MATHEMATICAL MODELING And RISK ANALYSIS

Man VM. Nguyen

Faculty of Science
Mahidol University

This page is left blank intentionally.

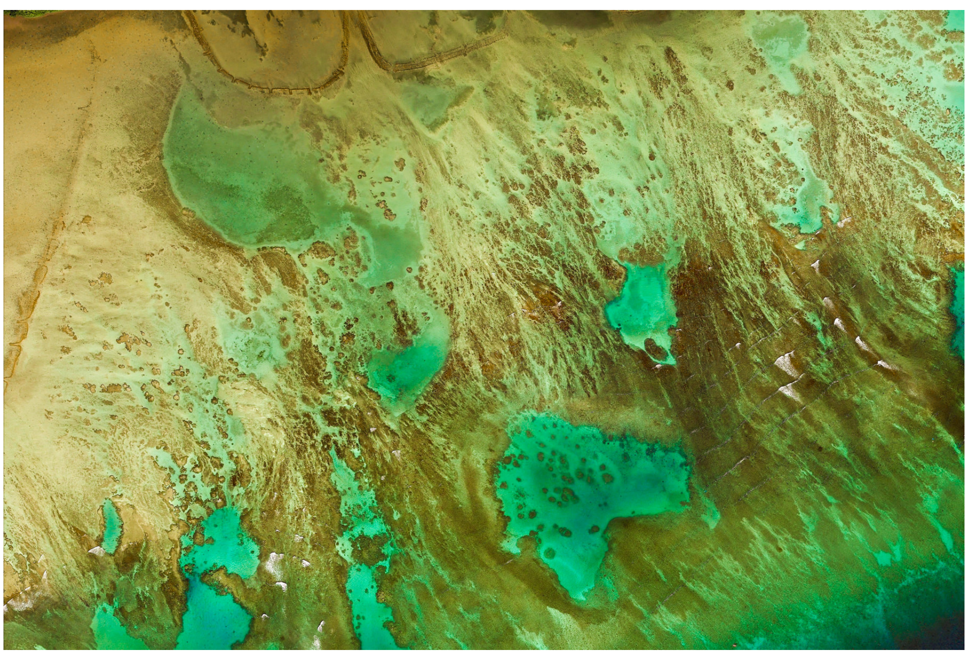
Contents

List of Figures

Chapter 5

PETRI NETWORKS -

From Process Mining View



INTRODUCTION

KEY TERMS reminded

- `Data Science` aims to answer the following four questions.

    1. Reporting: What happened?
    2. Diagnosis: Why did it happen?
    3. Prediction: What will happen?
    4. Recommendation: What is the best that can happen?

- `Process Science` refers to the broader discipline that combines knowledge from *information technology* (IT) and knowledge from *management sciences* (ManSci) to improve and run operational processes.
  WHY? Business processes have become more complex, heavily rely on information systems, and may span multiple organizations. Therefore, process modeling has become of the utmost importance.

OUTLINE

1. INTRODUCTION to Process Mining
2. PETRI NETWORKS- Background
3. BEHAVIOR of PETRI NETS
4. NETWORK STRUCTURES and TYPICAL PROBLEMS in Petri Nets
5. SUMMARIZED OUTCOMES and REVIEWED PROBLEMS
6. ASSIGNMENT on MODELING by PETRI NETS

Key References

- R1: Part I of **PROCESS MINING**, 2nd edition, 2016, Springer, by `W.M.P. Aalst, van der`[1]

- R2: **Process mining techniques and applications**- A systematic mapping study, in Expert Systems With Applications, **Vol 133** (2019) 260-295, Elsevier, by `Cleiton dos Santos Garcia et. al.`
- R3: Chapter 4-9 of the text
  DATA MINING- Concepts, Models, Methods, and Algorithms,
  3rd Edition, 2020, IEEE press and Wiley, by `Mehmed Kantardzic`.
- R4: **Modeling Business Processes**: A Petri Net-Oriented Approach, 2011 Massachusetts Institute of Technology, by Wil van der Aalst and Christian Stahl

[1] W.M.P. Aalst is a full professor in Process Analytics and a full professor in Process Science, both at TUe (the Eindhoven University of Technology). He is also a full professor at RWTH Aachen University.

5.1 INTRODUCTION to Process Mining

5.1.1 *WHAT is Process Mining (PM)?*

Process mining emerged as a new research field that focuses on the analysis of processes using **event data** since 1990s. Historically it was *work-flow mining*. Most relevant disciplines are

1. *Data science*: an interdisciplinary field aiming to turn data into real value. It is the core of many partially overlapping (sub)disciplines:
   - algorithms, business models, data mining
   - data transformation, storage and retrieval of information
   - (computing) infrastructures: databases, distributed systems
   - privacy- security law, predictive analytics (learning, predictions)
   - statistics (modeling+ inference), can be viewed as the origin of data science (**DS**)...

2. *Process science*- **PS**, a broad term aims to combine knowledge from information technology and knowledge from management sciences to run and improve operational processes.
   In process science, basic sub-processes are **event**, not number.
   Research objective is changed [from sample numerical/digital data to event data], so the coupling methods are changed, upgraded to a more sophisticated scale, and as a result, applications are broader.

3. *Process Mining*- the missing link between DS and PS.
   With the key study objective of **event & event data**, Process Mining (**PM**) can be seen as a mean (a research methodology with mixture of modern disciplines) to bridge the gap between data science and process science.
   The concept of EVENT (in process science), popularly called **Internet of Events**- IoE, [2] is a newly structural extension of *classical sample data*, i.e. *numerical-digital observations* in Data Science. In general, the **IoE** include 4 dimensions (constituents), possibly

[2] coined from 2014, early beginning of the AI-based disruptive technology era

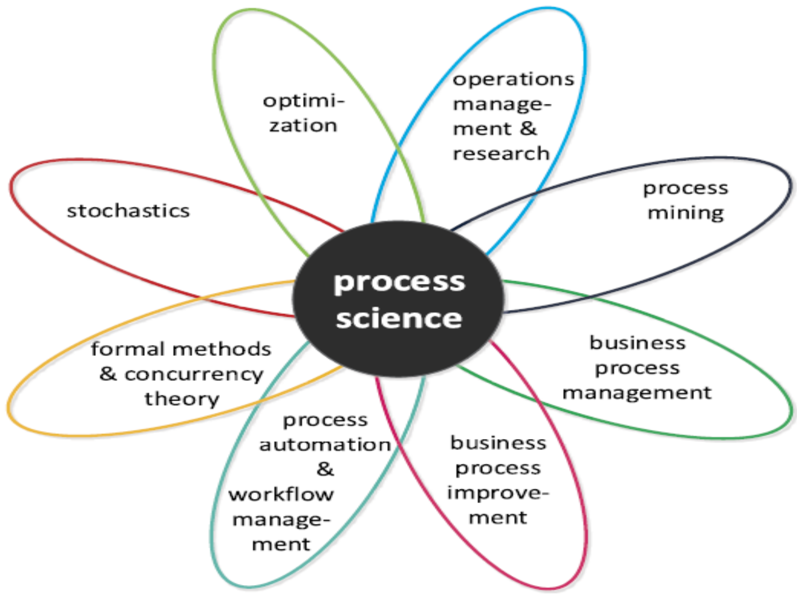
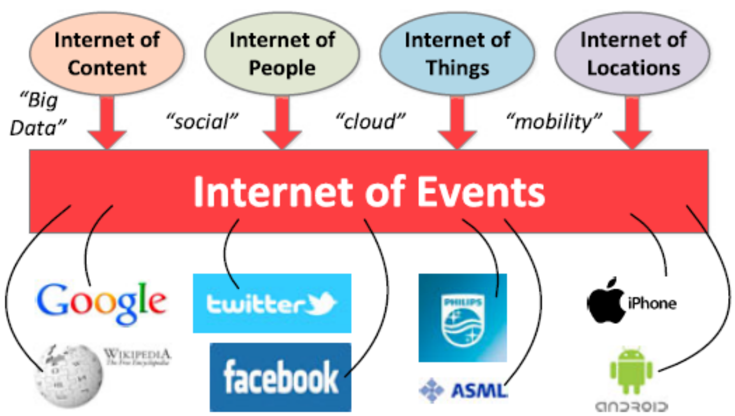


Figure 5.1: *Process science and its constituents*

overlapping, as follows.



Event data nowadays are generated from a variety of sources connected to the Internet [courtesy Wil van der Aalst (2016)]

Content, i.e., all information created by humans to increase knowledge on particular subjects. The IoC includes traditional web pages, articles, encyclopedia-Wikipedia, YouTube, e-books, newsfeeds...

People, i.e., all data related to personal, human-being subject with their social interaction (e-mail, Facebook, Twitter, forums, LinkedIn, etc.

The Internet of Things (IoT), i.e., all physical objects connected to the network, or broadly, relevant hardware which allow an event or sub-process happens.

Locations refers to all data that have a geographical or geo-spatial dimension. [3]

Discussion on drawbacks of approaches:

Data science and its key component, data mining, however are data-centric, **not** process-centric.

Data mining, Statistics, Machine learning and the likes technically do not consider end-to-end process models. Process science approaches are process-centric, but often focus on modeling rather than learning from event data, see next figures.

5.1.2 *WHY? DS and PS to Process Mining (PM)*

Structurally, PM is viewed as the core of a flower, with the branches

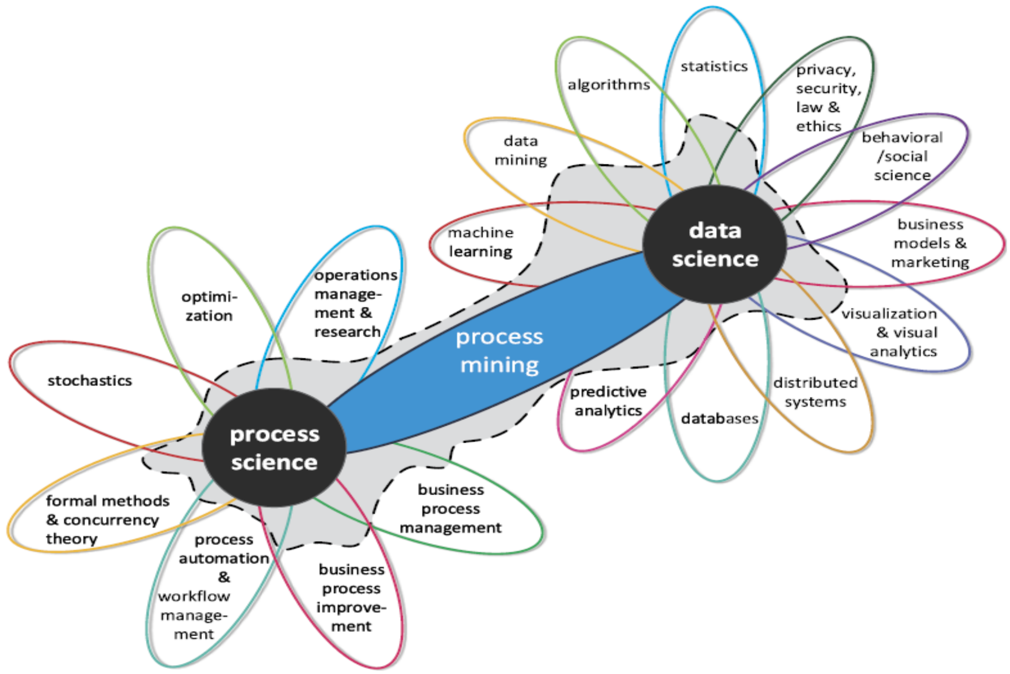
- business process management and improvement (BPM & I), business intelligence (BI), as Petri nets, Lean Six Sigma, TQM
- formal methods and concurrency theory, machine learning
- optimization, operations (management and research),
- process automation and work-flow management,
- stochastic process and analysis, as Markov models,
- statistical inference and optimization, time series model and analysis, and so on.

