 3.1. *A SiPN N^* which is conservative with respect to a weighing vector, $w = ((1,1,\ldots,1,1),\ (1,1,\ldots,1,1))$ is strictly conservative.*

Remark 3.2. *The equations* (3.2.1) *and* (3.2.2) *do not imply that SiPN N^* is positively or negatively conserved.*

3.2.3 Liveness

This property provides information about the working of a given SiPN, whether the given system will work smoothly or will enter a deadlock. If a transition t_j can never fire, it is called a *'dead'* transition otherwise, it is called a *'live'* transition.

A SiPN in which no transition is dead is called a *'Live'* SiPN.

3.2.4 Reachability Problem

This is the simplest problem which can be considered for a SiPN. In the problem, it is checked whether a marking $\mu \in R(N^*, \mu_0)$ or not. The reachability problem can be used to express some of the analysis problems, which will be seen in the next section.

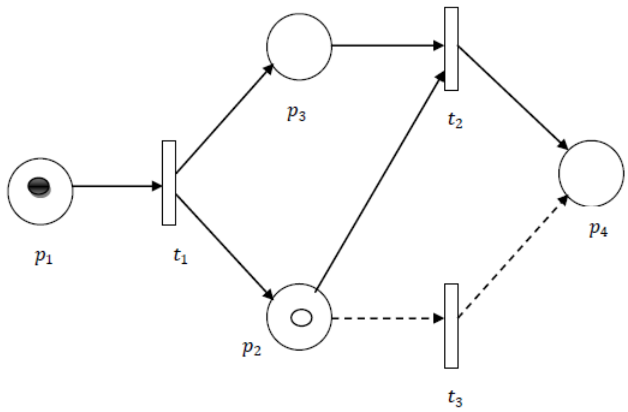


Figure 3.1: A SiPN with initial marking ((1,0,0,0),(0,1,0,0))

SiPN in **Figure 3.1** is 2-bounded, live and is not strictly conservative.

3.3 Analysis Techniques

Two major analysis techniques which provide solution mechanisms for the analysis problems mentioned in the previous section are discussed here. These techniques are:

1. Reachability Tree

2. Matrix Equations

3.3.1 Reachability Tree

A *'Reachability Tree'* represents the reachability set of a given SiPN. In a SiPN, reachability tree can be represented in one of the two forms:-

1. In the first form, the reachability tree is made by considering the complete marking $\mu = (\mu^{+}, \mu^{-})$. The change in the positive and negative tokens is shown as transitions are fired, in a single tree.

2. In the second form, two reachability trees are made, one by considering the positive marking μ^{+} and the other by taking negative marking μ^{-} into account. The advantage of this method is that, one can focus on the tokens which are important in a process and the changes taking place in the other type of tokens can be neglected. For example, consider a process modeled using SiPNs. In this model, two resources are present; the one in abundance is represented via a positive token and the other one present in scarce, is represented by a negative token. While analyzing this modeled system, the change in the distribution of negative tokens should be preferred over positive ones. Hence, here this method is useful as the reachability tree corresponding to only negative marking μ^{-} is considered while the reachability tree corresponding to the positive marking μ^{+} is not taken into account.

In **Figure 3.3**, a complete reachability tree and two reachability trees (one for positive marking and the other for negative marking) are shown for the SiPN in **Figure 3.2**. Now, analyzing this SiPN using the reachability trees, following observations can be made.

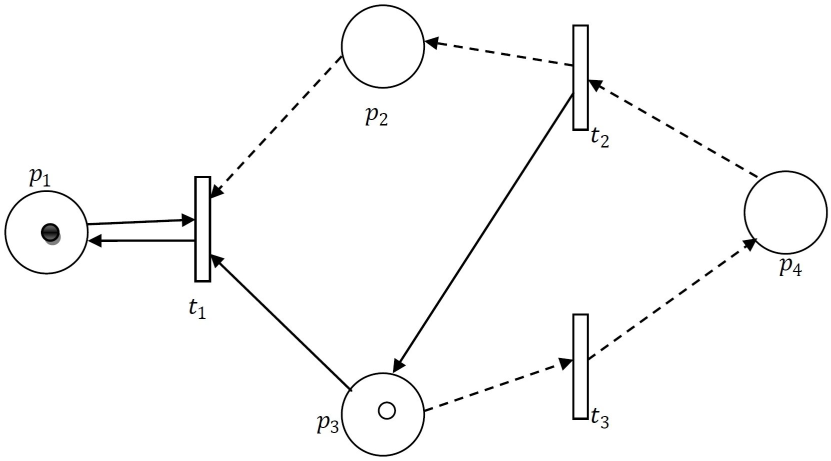
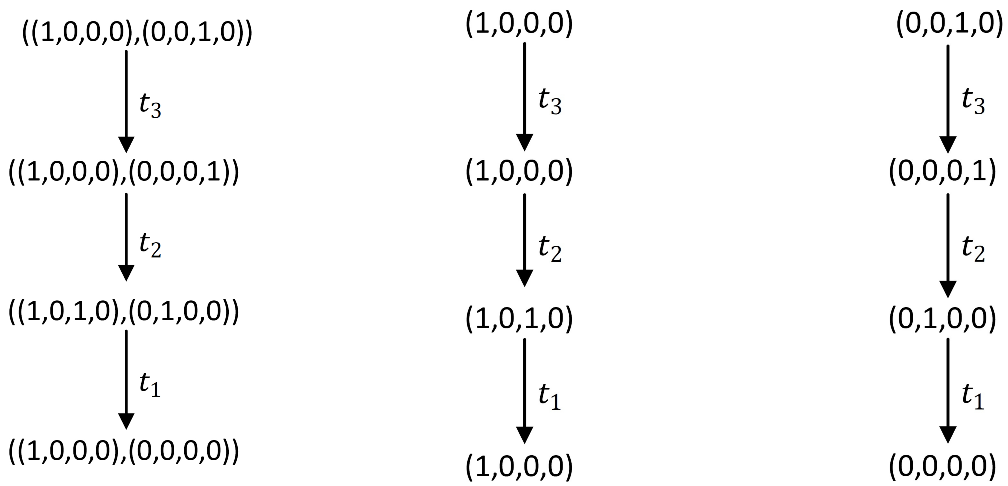


Figure 3.2: A SiPN with initial marking ((1,0,0,0),(0,0,1,0))



(a) Complete Reachability Tree

(b) Reachability Tree corresponding to positive marking

(c) Reachability Tree corresponding to negative marking

Figure 3.3: Reachability Trees for SiPN in **Figure 3.2**

- *Boundedness*: Clearly, seeing the reachability trees in **Figure 3.3**, one can see that the corresponding SiPN is safe (1-bounded).
- *Conservation*: By looking at the sum of tokens at places in markings, it is clear that SiPN is neither positively nor negatively conserved.
- *Liveness*: Since all transitions appear as arc labels in the reachability trees, therefore the SiPN is live.

Thus, the reachability problem can be used to express some of the analysis problems.

3.3.2 Matrix Equations

The second technique for analysis of SiPNs is based on matrices. Just like the incidence matrix used to represent the flow relation in a Petri net, two incidence matrices A and B are used to represent the flow relation in a SiPN. The matrix A provides information about the positive arcs in a SiPN and matrix B about the negative arcs in it. The matrices A and B are $m \times n$ matrices where $m = |T|$ and $n = |P|$.

$$A = [a_{ij}^{+}] - [a_{ij}^{-}] \quad \text{and} \quad B = [b_{ij}^{+}] - [b_{ij}^{-}]$$

where a_{ij}^{+}, a_{ij}^{-} give the number of positive arcs from transition t_i to place p_j and place p_j to transition t_i respectively and b_{ij}^{+}, b_{ij}^{-} give the number of negative arcs from transition t_i to place p_j and place p_j to transition t_i respectively.

The matrix A is termed as the *'positive arc incidence matrix'* and matrix B as the *'negative arc incidence matrix'*.

Next, the enabling conditions and firing rules of SiPNs using these incidence matrices are given.

Execution Rules for Signed Petri nets in Matrix Form

A transition t_i is *enabled* at a marking $\mu = (\mu^+, \mu^-)$ if

$$a_{ij}^- \leq \mu^+(p_j) \quad \text{and} \quad b_{ij}^- \leq \mu^-(p_j) \quad \forall\, j$$

Such an enabled transition t_i may *fire* at $\mu = (\mu^+, \mu^-)$ in one of the following cases:

1. If $a_{ij}^- \neq 0$ for some j and $b_{ij}^- = 0\ \forall j$ then $a_{ik}^+ \neq 0$ for some k.
2. If $a_{ij}^- = 0\ \forall j$ and $b_{ij}^- \neq 0$ for some j then $b_{ik}^+ \neq 0$ for some k.
3. If $a_{ij}^- \neq 0$ for some j and $b_{il}^- \neq 0$ for some l then $a_{ik}^+ \neq 0$ or $b_{im}^+ \neq 0$ for some k, m.

Let $e[i] = (0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)$ be the $1 \times m$-vector in which '1'occurs at the i^{th} position. A transition t_i is enabled at μ if

$$\mu^+ \geq e[i].A^- \quad \text{and} \quad \mu^- \geq e[i].B^-$$

where $A^- = [a_{ij}^-]$ and $B^- = [b_{ij}^-]$.

The result of firing an enabled transition t_i at a marking μ yields a new marking μ_1 given by:

$$\mu_1^+ = \mu^+ - e[i].A^- + e[i].A^+$$

$$\mu_1^- = \mu^- - e[i].B^- + e[i].B^+$$

which can be further simplified as:

$$\mu_1^+ = \mu^+ + e[i].A$$

$$\mu_1^- = \mu^- + e[i].B$$

where $A = A^+ - A^-$ and $B = B^+ - B^-$.

Next, if a sequence of transitions $\eta = t_{j_1} t_{j_2} \ldots t_{j_k}$ is fired at μ, then,

$$\begin{aligned} \mu_1^+ &= \mu^+ + (e[j_1] + e[j_2] + \ldots + e[j_k]).A \\ &= \mu^+ + f(\eta).A \end{aligned}$$

Therefore,

$$\mu_1^+ = \mu^+ + f(\eta).A \tag{3.3.1}$$

$$\mu_1^- = \mu^- + f(\eta).B \tag{3.3.2}$$

The vector $f(\eta) = e[j_1] + e[j_2] + \ldots + e[j_k]$ is called the *'firing vector'* of the sequence η. The i^{th} element of $f(\eta), f(\eta)_i$ is the number of times the transition t_{j_i} fires in the sequence η.

Result 3.1. *In a SiPN N^*, if a marking $\mu_1 = (\mu_1^+, \mu_1^-)$ is reachable from a marking $\mu = (\mu^+, \mu^-)$, then there exists a solution $x > 0$ for the marking equations for μ_1:*

$$\mu_1^+ = \mu^+ + x.A \tag{3.3.3}$$

$$\mu_1^- = \mu^- + x.B \tag{3.3.4}$$

But not conversely, i.e., if there exists a vector x which satisfies the marking equations (3.3.3) & (3.3.4)*, then it is not necessary that the marking μ_1 is reachable from μ.*

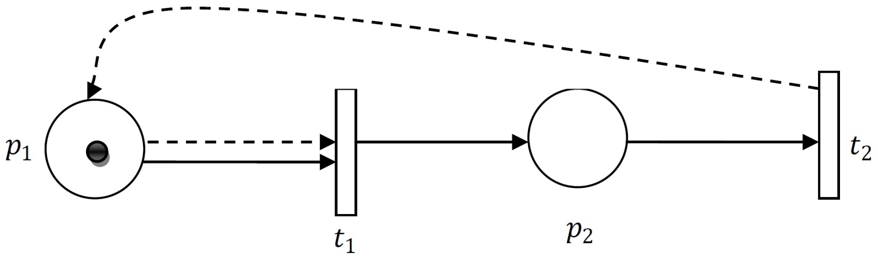


Figure 3.4: A SiPN with initial marking ((1,0),(0,0))

This can be seen through SiPN given in **Figure 3.4**. Here $\mu = \mu_0 = ((1,0),(0,0))$ and let $\mu_1 = ((0,0),(0,0))$. Using equations (3.3.3) and (3.3.4), a solution $x = (1,1)$ is obtained. But clearly, neither t_1t_2 is a firing sequence nor t_2t_1.

Next, by considering the conservation problem, it is seen how the matrix approach for analysis is useful.

Theorem 3.1. *In a SiPN N^*, if there exists a weighing vector $w = (w^+, w^-)$. Then,*

1. *$A.w^+ = 0$ if N^* is positively conserved with respect to w^+.*

2. *$B.w^- = 0$ if N^* is negatively conserved with respect to w^-.*

3. *$A.w^+ + B.w^- = 0$ if N^* is conserved with respect to $w = (w^+, w^-)$.*

Proof.

1. A SiPN N^* with initial marking μ_0 is said to be *positively conserved with respect to w^+*, if for all $\mu \in R(N^*, \mu_0)$,

$$\mu_0^+.w^+ = \mu^+.w^+ \tag{3.3.5}$$

Now using the equation (3.3.1), μ^+ can be replaced in equation (3.3.5) as:

$$\mu_0^+.w^+ = (\mu_0^+ + f(\eta).A).w^+$$

On simplifying,

$$f(\eta).A.w^+ = 0$$

This is true for all firing vectors $f(\eta)$, hence

$$A.w^+ = 0$$

2. A SiPN N^* with initial marking μ_0 is said to be *negatively conserved with respect to* w^-, if for all $\mu \in R(N^*, \mu_0)$,

$$\mu_0^-.w^- = \mu^-.w^- \tag{3.3.6}$$

Now using the equation (3.3.2), μ^- can be replaced in equation (3.3.6) as:

$$\mu_0^-.w^- = (\mu_0^- + f(\eta).B).w^-$$

On simplifying,

$$f(\eta).B.w^- = 0$$

This is true for all firing vectors $f(\eta)$, hence

$$B.w^- = 0$$

3. A SiPN N^* with initial marking μ_0 is said to be *conservative with respect*

to a weighing vector $w = (w^+, w^-)$, if for all $\mu \in R(N^*, \mu_0)$,

$$\mu_0^+.w^+ + \mu_0^-.w^- = \mu^+.w^+ + \mu^-.w^- \tag{3.3.7}$$

Now, using the equations (3.3.1) and (3.3.2), μ^+ and μ^- can be replaced in equation (3.3.7) as:

$$\mu_0^+.w^+ + \mu_0^-.w^- = (\mu_0^+ + f(\eta).A).w^+ + (\mu_0^- + f(\eta).B).w^-$$

On simplifying,

$$f(\eta).A.w^+ + f(\eta).B.w^- = 0$$

$$f(\eta).(A.w^+ + B.w^-) = 0$$

This is true for all firing vectors $f(\eta)$, hence

$$A.w^+ + B.w^- = 0$$

□

3.4 A Restaurant model

A restaurant system is modeled using SiPNs as shown in **Figure 3.5**. In the modeled system, a waiter is represented by negative tokens while positive tokens are used to represent the customers at all places except the place p_7. This place is a table counter where positive tokens gives the number of vacant tables in the restaurant. An ordered pair near a place gives the token count in the place, where first component tells about the number of positive tokens and the second component, the number of negative tokens in it.

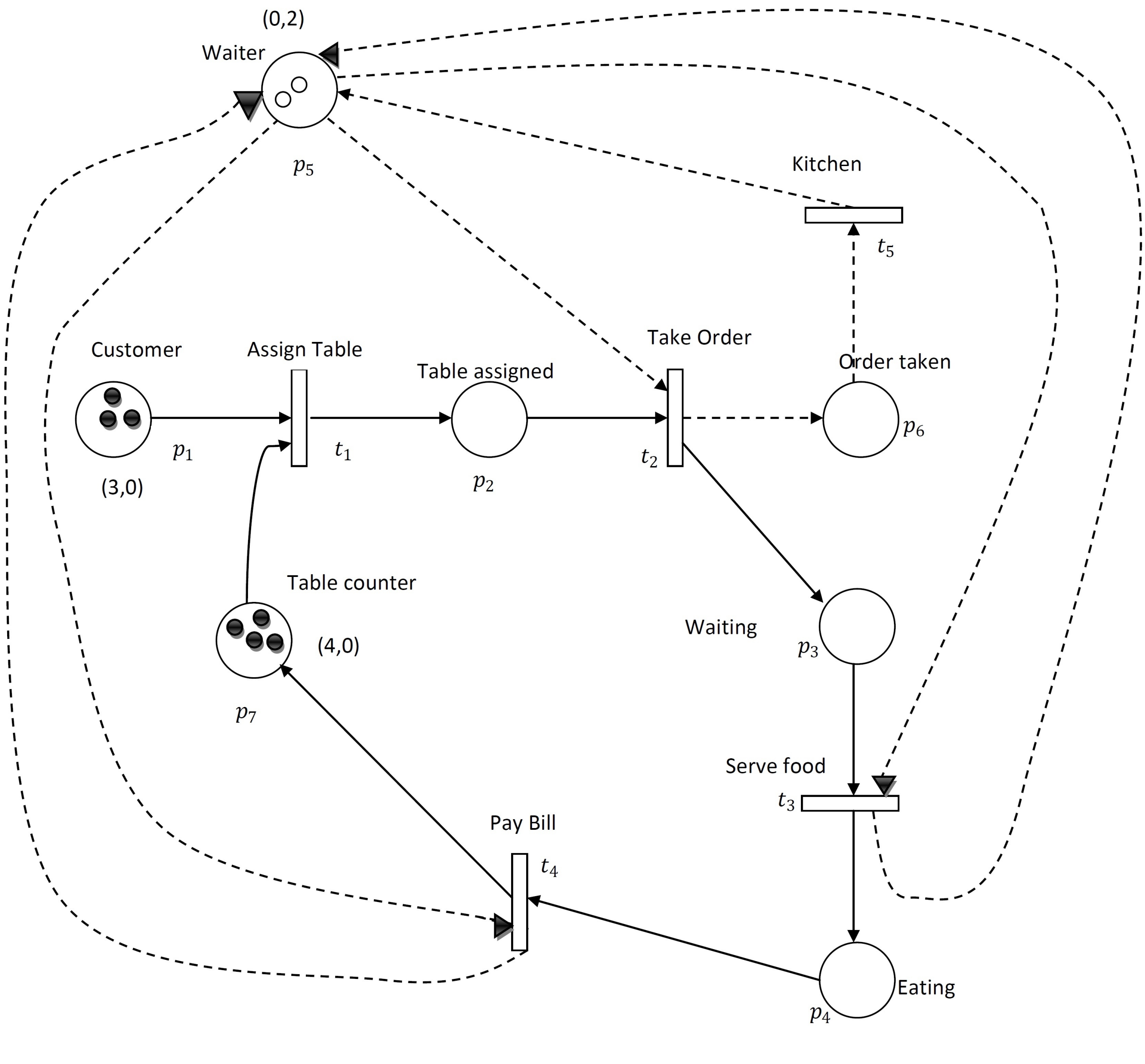


Figure 3.5: A Restaurant Model

3.4.1 Analysis of the Restaurant Model

In order to analyze this modeled system, look at the behavioral properties mentioned in section 3.2.

1. *Boundedness* :- This property is useful in identifying whether all the places are bounded or not. The places need to be positively bounded in order for the number of customers to be bounded,

since the number of customers can not exceed the number of tables present in the restaurant. Also, the SiPN should be negatively bounded as the number of waiters working in the restaurant in a particular shift are fixed and hence, all the places are negatively bounded.

2. *Conservation* :- Start with a fixed number of customers and waiters in the model. In order for all customers to leave the restaurant satisfied, i.e., after enjoying their food and not leaving the restaurant due to unavailability of food or bad ambiance, the SiPN should be positively conserved with respect to $w^+ = (1,1,1,1,1,1,k)$ where $k = 1, 4/3, 2, 4$ depending on if $\mu^+(p_7) = 4,3,2,1$ respectively. Also, the negative tokens in the model should be strictly conserved, so that the waiters don't leave the restaurant during their shift. Therefore, SiPN should be strictly negatively conserved.

3. *Liveness* :- The efficient working of the restaurant can be judged by the liveness property. If none of the transitions is dead then, the restaurant is working smoothly.

3.4.2 Advantages of using Signed Petri nets over Petri nets

- It is easier to differentiate among the tokens in a SiPN. For example, in the restaurant model, a customer and a waiter can be easily identified by the sign of tokens.

- It becomes relatively easier to make necessary modifications (in case of errors or improvements) in the modeled system by identifying the paths followed by the positive and negative tokens. Thus, the changes can be incorporated by focusing only on the paths where changes are required. For instance, to improve the restaurant model if changes are to be made in the path of a customer then, the focus is on the positive arcs only, thereby reducing the efforts required in modification of the system as compared to a model formulated using a Petri net in which all arcs are positive.

Modified Restaurant Model

In the modified restaurant model, changes are done in the path of the customer. If there are no vacant tables available at present, a customer is asked to wait and put in the waiting list. Now, in order to accomplish this task, focus is on the positive arcs in the modeled system in **Figure 3.5**, since negative arcs in the model represent movement of waiter which plays no role in the modification of customers path. If there are no vacant tables, transition t_1 can not fire. There is a new transition which can fire if a customer is willing to wait. Another transition is required which fires to assign a table to the waiting customer, when a table gets vacant. In the modified model given in **Figure 3.6**, these tasks are fulfilled by firing of transition t_6 and t_7 respectively. Thus, a modified system can be obtained, with minimum efforts.