Data science revisited, aims to answer the following four questions.

1. Reporting: What happened?
2. Diagnosis: Why did it happen?
3. Prediction: What will happen?
4. Recommendation: What is the best that can happen?

- However, without incorporating the methods of PS with those of DS, the answers of the above key fours could be **not fully** adequate.

[3] In Data Science and Data Mining (now a mature discipline) we have mostly focused on few dimensions, say Content and Locations, of the newly defined fours.



Process mining as the bridge between data science and process science

- To adequately answer such questions there is **not just** a need for raw data and computing power.
- Expertise in data+process mining: probability/statistics/stochastics, inference and causality analysis, and visualization
  to efficiently decode diverse complex datasets from biology, chemistry, computing, environment,
  epidemiology, medicine, nanotech, plant science, urban traffic, virology... are vital.

A brief spectrum of techniques for old and new applications

PM only recently emerged as a subdiscipline of both data science and process science, but the corresponding techniques can be applied to any type of operational processes (organizations/systems), as

- analyzing treatment processes in hospitals,
- improving customer service processes in a multinational corporation,
- understanding the behavior of customers choosing an insurance firm,
- improving the efficacy of a new vaccine to cope with fatal pandemics, [by firstly well

knowing lab's experimental designs, better modeling data obtained from labs/field trips/factories, in order to perform convincing statistical analyses, and finally to make meaningful guidelines or (nearly) optimal decisions].

A fact-based conclusion: The coworkers must know things at process level (at least at dimensions of EVENT DATA) to do the right thing!

5.1.3 *HOW to bring PM to life? Main challenges*

Possibly to be figured out by the upcoming interdisciplinary teams, or center built up to cope with urgent problems, to maintain Univ. X ...

What are key challenges?

I) Define the PM research spectrum (breath of topics),

II) Create a consistent and explicit process model given an *event log*,

III) Diagnose issues observing dynamic behavior with the use of tools?

1. Furthermore, processes and information need to be aligned perfectly in order to meet requirements related to compliance, efficiency, and customer service. How to define and build clear causal bonds so that decisions could be made?
2. The identification of issues and diagnoses needs explore the causal and casual (occasional) relations between activities, and this functionality is **not** present in a traditional *Workflow Management System* (WFMS) or Business Process Management System (BPMS).

First of first, we need to provide an overview of development history and then

describe the **process mining spectrum** (research topics).

5.1.4 *Process Mining - A brief development history and Top applications*

See a brief story in ref R2 summarizing these previous studies between 2003 and 2018. Notably

- 2015 - *Compliance monitoring in business processes*: Functionalities, application, and tool-support

- 2016 - *The State of the Art of Business Process Management Research*: related to research methods, quality discussion, maturity, citations index, and progress in the business process management
- 2016 - *Process mining in healthcare*, Biomedical Informatics.
- 2018 - *A systematic mapping study of process mining*: maps the relationship between data mining tasks in the process mining context...

Top seven areas with $> 80\%$ publications related to PM applications [ref. R2]:

α **Healthcare:** covering clinical path, patient treatment, or the primary processes of a hospital

β **ICT:** related to software development, IT operation services

γ **Manufacturing:** in industrial activities, realized by a factory that usually receives material and delivers partial or finished products

δ **Education:** e-learning, scientific applications, and centers with innovation process management.

ε **Finance and** η **Logistics:** see more in R1 and R2

λ **Robotics / Smart:** applications using advanced technologies related to smart buildings, industry 4.0...

5.1.5 *Process Mining (PM) key research topics*

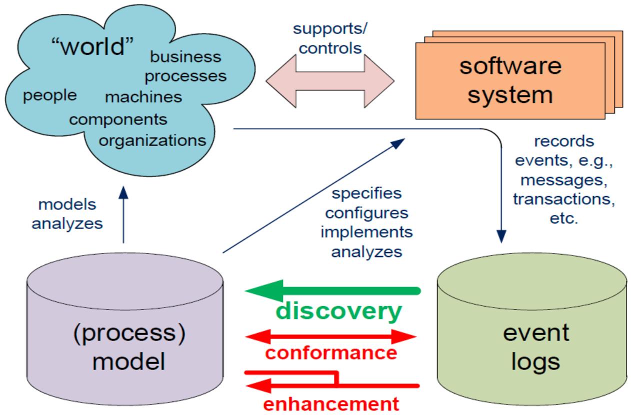
1. Process Discovery
2. Process Conformance
3. Process Enhancement... (kindly see ref. R1 and R2)

From the Quality Engineering view, including process control and improvement, PM **does not** replace the traditional process improvements methods, such as Business Process Improvement (BPI),

Continuous Process Improvement (CPI), Corporate Performance Management (CPM), Total Quality Management (TQM), Six Sigma, and others.

However, process miners are able (a) to check compliance, diagnose deviations, point out bottlenecks, and then (b) to perform, integrate, accelerate process improvements, as well as recommend actions and, last but not least redesign systems.

Process Discovery [see R1]



Process Discovery

The process discovery should balance four competing quality criteria:

simplicity, fitness (able to replay event log)

precision (avoid underfitting) and generalization (not overfitting the log).

5.1.6 *Conclusion*

1. Process mining is a new research discipline as well as cutting edge technology enabling evidence-based process analysis. The three basic types of process mining currently are Process Discovery, Process Conformance and Process Enhancement.
2. Nevertheless, there are still many open scientific challenges and most end-user organizations are not yet aware of the potential of process mining.
3. To cope with new real live challenges, IEEE, with Data Mining Technical Committee of the Computational Intelligence Society (CIS) created a **Task Force on Process Mining** since 2010s, see more at `https://www.tf-pm.org/`.

4. Two major concerns in process mining - Concurrency and Causality will be studied. For concurrency (i.e., parallelism), one of the most essential problems in PM, we are going to investigate in details the methodology and key techniques of **Petri Net** in next sections. The matter of causality will be the subject in Chapter **??**.

PETRI NETWORKS

[[42]]

5.2 PETRI NETWORKS- Background

5.2.1 *The Art of Modeling- motivated from Operation Research*

Operation research (OpRe), is a branch of management science heavily relying on **modeling**. Here a variety of mathematical models ranging from

- (I. deterministic modeling): integer linear programming, dynamic programming, to

- (II. stochastic modeling): Markov chains, queueing models, to stochastic dynamic programming, and
- (III. mixed type one): as simulation [discrete event simulation, MCMC simulation].

ELUCIDATION

- Models are used to reason about *processes* (redesign) and to make decisions *inside processes* (planning and control). The models used in operations management are typically tailored towards a particular analysis technique and only used for answering a specific question.

  In contrast, process models in `Business Process Management` typically serve *multiple purposes*.
- However, creating such models is therefore a difficult and delicate task, since concurrency exists. In such complex process models, **concurrency** must be handled properly. The most popular theory for studying concurrency is **Petri Net**, named after the German Carl Adam Petri. [4]

IMPACTS: Petri nets nowadays have brought engineers a breakthrough in their treatment of *discretely controlled systems*. Petri nets are a key to solve the design problem, as

[4] `Carl Adam Petri` (1926- 2010), the first computer scientist to identify *concurrency* as a fundamental aspect of computing (sketched largely in his seminal PhD thesis, title **Communication with Automata**, submitted to the Science Faculty of Darmstadt Technical University in 1962, where in fact he outlined a whole new foundations for computer science). Petri's father was a serious scholar. He had a PhD in mathematics and had met Minkowski and Hilbert.

Carl Adam Petri (1926- 2010)

A German scientist and engineer,
one of the renown pioneers in *Computing*,
and a visionary who founded an extraordinarily
fruitful domain of study in the field of
distributed discrete event systems.

[*courtesy* Karsten Wolf, [39]]

this is the first technique to allow for a unique description, as well as powerful analysis of **discrete control systems**.

■ **CONCEPT 1.** *WHAT is Petri Net (PetN)?*

A **Petri net** is a graphical tool [a bipartite graph consisting of *places* and *transitions*] for the description and analysis of *concurrent processes* which arise in systems with many components (distributed systems). The graphics, together with the rules for their coarsening and refinement, were invented in August 1939 by Carl Adam Petri.

5.2.2 *Formal definitions*

Definition 5.1 (Transition system or State transition system).

A transition system is a triplet $TS = (S, A, T)$ where S is the set of *states*, $A \subseteq \mathcal{A}$ is the set of *activities* (often referred to as *actions*), and $T \subseteq S \times A \times S$ is the set of *transitions*.

The following subsets are defined implicitly,

- $S^{start} \subseteq S$ is the set of *initial states* (sometimes referred to as 'start' states), and $S^{end} \subseteq S$ is the set of *final states* (sometimes referred to as 'accept' states).

For most practical applications the state space S is finite. In this case the transition system is also referred to as a Finite-State Machine or a finite automaton (FA), see Section **??**.

- WHY transition systems? The goal of (using transition systems in) a **process model** is to decide *which activities* need to be executed and in *what order*.
  Activities can be executed sequentially, activities can be optional or concurrent, and the repeated execution of the same activity may be possible.
- BEHAVIOR of a transition system: can be studied and expressed via its net structure and dynamic. The transition starts in one of the initial states. Any path in the graph starting in such a state corresponds to a possible *execution sequence*.
  * A path *terminates successfully* if it ends in one of the final states.
  * A path *deadlocks* if it reaches a non-final state without any outgoing transitions.
  (Note that the absence of deadlocks does not guarantee successful termination).

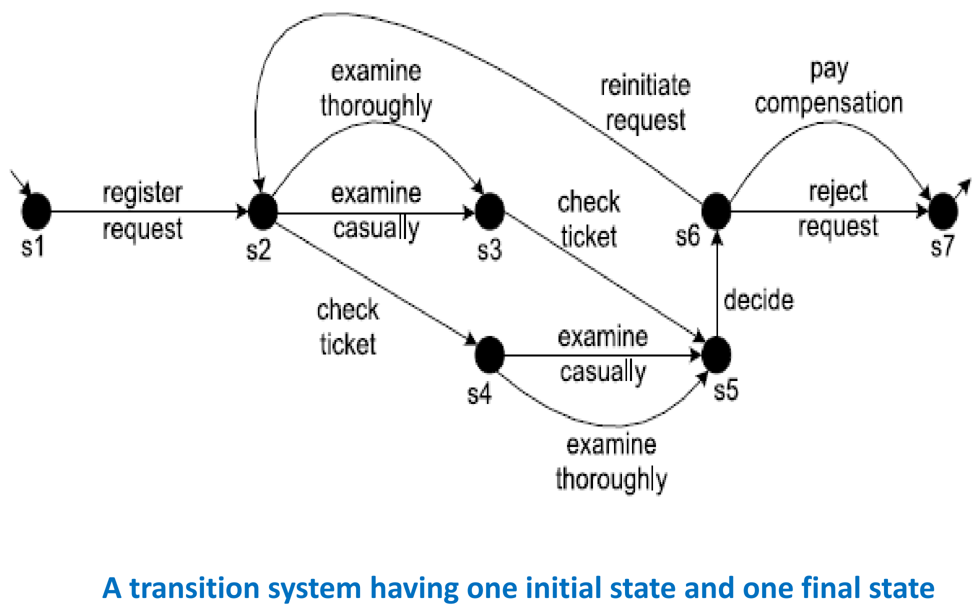


Figure 5.2: *A small size transition system*

♦ EXAMPLE 5.1.