In conclusion, the behavioral properties of SiPNs are given along with two analysis techniques. A restaurant system is modeled using SiPNs. The usage of the behavioral properties in analysis of the

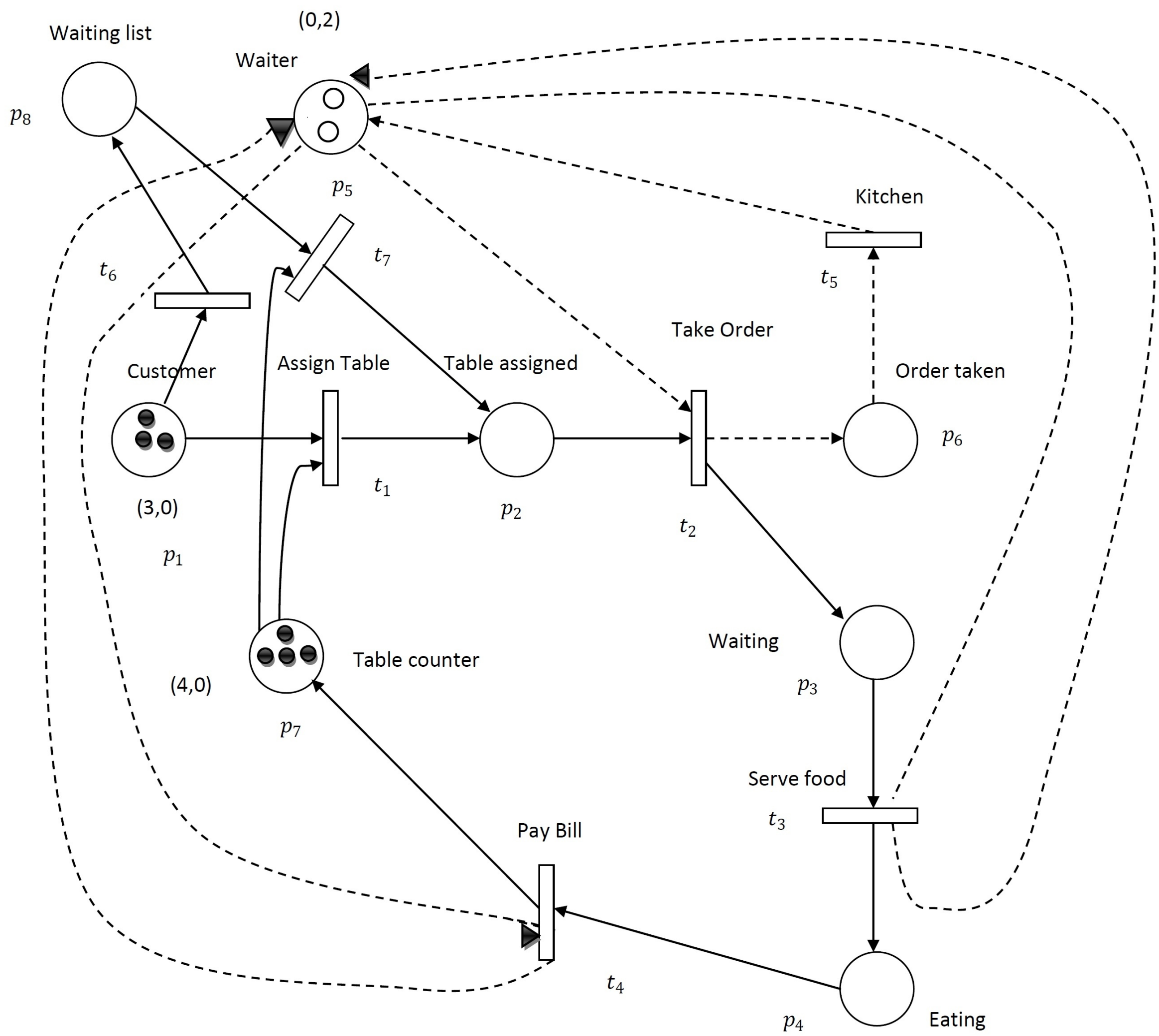


Figure 3.6: Modified Restaurant Model

modeled system are discussed. Some of the benefits of using SiPNs over Petri nets are also given, which helps in easier modification of the system. A modified restaurant model is given to show how it is easier to incorporate required changes in a system modeled using SiPNs.

References

[1] Jensen, K. (1987). *Coloured Petri nets*. Coloured Petri nets. In: Brauer, W., Reisig, W., Rozenberg, G. (eds) Petri nets: Central Models and Their Properties. ACPN 1986. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. , 254:248–299.

[2] Marsan, M. (1990). *Stochastic Petri nets: An Elementary Introduction*. Stochastic Petri nets: An elementary introduction. In: Rozenberg, G. (eds) Advances in Petri nets 1989. APN 1988. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 424:1–29.

[3] Murata, T. (1989). *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, 77(4):541–580.

[4] Peterson, J. (1981). **Petri net theory and modeling of systems**. *Prentice Hall-Inc*.

[5] Reisig, W. (1991). *Petri nets and algebraic specifications*. Theoretical Computer Science, 80(1):1–34.

[6] Reisig, W. (2013). *The Synthesis Problem*. Transactions on Petri Nets and Other Models of Concurrency VII, Lecture Notes in Computer Science, Springer, Heidelberg, 7480:303–313.

[7] Valk, R. (2004). *Object Petri nets*. Lectures on Concurrency and Petri nets: Advances in Petri nets, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3098:819–848.

Chapter 4

Structural Matrices for Signed Petri Nets

In this chapter, various matrices have been introduced for SiPNs which help in identifying relationships among the transitions and places of a SiPN. The transition precedence matrix or place precedence matrix defined in the chapter are utilized in finding a directed cycle in a SiPN. Various subclasses of SiPNs are given along with characterizations of these subclasses using the matrices introduced. Ordinary SiPNs (i.e., SiPNs without multiple arcs) are considered in this chapter.

4.1 Introduction

Many authors in the field of Petri net theory and Graph theory have utilized the concept of matrices to represent Petri nets and graphs [2, 5, 6] for computer implementation. Cantrell et al. [1] used matrix reduction method for automatic verification of a model in modeling and simulation. Saha et al. [3] utilized a relation between marking and firing vectors to obtain a simple and systematic technique of analyzing Petri nets. In [4], the author considers the pattern-type reachability analysis for the elastic dinning philosophers system based on Fractal Petri nets. The operations on fractal algebra are implemented by matrix operations.

Johnson et al. [2] introduced several matrices which show the relationship between transitions and places in a Petri net. This work inspired us to introduce such matrices for SiPNs which help in identifying the relation between its transitions and places. The matrix approach to SiPNs makes it easier to develop the algorithms related to various other concepts for SiPNs proposed in the thesis.

The structural matrices so introduced can be utilized for finding directed cycles in a SiPNs. The directed cycles can then be used further for the introducing the concept of dynamical balanceness. In section 4.2.2, three different matrices are defined by different products of the adjacency matrix of a SiPN with its transpose and with itself. In fact, if all these matrices are given, the SiPN structure can be obtained after analyzing them. In section 4.3, subclasses of SiPNs are defined and it is shown how various matrices introduced in the chapter can be used to characterize these subclasses.

4.2 A Matrix Approach to Signed Petri Nets

4.2.1 Adjacency Matrix

Since a SiPN is a directed bipartite graph in which vertices (places and transitions) are connected (places with transitions and vice-versa) via two types of arcs: positive and negative, one can associate an adjacency matrix with a SiPN.

Define an *Adjacency Matrix*, $A = [a_{ij}]$ of a SiPN, as a square matrix of order $(m+n)$ where $|T| = m$ and $|P| = n$, with

$$a_{ij} = \begin{cases} 1^+, & \text{if there exists a directed positive arc from vertex } i \text{ to vertex } j \\ 1^-, & \text{if there exists a directed negative arc from vertex } i \text{ to vertex } j \\ 0\ , & \text{else} \end{cases} \tag{4.2.1}$$

Therefore,

$$A = \begin{bmatrix} 0 & B \\ C & 0 \end{bmatrix}$$

where $B = [b_{ij}]$ is a $m \times n$ matrix in which b_{ij} represents an arc (positive or negative), if exists, from transition t_i to place p_j and $C = [c_{ij}]$ is a $n \times m$ matrix in which c_{ij} represents an arc (positive or negative), if exists, to transition t_j from place p_i.

4.2.2 Conflict, Confluence and Precedence Matrices

In order to introduce the conflict, confluence and precedence matrices of a SiPN, first the necessary operations on the entries of the adjacency matrix are defined.

- $1^{+}.1^{+} = 1^{+}$
- $1^{-}.1^{-} = 1^{-}$
- $1^{+}.0 = 0 = 0.1^{+}$
- $1^{-}.0 = 0 = 0.1^{-}$
- $1^{+}.1^{-} = 1^{*} = 1^{-}.1^{+}$
- $\underbrace{1^{i}+1^{i}\ldots+1^{i}}_{ktimes} = k.1^{i}, \quad \text{where} \quad i \in \{+,-,*\},\ k \in \mathbb{N}$
- $\underbrace{1^{+}+1^{+}\ldots+1^{+}}_{k_1 times}+\underbrace{1^{-}+1^{-}\ldots+1^{-}}_{k_2 times}+\underbrace{1^{*}+1^{*}\ldots+1^{*}}_{k_3 times}$
  $= k_1.1^{+}+k_2.1^{-}+k_3.1^{*} \text{ where } \quad k_1,k_2,k_3 \in \mathbb{N}$

Form three different matrices by different products of the adjacency matrix of a SiPN with its transpose and with itself as given below:

$$A'A = \begin{bmatrix} C'C & 0 \\ 0 & B'B \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \tag{4.2.2}$$

$$AA' = \begin{bmatrix} BB' & 0 \\ 0 & CC' \end{bmatrix} = \begin{bmatrix} \delta & 0 \\ 0 & \gamma \end{bmatrix} \tag{4.2.3}$$

$$AA = \begin{bmatrix} BC & 0 \\ 0 & CB \end{bmatrix} = \begin{bmatrix} \phi & 0 \\ 0 & \psi \end{bmatrix} \tag{4.2.4}$$

Any entry of the matrices defined above gives two information: Magnitude and Symbol. The magnitude of an entry helps in identification of cardinality while, the symbol helps to specify sign of arcs which is possible. For any entry $x = k.1^i$, $i \in \{+,-,*\}$, $k \in \mathbb{N}$, k is called as *'magnitude'* of x, denoted by $||x||$. $i \in \{+,-,*\}$ is called the *'symbol'* of a non-zero element x, which appears in the superscript of 1 and is denoted by $S(x)$. If $S(x) = +$, it represents a positive arc, if $S(x) = -$, it represents a negative arc. What $S(x) = *$ represents will be clear later in this section.

If $x = k_1.1^+ + k_2.1^- + k_3.1^*$, $k_1, k_2, k_3 \in \mathbb{N}$, then the magnitude of x is given by $||x|| = k_1 + k_2 + k_3$. $S(x)$ is not defined if $k_i, k_j \neq 0$ for $i \neq j$ or if $x = 0$.

Definition 4.1. ***Transition-Conflict Matrix***
An $m \times m$ symmetric matrix, $C'C = \alpha = [\alpha_{ij}]$, where $||\alpha_{ij}|| = |{}^\bullet t_i \cap {}^\bullet t_j| =$ the number of common input places of t_i and t_j and $||\alpha_{ii}|| = |{}^\bullet t_i| =$ the total number of input places of t_i.

Definition 4.2. ***Place-Conflict Matrix***
An $n \times n$ symmetric matrix, $B'B = \beta = [\beta_{ij}]$, where $||\beta_{ij}|| = |{}^\bullet p_i \cap {}^\bullet p_j| =$ the number of common input transitions of p_i and p_j and $||\beta_{ii}|| = |{}^\bullet p_i| =$ the total number of input transitions of p_i.

Definition 4.3. ***Transition-Confluence Matrix***
An $m \times m$ symmetric matrix, $BB' = \delta = [\delta_{ij}]$, where $||\delta_{ij}|| = |t_i^\bullet \cap t_j^\bullet| =$ the number of common output places of t_i and t_j and $||\delta_{ii}|| = |t_i^\bullet| =$ the total number of output places of t_i.

Definition 4.4. ***Place-Confluence Matrix***
An $n \times n$ symmetric matrix, $CC' = \gamma = [\gamma_{ij}]$, where $||\gamma_{ij}|| = |p_i^\bullet \cap p_j^\bullet| =$ the number of common output transitions of p_i and p_j and $||\gamma_{ii}|| = |p_i^\bullet| =$ the total number of output transitions of p_i.

Definition 4.5. ***Transition-Precedence Matrix***

It is an $m \times m$ matrix given by, $BC = \phi = [\phi_{ij}]$, where $||\phi_{ij}|| = |t_i^\bullet \cap {}^\bullet t_j| =$ the number of places that are in $t_i^\bullet$ as well as in ${}^\bullet t_j$ and $||\phi_{ii}|| = |t_i^\bullet \cap {}^\bullet t_i| =$ the number of self-loops at t_i.

Definition 4.6. ***Place-Precedence Matrix***

It is an $n \times n$ matrix given by, $CB = \psi = [\psi_{ij}]$, where $||\psi_{ij}|| = |p_i^\bullet \cap {}^\bullet p_j| =$ the number of transitions that are in $p_i^\bullet$ as well as in ${}^\bullet p_j$ and $||\psi_{ii}|| = |p_i^\bullet \cap {}^\bullet p_i| =$ the number of self-loops at p_i.

Consider a SiPN given in **Figure 4.1**. The adjacency matrix for the

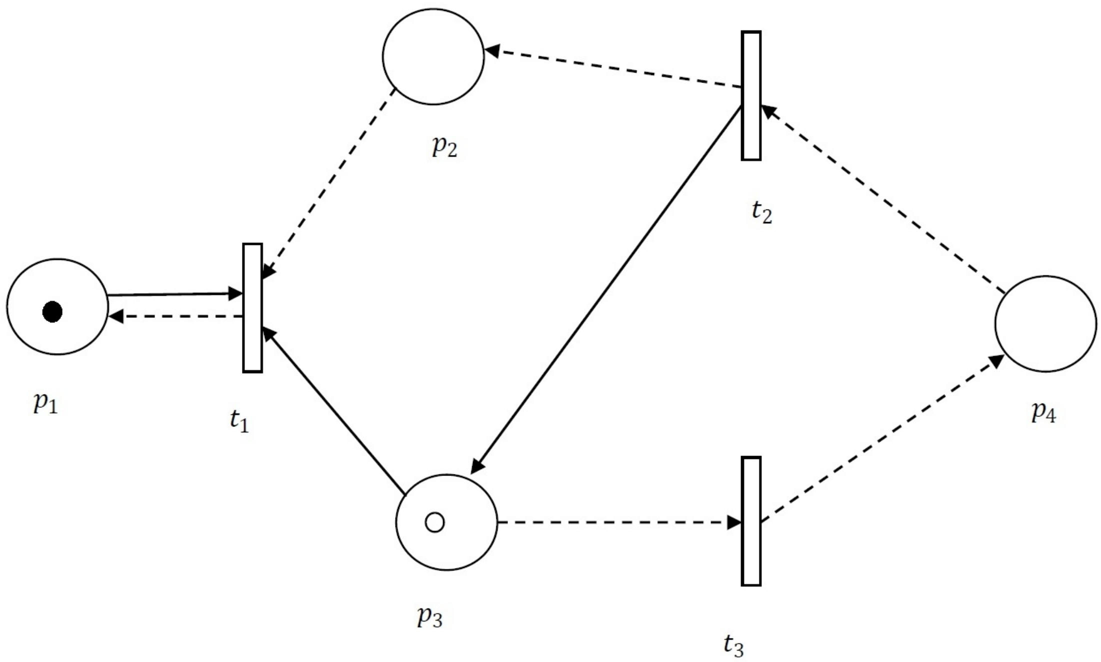


Figure 4.1: A SiPN with initial marking ((1,0,0,0),(0,0,1,0))

SiPN is given by:

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

Now, using the adjacency matrix A, the above mentioned six matrices for the given SiPN in **Figure 4.1** are formed and the following observations have been made.

1. First consider the product $A'A$, whose sub-matrices give the transition-conflict and place-conflict matrices which are as under:

$$\alpha = C'C = \begin{bmatrix} 2.1^+ + 1^- & 0 & 1^* \\ 0 & 1^- & 0 \\ 1^* & 0 & 1^- \end{bmatrix} \quad , \quad \beta = B'B = \begin{bmatrix} 1^- & 0 & 0 & 0 \\ 0 & 1^- & 1^* & 0 \\ 0 & 1^* & 1^+ & 0 \\ 0 & 0 & 0 & 1^- \end{bmatrix}$$

   Looking at the transition-conflict matrix α and place-conflict matrix β, one can infer the following:

   (a) Since $\alpha_{11} = 2.1^+ + 1^-$, it can be concluded that the transition t_1 has 3 input arcs, two positive and one of negative sign.

   (b) $\alpha_{13} = 1^*$ means that there exists a common input place of t_1 and t_3. Since $S(\alpha_{13}) = S(1^*) = *$, this implies that one of the arc from the common input place to t_1 or t_3 is positive and the other is negative.

(c) $\alpha_{22} = \alpha_{33} = 1^-$ shows that both t_2 and t_3 have one input arc. Also $S(\alpha_{22}) = S(\alpha_{33}) = S(1^-) = -$, which means that the input arc is negative in sign.

(d) $\beta_{11} = \beta_{22} = \beta_{44} = 1^-$ means that places p_1, p_2 and p_4 each have one input arc which is negatively signed.

(e) Since $\beta_{23} = 1^*$, there exists a common input transition of p_2 and p_3 and one of these input arcs from the common transition is positive and the other one is negative.

2. Next, look at the product AA', whose sub-matrices give the transition-confluence and place-confluence matrices which are as:

$$\delta = BB' = \begin{bmatrix} 1^- & 0 & 0 \\ 0 & 1^+ + 1^- & 0 \\ 0 & 0 & 1^- \end{bmatrix} \quad , \quad \gamma = CC' = \begin{bmatrix} 1^+ & 1^* & 1^+ & 0 \\ 1^* & 1^- & 1^* & 0 \\ 1^+ & 1^* & 1^+ + 1^- & 0 \\ 0 & 0 & 0 & 1^- \end{bmatrix}$$

Similarly observing the transition-confluence matrix δ and place-confluence matrix γ, one can make the following observations:

(a) $\delta_{22} = 1^+ + 1^-$ shows that t_2 has two output arcs one of positive sign and the other is of negative sign.

(b) $\delta_{11} = \delta_{33} = 1^-$ means that t_1, t_3 each have one negative output arc.

(c) Since $\gamma_{33} = 1^{+} + 1^{-}$, it can be concluded that the place p_3 has 2 output arcs, one positive and the other is of negative sign.

(d) $\gamma_{21} = 1^{*}$ means that there exists a common output transition of p_1 and p_2 while the symbol $*$ implies that one of the arc to the common output transition is positive and the other is negative.

3. Lastly, check the product AA. The sub-matrices of AA give the transition-precedence and place-precedence matrices which are as:

$$\phi = BC = \begin{bmatrix} 1^{*} & 0 & 0 \\ 1^{+}+1^{-} & 0 & 1^{*} \\ 0 & 1^{-} & 0 \end{bmatrix} \quad , \quad \psi = CB = \begin{bmatrix} 1^{*} & 0 & 0 & 0 \\ 1^{-} & 0 & 0 & 0 \\ 1^{*} & 0 & 0 & 1^{-} \\ 0 & 1^{-} & 1^{*} & 0 \end{bmatrix}$$

Observing the transition-precedence matrix ϕ and place-precedence matrix ψ, the following observations can be concluded:

(a) $\phi_{11} = 1^{*}$ means that transition t_1 has a self loop. The symbol $*$ implies that the self loop is of negative sign, i.e., one arc of self loop is positive while the other is negative.

(b) $\phi_{12} = 1^{+} + 1^{-}$ means that there exists a common place which is output of t_1 and input of t_2. One of the arc is positive and the other is negative.

(c) $\phi_{23} = 1^{*}$ shows that a common place is output of t_2 and input of t_3 with the arcs of opposite sign.

(d) $\psi_{12} = 1^-$ means that a transition exist which is output of p_1 and input of p_2, with each arc of negative sign.

Remark 4.1. *Since, the proposed work is based on finding various structural matrices from the product of adjacency matrix with itself and with its transpose, therefore, the time complexity of the work is same as the time complexity of the matrix multiplication, which in this case is, $\mathscr{O}(N^3)$, where $N = m+n$.*

The space complexity will be the complexity required to store a matrix, which is, $\mathscr{O}(n^2)$, where n is the order of the matrix.

4.3 Subclasses of Signed Petri net

In Petri net theory, various subclasses of Petri nets have been defined. In this section, such subclasses for a SiPNs have been introduced and it is shown how the matrices defined in the previous section can be used to characterize these subclasses.

Definition 4.7. ***Signed State Machine***
A SiPN is called a 'signed state machine' *if each transition is restricted to have only one input and one output arc, i.e., $|^{\bullet}t| = |t^{\bullet}| = 1 \ \forall \ t \ \in T$ and both the input and output arcs are of same sign.*

Theorem 4.1. *1. A SiPN N^* is a signed state machine iff α_{ii}, $\delta_{ii} \in \{1^+, 1^-\}$ with $\alpha_{ii} = \delta_{ii} \ \forall \ i$.*

2. If a SiPN N^* is a signed state machine then both the place-conflict matrix, β and the place-confluence matrix, γ are diagonal matrices and the corresponding diagonal entries of transition-conflict matrix, α and transition-confluence matrix, δ are same i.e., $\alpha_{ii} = \delta_{ii} \ \forall \ i$.

3. *If both β and γ are diagonal matrices and $\alpha_{ii} = \delta_{ii}\ \forall\ i$, then either N^* is a signed state machine or there exists a transition t such that $|^\bullet t| = 0$ or $|t^\bullet| = 0$.*

Proof. 1. Let $\alpha_{ii}, \delta_{ii} \in \{1^+, 1^-\}\ \forall\, i$. Then, $||\alpha_{ii}|| = ||\delta_{ii}|| = 1\ \forall i$. Therefore, $|^\bullet t_i| = |t_i^\bullet| = 1\ \forall\ t$. Next, $\alpha_{ii} = \delta_{ii}$, which implies that the superscript of 1 is same, i.e., the input and output arcs of any transition t_i are of same sign. Thus, N^* is a signed state machine.

For the converse part, let N^* be a signed state machine. By definition, $|^\bullet t_i| = |t_i^\bullet| = 1\ \forall t_i \in T$, therefore, $||\alpha_{ii}|| = ||\delta_{ii}|| = 1$. Also, input and output arcs of a transition are of same sign, thus the superscript of 1 in α_{ii}, δ_{ii} should be same for all i resulting in $\alpha_{ii} = \delta_{ii}$. Note that $\alpha_{ii}, \delta_{ii} \neq 1^*$, since then input and output arcs of transition t_i are of opposite sign. Therefore, $\alpha_{ii}, \delta_{ii} \in \{1^+, 1^-\}$.

2. N^* is a signed state machine, therefore for each transition t, $|^\bullet t| = |t^\bullet| = 1$. Thus $t \notin {^\bullet p_i} \cap {^\bullet p_j}\ for\ i \neq j$ otherwise $p_i, p_j \in t^\bullet$ which is a contradiction to the definition of a signed state machine. Hence, $|^\bullet p_i \cap {^\bullet p_j}| = 0$ for $i \neq j$, which implies $\beta_{ij} = 0$ for $i \neq j$. Likewise, $t \notin p_i^\bullet \cap p_j^\bullet\ for\ i \neq j$. Hence, $|p_i^\bullet \cap p_j^\bullet| = 0$ for $i \neq j$, implying $\gamma_{ij} = 0$ for $i \neq j$. Thus, both β and γ are diagonal matrices. Since the input and output arcs of a transition are of same sign, therefore, $\alpha_{ii} = \delta_{ii}\ \forall\ i$.

3. If both β and γ are diagonal matrices, then $|^\bullet p_i \cap {^\bullet p_j}| = 0$ and $|p_i^\bullet \cap p_j^\bullet| = 0$ for $i \neq j$. Thus for any transition $t, |^\bullet t| = |t^\bullet| \leq 1$. Also, $\alpha_{ii} = \delta_{ii}\ \forall\ i$ implies that the input and output arcs of a transition are of same sign. Thus, either N^* is a signed state machine or $\exists$ a transition t such that either $|^\bullet t| = 0$ or $|t^\bullet| = 0$.

$\square$

Definition 4.8. ***Signed Marked Graph***

A SiPN is called a 'signed marked graph' *if each place is restricted to have only one input and output arc, i.e.,* $|^{\bullet}p| = |p^{\bullet}| = 1 \; \forall \; p \; \in P$ *and both the input and output arcs are of same sign.*

Theorem 4.2. *1. A SiPN N^* is a signed marked graph iff $\beta_{ii}, \gamma_{ii} \in \{1^+, 1^-\}$ with $\beta_{ii} = \gamma_{ii} \; \forall \; i$.*

2. *If a SiPN N^* is a signed marked graph then both the transition-conflict matrix, α and the transition-confluence matrix, δ are diagonal matrices and the corresponding diagonal entries of place-conflict matrix, β and place-confluence matrix, γ are same i.e., $\beta_{ii} = \gamma_{ii} \; \forall \; i$.*

3. *If both α and δ are diagonal matrices and $\beta_{ii} = \gamma_{ii} \; \forall \; i$, then either N^* is a signed marked graph or there exists a place p such that $|^{\bullet}p| = 0$ or $|p^{\bullet}| = 0$.*

Proof. 1. Let $\beta_{ii}, \gamma_{ii} \in \{1^+, 1^-\} \; \forall \; i$. Then, $||\beta_{ii}|| = ||\gamma_{ii}|| = 1 \; \forall \; i$. Therefore, $|^{\bullet}p_i| = |p_i^{\bullet}| = 1 \; \forall \; p_i \in P$. Next, $\beta_{ii} = \gamma_{ii}$, which implies that the superscript of 1 is same, i.e., the input and output arcs of any place p are of same sign. Thus, N^* is a signed marked graph.

The converse part follows from the definition of a signed marked graph.

2. If N^* is a signed marked graph, then for each place p, $|^{\bullet}p| = |p^{\bullet}| = 1$. Thus $p \notin {}^{\bullet}t_i \cap {}^{\bullet}t_j \; for \; i \neq j$ otherwise $t_i, t_j \in p^{\bullet}$ which is a contradiction to the definition of a signed marked graph. Hence, $|^{\bullet}t_i \cap {}^{\bullet}t_j| = 0$ for $i \neq j$. Likewise, $p \notin t_i^{\bullet} \cap t_j^{\bullet}$ for $i \neq j$. Hence, $|t_i^{\bullet} \cap t_j^{\bullet}| = 0$ for $i \neq j$. Thus, $\alpha_{ij} = \delta_{ij} = 0 \; \forall i \neq j$ making both α and δ diagonal matrices. Next, by definition of signed marked graph $\beta_{ii} = \gamma_{ii} \; \forall \; i$.

3. If both α and δ are diagonal matrices, then $|^\bullet t_i \cap {}^\bullet t_j| = 0$ and $|t_i^\bullet \cap t_j^\bullet| = 0$ for $i \neq j$. Thus, for any place p, $|^\bullet p| = |p^\bullet| \leq 1$. Also, $\beta_{ii} = \gamma_{ii}\ \forall\ i$ which implies input and output arcs of a place are of same sign. Thus, either N^* is a signed marked graph or $\exists\ p$ such that either $|^\bullet p| = 0$ or $|p^\bullet| = 0$.

□

Definition 4.9. ***Signed Free-Choice net***

A SiPN is called a 'signed free-choice net' *if*

$$p_i^\bullet \cap p_j^\bullet \neq \emptyset \implies p_i^\bullet = p_j^\bullet \quad \& \quad |p_i^\bullet| = |p_j^\bullet| = 1$$

Remark 4.2. *Define sets Z and Z' as:*

$$Z = \{k_1.1^+ + k_2.1^- \ :\ k_1, k_2\ \in\ \mathbb{N}\}$$

$$Z' = Z \backslash \{1^+, 1^-, 0\}$$

Theorem 4.3. *1. A SiPN N^* is a signed free-choice net iff the following holds:-*

If $\alpha_{ii}\ \in\ Z'$, then $\alpha_{ij} = 0\ \forall\ i \neq j$. (If $\alpha_{ii} = 1^+$ or 1^-, then, $\alpha_{ij} \in \{0, 1^+, 1^-, 1^*\}$)

2. A SiPN N^* is a signed free-choice net iff $\gamma_{ij} \neq 0$ for some i and j implies $||\gamma_{ij}|| = ||\gamma_{ii}|| = ||\gamma_{jj}|| = 1$.

Proof. 1. Suppose N^* is a signed free-choice net. If $p_i^\bullet \cap p_j^\bullet \neq \emptyset$, then $p_i^\bullet = p_j^\bullet = \{t\}$ for some $t \in T$. Now $\alpha_{ii} \in Z'$ implies $\alpha_{ii} = k_1.1^+ + k_2.1^-$ for some non-zero $k_1, k_2 \in \mathbb{N}$ which further implies $||\alpha_{ii}|| = k_1 + k_2 = |^\bullet t_i|$. Let $p_1, p_2, \ldots p_{k_1+k_2}\ \in {}^\bullet t_i$, then, $t_i \in p_1^\bullet \cap p_2^\bullet \cap \ldots \cap p_{k_1+k_2}^\bullet$, i.e., $\cap_{i=1}^{k_1+k_2} p_i^\bullet \neq \emptyset$. Then, $p_1^\bullet = p_2^\bullet = \ldots = p_{k_1+k_2}^\bullet = \{t_i\}$, i.e., for every $p\ \in$

${}^{\bullet}t_i$, $p^{\bullet} = \{t_i\}$. Therefore, $||\alpha_{ij}|| = |{}^{\bullet}t_i \cap {}^{\bullet}t_j| = 0$, since any p in ${}^{\bullet}t_i$ cannot be in ${}^{\bullet}t_j$, otherwise $\{t_i, t_j\}$ would be contained in $p^{\bullet} = \{t_i\}$. Therefore, if $\alpha_{ii} \in Z'$, then $\alpha_{ij} = 0 \;\forall\; i \neq j$. Note that $||\alpha_{ij}|| \leq ||\alpha_{ii}|| \;\forall\; i, j$ (since ${}^{\bullet}t_i \cap {}^{\bullet}t_j \subseteq {}^{\bullet}t_i$). Thus, if $\alpha_{ii} \in \{1^+, 1^-\}$ then, $\alpha_{ij} \in \{0, 1^+, 1^-, 1^*\}$.

Conversely, suppose that $\alpha_{ii} \in Z'$, then $\alpha_{ij} = 0 \;\forall\; i \neq j$ i.e., if $|{}^{\bullet}t_i| > 1$, then $|{}^{\bullet}t_i \cap {}^{\bullet}t_j| = 0$. It is to be shown that N^* is a signed free-choice net. For this, assume that $p_i^{\bullet} \cap p_j^{\bullet} \neq \emptyset$ for $i \neq j$. Let $t_k, t_l \in p_i^{\bullet} \cap p_j^{\bullet}$. Then, ${}^{\bullet}t_k$ contains $\{p_i, p_j\}$ but so does ${}^{\bullet}t_l$, a contradiction (since $\alpha_{ij} = 0$). Thus, $p_i^{\bullet} \cap p_j^{\bullet}$ can contain at most one transition and $p_i^{\bullet} = p_j^{\bullet} = t$ for some $t \in T$. Hence, N^* is a signed free-choice net.

2. Suppose N^* is a signed free-choice net. If $\gamma_{ij} \neq 0$, then $|p_i^{\bullet} \cap p_j^{\bullet}| \neq 0 \;\forall i \neq j$ i.e., $p_i^{\bullet} \cap p_j^{\bullet} \neq \emptyset$. Then, $p_i^{\bullet} = p_j^{\bullet}$ and $|p_i^{\bullet}| = 1 = |p_j^{\bullet}|$ (Since, N^* is a signed free-choice net). Therefore, $||\gamma_{ii}|| = ||\gamma_{jj}|| = 1$. Now, $||\gamma_{ij}|| = |p_i^{\bullet} \cap p_j^{\bullet}| \neq 0$. Also, $|p_i^{\bullet} \cap p_j^{\bullet}| \leq |p_i^{\bullet}|$ (As $p_i^{\bullet} \cap p_j^{\bullet} \subseteq p_i^{\bullet}$). So, $|p_i^{\bullet} \cap p_j^{\bullet}| = 1 = ||\gamma_{ij}||$. Hence, $||\gamma_{ij}|| = ||\gamma_{ii}|| = ||\gamma_{jj}|| = 1$.

   Conversely, it is to be shown that N^* is a signed free-choice net. Let $p_i^{\bullet} \cap p_j^{\bullet} \neq \emptyset$, therefore, $|p_i^{\bullet} \cap p_j^{\bullet}| \neq 0$. By hypothesis, $||\gamma_{ij}|| = ||\gamma_{ii}|| = ||\gamma_{jj}|| = 1$. Therefore, $|p_i^{\bullet}| = |p_j^{\bullet}| = |p_i^{\bullet} \cap p_j^{\bullet}| = 1$. Let $p_i^{\bullet} \cap p_j^{\bullet} = \{t\}$ for some $t \in T$. Therefore, $p_i^{\bullet} \cap p_j^{\bullet} = p_i^{\bullet} = p_j^{\bullet} = \{t\}$ (As $p_i^{\bullet} \cap p_j^{\bullet} \subseteq p_i^{\bullet}$ & $p_i^{\bullet} \cap p_j^{\bullet} \subseteq p_j^{\bullet}$). Hence, N^* is a signed free-choice net.

$\square$

Definition 4.10. ***Extended Signed Free-Choice net***

A SiPN is called an 'extended signed free-choice net' *if*

$$p_i^{\bullet} \cap p_j^{\bullet} \neq \emptyset \implies p_i^{\bullet} = p_j^{\bullet}$$

Theorem 4.4. *A SiPN N^* is an extended signed free-choice net iff $\gamma_{ij} \neq 0$ for some i and j implies $||\gamma_{ii}|| = ||\gamma_{jj}|| = ||\gamma_{ji}||$.*

Proof. Suppose N^* is an extended signed free-choice net. Let $\gamma_{ij} \neq 0$ therefore, $|p_i^\bullet \cap p_j^\bullet| \neq 0$, i.e., $p_i^\bullet \cap p_j^\bullet \neq \emptyset$ which implies $p_i^\bullet = p_j^\bullet$ so, $p_i^\bullet = p_i^\bullet \cap p_j^\bullet$. Thus, $||\gamma_{ii}|| = ||\gamma_{jj}|| = ||\gamma_{ji}||$.

Conversely, suppose that $\gamma_{ij} \neq 0$. Then, $||\gamma_{ii}|| = ||\gamma_{jj}|| = ||\gamma_{ji}||$, i.e., $p_i^\bullet \cap p_j^\bullet \neq \emptyset$ which implies $||\gamma_{ij}|| = ||\gamma_{ii}||$. Therefore, $|p_i^\bullet \cap p_j^\bullet| = |p_i^\bullet|$ then, $p_i^\bullet = p_i^\bullet \cap p_j^\bullet$. Likewise, $||\gamma_{ij}|| = ||\gamma_{jj}||$ so that $p_j^\bullet = p_i^\bullet \cap p_j^\bullet$ implying $p_i^\bullet = p_j^\bullet$. Thus, N^* is an extended signed free-choice net. □

Definition 4.11. ***Signed Simple net***

A SiPN is called a 'signed simple net' *if*

$$p_i^\bullet \cap p_j^\bullet \neq \emptyset \implies \text{either} \quad p_i^\bullet \subset p_j^\bullet \quad \text{or} \quad p_j^\bullet \subset p_i^\bullet$$

Theorem 4.5. *A SiPN N^* is a signed simple net iff $\gamma_{ij} \neq 0$ for some i and j, then, either $||\gamma_{ij}|| = ||\gamma_{ii}||$ or $||\gamma_{ij}|| = ||\gamma_{jj}||$.*

Proof. Suppose that N^* is a signed simple net. If $\gamma_{ij} \neq 0$ then, $p_i^\bullet \cap p_j^\bullet \neq \emptyset$ which means, either $p_i^\bullet \subset p_j^\bullet$ or $p_j^\bullet \subset p_i^\bullet$, i.e., either $p_i^\bullet \cap p_j^\bullet = p_i^\bullet$ or $p_i^\bullet \cap p_j^\bullet = p_j^\bullet$. Therefore, either $||\gamma_{ij}|| = ||\gamma_{ii}||$ or $||\gamma_{ij}|| = ||\gamma_{jj}||$.

The converse can be obtained by reversing the steps. □

Remark 4.3. *The space complexity of finding any of the subclasses of SiPNs is same as the space complexity for storing a matrix, i.e., $\mathcal{O}(n^2)$, where n is the order of the matrix.*

The time complexity of finding signed state machine and signed marked graph is $\mathcal{O}(n)$ where n is the order of the matrix, since in these cases only the diagonal entries of the matrices are considered. The time complexity of finding *rest of the subclasses is $\mathscr{O}(n^2)$, where n is the order of the matrix because in these cases all the elements of the concerned matrix are considered.*

4.4 Finding Directed Cycle in Signed Petri nets

One can to find a directed cycle in a SiPN N^*, with the help of transition precedence matrix, ϕ or place precedence matrices, ψ. Any of the matrix ϕ or ψ can be used, depending on whether the identification of transitions/places in the cycle is to be done. Below is the procedure for finding the directed cycle using transition precedence matrix, ϕ.

Procedure

1. Find the adjacency matrix A, of order $(m+n)$, where $m = |T|$ and $n = |P|$.
2. Find the transition precedence matrix, ϕ .
3. Find all ϕ_{ij} such that $\phi_{ij} \neq 0$ for some pair $(i, j),\ i, j \in \mathbb{N}$ and $i \neq j$.
4. Form the set $\wp = \{(i, j)|\phi_{ij} \neq 0\ i, j \in \mathbb{N}\}$.
5. Find a sequence of ordered pairs in $\wp$ of the type $(i,i+1),(i+1,i+2),(i+2,i+3),\ldots(i+k,i)$ for $k,i \in \mathbb{N}$ & $k > 0$.
6. If such a sequence exists, there exists a cycle containing transitions from the set $T' = \{t_i,t_{i+1},t_{i+2},\ldots t_{i+k}\}$.
7. Finding all such sequences give a set of all directed cycles in the given SiPN.

Consider the SiPN in **Figure 4.1**, the matrix ϕ is as:

$$\phi = BC = \begin{bmatrix} 1^* & 0 & 0 \\ 1^+ + 1^- & 0 & 1^* \\ 0 & 1^- & 0 \end{bmatrix}$$

Using the above mentioned procedure, $\wp = \{(1,1),(2,1),(2,3),(3,2)\}$. Next, a sequence of ordered pairs in the set $\wp$ is $\{(2,3),(3,2)\}$. Therefore, there exists a cycle containing transitions from the given set, i.e., $T' = \{t_2, t_3\}$. This can be verified from the **Figure 4.1**, since there exists a cycle, $t_2 p_3 t_3 p_4 t_2$, containing the transitions t_2 and t_3.

If instead of transition precedence matrix, the place precedence matrix is used, a cycle containing places p_3 and p_4 will be formed.

In concluding remark, as the graphs /images are unsuitable for computer implementation and poses problems in storage since images use more memory as compared to the text, a matrix approach for SiPNs has been introduced. Hence, using the matrices defined in this chapter the structural information of SiPNs can be stored and utilized to find relationships between transitions and places and finding a directed cycle in it. The characterizations for subclasses of SiPNs are given using these matrices.

References

[1] Cantrell, W. A., Maxwell, K. P., Petty, M. D., and Whitaker, T. S. (2020). *Matrix Reduction Verification of Extended Petri Nets*. Proceedings of the 2020 ACM Southeast Conference, Association for Computing Machinery, New York, NY, USA, ACM SE ’20:296–299.

[2] Johnson, J. and Murata, T. (1985). *Structure matrices for Petri nets and their applications*. Journal of the Franklin Institute, 319(3):299–309.

[3] Saha, B. and Bandyopadhyay, S. (1988). *Representation and analysis of Petrinets via the matrix state equation approach*. International Journal of Electronics, 65(1):1–7.

[4] Semenov, A. S. (2020). *Matrix Approach to Pattern-type Reachability Analysis of Fractal Petri nets*. 7th International Conference on Control, Decision and Information Technologies (CoDIT), 1:211–216.

[5] Staines, S. A. (2016). *Modelling Simple Network Graphs Using the Matrix Vector Transition net*. International Journal of Computers and Communications, 10:11–17.

[6] Staines, S. A. (2019). *Alternative Matrix Representation of Ordinary Petri nets*. WSEAS Transactions on Computers, 18:11–18,.

Chapter 5

Structural and Dynamical Balanceness in Signed Petri Nets

In this chapter, the notion of balanceness in signed graphs, formulated by Cartwright and Harary, to study and analyze social networks is extended to Petri nets via signed Petri nets as such a concept has not been defined for already existing Petri nets. A structurally balanced (or simply balanced) SiPN has been defined and its characterization is given. The concept of dynamical balanceness is also proposed, which is advantageous as all the signed graphs (directed or undirected) can be simulated by firing of different sequence of transitions in a single SiPN. Also, dynamics associated with a system can be easily represented using SiPNs rather than a signed graphs.

5.1 Introduction

In the field of graph theory, Harary [7] introduced a signed graph and gave the characterization for a balanced signed graph. This concept of signed graph has been used in [8] for anticipating the existence of positive and negative links in online social networks, for determining the stability of the social model using the degree of balanceness [3] and various other applications [1, 2, 4, 5, 6, 9]. The positive and negative arcs which exist in a SiPN make it possible to extend the notion of balanceness in a signed graph to a Petri net which can be used to model social networks and study them using a single SiPN rather than multiple signed graphs.

The balanceness in signed graph, formulated by Cartwright and Harary [3], can be used to study and analyze social networks. This notion of balanceness is extended to Petri nets via SiPNs as such a concept has not been defined for already existing Petri nets. A structurally balanced SiPN (or simply balanced SiPN) has been defined and its characterization is given in section 5.2.2. As a SiPN has dynamics (i.e., the movement of tokens in the places) associated with it, the concept of dynamical balanceness has been introduced in section 5.2.3. Further, using a model of all possible interactions between three individuals, it is demonstrated which kind of interactions will lead to a stable/ balanced environment. The results obtained aligns with the results of balanceness in signed graph.

The advantage of using the proposed notion over a signed graph is that a single SiPN can simulate all signed graphs by change in the marking of SiPN due to firing of a set of transitions. Also, the dynamics of SiPN helps in studying all interactions in a single SiPN structure rather than multiple signed graphs. Thus, the efforts are reduced considerably to obtain the required results as analysis of a single SiPN need to be done rather than multiple signed graphs. Also, the movement of tokens, i.e., dynamic behavior of SiPNs makes it easier for changes to be incorporated at any instant of time with ease.

5.2 Balanceness in Signed Petri nets

The *'underlying graph'* of a SiPN N^*, denoted by N^v is the graph where directions of the arcs of N^* have been removed.
A *'path (cycle)'* in a SiPN N^* is the path (cycle) in its underlying graph N^v.
A *'Pure SiPN'* is a SiPN without self loops (cycle of length 2).

5.2.1 Sign of a cycle in Signed Petri nets

The *'sign of a cycle'* in a SiPN N^* can be defined in three ways given below. In cases 1 and 2, cycles are considered in the underlying graph of the SiPNs while in case 3, directed cycles are used for evaluating the sign.

1. **With respect to arc sign**
   In this case, sign of a cycle is given by product of sign of the arcs on the cycle. e.g. the sign of cycle $p_1t_1p_3t_2p_1$ in the **Figure 5.1** is positive.

2. **With respect to vertex sign**

   In this case, sign of a cycle is given by product of sign of the vertices on the cycle. e.g. the sign of cycle $p_1t_1p_3t_2p_1$ in the **Figure 5.1** is negative.

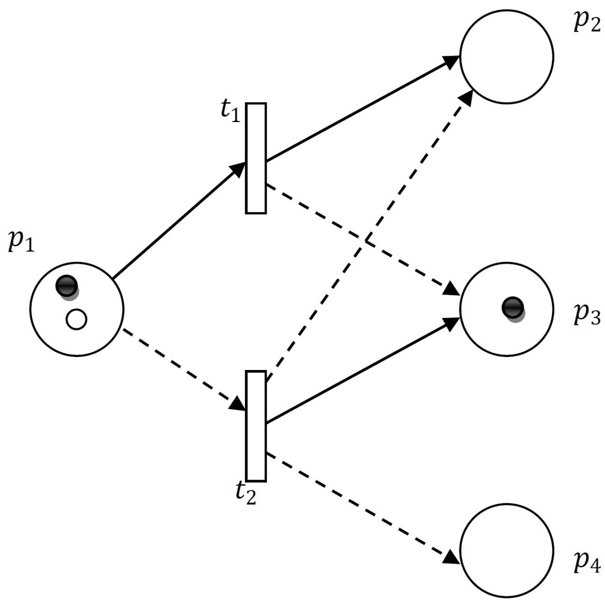


Figure 5.1: A SiPN with initial marking $((1,0,1,0),(1,0,0,0))$

3. **With respect to flow of tokens**

   In this case, directed cycles are considered.

   The sign of a cycle with respect to flow of tokens is defined as the product of sign of tokens that flow on the output arcs of the transitions lying on the cycle, when all the transitions on the cycle fire. In **Figure 5.2**, the sign of cycle $p_1t_1p_2t_2p_3t_5p_1$ is positive since, when t_1, t_2 and t_5 fire, one positive and two negative tokens flow on the output arcs of these transitions. The product of the sign of these tokens is positive. Similarly, the sign of cycle $p_1t_1p_2t_2p_3t_3p_4t_4p_1$ is negative since, when t_1,t_2,t_3 and t_4 fire, three positive and one negative token flow on the output arcs of these

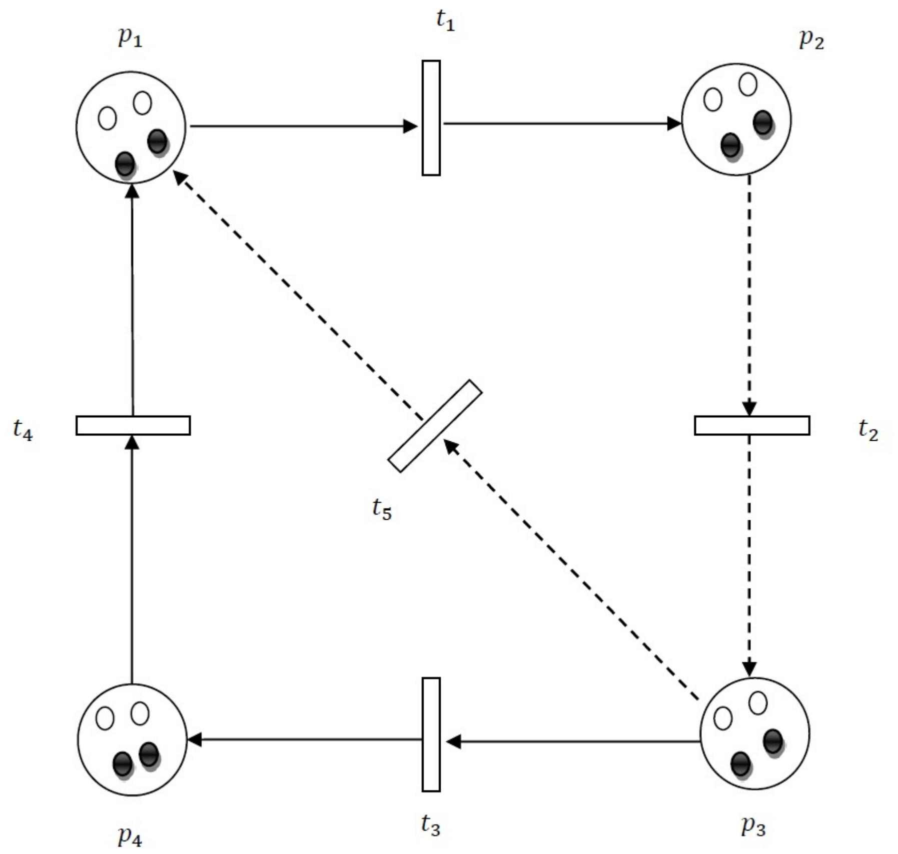


Figure 5.2: A SiPN

transitions and the product of sign of these tokens is negative, hence the sign of cycle $p_1t_1p_2t_2p_3t_3p_4t_4p_1$ w.r.t. flow of tokens is negative.