Observe a transition system in Figure 5.2 we see the state space $S = \{s_1, s_2, \dots, s_7\}$.

Here, $S^{start} = \{s_1\}$, $S^{end} = \{s_7\}$. Could the reader fill in the set of *activities*

$A = \{$ register request, examine thoroughly, examine casually, $\dots\}$, and determine the set $T = \{(s_1, \text{register request}, s_2), (s_2, \text{examine casually}, s_3), \ldots\}$ of all *transitions*?

■

NOTES on using transition system:

1. **Any process model with executable semantics** can be mapped onto a transition system. Therefore, many notions defined for transition systems can easily be translated to higher-level languages such as Petri nets...
2. Transition systems, however, are simple but have problems expressing concurrency succinctly, as 'state explosion'. But a **Petri Net** can be used much more compactly and efficiently.

3. Indeed, suppose that there are n parallel activities, i.e., all n activities need to be executed but any order is allowed.
   There are $n!$ possible execution sequences. The transition system requires 2^n states and $n \times 2^{n-1}$ transitions. When $n = 10$, the number of reachable states is $2^n = 1024$, and the number of transitions is $n \times 2^{n-1} = 5120$. A **Petri Net** needs only 10 transitions and 10 places to model the 10 parallel activities.

On the set of all multisets $\mathcal{M}$ over a domain D

Given a finite domain $D = \{x_1, x_2, \cdots, x_k\}$, a map $X : D \longrightarrow \mathbb{N}$ defines a multi-set on D as follows: for each $x \in D$, $X(x) = m$ denotes the number of times x is included in the multi-set, i.e.,

$$M = \{\underbrace{x_1, \ldots, x_1}_{m_1\ times}, \cdots, \underbrace{x_k, \ldots, x_k}_{m_k\ times}\}. \tag{5.1}$$

Evidently, $support(M) \subseteq D$ and we could use multiplicative format for

$$M = \{x_1^{m_1}, x_2^{m_2}, \ldots, x_k^{m_k}\},$$

and so M is identified with the list $[m_1, m_2, \ldots, m_k]$. Here frequencies $m_i \geq 0$, $m_i = 0$ means that

x_i does not appear in M, and $support(M)$ consists of different elements in the multi-set M.

♦ **EXAMPLE 5.2.**

A multi-set (also referred to as *bag*) is like a set in which each element may occur multiple times, and the order is **not** matter.

Given domain $D = \{a, b, c, d, e\}$, $k = |D| = 5$, we observe a multi-set with $n = 9$ elements: one a, two b's, three c's, two d's, and one e: $M = [a, b, b, c, c, c, d, d, e] = \{a, b^2, c^3, d^2, e\} = \{e, d^2, c^3, b^2, a\} \equiv [1, 2, 3, 2, 1]$.

Definition 5.2 (**Petri Net** is a bipartite directed graph N of *places* and *transitions*)**.**

A Petri net is a triplet $N = (P,\ T,\ F)$ where P is a finite set of *places*,
T is a finite set of *transitions* such that $P \cap T = \emptyset$, and
$F \subseteq (P \times T)\ \cup (T \times P)$ is a set of directed arcs, called the *flow relation*.

1. A *token* is a special *transition node*, being graphically rendered as a black dot, •. The symbolic tokens generally denote elements of the real world. Places can contain tokens, and transitions **cannot**.
2. A transition is *enabled* if each of its input places contains a **token**.
   [Example: node `enter` in figure 5.3 has input places `wait` and `free`.]
   An **enabled transition** can *fire*, thereby consuming (*energy* of) one token [e.g., node `enter` in figure 5.3] from each *input place* and producing at least one token for each *output place* next. consumed one from each to produce at least one for each
3. A *marking* is a distribution of tokens across places. A *marking* of net N is a function $m : P \to \mathbb{N}$, assigning to each place $p \in P$ the number $m(p)$ of tokens at this place. [e.g. $m(\texttt{wait}) = 3$.]
   Denote $M = m(P)$, the range of map m, viewed as a multiset.
4. A *marked* Petri net is a pair (N, M), where $N = (P, T, F)$ is a Petri net and where M is a *multi-set* (or *bag*, defined generally in Equation 5.1) over P denoting the *marking* of the net.
   ♦ We write the set of all multisets over P as $\mathcal{M}(P)$ or $\mathcal{M}$ for short.
   ♦ The set of all marked Petri nets is denoted $\mathcal{N}$.

$P \to T$

$T \to P$

~~$T \to T$~~

~~$P \to P$~~

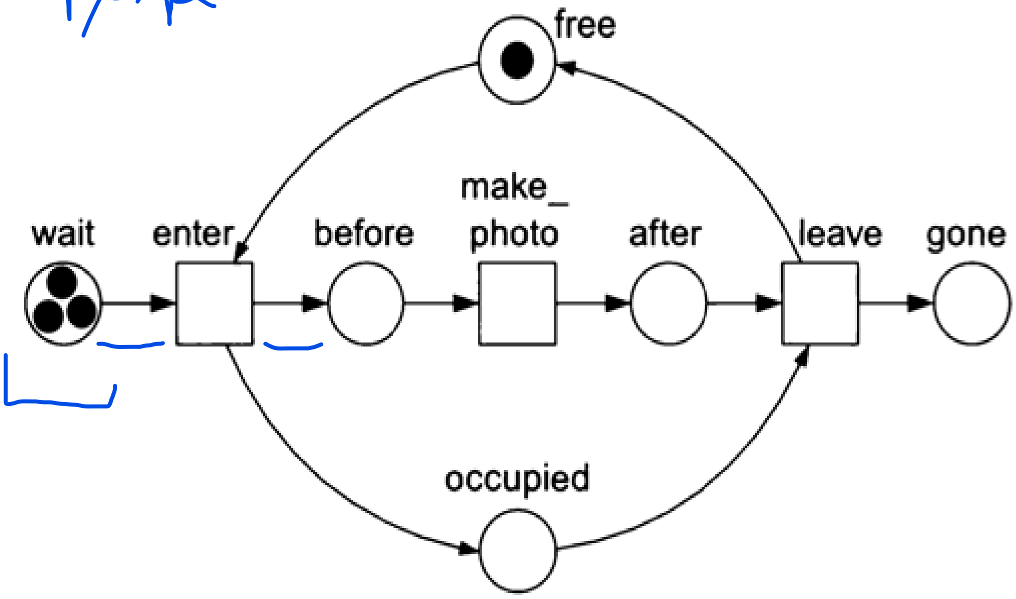


Figure 5.3: *A Petri net for the process of an X-ray machine*

♦ **EXAMPLE 5.3.**

The Petri net in figure 5.3 has three transitions (drawn as □): $T = \{enter,\ make-photo,\ leave\}$.

- Transition `enter` is **enabled** if there is at least one token in place `wait` and at least one token in place `free`. In the marking of this net, these conditions are fulfilled. Transition `make-photo` is enabled if place `before` holds at least one token. This condition is **not** fulfilled.
- List the places P, and give the marking of the Petri Net in figure 5.3.
- Determine fully the flow relation $F \subseteq (P \times T) \cup (T \times P)$. ■

ELUCIDATION

1. PLACES: In a Petri net, graphically, a place $p \in P$ is represented by a circle or ellipse. A place p always models a *passive* component:
   p can store, accumulate or show things. A place has discrete states.

2. TRANSITIONS: The second kind of elements of a Petri net are *transitions*.

Graphically, a transition $t \in T$ is represented by a square or rectangle.

A transition t always models an *active* component:

t can produce things/tokens, consume, transport or change them.

After each firing of a transition [consuming energy of token] the tokens are reallocated on places, henceforth building up the dynamic of **Petri net**.

A redistribution or reallocation of tokens across places is a **marking** [Figure 5.4].

3. ARCS: Places and transitions are connected to each other by directed arcs, graphically, represented by an arrow. An arc **never** models a system component, but an abstract, sometimes only notional relation between components such as *logical connections*, or *access rights*.
4. OPERATIONS: The sum of two multi-sets ($A \uplus B$), the difference ($A \setminus B$), the presence of an element in a multi-set ($x \in M$), and the notion of subset ($X \leq Y$) are defined in the classic way of set theory. [5]

♦ **EXAMPLE 5.4.**

Figure 5.4 shows a **Petri net** with three different markings.

Find the places P, and give the transitions T of the net.

Write down completely three different markings in format of lists or tables.

Definition 5.3 (Input is place, output is transition.).

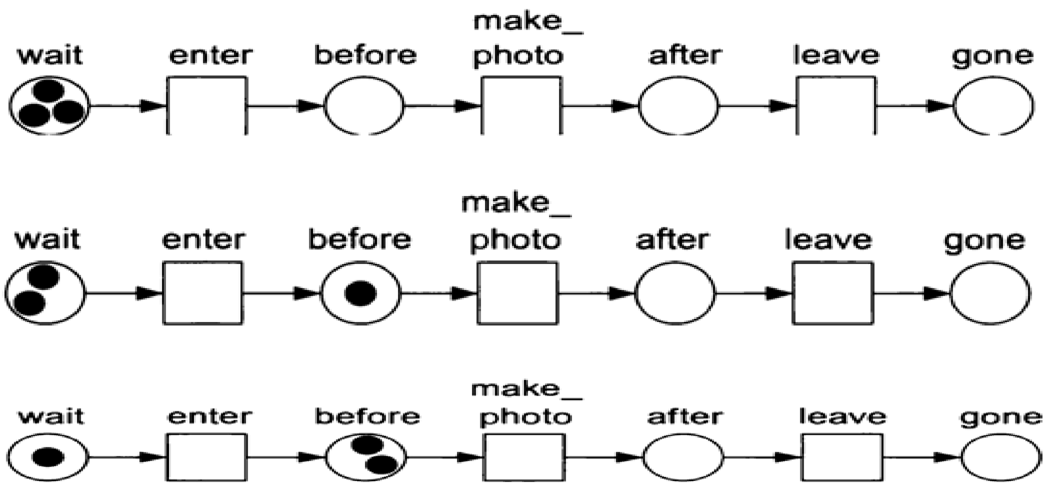
Let $N = (P, T, F)$ be a Petri net. Elements of $P \cup T$ are called nodes.

- A node x is an *input node* of another node y if and only if there is a directed arc from x to y (i.e., $(x, y) \in F$). Node x is an *output node* of y if and only if $(y, x) \in F$.

set of arcs

- For any $x \in P \cup T$, write $\bullet\, x = \{y |\ (y, x) \in F\}$- the preset of x, and $x \bullet = \{y |\ (x, y) \in F\}$- the postset x.

[5] What is multi-set used for? A marking corresponds to a multi-set of **tokens**. However, multi-sets are **not only** used to represent markings; later we will use multi-sets to model *event logs* where the same trace may appear multiple times.