Remark 5.1. *If there exist at least one transition on a directed cycle which does not fire then, sign of such a cycle is not defined with respect to flow of tokens.*

5.2.2 Structurally Balanced Signed Petri Nets

A SiPN N^* is said to be *'Structurally balanced 'or simply 'balanced '* if all of its cycles are positive w.r.t. arc sign in the underlying graph N^ν. A SiPN without cycles in its underlying graph N^ν is balanced.

Theorem 5.1. *A partition of the set of vertices $V = (P \cup T)$ in a SiPN N^* into two disjoint subsets V_1 and V_2; one of which may be empty, such that all the arcs between vertices of the same subset are positive and all arcs between vertices of different subsets are negative is possible in the following cases:*

1. *All the arcs in N^* are positive.*

2. *All the arcs in N^* are negative.*

3. *Any place in N^* has all incoming as well as all outgoing arcs of one sign (either all positive or all negative).*

4. *Any transition in N^* has all incoming as well as all outgoing arcs of one sign (either all positive or all negative).*

Proof.

1. In this case, take $V_1 = V(= P \cup T)$ and $V_2 = \emptyset$.

2. In this case, begin with a place p_i in V and put it in subset V_1. Since only negative arcs exist among all the vertices, therefore, all the transitions connected to this place p_i must lie in the subset V_2. Now, the places connected to these transitions must lie in the subset V_1. Continuing in this manner, until all the vertices are put in either V_1 or in V_2, thus, $V_1 = P$ and $V_2 = T$.

3. WLOG, begin with a place with positive incoming and outgoing arcs and put it in the subset V_1. Since the transitions which are connected to this place are connected by positive arcs only, put these transitions in V_1 as well. Now the transitions can have both types (positive and negative) of arcs incident on them. The places connected to these transitions via positive arcs lie in subset V_1 and those connected through negative arcs lie in subset V_2. Again these places are connected by only one kind

of arcs so places in subset V_1 will have connections to transitions via positive arcs only and put those transitions in subset V_1. The places placed in subset V_2 have negative arcs through which they are connected to other transitions, which then will also lie in subset V_1. Continue this process unless all vertices (places and transitions) are exhausted.

4. In this case, proceed as in case 3. WLOG, begin with a transition having positive incoming and outgoing arcs.

□

Observation: In addition to the four cases mentioned in Theorem 5.1, such a partition of vertices into two disjoint subsets in a Pure SiPN N^* is possible in the following two cases also.

1. Any place in a pure SiPN N^* has incoming and outgoing arcs of opposite signs (i.e., if incoming arcs are positive then, outgoing arcs are negative and vice-versa).

2. Any transition in a pure SiPN N^* has incoming and outgoing arcs of opposite signs.

Corollary 5.1. *A SiPN N^* is negative iff such a partition of the set of vertices, V, is a bipartite partition.*

Proof. If N^* is negative, then such a bipartite partition exists has been proved in the Theorem 5.1 case 2.

Conversely, suppose such a bipartite partition of vertex set, V exists. Because of this partition, no arc exists within the set. Hence, no positive arcs exist resulting in an all negative SiPN. □

Theorem 5.2. *A complete SiPN N^* is balanced iff its vertex set $V = (P \cup T)$ can be partitioned into two disjoint subsets V_1 and V_2; one of which may be empty such that all arcs between vertices of the same subset are positive and all arcs between vertices of different subsets are negative.*

Proof. Suppose such a partition of the vertex set $V = P \cup T$ exists in a complete SiPN. It is enough to show that all the cycles in N^v are positive. Because of this partition, any cycle in N^v always has even number of negative arcs, if exists, thus the product of sign of arcs in the cycle is positive, making the SiPN balanced.

In order to prove the converse part, the method of contradiction is used. Suppose a complete SiPN N^* is balanced and such a partition doesn't exist. Let $|V| > 3$ (For $|V| = 2,3$ result is obvious). WLOG, assume that there exists at least one negative arc within a subset (say V_1) while trying to partition V. Let this negative arc exists between vertices v_i and v_j (place-transition pair). Since $|V| > 3$, there exist at least two more vertices v_k and v_l in V (place-transition pair). Since N^* is complete, v_k is connected to v_j and v_l is connected to v_i. Now, three cases arise:

1. If $v_k, v_l \in V_1$

   In this case, the arcs between v_i and v_l and between v_j and v_k are positive. The cycle of length four so formed by vertices v_i, v_j, v_k and v_l is negative.

2. If $v_k, v_l \in V_2$

   In this case, the arcs between v_i and v_l and between v_j and v_k are negative. The cycle of length four so formed by v_i, v_j, v_k and v_l is negative.

3. If $v_k \in V_1$ & $v_l \in V_2$

> Here, the arc between v_i and v_l is negative while that between v_j and v_k is positive. Also, the arc between v_k and v_l is negative. The cycle of length four so formed by v_i, v_j, v_k and v_l is negative.

If the partition doesn't exist on account of a positive arc across the subsets then, the result can be proved on similar lines.
In any of the cases, the negative cycle so formed will contradict the balanceness of N^*. Thus, the assumption is wrong and there always exists such a partition.

□

Corollary 5.2. *A sub-SiPN of a balanced SiPN is balanced.*

Proof. Since every cycle of a sub-SiPN is also a cycle of the given balanced SiPN and is therefore positive. Hence, sub-SiPN is also balanced. □

Theorem 5.3. *A SiPN N^* is balanced iff for each pair of distinct vertices v_1 and v_2 in the underlying graph N^ν, all paths joining v_1 and v_2 have the same sign.*

Proof. Let a balanced SiPN N^* be given. Consider any two paths A and B joining v_1 and v_2. If any common arcs which exist in these two paths are removed, a collection of arc-disjoint cycles are obtained. Each of these cycles consists of a subpath of A and a subpath of B. The cycle must be positive in sign which implies these subpaths must be of the same sign. Joining these subpaths with common arcs removed earlier leads to paths A and B having the same sign.

Conversely, let all paths joining any two distinct vertices v_1 and v_2 in N^* are of same sign. Hence, all cycles containing v_1 and v_2 must be positive. Since, v_1 and v_2 are arbitrary, all cycles in N^* are positive. Thus, N^* is balanced. □

Theorem 5.4. *A SiPN N^* is balanced iff its set of vertices $V = (P \cup T)$ can be partitioned into two disjoint subsets V_1 and V_2; one of which may be empty such that all the arcs between vertices of the same subset are positive and all arcs between vertices of different subsets are negative.*

Proof. A partition V_1 and V_2 of the set of vertices of N^* is given. N^* can be extended to form a complete SiPN. Take a pair of non-connected place and transition. If they lie in the same subset join them with a positive bi-directed arc, otherwise with a negative one. By Theorem 5.2, the complete SiPN so obtained is balanced and by its Corollary 5.2, given SiPN is balanced.

Conversely, let N^* be balanced. Consider any non-adjacent pair (place-transition) of vertices v_1 and v_2. By Theorem 5.3, all the paths joining v_1 and v_2 have the same sign. Join v_1 and v_2 by a bi-directed arc of the same sign as the sign of these paths. In this way, all the cycles thus introduced are positive, resulting in a balanced SiPN. Once, all the non-adjacent vertices are joined using this method, a complete SiPN is obtained. The result follows from Theorem 5.2. □

NOTE: The theorems 5.2-5.4 and corollary 5.1 have been formulated using the results given by Harary for a signed graph [7].

5.2.3 Dynamical Balanceness in Signed Petri nets

While modeling social interactions through SiPNs, the method of assigning sign to a cycle w.r.t. flow of token is used. Here, the input and output arcs for any transition are of same sign.

Dynamically Balanced Signed Petri nets

A SiPN N^* is said to be *'dynamically balanced'* with respect to $T' \subseteq T$, if the sign of all the directed cycles consisting of transitions from T' only, is positive with respect to flow of tokens else, the SiPN is not dynamically balanced. Consider the SiPN given in **Figure 5.2** with $T' = \{t_1, t_2, t_5\}$. The SiPN is dynamically balanced w.r.t. T' since the cycle $p_1t_1p_2t_2p_3t_5p_1$ is positive. If SiPN with $T' = \{t_1, t_2, t_5, t_4, t_3\}$ is considered, it is not dynamically balanced w.r.t. T' since out of the two cycles $p_1t_1p_2t_2p_3t_3p_4t_4p_1$ and $p_1t_1p_2t_2p_3t_5p_1$, the former is negative.

Equivalence between Balanced Signed Graphs and Dynamically Balanced Signed Petri net

Consider a SiPN N^* given in **Figure 5.3** which represents all possible interactions between three individuals represented by the places $p_1,\ p_2,\ \&\ p_3$. An interaction between p_i and p_j can be positive or negative $\forall\ i, j \in\ \{1,2,3\}, i \neq j$. All the transitions in N^* represent the initiation of an interaction between any two individuals. The positive and negative arcs in N^* indicate positive (friendly) and negative (antagonistic) interactions respectively.

The positive and negative tokens in the places are the positive and negative thoughts of the person which leads to positive and negative interactions respectively. A subset of transition set of SiPN N^*, say T' can be used to obtain a sub-SiPN from a given SiPN which contains all the transitions from T' only along with the places connected to these transitions via incident arcs of the transitions. The given sub-SiPN obtained using T' can describe a situation in signed graph, where

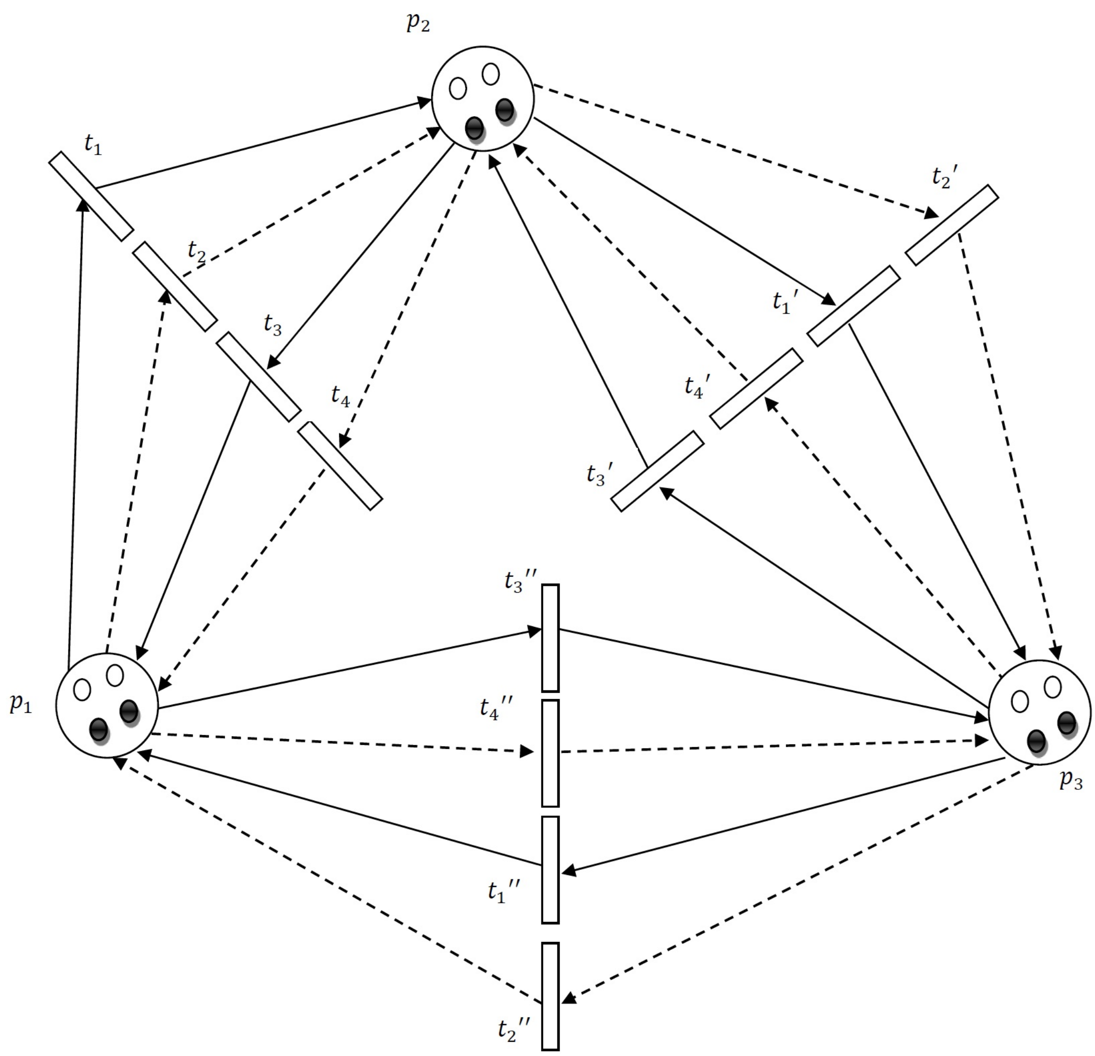


Figure 5.3: A SiPN representing three persons interactions

between three individuals some type of interactions are taking place. On changing T', other situations can be represented. Hence, using a single SiPN all the possible interactions can be represented by change of this subset of transitions.

Consider the SiPN given in Figure **Figure 5.3**. If $T' = \{t_1, t_3, t_2', t_4', t_1'', t_3''\}$, then corresponding sub-SiPN given in **Figure 5.4** is not dynamically balanced w.r.t. T', since, cycle $p_1 t_1 p_2 t_2' p_3 t_1'' p_1$ is negative in sign w.r.t. flow of tokens. The signed graph equivalent to this sub-SiPN is represented in **Figure 5.5** which by theory of structural balance for signed graphs is unbalanced and agrees with the result for SiPN . Similarly, all the possible signed graphs can be

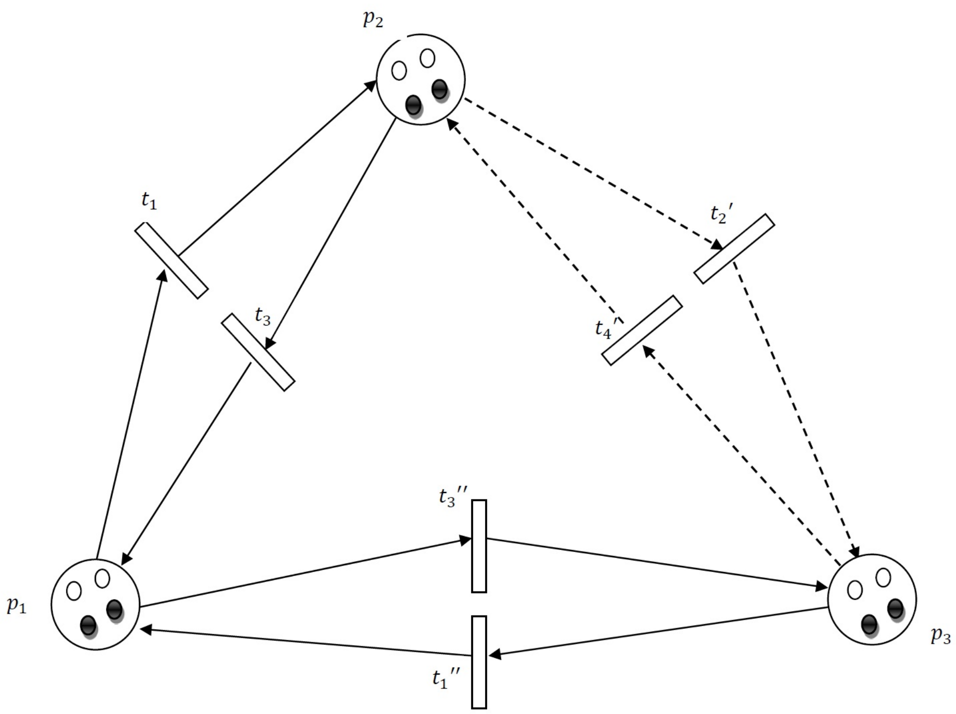


Figure 5.4: Sub-SiPN corresponding to $T' = \{t_1, t_3, t_1'', t_3'', t_2', t_4'\}$

obtained if different subsets of transition set of SiPN are considered. This is shown in the **Table 5.1**. The results in the table are obtained for

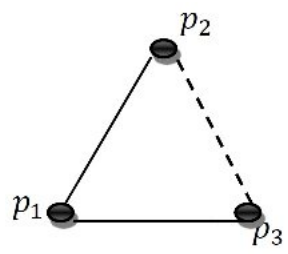
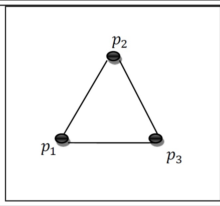
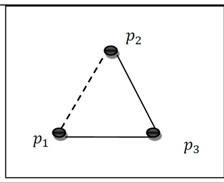
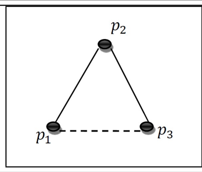
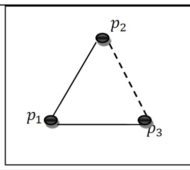
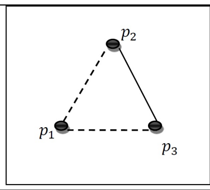
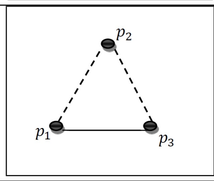
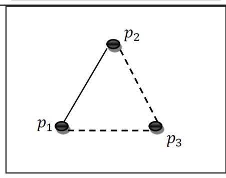
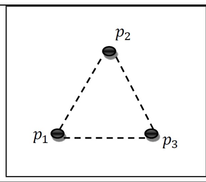


Figure 5.5: Equivalent Signed Graph

bi-directed signed graphs, however, for the same SiPN the results for uni-directed signed graphs can also be obtained.

This concept of dynamical balanceness can be extended if the number of individuals increase. As seen above, a single SiPN can simulate all signed graphs by change in the marking of SiPN due to firing of a set of transitions. It can be seen that exactly same results are obtained with less efforts. Also, the dynamics of SiPNs help in studying all interactions in a single SiPN structure rather than multiple signed graphs.

Table 5.1: Table for nature of a SiPN for a subset of transitions and the corresponding signed graph

Subset of transitions T'	Corresponding signed graph	Nature
$\{t_1, t_3, t'_1, t'_3, t''_1, t''_3\}$		Balanced
$\{t_2, t_4, t'_1, t'_3, t''_1, t''_3\}$		Not Balanced
$\{t_1, t_3, t'_1, t'_3, t''_2, t''_4\}$		Not Balanced
$\{t_1, t_3, t'_2, t'_4, t''_1, t''_3\}$		Not Balanced
$\{t_2, t_4, t'_1, t'_3, t''_2, t''_4\}$		Balanced
$\{t_2, t_4, t'_2, t'_4, t''_1, t''_3\}$		Balanced
$\{t_1, t_3, t'_2, t'_4, t''_2, t''_4\}$		Balanced
$\{t_2, t_4, t'_2, t'_4, t''_2, t''_4\}$		Not Balanced

References

[1] Akiyama, J., Avis, D., Chvátal, V., and Era, H. (1981). *Balancing signed graphs*. Discrete Applied Mathematics, 3:227–233.

[2] Aref, S. and Wilson, M. (2019). *Balance and Frustration in Signed Networks*. Journal of Complex Networks, 7(2):163–189.

[3] Cartwright, D. and Harary, F. (1956). *Structural Balance:A Generalization of Heider's Theory*. Psychological Review, 63(5):277–293.

[4] Davis, J. A. (1967). *Clustering and Structural Balance in Graphs*. Human Relations, 20:181–187.

[5] Doreian, P. and Mrvar, A. (1996). *A partitioning approach to structural balance*. Social Networks, 18(2):149–168.

[6] Frank, O. and Harary, F. (1979). *Balance in stochastic signed graphs*. Social Networks, 2:155–163.

[7] Harary, F. (1953). *On the notion of balance of a signed graph*. Michigan Math. Journal, 2:143–146.

[8] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). *Predicting Positive and Negative Links in Online Social Networks*. Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, WWW '10:641–650.

[9] Wu, B. Y. and Chen, J.-F. (2013). *Balancing a Complete Signed Graph by Editing Edges and Deleting Nodes*. Advances in

Intelligent Systems and Applications, Smart Innovation, Systems and Technologies, Springer Berlin Heidelberg, 1:79–88.

Chapter 6

Domination in Signed Petri Nets

In this chapter, domination in signed Petri nets has been introduced, as the concept of domination is not prevalent in dynamic systems. The theory of domination has been applied to find the highest and lowest ranking officials in an institute based on a certain activity, producer-consumer problem, searching of food by bees and finding similarity in research papers. A C++ code is also given for finding highest and lowest ranking officials in an institute based on a certain activity.

6.1 Introduction

The foundation of theory of domination can be traced back to the chess problem of finding the minimum number of queens required such that all the squares are either occupied or can be attacked by a queen [7]. The applications of theory of domination includes communication network problems, facility location problem, routings, etc [4, 10]. The domination in graphs and signed graphs have been well studied by various authors in different forms viz. roman domination, double domination, total domination, signed domination, signed total domination etc. [1, 2, 3, 5, 6, 8, 9, 11]. However, such a concept does not exist for dynamic systems. Therefore, the concept of domination has been introduced in SiPNs and it is shown how the proposed concept is beneficial using the applications of finding the highest and lowest ranking officials in an institute based on a certain activity, producer-consumer problem, searching of food by bees and finding similarity in research papers. A C++ code is also given for finding highest and lowest ranking officials in an institute based on a certain activity.

6.2 Domination in Signed Petri Nets

In this chapter, an ordinary SiPN (without multiple arcs) is considered unless stated otherwise.

Definition 6.1. ***Dominating Set (or Absorbent)***

A set $D \subseteq V = P \cup T$ in a SiPN N^* is called a 'Dominating set with respect to a marking' $\mu_1 \in R(N^*, \mu_0)$ *if either all the vertices of* V *are in* D *or* $\forall\ v \in V \backslash D$

$$v^\bullet \cap D \neq \emptyset \text{ and } \sigma(v,u) = S(v)S(u)\ \forall\ u \in v^\bullet \cap D$$

where $S(u), S(v)$ *are sign of vertices* u, v *with respect to marking* μ_1.

Remark 6.1. *It may be noted that sign of a transition remains same irrespective of marking of the SiPN while sign of a place may vary if the marking of SiPN changes.*

Definition 6.2. ***Dominating Set with respect to a set of markings***

A set $D_M \subseteq V$ *in a SiPN* N^* *is called a* 'Dominating set with respect to a set of markings' $M \subseteq R(N^*, \mu_0)$ *if* D_M *is a dominating set with respect to all the markings* $\mu \in M$ *(Clearly,* $|M| \geq 2$*).*

Definition 6.3. ***Dependent (or connected) Dominating Set***

A dominating set D_M *with respect to a set of markings* M *in a SiPN* N^* *is called a* 'dependent Dominating set' *if there exists a subtree of the reachability tree of* N^* *whose nodes are all the markings of* M *only.*

Definition 6.4. ***Minimal Dominating Set***

A dominating set D *is called* 'Minimal dominating set' *if no proper subset of it is a dominating set or it is a dominating set with minimum number of vertices.*

Remark 6.2. *For application purposes it is best to find a minimal dependent dominating set* D *w.r.t. a set of markings* M *of maximum cardinality.*

Theorem 6.1. *For a SiPN structure* N'' *in which all the transitions are positively signed and each place has incident (input/output) arcs of one kind only, there exists a marking* μ *w.r.t. which* $V \backslash A$ *is a dominating set where* A *is a set of source vertices.*

Proof. If $A = \emptyset$, then $V \backslash A = V$ is a dominating set by definition.

If $A \neq \emptyset$, then aim is to find a marking μ such that $V \backslash A$ is a dominating set w.r.t. μ.

For any $x \in V \backslash (V \backslash A) = A$, $x^{\bullet} \cap (V \backslash A) \neq \emptyset$ (Since, x is a source vertex). Then, find a marking μ such that $\forall y \in x^{\bullet} \cap (V \backslash A)$

$$\sigma(x,y) = S(x)S(y) \tag{6.2.1}$$

where $S(x)$ and $S(y)$ are sign of vertices w.r.t. μ. Let $y \in x^{\bullet} \cap (V \backslash A)$. Now, two cases arise:

1. **x is a place.**

   Therefore, y is a transition and $S(y) = +$ (By hypothesis).

   (a) Now, if $\sigma(x,y) = +$, then, for equation (6.2.1) to hold ; $S(x) = +$. Therefore, $\mu(x) = (\mu^+(x), \mu^-(x))$ where $\mu^+(x) \in \mathbb{N}$ and $\mu^-(x) \in \mathbb{N}_e$, where $\mathbb{N}_e$ is the set of positive even integers.

   (b) Else, if $\sigma(x,y) = -$, then, for equation (6.2.1) to hold ; $S(x) = -$. Therefore, $\mu(x) = (\mu^+(x), \mu^-(x))$ where $\mu^+(x) \in \mathbb{N}$ and $\mu^-(x) \in \mathbb{N}_o$, where $\mathbb{N}_o$ is the set of positive odd integers.

2. **x is a transition.**

   Therefore, y is a place. Since x is a transition, $S(x) = +$ (By hypothesis). Now, $\sigma(x,y)$ can be positive or negative.

   (a) If $\sigma(x,y) = +$ then, for equation (6.2.1) to hold ; $S(y) = +$ and hence, $\mu(y) = (\mu^+(y), \mu^-(y))$ where $\mu^+(y) \in \mathbb{N}$ & $\mu^-(y) \in \mathbb{N}_e$.

   (b) Else, if $\sigma(x,y) = -$ then, for equation (6.2.1) to hold ; $S(y) = -$ and hence, $\mu(y) = (\mu^+(y), \mu^-(y))$ where $\mu^+(y) \in \mathbb{N}$ & $\mu^-(y) \in \mathbb{N}_o$.

Hence, for all $p_i \in A \cup \{z \in V \backslash A \mid z \in A^{\bullet}\}$, $\mu^{+}(p_i) \in \mathbb{N}$ &

$$\mu^{-}(p_i) \in \begin{cases} \mathbb{N}_e & \text{, if } p_i \text{ has positive incident arcs.} \\ \mathbb{N}_o & \text{, if } p_i \text{ has negative incident arcs.} \end{cases}$$

All the remaining places can have any number of positive and negative tokens without any restrictions. □

Theorem 6.2. *If a SiPN structure N'' with no source/sink vertices and in which any place has only one type of incident arcs then, there exists a marking μ such that P and T are dominating sets w.r.t. μ, provided all the transitions are of same sign.*

Proof. Since all the transitions are of same sign, two cases arise:

1. **Transitions are positively signed .**

   Find a marking μ w.r.t. which P and T are dominating sets.

   (a) P **is a dominating set**.

   Let $t \in V \backslash P$, therefore, $t^{\bullet} \cap P \neq \emptyset$ (since there are no source/sink vertices). Find a marking μ such that $\forall\ p \in t^{\bullet} \cap P$;

$$\sigma(t,p) = S(t)S(p) \tag{6.2.2}$$

   where $S(p)$ is sign of place p w.r.t. marking μ and $S(t)$ is the sign of transition t.

   Let $p \in t^{\bullet} \cap P$. Then,

   i. $\sigma(t,p) = +$

   Since $S(t) = + \ \forall t \in T$. Then, for equation (6.2.2) to hold; $S(p) = +$. Then, $\mu(p) = (\mu^{+}(p), \mu^{-}(p))$ where, $\mu^{+}(p) \in \mathbb{N}$, $\mu^{-}(p) \in \mathbb{N}_e$.

ii. $\sigma(t,p) = -$

Since $S(t) = + \ \forall t \in T$. Then, for equation (6.2.2) to hold; $S(p) = -$. Then, $\mu(p) = (\mu^+(p), \mu^-(p))$ where, $\mu^+(p) \in \mathbb{N}$, $\mu^-(p) \in \mathbb{N}_o$.

(b) T **is a dominating set**.

Let $p \in V \backslash T$, therefore, $p^\bullet \cap T \neq \emptyset$ (since there are no source/sink vertices). Find a marking μ such that $\forall\, t \in p^\bullet \cap T$;

$$\sigma(p,t) = S(p)S(t) \tag{6.2.3}$$

where, $S(p)$ is sign of place p w.r.t. marking μ and $S(t)$ is the sign of transition t.

Let $t \in p^\bullet \cap T$. Then,

i. $\sigma(p,t) = +$

Since $S(t) = +$. Then, for equation (6.2.3) to hold $S(p) = +$. Then, $\mu(p) = (\mu^+(p), \mu^-(p))$ where, $\mu^+(p) \in \mathbb{N}$ and $\mu^-(p) \in \mathbb{N}_e$.

ii. $\sigma(p,t) = -$

Since $S(t) = +$. Then, for equation (6.2.3) to hold $S(p) = -$. Then, $\mu(p) = (\mu^+(p), \mu^-(p))$ where, $\mu^+(p) \in \mathbb{N}$ and $\mu^-(p) \in \mathbb{N}_o$.

Hence, for all p_i, $\mu^+(p_i) \in \mathbb{N}$ &

$$\mu^-(p_i) \in \begin{cases} \mathbb{N}_e & \text{, if } p_i \text{ has positive incident arcs.} \\ \mathbb{N}_o & \text{, if } p_i \text{ has negative incident arcs.} \end{cases}$$

2. **Transitions are negatively signed .**

By following the same procedure as in the case when transitions are positively signed , it is seen that for all p_i, $\mu^+(p_i) \in \mathbb{N} \cup \{0\}$ &

$$\mu^-(p_i) \in \begin{cases} \mathbb{N}_o & \text{, if } p_i \text{ has positive incident arcs.} \\ \mathbb{N}_e & \text{, if } p_i \text{ has negative incident arcs.} \end{cases}$$

Thus, a marking μ is obtained w.r.t. which P and T are dominating sets. $\square$

Remark 6.3. *Theorems 6.1 and 6.2 given above show that:*

1. *A dominating set can be obtained if the initial marking is taken as the one mentioned in the proof.*

2. *The sets P and T in a Petri net represent conditions and events respectively of the system modeled. So, it can be checked whether conditions or events dominate in the given system w.r.t. a given marking.*

3. *If the structure of a SiPN and its initial marking μ_0 are known, then it can be checked whether domination can occur or not. This can be done by verifying the existence of a marking $\mu \in R(N^*, \mu_0)$ w.r.t. which there exists a dominating set. So, in order to avoid or force domination a SiPN can be restricted (or forced) to avoid (or reach) such a marking.*

Theorem 6.3. *In a live signed state machine with no sink places:*

1. *P is a dominating set w.r.t. a marking, provided input arcs for any place are of one type only.*

2. *T is a dominating set w.r.t. a marking, provided output arcs for any place are of one type only.*

Proof. 1. **P is a dominating set.**

For any $t \in T\ \exists\ p \in P$ such that $t^{\bullet} \cap P \neq \emptyset$ (since $|{}^{\bullet}t| = |t^{\bullet}| = 1\ \forall\ t\ \in\ T$). If $\sigma(t,p) = +$, then, take $\mu_0(p) = (m, even)$ else, if $\sigma(t,p) = -$, then, take $\mu_0(p) = (m, odd), m \in \mathbb{N}$ and $odd, even$ means any odd or even positive integer respectively.

For any $t \in T$, there does not exist $p_i, p_j \in P$ s.t. $t^{\bullet} \cap P = \{p_i.p_j\}, i \neq j$ (since $|{}^{\bullet}t| = |t^{\bullet}| = 1$), however there might exist $t_i, t_j \in T$ such that $t_i^{\bullet} \cap P = \{p\} = t_j^{\bullet} \cap P$. Then, two cases can arise:

(a) $\sigma(t_i, p) = +$ and $\sigma(t_j, p) = +$. Here, take $\mu_0(p) = (m, even)$.

(b) $\sigma(t_i, p) = -$ and $\sigma(t_j, p) = -$. Here, take $\mu_0(p) = (m, even)$.

where $m \in \mathbb{N}$.

Similarly, the result can be proved if there exist three or more transitions which are input of a place.

2. **T is a dominating set.**

For any $p \in P\ \exists\ t \in T$ such that $p^{\bullet} \cap T \neq \emptyset$ (since there are no sink places).

Now, if $\sigma(p,t) = +$, then, take $\mu_0(p) = (m, even)$ else, if $\sigma(p,t) = -$, then, take $\mu_0(p) = (m, odd)$, $m \in \mathbb{N}$.

For any $p \in P\ if\ \exists\ t_1, t_2 \in T$ such that $p^{\bullet} \cap T = \{t_1, t_2\}$, then, either $\sigma(p, t_i) = \sigma(p, t_j) = +$ or $\sigma(p, t_i) = \sigma(p, t_j) = -$ because a place has output arcs of one kind only. When sign of the arcs is positive, take $\mu_0(p) = (m, even)$, and when sign is negative, take $\mu_0(p) = (m, odd), m \in \mathbb{N}$.

If $|p^{\bullet} \cap T| > 2$, similar process can be followed. In either case a marking μ_0 is obtained with respect to which P or T is a dominating set.

□

Remark 6.4. *A live signed state machine is balanced. Since $|^{\bullet}t| = |t^{\bullet}| = 1\ \forall\, t \in T$ and SiPN is live, therefore all transitions must fire at least once which implies input and output arcs of any transition are of one kind only. Also, this implies that all transitions are positive in sign.*

Theorem 6.4. *In a signed marked graph N^* with no sink transitions, P and T are dominating sets of N^* w.r.t. some marking.*

Proof. 1. P **is a dominating set**.

For any $t \in T\ \exists\ p \in P$ such that $t^{\bullet} \cap P \neq \emptyset$ (since no sink transitions). If $\sigma(t,p) = +$ then, take $\mu_0(p) = (m, even)$ when $S(t) = +$ and $\mu_0(p) = (m, odd)$ when $S(t) = -$ else, if $\sigma(t,p) = -$ then, take $\mu_0(p) = (m, even)$ when $S(t) = -$ and $\mu_0(p) = (m, odd)$ when $S(t) = +$, where $m \in \mathbb{N}$ and $odd, even$ means any odd or even positive integer respectively.

For any $t \in T$ if $t^{\bullet} \cap P = \{p_i.p_j\}, i \neq j$, four cases arise.

(a) $\sigma(t,p_i) = +\ \&\ \sigma(t,p_j) = +$

Here, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = -$.

(b) $\sigma(t,p_i) = +\ \&\ \sigma(t,p_j) = -$

Here, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = -$.

(c) $\sigma(t,p_i) = -\ \&\ \sigma(t,p_j) = +$

Here, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, even)$ if

$S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = -$.

(d) $\sigma(t, p_i) = -$ & $\sigma(t, p_j) = -$

Here, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = -$.

where, $m_1, m_2 \in \mathbb{N}$.

Similarly, the result can be proved if there exist three or more places which are output of a transition.

2. T **is a dominating set**.

For any $p \in P\ \exists\ t \in T$ such that $p^\bullet \cap T \neq \emptyset$.

(since $|{}^\bullet p| = |p^\bullet| = 1\ \forall\ p\ \in\ P$).

Now, if $\sigma(p,t) = +$ then, take $\mu_0(p) = (m, even)$ when $S(t) = +$ and $\mu_0(p) = (m, odd)$ when $S(t) = -$. However, if $\sigma(p,t) = -$ then, take $\mu_0(p) = (m, even)$ when $S(t) = -$ and $\mu_0(p) = (m, odd)$ if $S(t) = +$, where $m \in \mathbb{N}$.

If $\exists\ p_i, p_j\ \in P$ such that $p_i^\bullet \cap T = \{t\} = p_j^\bullet \cap T,\ i \neq j$, then, four cases arise.

(a) $\sigma(t, p_i) = +$ & $\sigma(t, p_j) = +$

Here, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = -$.

(b) $\sigma(t, p_i) = +$ & $\sigma(t, p_j) = -$

Here, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, odd)$ and

$\mu_0(p_j) = (m_2, even)$ if $S(t) = -$.

(c) $\sigma(t, p_i) = -$ & $\sigma(t, p_j) = +$

Here, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = -$.

(d) $\sigma(t, p_i) = -$ & $\sigma(t, p_j) = -$

Here, take $\mu_0(p_i) = (m_1, odd)$ and $\mu_0(p_j) = (m_2, odd)$ if $S(t) = +$ otherwise, take $\mu_0(p_i) = (m_1, even)$ and $\mu_0(p_j) = (m_2, even)$ if $S(t) = -$.

where, $m_1, m_2 \in \mathbb{N}$.

If $|^\bullet t \cap P| > 2$, a similar process can be followed. In either case a marking μ_0 is obtained with respect to which P or T is a dominating set.

□

6.3 Applications of Domination

6.3.1 Determining highest and lowest ranking official in an institution

Consider an institution with a given number of individuals and take all possible interactions that can happen among them. This scenario is represented by SiPN in **Figure 6.1** for an institution with three individuals. If there is a positive arc connecting an individual p_i to an individual p_j then p_i is lower in hierarchy than p_j and if there is a negative arc connecting an individual p_i to an individual p_j then p_i is higher in hierarchy than p_j. Thus, a positive arc can be considered as a compliance to some order or request, while a negative arc may be considered as an order. The following assumptions are made in order to find out the highest and lowest ranking official in the given institution.

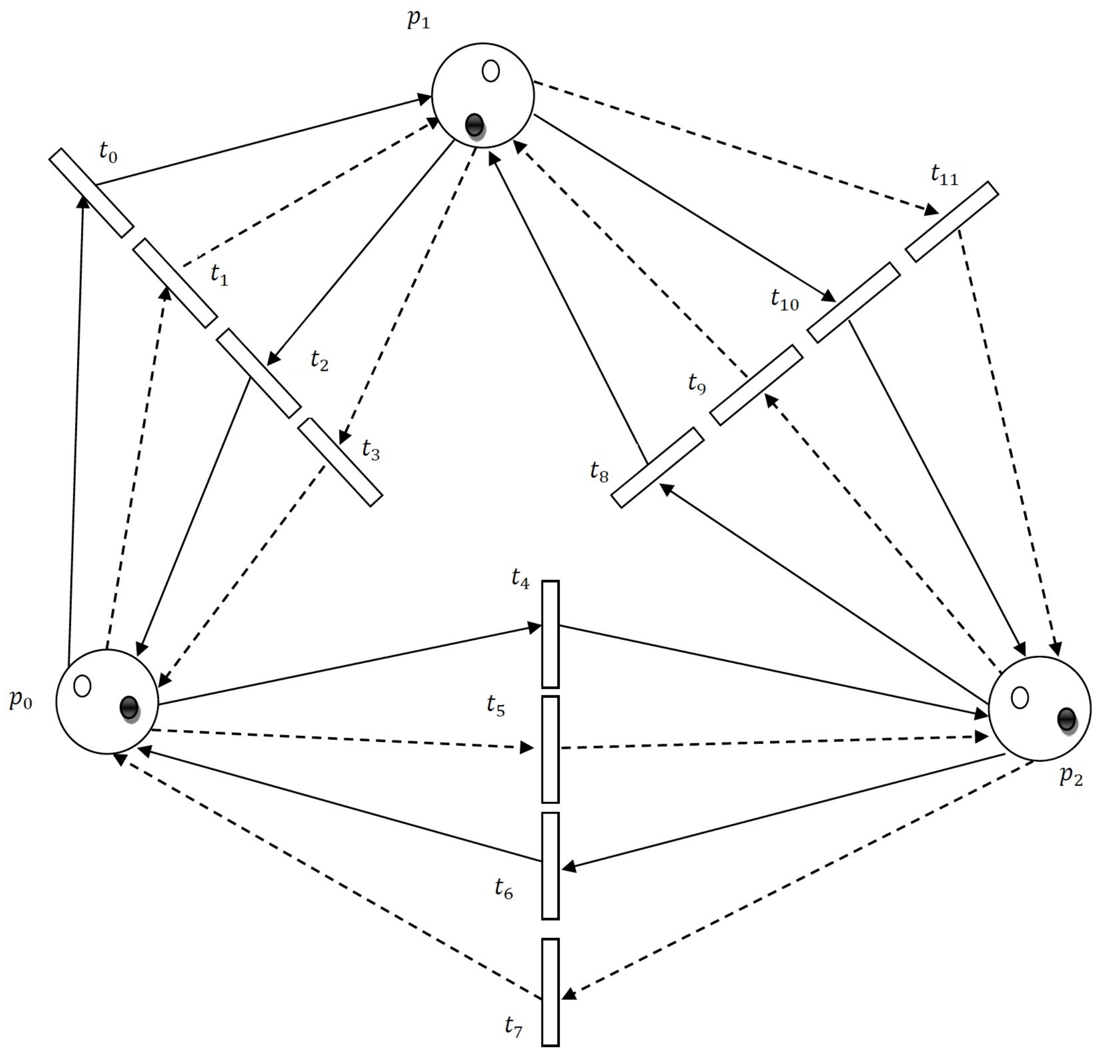


Figure 6.1: A SiPN representing three individuals interactions

1. The transitions of the firing sequence that will be considered in this application should lie on a directed cycle, i.e., a firing sequence η is given which represent cyclic interactions between all the individuals in the institutions. This implies that $I^{-}(p,t) \neq 0$ & $I^{+}(p,t) \neq 0$ $\forall p \in P$ i.e., every place has an

incoming as well as outgoing arc.

2. The initial marking of any place in the SiPN is $(1,1)$.

Procedure to find the highest and lowest ranking official in an Institution for a given activity

Given a SiPN with all possible interactions that exist between a group of individuals, where places represent the number of individuals in an institution and transitions represent the interactions between these individuals.

1. Start with a set of transitions $T' \subset T$ which includes transitions from the firing sequence η (representing an activity). Note that the number of transitions in the firing sequence is equal to the number of places (individuals) in the SiPN (institution).

2. Find the sign of all the places w.r.t. the new marking μ_1 obtained after firing of sequence η from μ_0. All the places in the initial marking μ_0 have one positive token and one negative token.

3. Find the dominating set of the sub-SiPN, which have all the places of the SiPN but transitions from η only.

   (a) First, check whether the set T' is the dominating set of sub-SiPN or not. Clearly, $p^{\bullet} \cap T' \neq \emptyset \; \forall p \in P$. The only property left to check is whether $\sigma(p,t) = S(p)S(t) \; \forall \; p \in P$. Since, in case of social interactions $S(t) = + \; \forall \; t \in T$, therefore only the equation given below need to be checked:

$$\sigma(p,t) = S(p) \; \forall \; p \in P \tag{6.3.1}$$

(b) If the equation (6.3.1) holds for all places $p \in P$, then the dominating set is the set T', else the dominating set D is given by:

$$D = T' \cup \{p_i \mid p_i \in P, \sigma(p_i,t) \neq S(p_i)\}$$

4. Next, based on the dominating set found in the previous step, find the highest and lowest ranking official in the institution using the steps below:

   (a) If $D = T'$ or $D = T' \cup P$, then there does not exist a hierarchy in the institution.

   (b) If places $p_i, p_j \in D$ such that $S(p_i) = +$, $S(p_j) = -$, then p_i is the highest ranking official while p_j is an intermediate between highest and lowest ranking official.

   (c) If places $p_i, p_j \notin D$ such that $S(p_i) = +$, $S(p_j) = -$, then p_i is the lowest ranking official while p_j is an intermediate between highest and lowest ranking official.

Look at how the procedure works with the help of some examples:

Example 1: Consider the firing sequence $\eta_1 = t_0t_{11}t_7$ in the SiPN given in the **Figure 6.1**. This firing sequence represents the interactions between three individuals p_0, p_1, and p_2 where p_0 interacts positively with p_1, p_1 interacts negatively with p_2 and p_2 interacts negatively with p_0. The corresponding sub-SiPN is represented in the **Figure 6.2**. The marking μ_1 obtained after firing of the sequence η_1 is $((0,2,1),(2,0,1))$. Find a dominating set for the

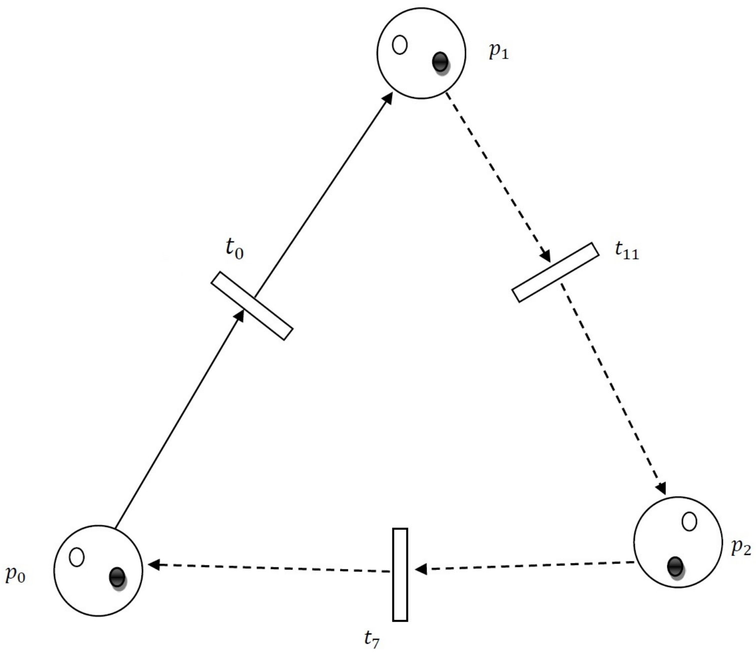


Figure 6.2: Sub-SiPN corresponding to the firing sequence $\eta_1 = t_0t_{11}t_7$

sub-SiPN.