The first three markings in a process of the X-ray machine

a) [top, **transition *enter*** not fired**];** b) [middle**, transition *enter*** fired**];** and

c) [down, **transition *enter*** has fired again**]**

Figure 5.4: *Three different markings on a Petri net, modeling of a process of an X-ray machine*

- Now for any set X, define X^* to be the set of sequences containing elements of X, i.e., for any $n \in \mathbb{N}$ and $x_1, x_2, \cdots, x_n \in X$: $(x_1, x_2, \cdots, x_n) \in X^*$.

Q: Can you give $\bullet\, c1 =?$, $c5 \bullet =?$ in Figure 5.5.

5.2.3 *On Enabled transition and Marking changes*

♦ **EXAMPLE 5.5.**

The marked Petri net in Figure 5.5 has the marking with only one token, node **start**.

- Hence, transition **a** is enabled at marking [start], now a becomes new token (with full energy)!
- Firing **a** results in the marking [`c1`, `c2`]: one token is consumed and two tokens are produced.

At marking [c1, c2], transition **a** is no longer enabled (spent all energy now). However, transitions **b**, **c**, and **d** have become enabled.

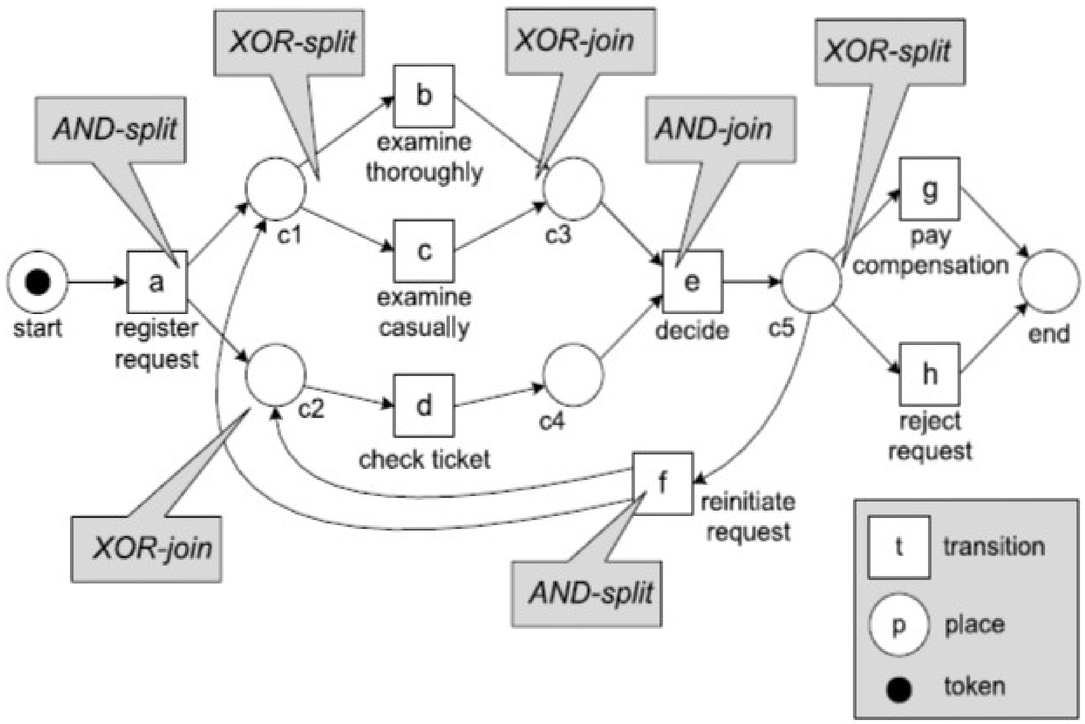


Figure 5.5: *A marked Petri net with one initial token*

[*courtesy* Wil van der Aalst, [**?**]]

- From marking c1, c2, firing **b** results in marking [c2, c3].
  Here, **d** is still enabled, but **b** and **c** not anymore.
  Because of the loop construct involving f there are infinitely many firing sequences starting in [*start*] and ending in [*end*].

- Now with multiple token at the beginning, assume that the initial marking is: [$start^5$].
  Firing **a** now results in the marking [$start^4$, c1, c2]. At this marking **a** is still enabled.
  Firing **a** again results in marking [$start^3$, $c1^2$, $c2^2$].
  Transition **a** can fire five times in a row resulting in marking [$c1^5$, $c2^5$].

- Note that after the first occurrence of **a**, also **b**, **c**, and **d** are enabled and can fire concurrently. ■

PRACTICE 5.1.

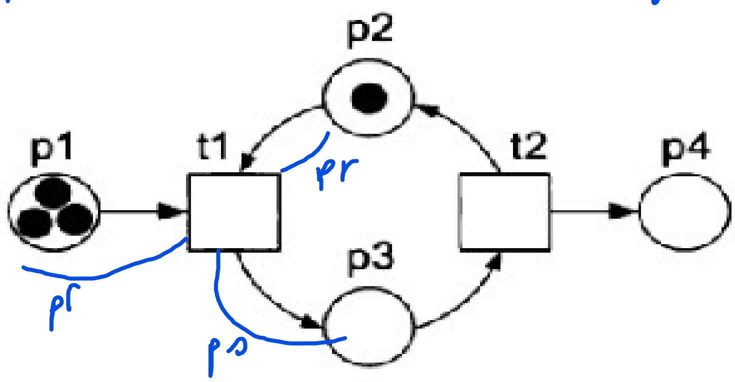
$\forall x \in P \cup T$: Preset of x is defined st
$x = \{y \mid (y,x) \in F\}$

Consider the Petri net in figure below.

Poset of x is defined st:
$x = \{y \mid (x,y) \in F\}$

t1: preset : (p1,t1) (p2,t1)
poset : (t1, p3)

t2: preset (p3,t2)
poset : (t2,p2)
(t2, p4)



A Petri net showing transitions t1 and t2.

Figure 5.6: *A simple Petri net, with only two transitions*

1. Define the net formally as a triple (P, T, F).
2. List presets and postsets for each transition.
3. Determine the marking of this net. $(p_1^3\, p_2)$
4. Are the transitions t1 and t2 enabled in this net.

only t1 is enabled

5.2.4 *Important usages of* **Petri net** *via explaining Figure 5.5*

Important usages of Petri nets: Information systems (IS) and business process modeling (BPM).

Business Process (BP)

- An *organization* is a system consisting of humans, machines, materials, buildings, data, knowledge, rules and other means, with a set of goals to be met. Most organizations have, as one of their main goals, the *creation* or *delivery* of (physical) *products* or (abstract) *services*.

- The creation of services and products is performed in *business processes* (BP). A BP is a set of *tasks* with *causal dependencies* between tasks.

Task ordering principles The five basic task ordering principles (Figure 5.5) are

1. *Sequence* pattern: putting tasks in a linear order;
2. *Or-split* pattern: selecting one branch to execute;
3. *And-split* patterns: all branches will be executed;
4. *Or-join* patterns: one of the incoming branches should be ready in order to continue; and
5. *And-join* pattern: all incoming branches should be ready in order to continue.

Execution of tasks For the execution of tasks resources are required.

- Resources can either be *durable* or *consumable*. The first kind is available again after execution of one or more tasks, like a catalyst in a chemical process. Typical examples of this kind are

  humans, machines, computer systems, tools, information and knowledge.

  Consumable resources disappear during the task execution. Examples are energy, money, materials, components and data. ?

- The results or output of a task can be considered as resources for subsequent tasks or as final products or services. Two kinds of durable resources are of particular importance: the humans as a resource, called *human resources*, and information and knowledge, which we will call *data resources*.

- Since human activities are sometimes replaced by computer systems, we use the term *agents* as a generic term for human and system resources.

Information Systems and Modeling Business Processes

Definition 5.4. *To study* **Petri net** *we first formalize the above concepts.*

1. A **business process** consists of a *set of activities* that is performed in an organizational and technical environment. These activities are coordinated to jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.
2. An **information system** is a software system to capture, transmit, store, retrieve, manipulate, or display information, thereby supporting people, organizations, or other software systems.

The awareness of the importance of business processes has triggered the introduction of the concept of *process-aware information systems*. The most notable implementations of the concept of process-aware information systems are workflow management systems. A workflow management system is configured with a **process model**, its graphical visualization is **workflow net**.

SUMMARY 1.

A **Petri net** is a triplet structure (P, T, F). The structure of a Petri net is determined if we know the places P, the transitions T, and the flow relation F of the ways in which they are connected with each other (i.e. arcs connecting places and transitions.).

1. A **Petri net** contains zero or more places. Each place has a unique name, the place label. We can describe the places of a Petri net by the set P of *place labels*.
   We can describe the transitions in a Petri net in the same way. Each transition has a unique name, the transition label.
   We describe the transitions of a Petri net by the set T of *transition labels*.

2. **Transitions** are the *active* nodes of a Petri net, because they can change the marking through firing. We therefore name transitions with *verbs* to express action.
   E.g., see node **b**, **c**, and **d** in Figure 5.7.
   Places are the *passive* nodes of a Petri net because they **cannot** change the marking.
   We name places using nouns, adjectives, or adverbs.

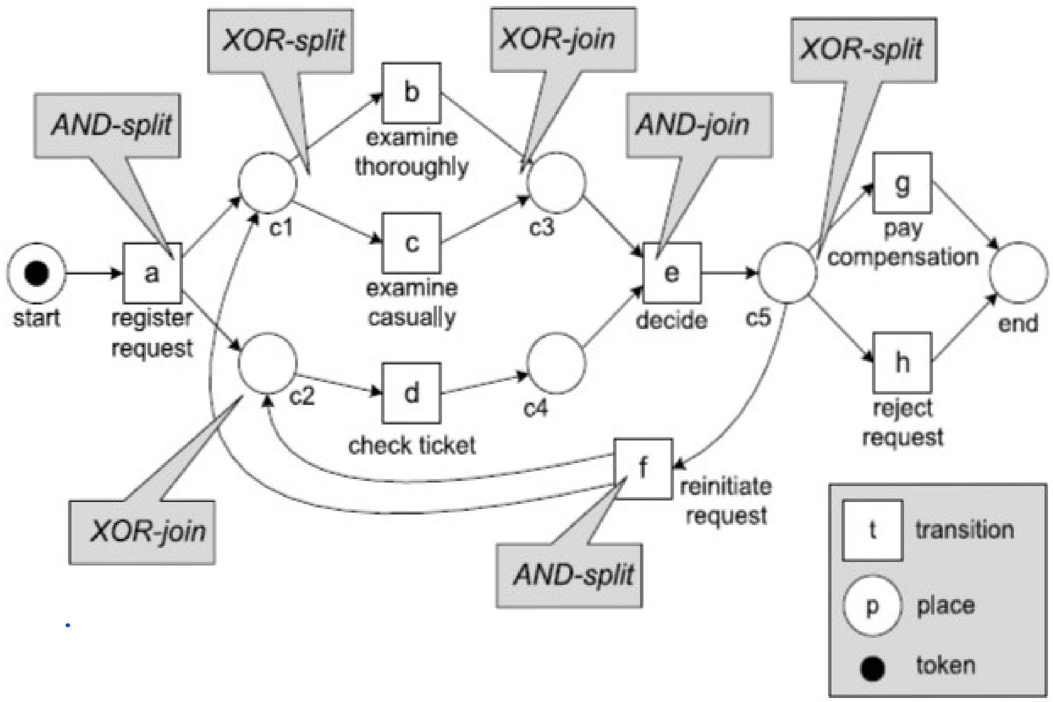
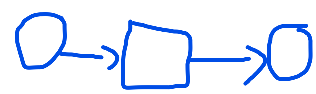


Figure 5.7: *A marked Petri net with one initial token*

3. In addition to the places and transitions, we must describe the *arcs*. Like a transition in a transition system, we can represent an arc as an ordered pair (x, y). The set of arcs is a *binary relation*.



As a **Petri net** has two kinds of arcs, we obtain two binary relations.
(i) The binary relation $R_I \subseteq P \times T$ contains all arcs connecting transitions and their input places.
(ii) The binary relation $R_O \subseteq T \times P$ contains all arcs connecting transitions and their output places.
The union $R_I \cup R_O$ represents all arcs of a Petri net. This union is again a relation

$$F = R_I \cup R_O \subseteq (P \times T) \cup (T \times P) \tag{5.2}$$

called the **flow relation**, where $(p, t) \in F$ defines the arc from p to t.

4. **Labeling of Arcs and Transitions**: Arcs and transitions can be labeled with *expressions* (for instance, $-$, a subtraction, and variables x and y). These expressions have two

central properties:

(i) If all variables in an expression are replaced by elements, it becomes possible to evaluate the expression in order to obtain yet another element.

(ii) The variables in these expressions are *parameters* describing different instances ("modes") of a transition. Such a transition can only occur if its labeling evaluates to the logical value '**true**'.

5. We can summarize the possible roles of tokens, places, and transitions with the following modeling guideline:

   We represent events as transitions, and we represent states as places and tokens. We represent the evolving states of a system by the distribution of tokens over the places. Each token in a place is part of the state. A token can model a physical object, information, or a combination of the two, but it can also model the state of an object or a condition.

   In the remainder we only consider Petri nets (special class of workflow nets), to model BPs. For many purposes it is sufficient to consider classical Petri nets, i.e. with '**black**' tokens. **The modeling power of *colored Petri nets* is postponed till next chapters.**

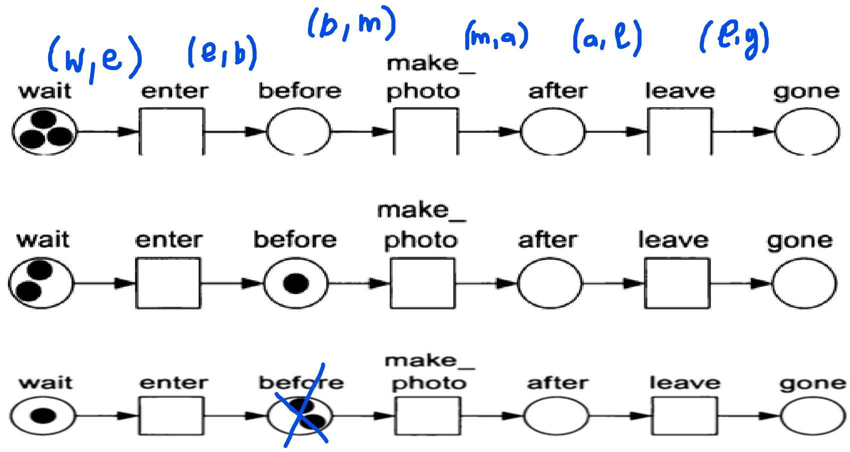
Practical Problem 1.

Given a process of a X-ray machine in which we assume the first marking in Figure 5.8.a shows that there are three patients in the queue waiting for an X-ray. Figure 5.8.b depicts the next marking, which occurs after the firing of transition enter.

1. Determine the two relations R_I and R_O, and the flow relation $F = R_I \cup R_O$. [HINT: find sets P, T.]
2. A patient may enter the X-ray room only after the previous patient has left the room. We must make sure that places **before** and **after** together do not contain more than one token.

   There are two possible states: the room can be **free** or **occupied**. We model this by adding theses two places to the model, to get the improved the Petri net, in Figure 5.9.

R_0: w b a g R_1: e m l



The first three markings in a process of the X-ray machine

a) [top, **transition *enter*** not fired**]**; b) [middle**, transition *enter*** fired**]**; and

c) [down, **transition *enter*** has fired again**]**

Figure 5.8: *A Petri net model of a business process of an X-ray machine*

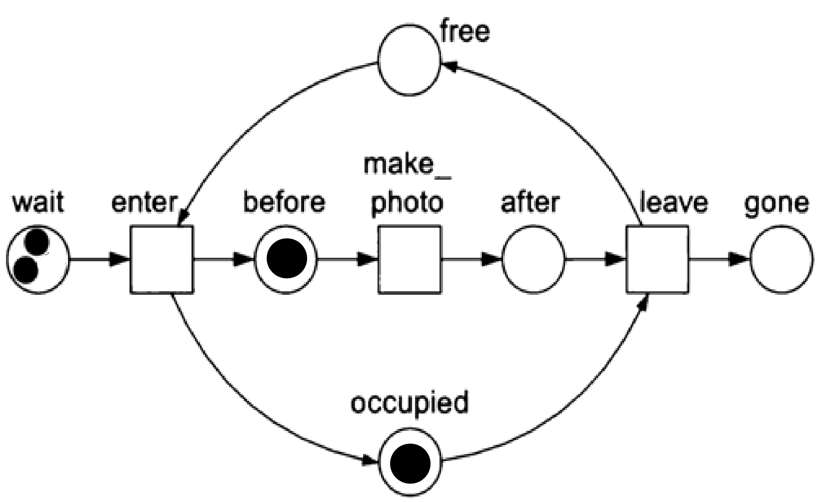


Figure 5.9: *An improved Petri net for the business process of an X-ray machine*

Now for this Petri net, can place **before** contain more than one token? Why? Rebuild the set P of place labels.

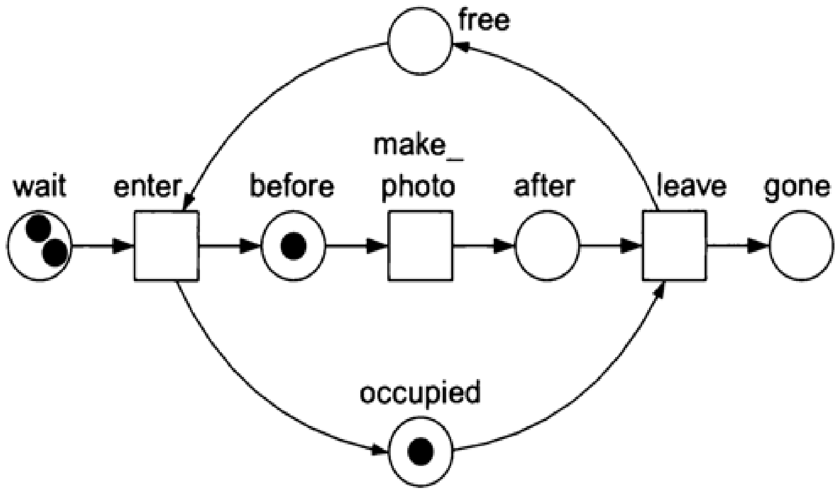


Figure 5.10: *The marking of the improved Petri net for the working process of an X-ray room after transition* ***enter*** *has fired.*

They can't

3. As long as there is no token in place **free** [Figure 5.10], can transition **enter** fire again? Explain why or why not. Remake the two relations R_I and R_O.

BC after firing the first token the free place is empty hence transition enter is disabled

The Behavior of Petri Nets

[[42]]

The behavior of a **Petri Net** is defined by the net structure, the distribution of tokens over the places P, and the firing of transitions T.

5.3 PETRI NETWORKS- Behaviors

5.3.1 *From Firing, Reachability to Labeled Petri net*

A token is graphically rendered as a black dot in the graph of a Petri net.

Definition 5.5 (*Firing rule*).

Let $(N, M) \in \mathcal{N}$ be a marked Petri net with $N = (P, T, F)$ and $M \in \mathcal{M}$.

- A transition is enabled if there is **at least one token in each of its input places**.
- Transition $t \in T$ is *enabled* at marking M, denoted $(N, M)\ [t\rangle$, if and only if $\bullet\, t \leqslant M$.
- The firing rule $\alpha\ [t\rangle\ \beta \subseteq \mathcal{N} \times T \times \mathcal{N}$ is the smallest relation satisfying

$$(N, M)[t\rangle \implies \underbrace{(N, M)}_{\alpha}\ [t\rangle\ \underbrace{(N, (M \setminus \bullet\, t)\ \uplus\ t\ \bullet)}_{\beta} \tag{5.3}$$

for any $(N, M) \in \mathcal{N}$ and any $t \in T$.

♣ **OBSERVATION 1.**

1. Places can contain tokens, transitions cannot. But transitions can change the marking through firing. That is places are passive, and transitions are active.
   To fire a transition, it must be enabled.

2. **Enabledness**: A marking $M = [m_1, m_2, \ldots, m_k] \equiv m : P \to \mathbb{N}$, by (5.1), has the form

$$M = \{\underbrace{x_1, \ldots, x_1}_{m_1\ times},\ \cdots, \underbrace{x_k, \ldots, x_k}_{m_k\ times}\},$$

   so we can equivalently say $t \in T$ is *enabled* at M
   if and only if for all $p \in \bullet\, t,\ m(p) > 0$.

3. An enabled transition t can fire, thereby changing the marking M to a marking M_1.

If an enabled transition fires, then it consumes one token from each of its input places and produces one token in each of its output places.

(P, T, F)

4. $(N, M)[t\rangle$ means that transition t is enabled at marking M.
   E.g., $(N, [start])\ [a\rangle$ in Figure 5.11.

5. $(N, M)\ [t\rangle\ (N, M_1)$ denotes that firing this enabled transition t results in marking M_1.

   For example, $(N, [start])\ [a\rangle\ (N, [c1, c2])$, how about $(N, [c3, c4])\ [e\rangle\ (N, [c5])$?