$\sigma(p_0,t_0) = +$ and $S(p_0) = +$, i.e., $\sigma(p_0,t_0) = S(p_0)$

$\sigma(p_1,t_{11}) = -$ and $S(p_1) = +$, i.e., $\sigma(p_1,t_{11}) \neq S(p_1)$

$\sigma(p_2,t_7) = -$ and $S(p_2) = -$, i.e., $\sigma(p_2,t_7) = S(p_2)$

Clearly, here the set $T' = \{t_0,t_{11},t_7\}$ is not a dominating set w.r.t. μ_1 of the sub-SiPN. The dominating set w.r.t. μ_1 for the given sub-SiPN is $D = T' \cup \{p_1\}$. Using the above mentioned procedure it can be seen that since $p_1 \in D$, $S(p_1) = +$, this implies that p_1 is highest ranking official. The places p_0 , $p_2 \notin D$ and $S(p_0) = +$ and $S(p_2) = -$ at the marking μ_1, therefore p_0 is the lowest ranking official while p_2 is intermediate between highest and lowest ranking official.

Example 2: Consider the firing sequence $\eta_2 = t_0t_{11}t_6$ in the SiPN given in the **Figure 6.1**. This firing sequence represents the interactions between three individuals p_0, p_1, and p_2 where p_0

interacts positively with p_1, p_1 interacts negatively with p_2 and p_2 interacts positively with p_0. The corresponding sub-SiPN is represented in the **Figure 6.3**. The marking μ_2 obtained after firing of

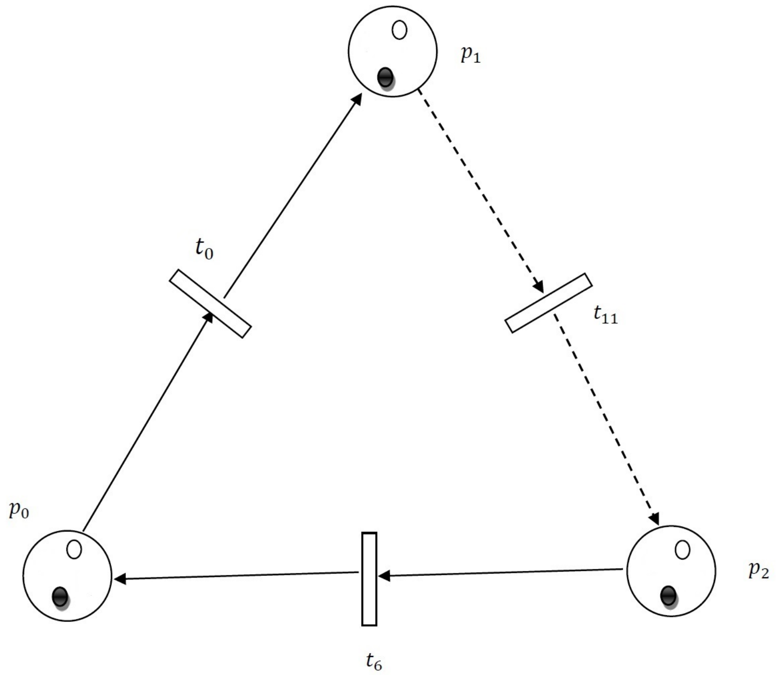


Figure 6.3: Sub-SiPN corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_6$

the sequence η_2 is $((1,2,0),(1,0,2))$. Find a dominating set for the sub-SiPN given in **Figure 6.3**.

$\sigma(p_0,t_0) = +$ and $S(p_0) = -$, i.e., $\sigma(p_0,t_0) \neq S(p_0)$

$\sigma(p_1,t_{11}) = -$ and $S(p_1) = +$, i.e., $\sigma(p_1,t_{11}) \neq S(p_1)$

$\sigma(p_2,t_6) = +$ and $S(p_2) = +$, i.e., $\sigma(p_2,t_6) = S(p_2)$

Clearly, here the set $T' = \{t_0,t_{11},t_6\}$ is not a dominating set w.r.t. μ_2 of sub-SiPN. The dominating set w.r.t. μ_2 is $D = T' \cup \{p_0, p_1\}$. Using the above mentioned procedure, it can be seen that since $p_0, p_1 \in D$ and $S(p_0) = -$ & $S(p_1) = +$, this implies that p_1 is highest ranking official while p_0 is intermediate between highest and lowest ranking official.

The place $p_2 \notin D$ and $S(p_2) = +$ at the marking μ_2, therefore p_2 is the lowest ranking official.

C++ Code to find the highest and lowest ranking official in an institution for a given activity

The C++ code for the procedure mentioned above is given in the Appendix 1 along with the outputs received corresponding to the Examples 1 and 2. Next, some applications of domination for SiPNs in various other areas are briefly discussed.

6.3.2 Producer-Consumer Problem

Consider, a standard Producer-Consumer problem with two producers producing a same product (assuming quality, price and other conditions are same), it need to checked whether one of the producer can dominate the market over the other. This can happen due to availability of product is greater for one producer as compared to other or because one product is well known due to its better marketing, etc. Consider a SiPN model for the problem given in **Figure 6.4**. Here, as seen in the figure, the left and right side of SiPN represent the producers while the center part of the SiPN represents the consumer. The transitions t_1, t_3 represent events of production of a product while the transitions t_2, t_4 represent putting the product in buffer for future use, as and when demand is raised by the consumer. When a consumer demands for a product, the product is provided by firing of transition t_5 or t_6, which represent removal of product from the buffer places p_7 or p_8. Then, finally consumer consumes the product by firing the transition t_7.

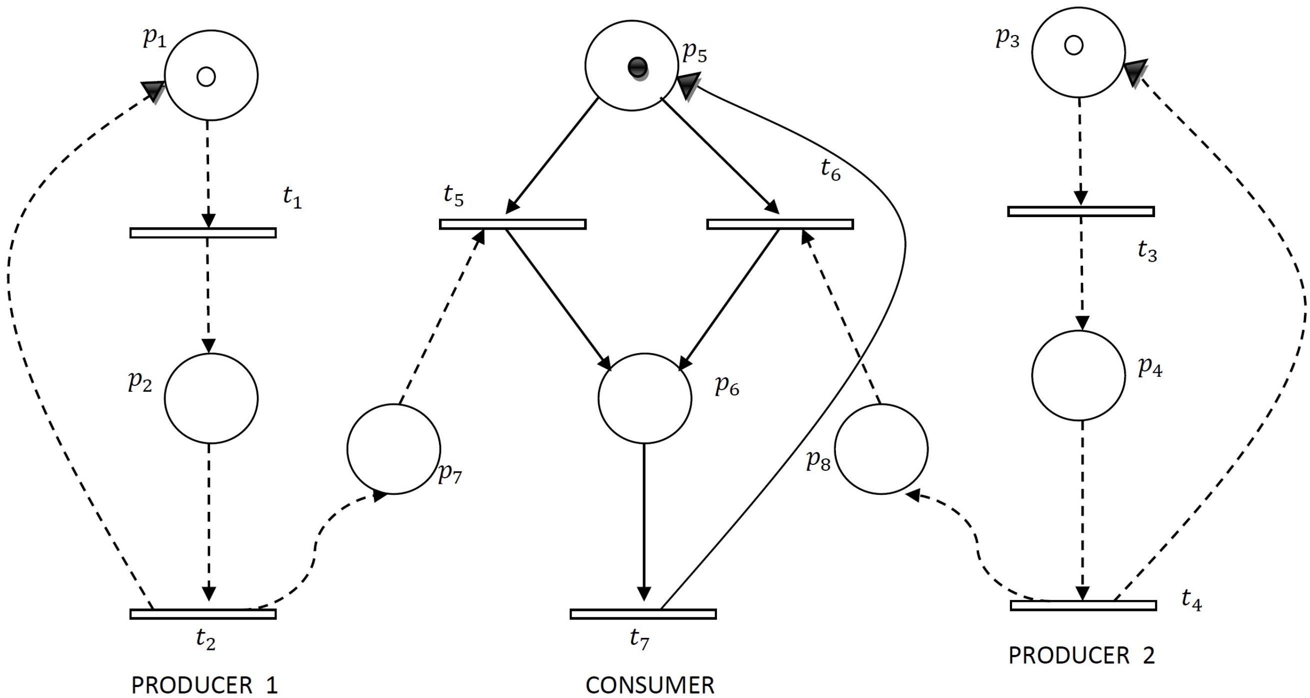


Figure 6.4: Producer-Consumer Problem

Now, in order to check whether the producer 2 dominates the market, it is enough to find a set of markings M of maximum cardinality with $\mu \in M$ ($\mu(p_6) \neq 0$, so that consumer always receives a product) such that $D_1 = V \backslash \{p_7, t_7\}$ is a dominating set w.r.t. M. The set D_1 should be a dependent dominating set over a set of markings M of maximum cardinality. A dependent dominating set is considered because one producer is said to dominate over the other if such a domination exists over a period of time, not for just an instant. Similarly, to check whether producer 1 dominates the market, it is enough to check the domination of set D_2 w.r.t. a set of markings where $D_2 = V \backslash \{p_8, t_7\}$.

6.3.3 Search of food by bees

Bees (Scout bees) go out in search of food. The one which find the food will return to the hive and celebrate. This scout bee can be considered to dominate other scout bees. This problem can be modeled using SiPNs and then it can be identified which scout bee will dominate the bee-hive. Consider the **Figure 6.5** where place p_1 represents the bee-hive while p_2, p_3 represent possible food locations where, scout bees, say A and B respectively search for food. The positive tokens are used to represent scout bees and negative ones to represent the availability of food. The transitions t_2, t_3 represent events of food search while t_1, t_4 represent events of food search completion.

According to the initial marking of SiPN, location p_2 has food while location p_3 does not. Therefore, bee A must dominate. This can be verified by checking that the set $D_1 = \{p_1, p_3, t_1, t_2, t_3, t_4\}$ is a dominating set w.r.t. the initial marking μ_0 rather than set $D_2 = \{p_1, p_2, t_1, t_2, t_3, t_4\}$. In the later case, bee B will dominate the bee-hive. The model can be

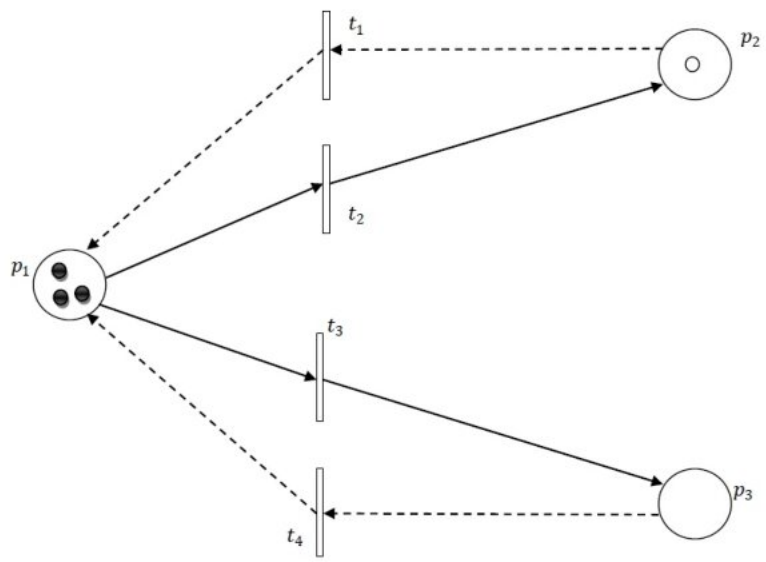


Figure 6.5: A SiPN model with initial marking $((3,0,0),(0,1,0))$ for finding food by Scout Bees

extended, if more than two bees search for food.

6.3.4 Finding papers with similarity to a given paper

Consider a paper which need to be checked for similarity using a software. This paper is compared with web content that is publicly available, books, papers in journals, articles and other content which is present in a software database. Let the paper to be checked be represented by a place p_0 and the rest of the content be represented by places $p_1, p_2, p_3, \dots p_k$. To check for similarity, a SiPN model is formed by connecting place p_0 to all other places via transitions $t_1, t_2, t_3, \dots t_k$ and by using negative arcs as in **Figure 6.6**.

While comparing paper p_0 with another paper represented by place p_i (say), $1 \leq i \leq k$, a matching algorithm is used to find a set of strings within submitted paper p_0 that matches with the papers maintained in its database. If a similarity exists, a negative token is generated in place p_0 which can be used to fire corresponding transition t_i. In this way, the submitted paper is checked for similarity with all the content present in the database. After the comparisons are completed, a new marking for the SiPN is obtained in which all the articles that have some similarity with the submitted paper get a negative token in the place representing it. All such places will form a list of articles that are similar to the paper submitted which is to be tested for similarity.

Now, in order to find the list of all articles which have some similarity with the submitted paper, the concept of domination can be used instead of finding all the places having negative token. Begin with set $D_1 = P = \{p_0, p_1, p_2, \dots p_k\}$. Check whether set D_1 is a dominating set w.r.t. the final marking (say μ') obtained after the matching algorithm is complete. If yes, then all the papers $p_i, 1 \leq i \leq k$ have some similarity with the submitted paper. If not, find $D_2 \subseteq T$ such that $D' = D_1 \cup D_2$ is a dominating set w.r.t. μ'. Then, the set $\{p_i | \ t_i \in T \backslash D_2\}$ will form the set of all the articles which have similarity with the submitted paper.

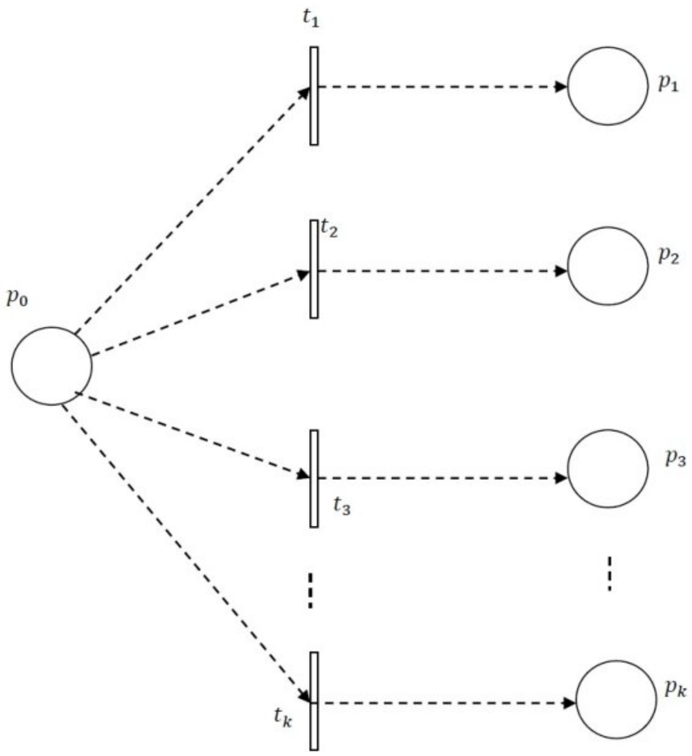


Figure 6.6: Finding similarity in papers

In conclusion, the concept of domination in ***signed Petri nets*** is introduced and it is shown how the proposed concept is useful with the help of applications.

References

[1] Acharya, B. D. (2013). *Domination and absorbance in signed graphs and digraphs I: Foundations*. Journal of Combinatorial Mathematics and Combinatorial Computing, 84:5–20.

[2] Ashraf, P. K. and Germina, K. A. (2015). *On minimal dominating sets for signed graphs*. Advances and Applications in Discrete Mathematics, 15:101–112.

[3] Blidia, M., Chellali, M., Haynes, T. W., and Henning, M. (2006). *Independent and double domination in trees*. Utilitas Mathematica, 70:159–173.

[4] Gupta, P. (2013). *Domination in Graph with Application*. Indian Journal of Research, Paripex, 2(3):115–117.

[5] Harary, F. and Haynes, T. (2000). *Double domination in graphs*. Ars Combinatoria, 55:201–213.

[6] Haynes, T., Hedetneimi, S. T., and Slater, P. J. (1998). **Fundamentals of domination in graphs**. *1st edition, CRC Press*.

[7] Jaenisch, C. (1862). **Trait des Applications de l'Analyse Mathematique au Jeu des Echecs**. Petrograd.

[8] Palakkutty, A., Germina, A., and Lishan, L. (2016). *Double domination in signed graphs*. Cogent Mathematics, 3(1):1186135.

[9] Sampathkumar, E. (1998). *Domination Parameters of A Graph*. Domination in Graphs: Advanced Topics, 2:271–300.

[10] Sasireka, A. and Nandhu Kishore, A. H. (2014). *Applications of Dominating Set of Graph in Computer Networks*. International Journal of Engineering Sciences and Research Technology, 10(9):2801–2810.

[11] Zelinka, B. (2001). *Signed total domination number of a graph*. Czechoslovak Mathematical Journal, 51(2):225–229.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

In the thesis work, an extension for Petri nets called "Signed Petri net" has been introduced. All the terminology of SiPNs and various other concepts related to it are also provided. The applications of SiPNs are given by modeling a message transmission system, a production unit. Along with this, by showing equivalence between a logic Petri net and a logic SiPN, it is shown how a modified SiPN can simulate a logic Petri net.

In chapter 3, various behavioral properties of SiPNs are defined which can help in analyzing the modeled system. For analysis, two techniques are provided: Reachability Tree and Matrix Equations with main focus on the matrix equations. Further, in chapter 3, it can be seen how the proposed concept of SiPNs is beneficial by modeling a restaurant system using SiPNs and analyzing it using various behavioral properties. Then, a modified restaurant system is modeled. The modifications incorporated show the ease of making changes in a system modeled using SiPNs rather than a basic Petri nets .

It has been shown in chapter 4 how various matrices help to store the information of a SiPN structure. Such matrices can be utilized for storage and used in various algorithms, since it is easier to work with matrices than images/graphs. Various subclasses for SiPNs have been given along with their characterizations using the matrices introduced in the chapter.

In chapter 5, structural and dynamical balanceness are defined. Various theorems are provided to identify the SiPNs which are structurally balanced. The concept of dynamical balanceness helps to reduce the efforts in studying social interactions, since in order to check for stability of interactions among individuals, only a single SiPN is studied as compared to multiple signed graphs being analyzed in Graph theory.

The concept of domination is defined in chapter 6 of the thesis. Such a concept does not exist for dynamic systems and using the dynamic nature of SiPNs and presence of positive and negative tokens in its place, it becomes easier to define such a concept for SiPNs. Further, the concept of domination is utilized to find the highest and lowest ranking official in an institute based on certain activity. The applications of producer-consumer problem, searching of food by bees and finding similarity in a research paper are also discussed using the domination in SiPNs.

7.2 Future Scope

Our motive is to further develop theory of signed Petri nets. Various concepts related to SiPNs have been defined and utilized in several applications. Along with the notions already defined in the thesis, we are of the view that this extension has a great scope in the study of various dynamic systems as well as modeling real-life applications. In fact, using SiPNs various systems where dynamics are involved can be studied with great ease as compared to a signed graph or a basic Petri net.

In fact, it is not a far fetched dream to combine the concepts of dynamical balanceness and domination and define a relationship among the two. It would make it easier to check the modeled system for one of the concept and infer from it whether the other concept also applies to it or not. Since, SiPNs can be used to model and analyze the social interactions among individuals, in future, this concept can become a great tool in modeling and analyzing social networks. The message diffusion process in on-line social networks is of great importance as this process helps to understand how and why some messages become viral.

In most of the research that focuses on On-line social networks, the theory of signed graph is extensively utilized. But since social media is ever changing, the signed graphs fails to incorporate the dynamics of the process. In order to involve the dynamics associated, we are of the view that SiPNs can serve as a better tool for modeling and analyzing such systems.

Bibliography

[1] Aalst van der, W., Stahl, C., and Westergaard, M. (2013). *Strategies for Modeling Complex Processes Using Colored Petri nets*. Transactions on Petri Nets and Other Models of Concurrency VII, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 7480:6–55.

[2] Aalst van der, W. M. P. (1993). *Multi-dimensional Petri nets*. Computing science notes Eindhoven:Technische Universiteit Eindhoven, 9326.

[3] Acharya, B. D. (2013). *Domination and absorbance in signed graphs and digraphs I: Foundations*. Journal of Combinatorial Mathematics and Combinatorial Computing, 84:5–20.

[4] Acharya, M., Jain, R., and Kansal, S. (2014). *Some results on the splitting signed graphs*. Journal of Combinatorics,Information and System Sciences, 39(1-4):23–32.

[5] Akiyama, J., Avis, D., Chvátal, V., and Era, H. (1981). *Balancing signed graphs*. Discrete Applied Mathematics, 3:227–233.

[6] Alla, H. and David, R. (1998). *Continuous and Hybrid Petri Nets*. Journal of Circuits,Systems and Computers, 8(1):159–188.

[7] Aref, S. and Wilson, M. (2019). *Balance and Frustration in Signed Networks*. Journal of Complex Networks, 7(2):163–189.

[8] Ashraf, P. K. and Germina, K. A. (2015). *On minimal dominating sets for signed graphs*. Advances and Applications in Discrete Mathematics, 15:101–112.

[9] Bartoletti, M., Cimoli, T., and Pinna, G. (2015). *Lending Petri Nets*. Science of Computer Programming, 112:75–101.

[10] Blidia, M., Chellali, M., Haynes, T. W., and Henning, M. (2006). *Independent and double domination in trees*. Utilitas Mathematica, 70:159–173.

[11] Brunsch, J. L., Huttenlocher, D., and Kleinberg, J. (2010a). *Signed Networks in Social Media*. 2:1361–1370.

[12] Brunsch, T., Hardouin, L., and Raisch, J. (2010b). *Control of cyclically operated High-Throughput Screening Systems*. IFAC Proceedings Volumes, 43(12):167–172.

[13] Brunsch, T., Raisch, J., Hardouin, L., and Boutin, O. (2013). *Discrete- Event Systems in a Dioid Framework:Modeling and Analysis*. Control of Discrete-Event Systems: Automata and Petri Net Perspectives. Lecture Notes in Control and Information Sciences, Springer, London, 433:431–450.