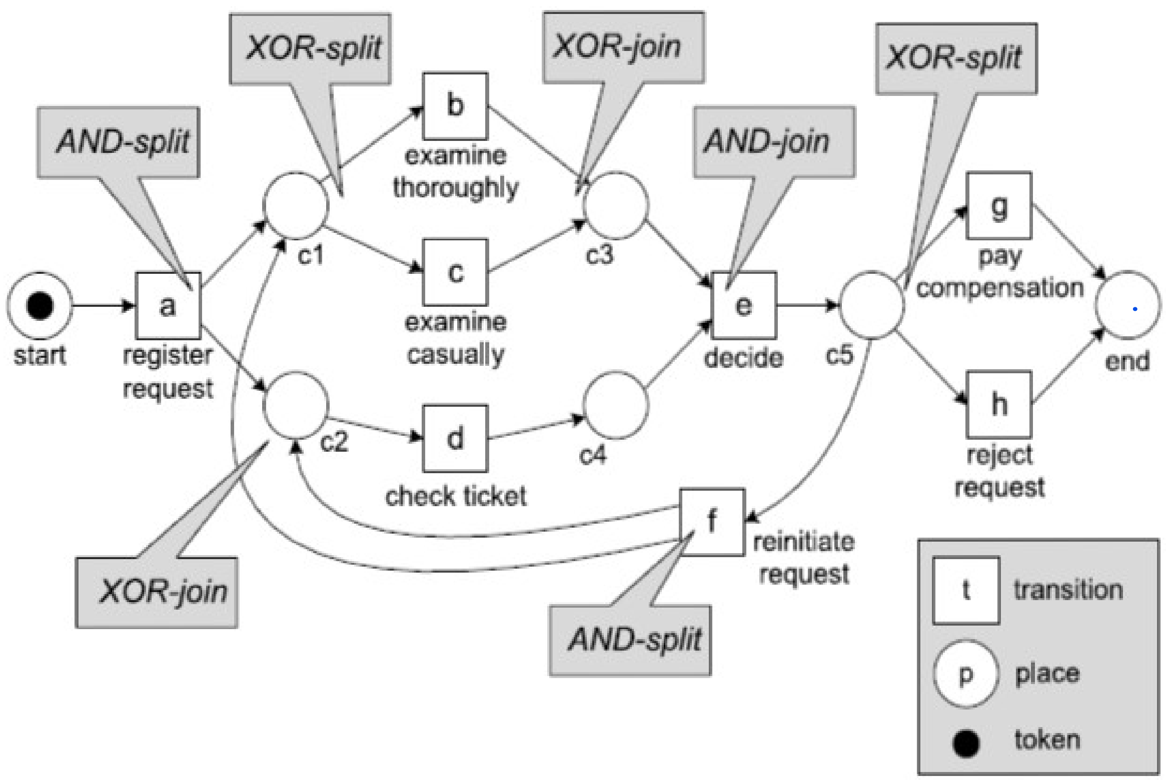


Figure 5.11: *A marked Petri net with one initial token*

Definition 5.6 (*Firing sequence*).

Let $(N, M_0) \in \mathcal{N}$ be a marked Petri net with $N = (P, T, F)$.

1. A sequence $\sigma \in T^*$ is called a *firing sequence* of (N, M_0) **if and only if**, for some natural number $n \in \mathbb{N}$, there exist markings $M_1, M_2, \ldots, M_n$ and transitions $T_1, T_2, \ldots, T_n$ such that
   - $\sigma = (t_1, t_2, \ldots, t_n) \in T^*$
   - and for all i with $0 \le i < n$, then $(N, M_i)[t_{i+1}\ \rangle$ and $(N, M_i)[t_{i+1}\ \rangle\ (N, M_{i+1})$.

2. A marking M is *reachable* from the initial marking M_0 **if and only if** there exists a sequence of enabled transitions whose firing leads from M_0 to M.
   The set of reachable markings of (N, M_0) is denoted $[N,\ M_0\rangle$.
   [E.g., the marked Petri net shown in Fig. 5.11 has seven reachable markings.]

3. (Petri net system) A Petri net system (P, T, F, M_0) consists of a Petri net (P, T, F) and a distinguished marking M_0, the *initial marking*.

♦ **EXAMPLE 5.6.**

In Fig. 5.11, write marking $M_0 = [start] = [1, 0, 0, 0, 0, 0, 0]$,
we get the marked Petri net (N, M_0) and see that

- The empty sequence $\sigma_0 = \langle\ \rangle$ - being enabled in (N, M_0)- obviously is a firing sequence of (N, M_0).
- The sequence $\sigma_1 = \langle a,\ b\rangle$ is also enabled in (N, M_0), and firing σ_1 results in marking $[c2, c3]$. We can write $(N, [start])\ [a\ b\rangle\ (N, [c2, c3])$ or $(N, M_0)\ [\sigma_1\rangle\ (N, [c2, c3])$.
- The sequence $\sigma_2 = \langle a,\ b,\ d,\ e\rangle$ is another possible firing sequence, and should we get $(N, M_0)\ [\sigma_2\rangle\ (N, [c5])$?
- Is $\sigma = \langle a,\ c,\ d,\ e,\ f,\ b,\ d,\ e,\ g\rangle$ a firing? What is the reachable marking M in the output $(N, M_0)\ [\sigma\rangle\ (N, M)$?
- Check that the set $[N,\ M_0\rangle$ (of reachable markings of (N, M_0)) has seven reachable markings. ■

Definition 5.7 (*Labeled Petri net*).

Often transitions are identified by a single letter, but also have a longer label describing the corresponding activity. A **labeled Petri net** with $N = (P, T, F, A, l)$ where (P, T, F) is a Petri net as defined in Definition 5.2.

- $A \subseteq \mathcal{A}$ is a set of *activity labels*, and the map $l \in \{L : T \longrightarrow A\}$ is a *labeling function*. [One can think of the transition label as the *observable action*. Sometimes one wants to express that particular transitions are **not** observable, or invisible.]
- Use the label τ for a special activity label, called 'invisible'. A transition $t \in T$ with $l(t) = \tau$ is said to be unobservable, silent or invisible.

♣ **OBSERVATION 2.**

1. The X-ray example in Practical Problem 1 illustrates that it is possible to go through several markings in a Petri net by a series of firings.
   The transitions keep firing until the net reaches a marking that **does not** enable any transition. Like the terminal state in transition systems [Definition 5.1], this marking is a *terminal marking*.
2. We may convert any Petri net into a labeled Petri: just take $A = T$ and $l(t) = t$ for any $t \in T$. The reverse is **not always** possible, (many transitions have the same label).

♣ **QUESTION 5.1.** *On markings in nets, when we have modeled a system as a* **Petri net** *system* (N, M_0) *(see Definition 5.6.3) then some matter occur, including*

1. How many markings are reachable?
2. Which markings are reachable?
3. Are there any reachable terminal markings?

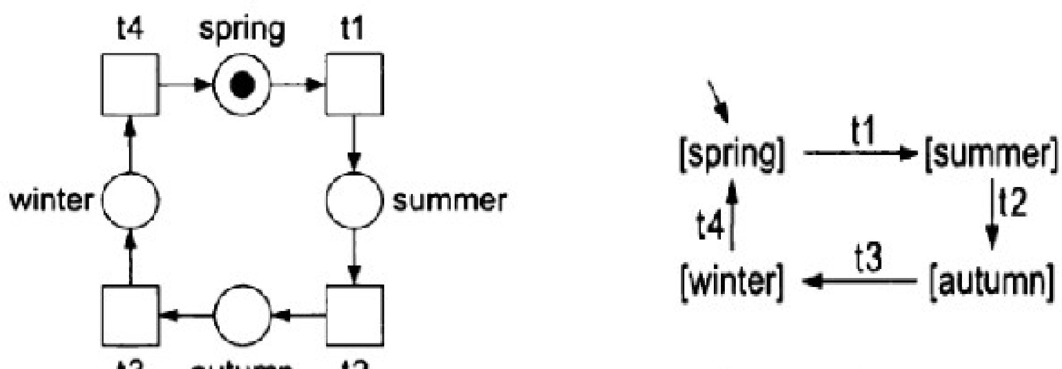
To answer these questions reachability graph is introduced

As we know the *initial marking* M_0 for the given system (N, M_0), we answer such questions by calculating the set of markings reachable from M_0. We represent this set as a graph- the **reachability graph** of the net. Its nodes correspond to the reachable markings and its edges to the transitions moving the net from one marking to another.

The key structure is **reachability graph** via transition systems.[6]

♦ **EXAMPLE 5.7** (Reachability graph).

Consider the **Petri net** system in Figure 5.12 modeling the four seasons.



(a) The four seasons

(b) Reachability graph

Figure 5.12: *A Petri net system and its reachability graph*

Recall that, each of the reachable markings is represented as a *multiset* (where the same element may appear multiple times). Multiset [spring] thus represents the marking in figure 5.12(a).

- The incoming edge **without source** pointing to this node denotes that this marking is the initial marking. We labeled each edge of the reachability graph on the right with the transition that fired in the corresponding marking.

- Figure 5.12(b) depicts the accompanying reachability graph that represents the set of markings that are reachable from the initial marking shown in figure 5.12(a). We can conclude that the net in figure 5.12(a) has four reachable markings.

[6]Transition system is the primal model of process modeling, they are the most elementary formalism with which we can describe systems.

- If a marking M is reachable from the initial marking M_0, then the reachability graph has a path from the start node to the node representing marking M. This path represents a sequence of transitions that **have to be fired** to reach marking M from M_0.

We refer to this transition sequence as a *run* (as an execution in finite automaton). A run is *finite* if the path and hence the transition sequence is finite.

Otherwise, the run is *infinite*.

The path from marking [`spring`] to marking [`winter`] is a finite run (t1, t2, t3) of the net in figure 5.12(a). Does it have infinite run? ■

5.3.2 *Representing Petri Nets as Special Transition Systems*

Our discussion shows that we can verify certain properties of a Petri net system by inspecting its reachability graph. For a simple Petri net system, it is easy to construct the accompanying reachability graph, but for more complex nets, reachability graphs can become huge, and it is possible to **forget markings**.

GENERIC AIM:

We describe the behavior of a Petri net system $(N, M_0) \equiv (P, T, F,\ M_0)$ as a state transition system (S, TR, S_0) by showing how to determine the state space S, the transition relation TR, and the initial state S_0 for the system (P, T, F, M_0).

Why are state transition systems suitable for representing Petri Nets?

The transition system represents the state space of the modeled system, thus representing all possible markings M of the net.

Definition 5.8 (Reachability graph).

Let (N, M_0) with $N = (P, T, F, A, l)$ be a marked labeled Petri net.

The set of reachable markings of (N, M_0) is denoted $[N, M_0\rangle$

(N, M_0) defines a transition system $TS = (S, A_1, TR)$ with

$S = [N, M_0\rangle$, $S^{start} = \{M_0\}$, $A_1 = A$, and

$TR = \Big\{(M, M_1) \in S \times S \mid \exists t \in T \quad (N, M)[\, t\rangle \quad (N, M_1)\Big\}$, or with label $l(t)$:

$$TR = \Big\{(M, l(t), M_1) \in S \times A \times S \mid \exists t \in T \quad (N, M)[\, t\rangle \quad (N, M_1)\Big\}. \tag{5.4}$$

TS is often referred to as the **reachability graph** of (N, M_0).

ELUCIDATION

- Elements of A in net N are *labels*, but when transformed to A_1 they are called *actions* in the output transition system TS.
- The initial state S_0 is defined by the initial marking : $S_0 = S^{start} = \{M_0\}$.
  The *markings* M (multisets) in N becomes the *states* in TS.
- The description of the transition relation TR for the Petri net system is more delicate. Let us consider two arbitrary states in state space S- that is, two markings M and M_1.
- Transition (M, M_1) is an element of the transition relation TR if there is a transition $t \in T$ enabled at marking M, and the firing of t in marking M yields marking M_1.

  We formalize this by defining transition relation TR as the set of all pairs $(M, M_1) \in S \times S$ satisfying

  $$\exists t \in T : \quad (N, M)[\, t\rangle\; (N, M_1).$$

  Otherwise, transition (M, M_1) is **not** possible, and (M, M_1) is **not** an element of TR.
- The set $\mathcal{M}$ (all markings) contains markings that are reachable from the given initial marking M_0, but also markings that are not. As a result, for a given marked Petri net (N, M_0), its reachability graph TS is a subgraph of the full transition system.

PRACTICE 5.2. *Build up the transition system TS generated from the labeled marked Petri net shown in Fig. 5.13.*

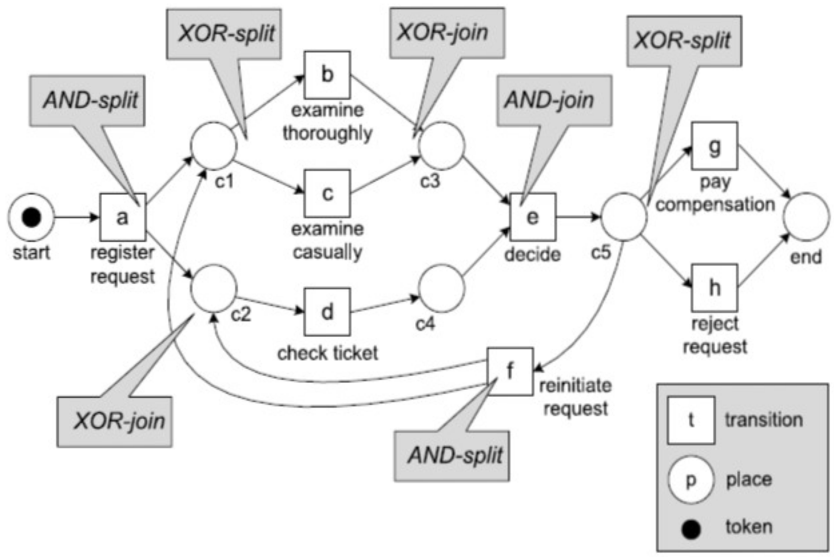


Figure 5.13: *A labeled marked Petri net*

HINT: DIY first, follow ideas in EXAMPLE 5.7.

States of TS correspond to reachable markings, i.e., multi-sets of tokens.

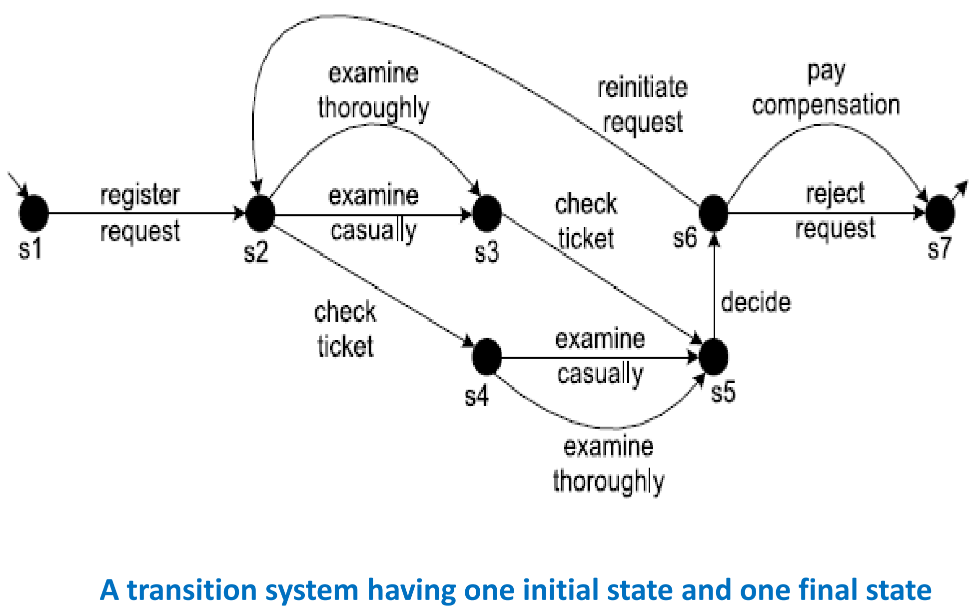


Figure 5.14: *The **reachability graph** TS of the above labeled marked Petri net*

Note that $S^{start} = \{[start]\}$ is a singleton containing the initial marking of the Petri net. The Petri net does not explicitly define a set of final markings S^{end}.

However, in this case it is obvious to take $S^{end} = \{[end]\}$.

The outcome should be as in Fig. 5.14,

but you must give specific values of $s2$, $s3$, $s4$, $s5$, $s6$ in terms of places ci. ■

5.4 PETRI NETWORKS - Structures and Basic Problems

Places and transitions in a Petri net are connected by *arcs*. Connecting of nodes determines the behavior of the network. Formally, the way in which transitions are connected determines the order in which they can fire. First we recall key terms, rules and relevant ideas, in a **Petri net** $N = (P, T, F)$.

1. When a transition t fires, the resulting number of tokens in any place p is equal to the initial number of tokens **minus** the number of consumed tokens plus the number of produced tokens.
2. The total number of tokens in the net changes if the number of input places of transition t is **not** the same as the number of output places of transition t. Accordingly, the *firing of a transition* may increase or decrease the overall number of tokens.
3. When several transitions are *enabled* at the same moment, it is **not** determined which of them will fire. This situation is a nondeterministic choice.

   Even though we **do not** know in this case which transition will fire, we know that one of them will be fired.

5.4.1 *Causality, Concurrency and Synchronization*

In general, transitions represent *events*. Let us assume that there are three events: x, y and z. We first discuss typical network structures to state that:

1. Event y happens after event x.
2. Event x and event y take place concurrently (at the same time or in any order).
3. Event z happens after both event x and y.

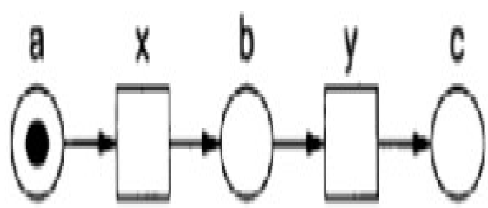
The first case of causality is defined by Item 1. and shown in figure 5.15(left).

■ **CONCEPT 2.**

1. **Causality** is formally understood as a relationship between two events in a system that must take place in a certain order. In a **Petri net** N, we may represent this relationship by two transitions connected through an intermediate place.

2. Concurrency (i.e., parallelism) is an important feature of (information) systems. In a concurrent system, several events **can occur simultaneously**. For example, several users may access an information system like a database at the same time.

- In Figure 5.15(right) transitions x and y can fire independently of each other; that is, there is **no causal relationship** between the firing of x and of y. With network structures, as those shown in Figure 5.15(right), we can model concurrency. In concurrent models, there is often a need for **synchronization**.
  Question: Can we find the reachability graph of the net N in Figure 5.15(right)? **Hint**: N has 4 places, initial marking is $M_0 = [1, 1, 0, 0]$, two transitions $x, y \in T$.



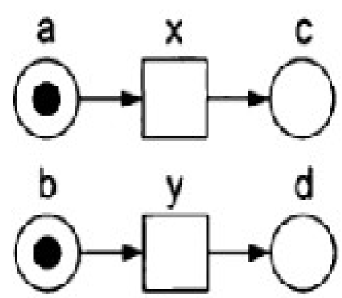


Figure 5.15: *Causality and Concurrency in a* **Petri net**

- We can model synchronization in a **Petri net** as a transition with at least two input places. In figure 5.16, transition z has two input places and **can only fire** after transitions x and y have fired.
  In industry or any process, assume that transitions x and y represent two concurrent production steps. Transition z can then represent an assembly step that can take place only after the results of the two previous production steps are available, see fig. 5.16.

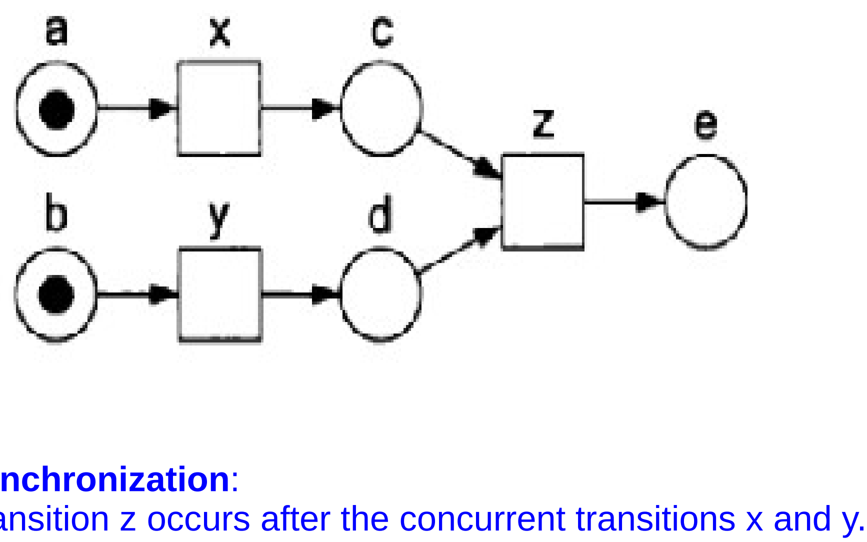


Figure 5.16: *Synchronization in a* **Petri net**

5.4.2 *Effect of Concurrency*

Figure 5.15(right) shows a simple concurrency occurring in a Petri net.

♣ **QUESTION 5.2.** *Could we quantify the concurrency for a given process or* **Petri net***?*

The answer is yes, and the tool is transition system. Remind that, a transition system is formally a triplet $TS = (S, A, T)$ where S is the set of *states*, $A \subseteq \mathcal{A}$ is the set of *activities* (often referred to as *actions*), and $T \subseteq S \times A \times S$ is the set of *transitions*.

Concurrency via transition systems

Fact 5.1.

If the model of a process contains a lot of concurrency or multiple tokens reside in the same place, then the transition system TS is much bigger than the Petri net $N = (P, T, F)$.
Generally, a marked Petri net (N, M_0) may have infinitely many reachable states.

Problem 5.3 illustrates this fact.

5.5 SUMMARY and REVIEWED PROBLEMS

After studying this chapter you should be able to:

1. Explain the terms
   "place," "transition," "flow relation," "token," "input place," "output place,"
   "preset," "postset," "enabled," "firing," "consumption," "production,"
   "nondeterminism," "initial marking," "terminal marking," and "reachable marking."

2. Model simple systems and business processes as Petri nets.
3. Draw the accompanying Petri net system when place set P, transition set T, flow relation F, and the initial marking M_0 are given.
4. Indicate for a given Petri net system
   which transitions are enabled, which transitions can fire, and
   which markings are reachable from a given marking.

PROBLEM 5.1. *Explain the following terms for Petri nets, and provide a specific example for each term:*

1. "enabled transition" : A transition is enabled meaning that it can consume tokens in it's presets to produce tokens in it's postsets
2. "firing of a transition"
3. "reachable marking," a marking that is potentialy produced from 1 marking via a transition
4. "terminal marking," and
5. "nondeterministic choice"?

when many transitions are enabled to fire at the same time

PROBLEM 5.2. *Consider the Petri net system in figure below.*

1. Formalize this net as a quadruplet $(\underbrace{P}_{\text{places}}, T, F, M_0)$.
2. Give the preset and the postset of each transition.
3. Which transitions are enabled at M_0?