[14] Cantrell, W. A., Maxwell, K. P., Petty, M. D., and Whitaker, T. S. (2020). *Matrix Reduction Verification of Extended Petri Nets*. Proceedings of the 2020 ACM Southeast Conference, Association for Computing Machinery, New York, NY, USA, ACM SE ’20:296–299.

[15] Cartwright, D. and Harary, F. (1956). *Structural Balance:A Generalization of Heider's Theory*. Psychological Review, 63(5):277–293.

[16] Chen, S.-M., Ke, J.-S., and Chang, J.-F. (1990). *Knowledge representation using fuzzy Petri nets*. IEEE Transactions on Knowledge and Data Engineering, 2(3):311–319.

[17] Commoner, F. G. (1972). **Deadlocks in Petri-nets**. *Massachusetts Computer Assoc., Incorporated*.

[18] Cong, X., Fanti, M. P., Mangini, A. M., and Li, Z. (2017). *Decentralized Fault Diagnosis by Petri nets and Integer Linear Programming*. IFAC-PapersOnLine, 50(1):13624–13629.

[19] Davis, J. A. (1967). *Clustering and Structural Balance in Graphs*. Human Relations, 20:181–187.

[20] Doreian, P. and Mrvar, A. (1996). *A partitioning approach to structural balance*. Social Networks, 18(2):149–168.

[21] Du, Y. Y. and Guo, B. (2009). *Logic Petri Nets and Equivalency*. Information Technology Journal, 8:95–100.

[22] Febbraro, A. D., Giglio, D., and Sacco, N. (2017). *On analyzing the vulnerabilities of a railway network with Petri nets*. Transportation Research Procedia, 27:553–560.

[23] Frank, O. and Harary, F. (1979). *Balance in stochastic signed graphs*. Social Networks, 2:155–163.

[24] Gupta, P. (2013). *Domination in Graph with Application*. Indian Journal of Research, Paripex, 2(3):115–117.

[25] Harary, F. (1953). *On the notion of balance of a signed graph*. Michigan Math. Journal, 2:143–146.

[26] Harary, F. and Haynes, T. (2000). *Double domination in graphs*. Ars Combinatoria, 55:201–213.

[27] Haynes, T., Hedetneimi, S. T., and Slater, P. J. (1998). **Fundamentals of domination in graphs**. *1st edition, CRC Press*.

[28] Helouet, L. and Kecir, K. (2018). *Realizability of schedules by stochastic time Petri nets with blocking semantics*. Science of Computer Programming, 157:71–102.

[29] Higuchi, Y. and Sato, I. (2015). *A Balanced Signed Digraph*. Graphs and Combinatorics, 31:2215–2230.

[30] Jaenisch, C. D. (1862). **Trait des Applications de l'Analyse Mathematique au Jeu des Echecs**. Petrograd.

[31] Jensen, K. (1987). *Coloured Petri nets*. Coloured Petri nets. In: Brauer, W., Reisig, W., Rozenberg, G. (eds) Petri nets: Central Models and Their Properties. ACPN 1986. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. , 254:248–299.

[32] Johnson, J. and Murata, T. (1985). *Structure matrices for Petri nets and their applications*. Journal of the Franklin Institute, 319(3):299–309.

[33] Kansal, S., Singh, G. P., and Acharya, M. (2010). *On the Petri nets generating all the binary n-vectors*. Scientiae Mathematicae Japonicae Online, 71(2):113–120.

[34] Kansal, S., Singh, G. P., and Acharya, M. (2011). *1-safe Petri nets generating every binary n-vector exactly once*. Scientiae Mathematicae Japonicae Online, 74:127–134.

[35] Lautenbach, K. (1975). **Liveness in Petri nets**. *Selbstverl. GMD*.

[36] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). *Predicting Positive and Negative Links in Online Social Networks*. Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, WWW '10:641–650.

[37] Liu, H.-C., You, J.-X., Li, Z., and Tian, G. (2017). *Fuzzy Petri nets for knowledge representation and reasoning: A literature review*. Engineering Applications of Artificial Intelligence, 60:45–56.

[38] Marsan, M. (1990). *Stochastic Petri nets: An Elementary Introduction*. Stochastic Petri nets: An elementary introduction. In: Rozenberg, G. (eds) Advances in Petri nets 1989. APN 1988. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 424:1–29.

[39] Murata, T. (1989). *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, 77(4):541–580.

[40] Murata, T. and Yamaguchi, H. (1990). *A Petri net with Negative tokens and its application to Automated Reasoning*. Proceedings of the 33rd Midwest Symposium on Circuits and Systems, 2:762–765.

[41] Palakkutty, A., Germina, A., and Lishan, L. (2016). *Double domination in signed graphs*. Cogent Mathematics, 3(1):1186135.

[42] Patil, S. (1970). **Coordination of asynchronous events**. Ph.D. dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[43] Peterson, J. (1981). **Petri net theory and modeling of systems**. *Prentice Hall-Inc*.

[44] Petri, C. (1962). **Kommunikation mit automaten**. Schriften des Institutes fur Instrumentelle Mathematik, Bonn.

[45] Reddy, P., Sampathkumar, E., and Subramanya, M. (2010). *Common-edge signed graph of a signed graph*. Journal of the Indonesian Mathematical Society , 16:105–113.

[46] Reisig, W. (1991). *Petri nets and algebraic specifications*. Theoretical Computer Science, 80(1):1–34.

[47] Reisig, W. (2013). *The Synthesis Problem*. Transactions on Petri Nets and Other Models of Concurrency VII, Lecture Notes in Computer Science, Springer, Heidelberg, 7480:303–313.

[48] Saha, B. and Bandyopadhyay, S. (1988). *Representation and analysis of Petrinets via the matrix state equation approach*. International Journal of Electronics, 65(1):1–7.

[49] Sampathkumar, E. (1984). *Point-signed and line-signed graphs*. National Academy Science Letters, 7(3):91–93.

[50] Sampathkumar, E. (1998). *Domination Parameters of A Graph*. Domination in Graphs: Advanced Topics, 2:271–300.

[51] Sasireka, A. and Nandhu Kishore, A. H. (2014). *Applications of Dominating Set of Graph in Computer Networks*. International Journal of Engineering Sciences and Research Technology, 10(9):2801–2810.

[52] Semenov, A. S. (2020). *Matrix Approach to Pattern-type Reachability Analysis of Fractal Petri nets*. 7th International Conference on Control, Decision and Information Technologies (CoDIT), 1:211–216.

[53] Silva, J. and Foyo, P. M. G. D. (2012). *Timed Petri nets*. Petri nets - Manufacturing and Computer Science, InTech:359–378.

[54] Staines, S. A. (2016). *Modelling Simple Network Graphs Using the Matrix Vector Transition net*. International Journal of Computers and Communications, 10:11–17.

[55] Staines, S. A. (2019). *Alternative Matrix Representation of Ordinary Petri nets*. WSEAS Transactions on Computers, 18:11–18,.

[56] Valk, R. (2004). *Object Petri nets*. Lectures on Concurrency and Petri nets: Advances in Petri nets, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 3098:819–848.

[57] Wang, J. (1998). *Time Petri nets. In: Timed Petri nets*. The Kluwer International Series on Discrete Event Dynamic Systems, Springer, Boston MA, 9:63–123.

[58] Winskel, G. (1985). *Categories of models for concurrency*. Seminar on Concurrency, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 197:246–267.

[59] Wu, B. Y. and Chen, J.-F. (2013). *Balancing a Complete Signed Graph by Editing Edges and Deleting Nodes*. Advances in Intelligent Systems and Applications, Smart Innovation, Systems and Technologies, Springer Berlin Heidelberg, 1:79–88.

[60] Zaslavsky, T. (1982). *Signed Graphs*. Discrete Applied Mathematics, 4(1):47–74.

[61] Zelinka, B. (2001). *Signed total domination number of a graph*. Czechoslovak Mathematical Journal, 51(2):225–229.

Appendix 1

```
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;
int main()
{

   int m,n,i,j,k;
   int A[15][15],B[15][15],eta[5];
   cout<<"Enter m and n and k values (i.e., |T|, |P|, number of
      transitions in firing sequence eta)";
   cin>>m>>n>>k;

   cout<<"\nEnter Matrix A \n";
   for(i = 0; i < m; ++i)
   for(j = 0; j < n; ++j)
   {
      cout << "Enter element a" << i << j << " : ";
      cin >> A[i][j];
   }

   cout<<"\nMatrix A is \n";
   for(i=0; i<m; i++)
   {
```

```
    cout<< "\n";
    for(j=0; j<n; j++)
    cout<<A[i][j]<< " ";
}
cout<<"\nEnter Matrix B \n";
for(i = 0; i < m; ++i)
for(j = 0; j < n; ++j)
{
    cout << "Enter element b" << i << j << " : ";
    cin >> B[i][j];
}

cout<<"Matrix B is \n";
for(i=0; i<m; i++)
{
    cout<< "\n";
    for(j=0; j<n; j++)
    cout<<B[i][j]<< " ";
}


cout << "\nEnter firing sequence eta \n";
for(i = 0; i < k; ++i)
{
    cout << "Enter element eta" << i << " : ";
    cin >>eta[i];
}

cout<<"\nFiring sequence is \n";
for(i=0; i<k; i++)
{
```

```
   cout<<eta[i]<< " ";
}

int mu01[n],mu02[n];
int mu11[n],mu12[n];
for (i=0; i<n; i++)
{
   mu01[i]=1;
   mu02[i]=1;
}

int feta[m];
for (i=0; i<m; i++)
{
   feta[i]=0;
}

for(int l=0;l<m;l++)
{  for (j=0; j<k; j++)
   {
      if(eta[j]==l)
      {

         feta[l]=feta[l]+1;


      }


   }

}

cout<<"\nFunction of firing sequence is \n";
```

```
for(i=0; i<m; i++)
{

   cout<<feta[i]<< " ";
}




for(i = 0; i < n; i++)
{
   mu11[i]=mu01[i];
   mu12[i]=mu02[i];
   for(j=0;j<m;j++)
   {
      mu11[i]+=feta[j]*A[j][i];
      mu12[i]+=feta[j]*B[j][i];
   }

}

cout<<")";
cout << "\nNew positive marking after firing of the sequence eta \n";
cout<< "mu11 =(";
for(i = 0; i < n; i++)
{
   cout<<   mu11[i]<<" ";
}
cout<<")";

cout << "\nNew negative marking after firing of the sequence eta \n";
cout<< "mu12 =(";
for(i = 0; i < n; i++)
```

```
{
   cout<<   mu12[i]<<" ";
}
cout<<")";


//PROCEDURE

for (j = 0; j < n; ++j)
{
   int check1=0,check2=0;

   {
      for(i = 0;i < m; ++i)
      {
         if(feta[i]!=0)
         {
            if((A[i][j]<0) )
            check1+=1;
            if((B[i][j]<0 ))
            check2+=1;
         }
      }

      if((check1==0)&&(check2==0))
      {
         cout<<"\nAssumptions for the Procedure are not met. \n";
         exit(1);
      }

   }
}
```

```
int D[n];

for(i = 0; i < m; ++i)
{
   if(feta[i]!=0)
   {
      for(j = 0; j < n; ++j)
      {
         if((A[i][j] <0) && (mu12[j]%2==0))
         {
            D[j]=0;

         }

         else if((B[i][j]) <0 && (mu12[j]%2!=0))
         {
            D[j]=0;

         }
         else if(((A[i][j] <0) && (mu12[j]%2!=0)) || ((B[i][j]) <0 &&
             (mu12[j]%2==0)))
         {

            D[j]=1;

         }
      }
   }
}
```

```
cout<<"\n Vector D corresponding to an entry 1 for a place in
   Dominating set else 0 \n";
for(i=0;i<n;++i)
{
   cout<<"\n"<<D[i]<<"  ";

}
int SD=0;
for(i=0;i<n;++i)
{
   if(D[i]!=0)
   {

      SD+=1;
   }

}

cout<<"\n RESULT OF PROCEDURE\n ";

if(SD==0 || SD==n)
{
   cout<<"\nNo hierarchy exists in the institution\n";

}



for(i=0;i<n;i++)
{
   if(mu12[i]%2==0)
   {
      if(D[i]!=0)
```

```
        cout<<"\n p"<<i<<" is the highest ranking official in the
           institution.\n";

        else if(D[i]==0)
        cout<<"\n p"<<i<<" is the lowest ranking official in the
           institution.\n";
      }
   }

   return 0;
}
```

The outputs received corresponding to the Example 1 using the C++ code is given in **Figures 1** to **4** respectively.

The outputs received corresponding to the Example 2 using the C++ code is given in **Figures 5** to **8** respectively.

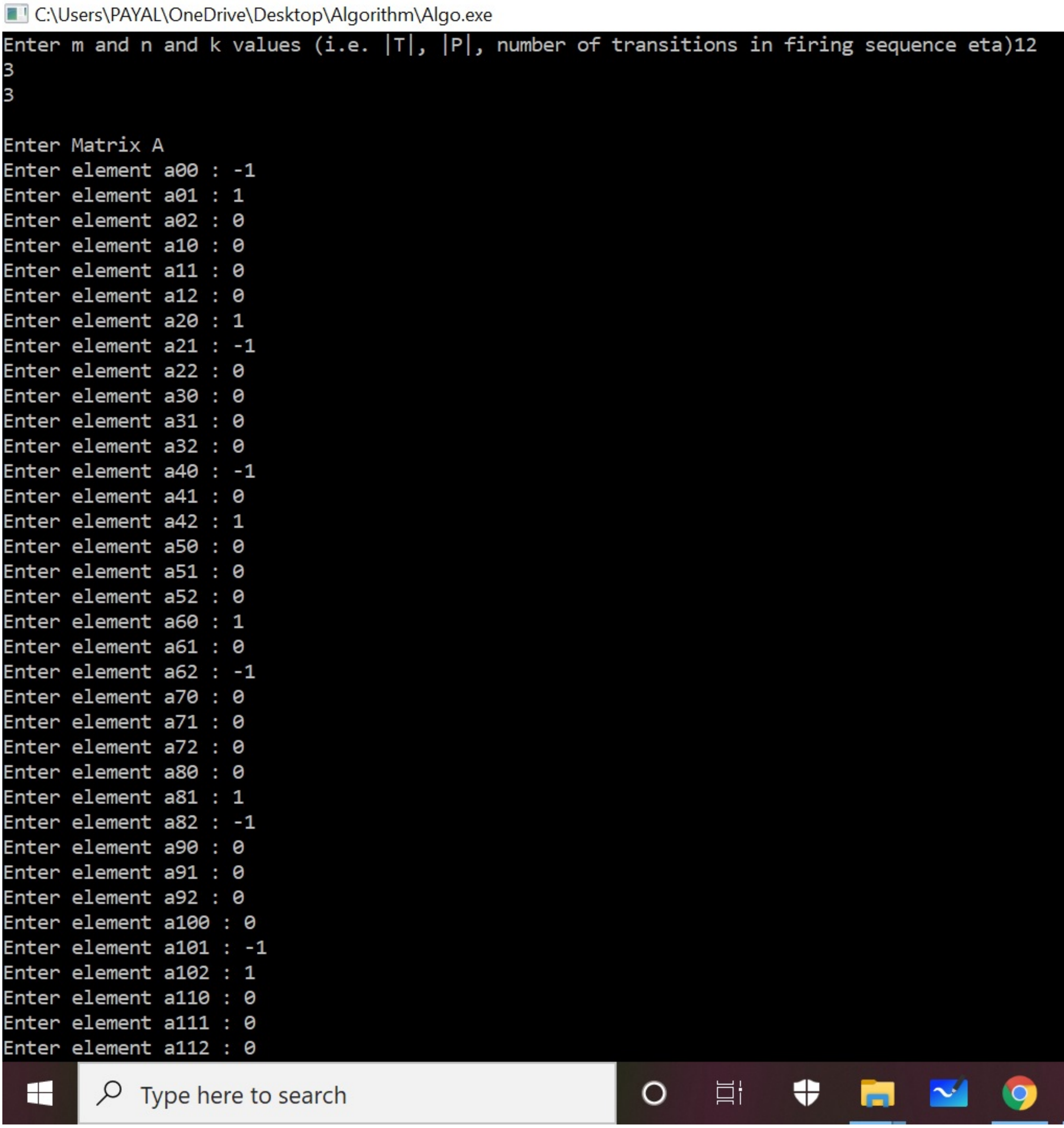


Figure 1: Output using C++ code for the Example 1 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_7$

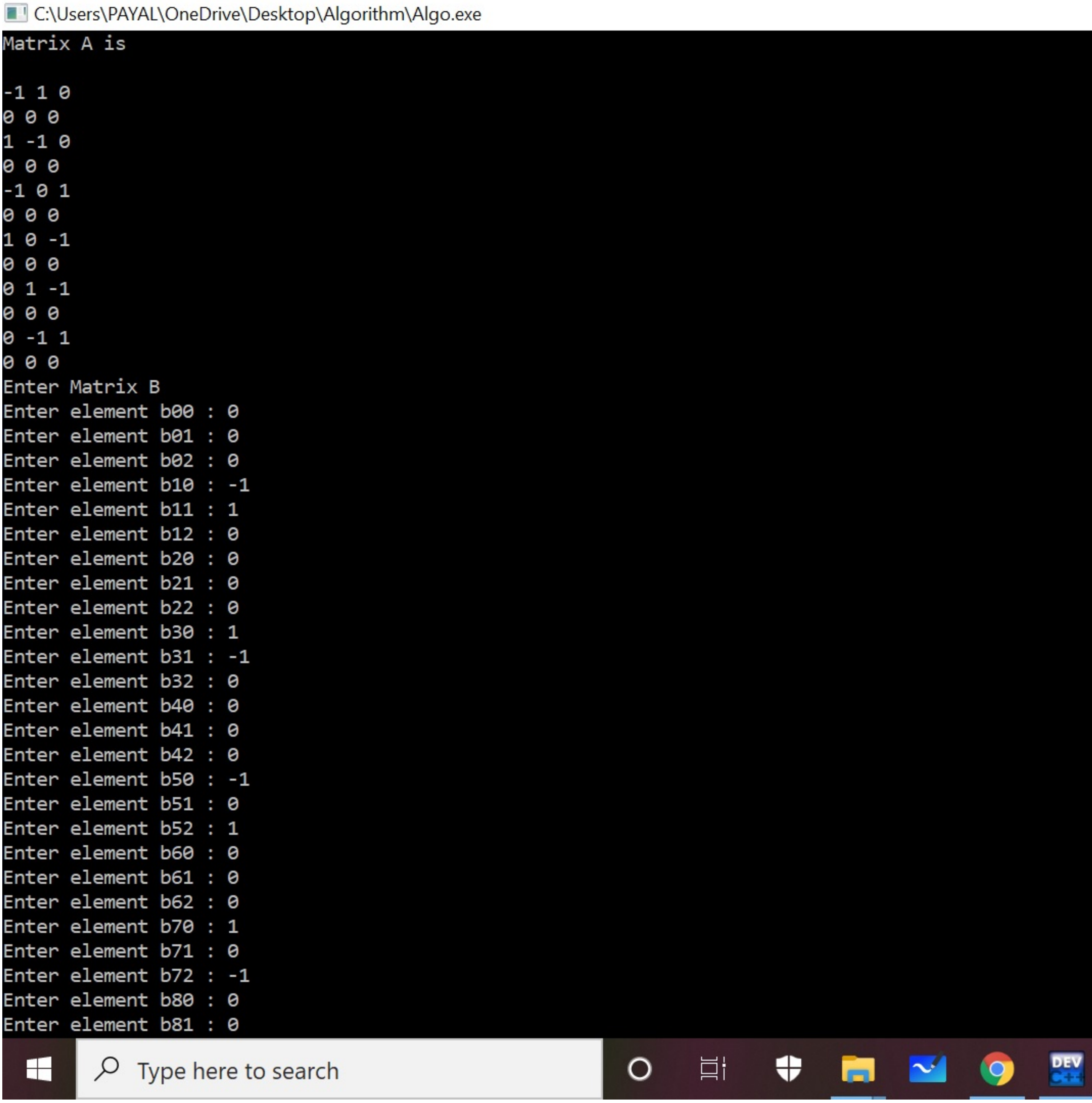
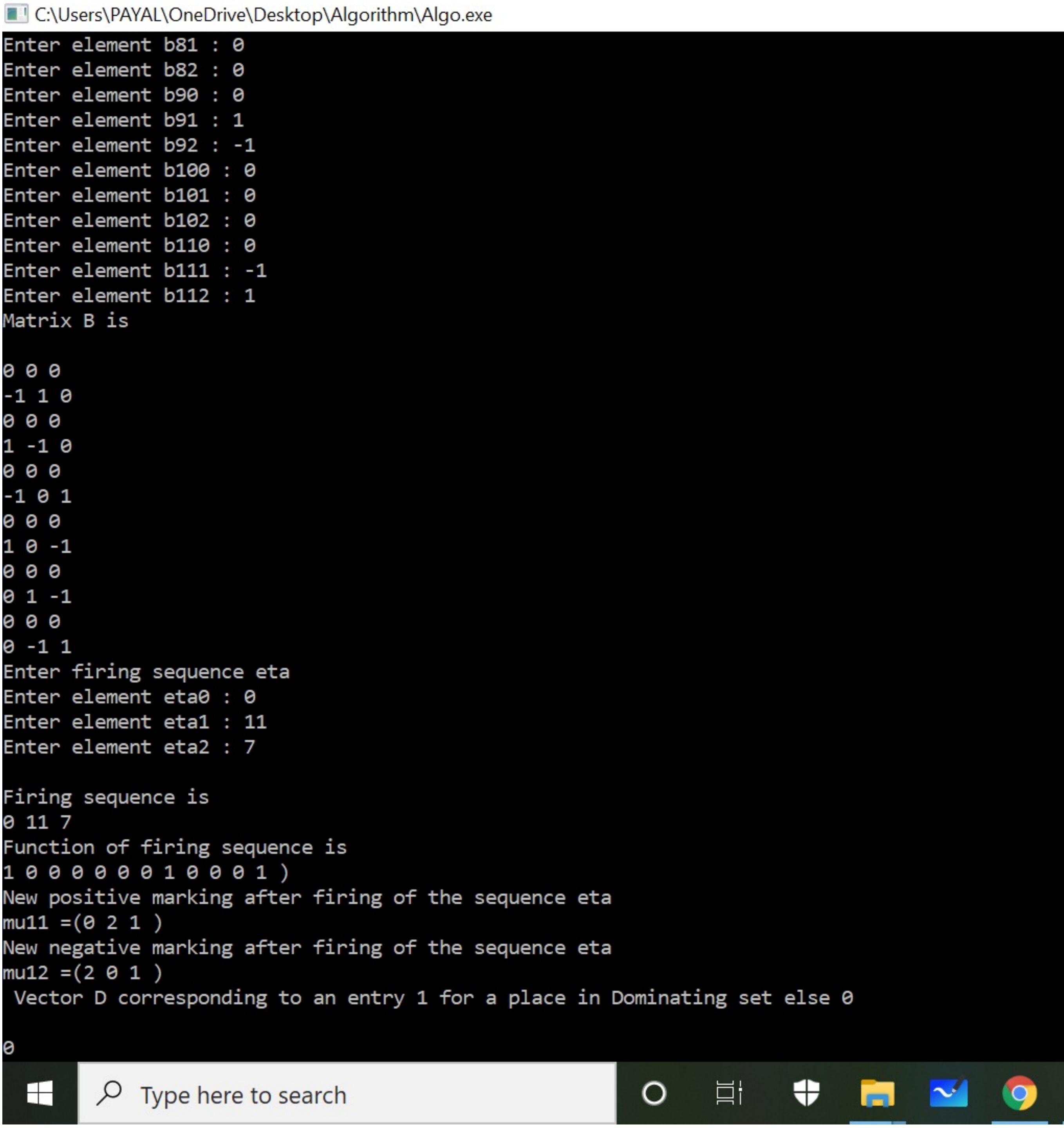


Figure 2: Output using C++ code for the Example 1 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_7$



```
C:\Users\PAYAL\OneDrive\Desktop\Algorithm\Algo.exe
Enter element b81 : 0
Enter element b82 : 0
Enter element b90 : 0
Enter element b91 : 1
Enter element b92 : -1
Enter element b100 : 0
Enter element b101 : 0
Enter element b102 : 0
Enter element b110 : 0
Enter element b111 : -1
Enter element b112 : 1
Matrix B is

0 0 0
-1 1 0
0 0 0
1 -1 0
0 0 0
-1 0 1
0 0 0
1 0 -1
0 0 0
0 1 -1
0 0 0
0 -1 1
Enter firing sequence eta
Enter element eta0 : 0
Enter element eta1 : 11
Enter element eta2 : 7

Firing sequence is
0 11 7
Function of firing sequence is
1 0 0 0 0 0 0 1 0 0 0 1 )
New positive marking after firing of the sequence eta
mu11 =(0 2 1 )
New negative marking after firing of the sequence eta
mu12 =(2 0 1 )
 Vector D corresponding to an entry 1 for a place in Dominating set else 0

0
```


Figure 3: Output using C++ code for the Example 1 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_7$

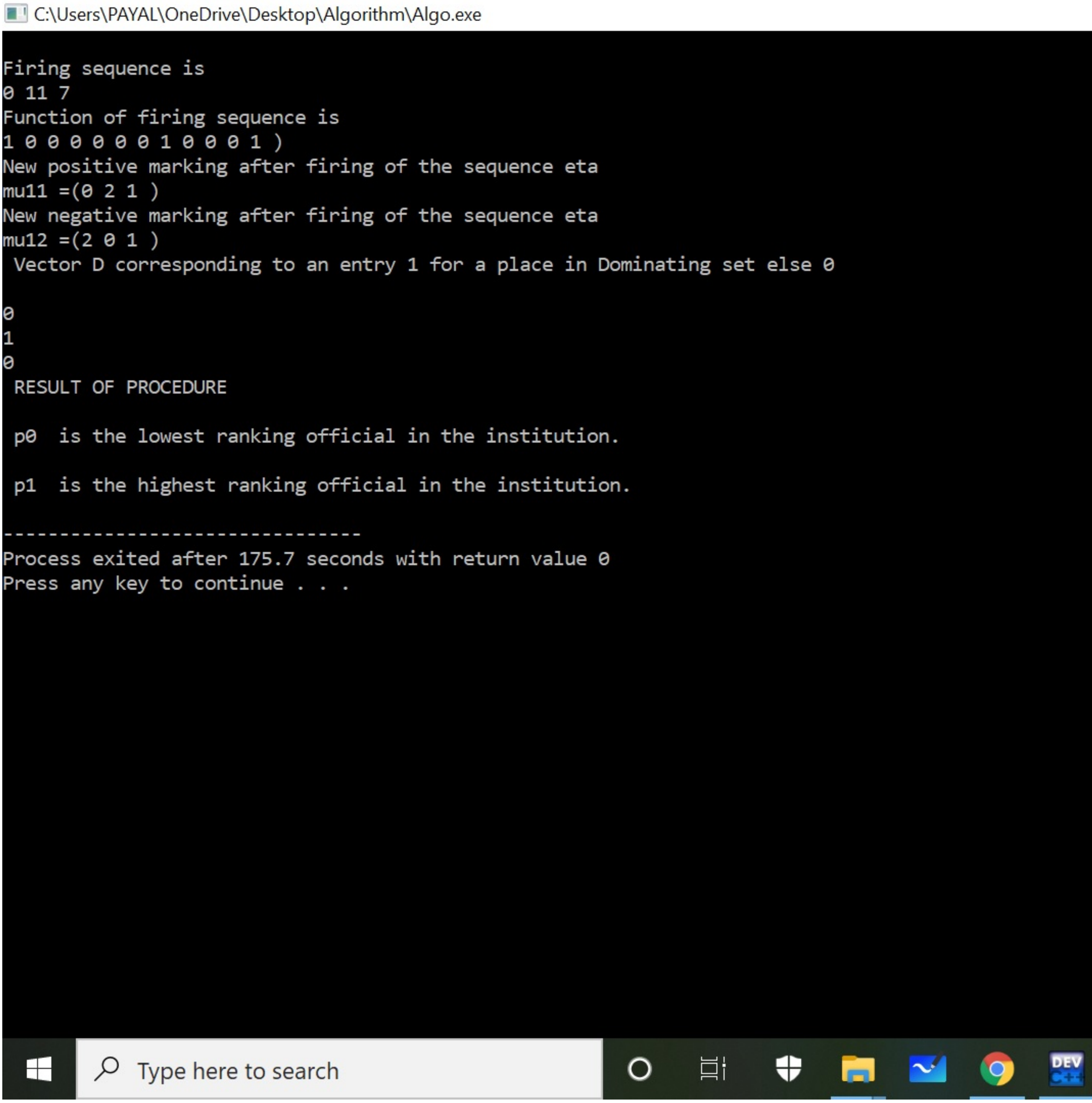


Figure 4: Output using C++ code for the Example 1 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_7$

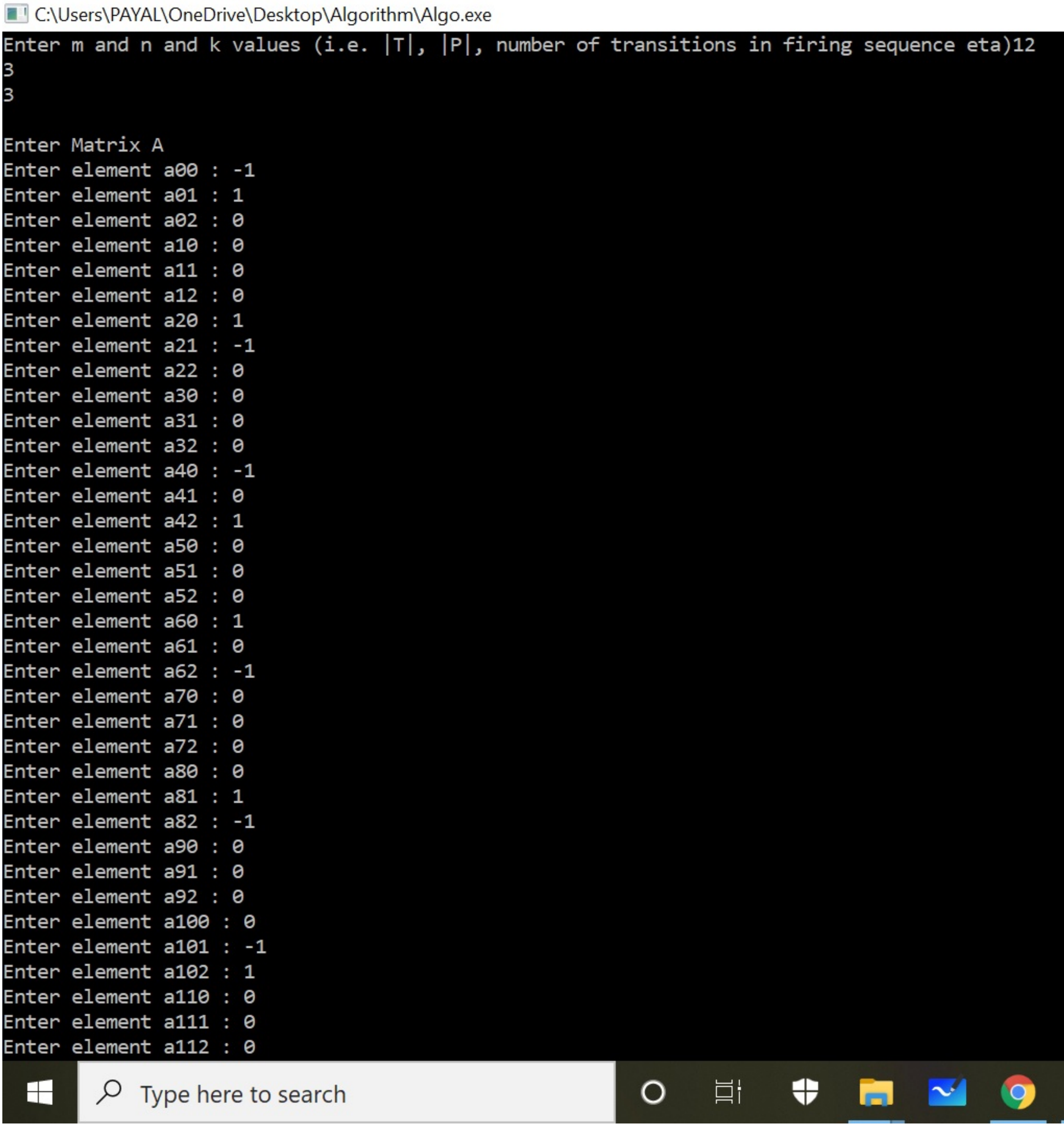


Figure 5: Output using C++ code for the Example 2 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_6$

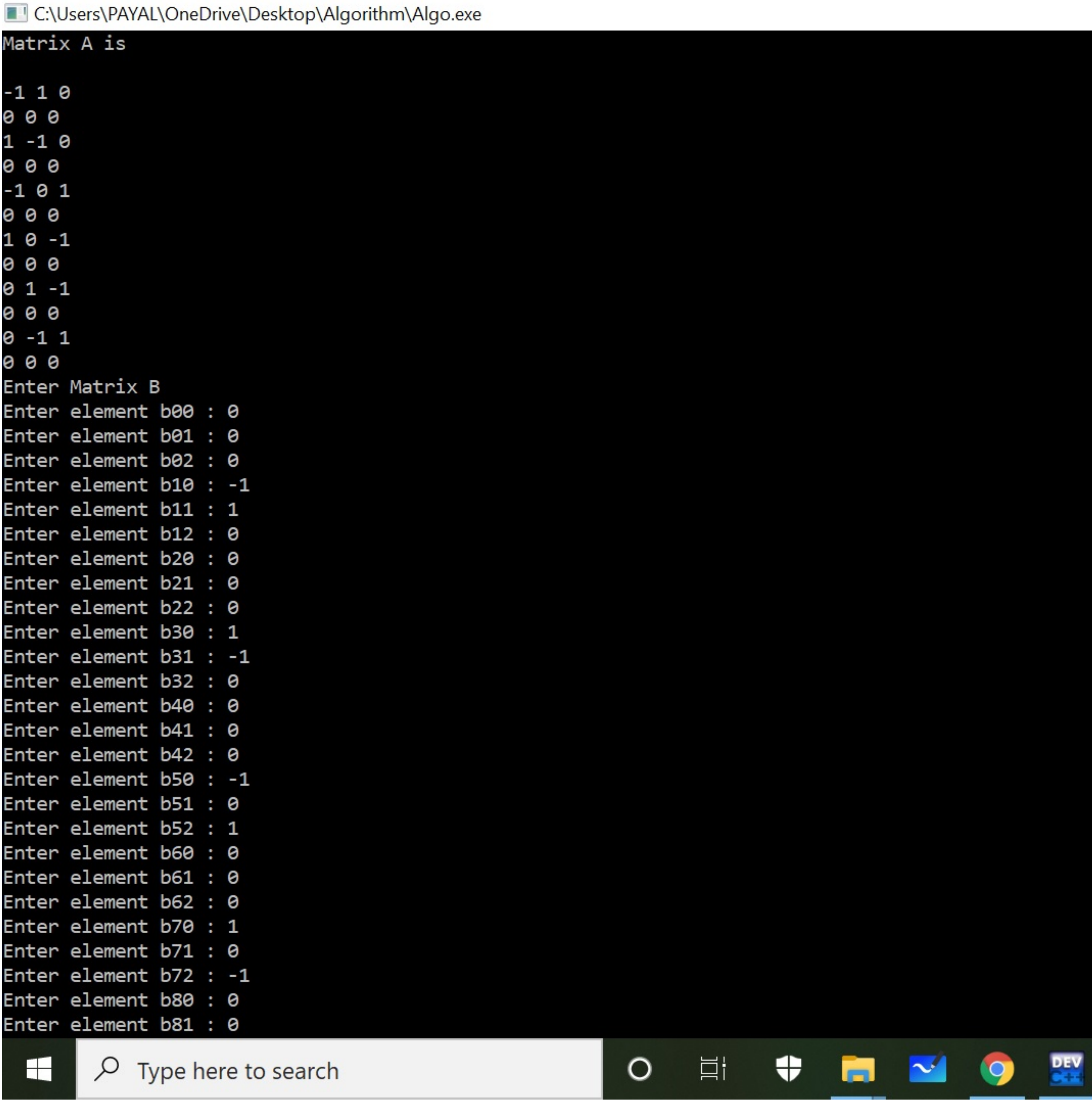


Figure 6: Output using C++ code for the Example 2 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_6$

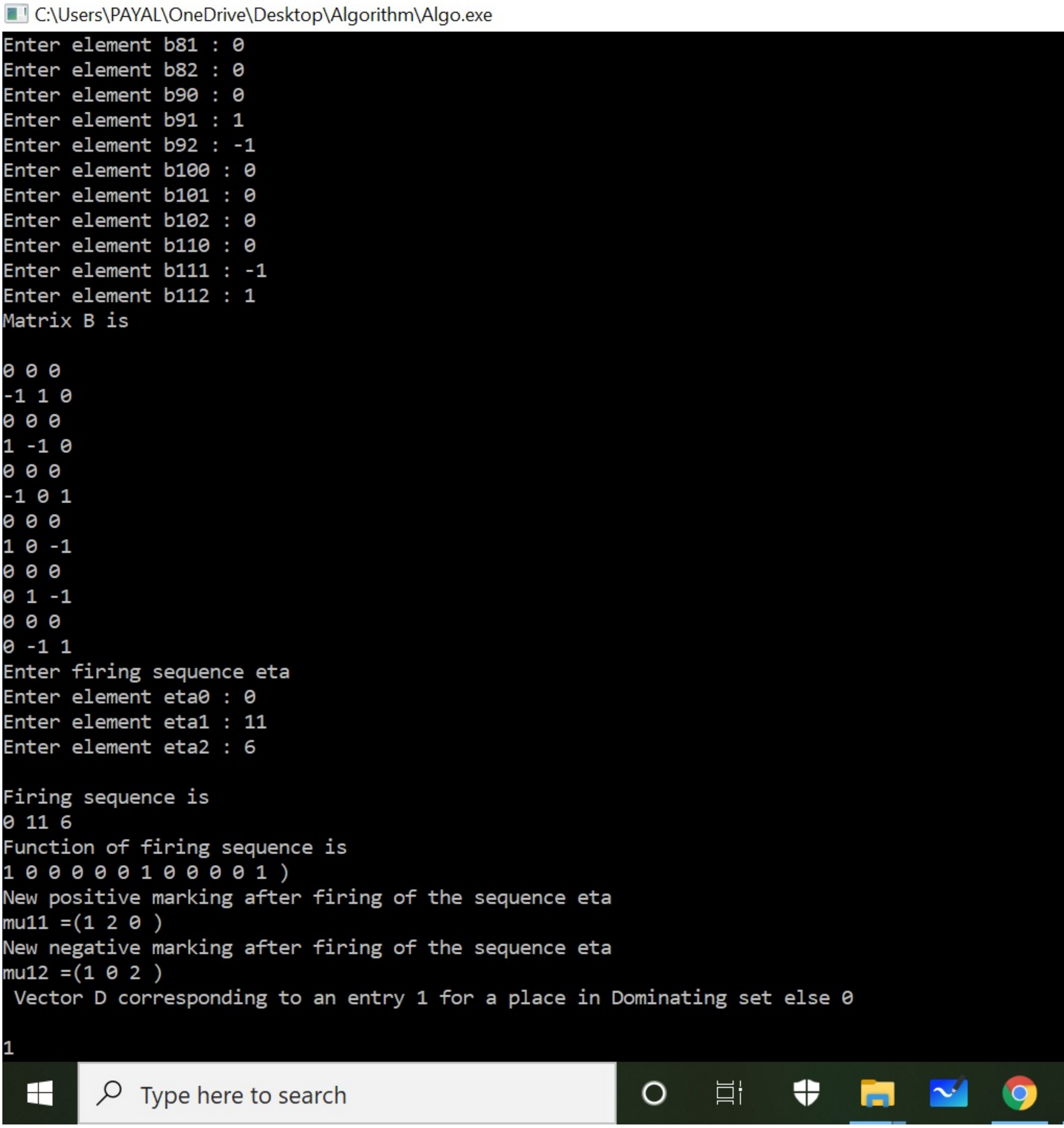


Figure 7: Output using C++ code for the Example 2 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_6$

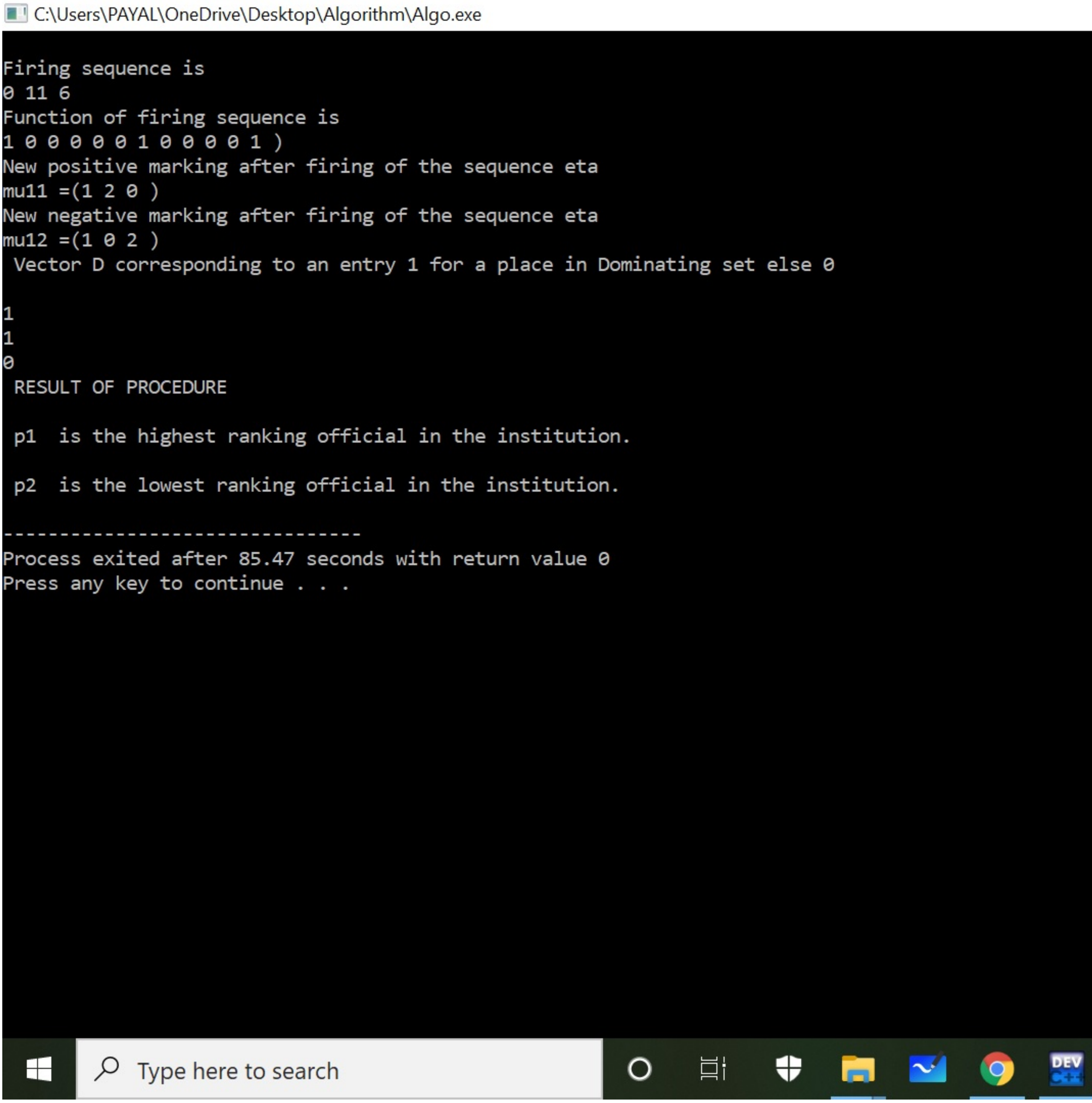


Figure 8: Output using C++ code for the Example 2 corresponding to the firing sequence $\eta_2 = t_0 t_{11} t_6$

List of Publications

1. Payal and Sangita Kansal; *Analysis of Signed Petri Net* , International Journal of Computing Science and Mathematics, DOI: 10.1504/IJCSM.2020.10047015, **2020**. (Accepted) ESCI

2. Sangita Kansal and Payal; *An introduction to Signed Petri Net*, Journal of Mathematics, vol. 2021, Article ID 5595536, DOI: 10.1155/2021/5595536, **2021**. SCIE, Impact Factor (0.971)

3. Payal and Sangita Kansal; *Structural matrices for Signed Petri net*, AKCE International Journal of Graphs and Combinatorics, DOI: 10.1080/09728600.2022. 2070718, **2022**. SCIE, Impact Factor (0.867)

4. Payal and Sangita Kansal; *Logic Signed Petri Net*, Communicated to a journal of high repute

5. Payal and Sangita Kansal; *Social Interactions through Signed Petri Net*, Communicated to a journal of high repute

6. Payal and Sangita Kansal; *Domination in Signed Petri Net*, Communicated to a journal of high repute

Papers presented in International Conferences

1. Payal and Sangita Kansal; *Logic Signed Petri Net* , International Conference on Recent Advances in Pure and Applied Mathematics, **23-25 October, 2018** held at Delhi Technological University, Delhi.

2. Payal and Sangita Kansal; *Dynamical Balanceness in a Signed Petri Net* , 4^{th} International Conference on Recent Advances in Mathematical Sciences and its Applications, **09-11 January, 2020** held at Jaypee Institute of Information Technology, Noida.

REPRINTS OF AUTHORS' PAPERS