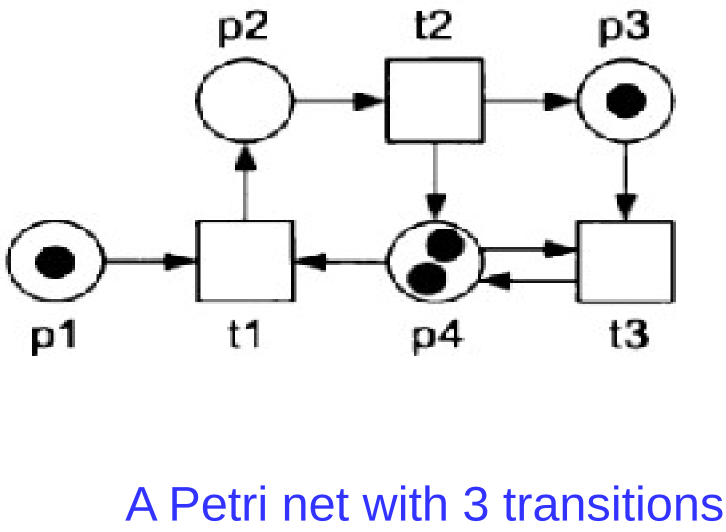


Figure 5.17: *A Petri net with small numbers of places and transitions*

4. Give all reachable markings. What are the reachable terminal markings? (p_4^2)

5. Is there a reachable marking in which we have a nondeterministic choice? yes

6. Does the number of reachable markings increase or decrease if we remove
   (1) place $p1$ and its adjacent arcs and
   (2) place $p3$ and its adjacent arcs?

PROBLEM 5.3 (From small marked Petri net to bigger transition system)**.**

Consider a marked Petri net $N = (P, T, F)$ with $|P| = 4 = n$, with the 4 start places each initially get 3 tokens, and the exit node is specially designated as place `OUT`. Besides, assume $|T| = 4$, and the initial marking is M_0, as seen in Figure 5.18.

Denote TS for the whole transition system made by the Petri net $N = (P, T, F)$.

1. Write down M_0, P, T of N.

$M_0(a^3, b^3, c^3, d^3)$

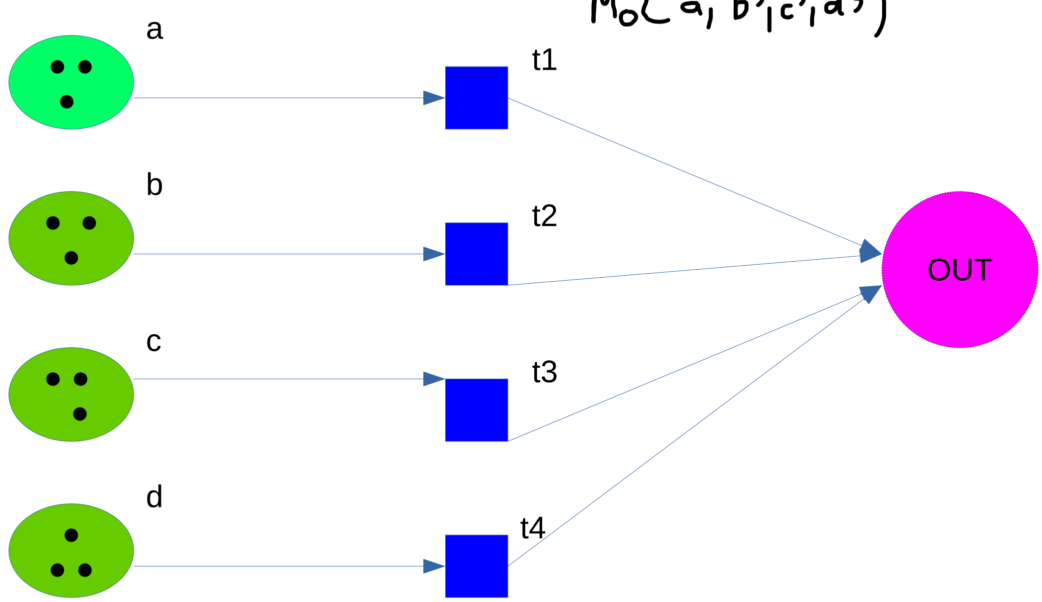


Figure 5.18: *A* **Petri net** *allows concurrency*

2. If **not allow concurrency** in this process (marked Petri net) then how many states of the transition system TS can be created? ? How many transitions are there?

   NOTE: The states of TS are in fact the *reachable markings* of the Petri net N.

   HINT:
   Extend the ideas of Hamming distance in the hypercube $H_n = (\{0,1\}^n, E)$ to "quarternary cube" $K_n = (\{0,1,2,3\}^n, E)$, and modify the concept of edge in K_n to capture the transitions in TS.

3. (* Optional) If we **allow concurrency** in this net, how many states and how many transitions of the transition system TS could be built? ■

5.6 PETRI NETS- ASSIGNMENT on MODELING

5.6.1 *Essential notion for Modeling with Petri Nets*

A **token** can model various things. A token can play the following roles:

- A *physical object*- for example, a product, a part, a drug, or a person;
- An *information object*- for example, a message, a signal, or a report;
- A *collection of objects*-as a truck with goods, a warehouse with parts, an address file;
- An *indicator of a state*- for example, the indicator of the state in which a business process is or the state of an object, such as a traffic light; and
- An *indicator of a condition*: the presence of a token indicates whether a certain condition is fulfilled.

Places may contain tokens. The role of a place in the network structure of a Petri net is, therefore, strongly connected with the tokens it can contain. A place can model:

- A *buffer*- for example, a depot, a queue, or a post bin;
- A *communication medium*: a telephone line, a middleman, or a communication net;
- A *geographic location*- a place in a warehouse, in an office, or in a hospital; and
- A possible *state or state condition* - the condition that a specialist is available.

Places are the passive elements of a Petri net. The tokens in a place represent a part of the state of the net, but a place **cannot** change the state. Transitions, in contrast, are the active elements of Petri nets. When a transition fires, the state of the net changes.

The role of a transition is, therefore, to represent:

- *An event*- for example, starting an operation, the death of a patient, a season change, or the turning of a traffic light from red to green;
- *A transformation of an object*, as repairing a product, updating a database, stamping a document; and
- *A transport of an object*- for example, transporting goods or sending a file.

The roles of tokens, places, and transitions give us the following guideline:
We represent events as transitions, and we represent states as places and tokens.

5.6.2 *Dynamic of* **Petri nets** *via Special properties*

Some generic properties are typically investigated in the context of a marked Petri net.

1. A marked Petri net (N, M_0) is k-**bounded** if no place ever holds more than k tokens. Formally, for any $p \in P$ and any $M \in [N, M_0\rangle : M(p) \leq k$.

   The marked Petri net in Fig. 5.19 is 12-bounded because in none of the reachable markings there is a place with more than 12 tokens. It is not 11-bounded, because in the final marking place out contains 12 tokens.

2. A marked Petri net is **safe** if and only if it is 1-bounded.
   The marked Petri net shown in Fig. 5.13 is safe because in each of the seven reachable markings there is no place holding multiple tokens.

3. A marked Petri net is **bounded** if and only if there exists a $k \in \mathbb{N}$ such that it is k-bounded. The marked Petri net in Figure 5.19 is bounded.

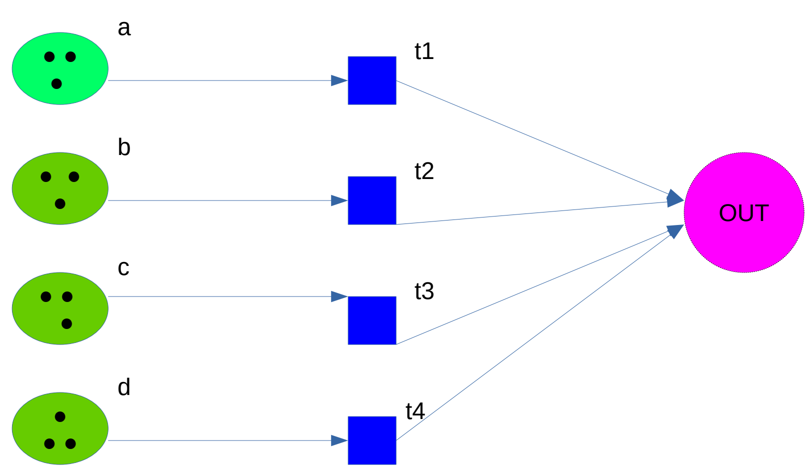


Figure 5.19: *Is this* **Petri net** *strongly dynamic?*

To represent the dynamic of **Petri nets** we could talk about the followings.

- A marked Petri net (N, M_0) is **deadlock free** if at every reachable marking at least one transition is enabled. Formally,
  for any $M \in [N, M_0\rangle$ there exists a transition $t \in T$ such that $(N, M)[\, t\rangle$. E.g.,

  (a) Figure 5.19 shows a net that is **not** deadlock free because at marking $[\mathtt{OUT}^{12}]$ no

  transition is enabled. (b) The marked Petri net below is deadlock free,

  $$\square \leftrightarrows \odot \; p,$$

  because there uniquely exists the transition $t \in T$ such that $(N, M)[\, t\rangle$. This has the only reachable state $[p] \equiv [1] = M$ (also the initial marking). Also $|[N, M_0\rangle| = 1$.

- A transition $t \in T$ in a marked Petri net (N, M_0) is **live** if from every reachable marking it is possible to enable t. Formally,
  for any $M \in [N, M_0\rangle$ there exists a marking $M_1 \in [N, M\rangle$ such that $(N, M_1)[\, t\rangle$.
- A marked Petri net is **live** if each of its transitions is live.
  Note that a deadlock-free Petri net does **not need to be** live.

5.6.3 *Modeling by Petri networks- Problem*

In a process there often are many agents, they all have different business activities, as a result, their Petri nets have distinct places/ tokens and transitions.

However, the dynamic of entire system (by all agents) is made up by mutual interactions, from which many places should usually have/share the same tokens, and sometimes do the transitions.

Therefore, the grand **Petri net** of the whole system is not simply the disjoint union of the constituents' nets, it should be the **superimposition** of the smaller nets. Examples include:

- Clinic or hospital systems: in which at least two agent types- the medical officers and patients interact with each other, share the same resources (places) and moving forward together (transitions).

- Educational systems: at least three agent types- the students, management & staff.
- Industrial factories: at least three agent types- the workers, management staffs and transporters or logistic personnel...

♣ **QUESTION 5.3.**

How could we build the grand **Petri net** of a large system without losing essential and useful information/knowledge of constituents' nets, as well as showing the true dynamic of the whole process/system?

Definition 5.9 (The superimposition (or merging) operator).

Formally, consider a system of only two agent types, denote N_1, N_2 to be their own **Petri nets**. Assume that $N_1 = (P_1, T_1, F_1, M_0)$ and $N_2 = (P_2, T_2, F_2, M_0)$, where the places P_i could be disjoint, but with the same initial marking M_0.

a) The **superimposition** operator, $\oplus : T_1 \times T_2 \longrightarrow T$, where $T = T_1 \cup T_2$, defined as follows.

   If $\bullet\, t_1 = \bullet\, t_2$ then $(t_1, t_2) \mapsto \oplus(t_1, t_2) = t \in T$ with $\bullet\, t := \bullet\, t_1$, (the presets of t_i are the same, keep one version only in the merged net),

   Else $\bullet\, t_1 \neq \bullet\, t_2$ $(t_1, t_2) \mapsto \oplus(t_1, t_2) = \{t_1, t_2\} \subseteq T$, keep both presets.

   Namely we can identify two transitions/events of two nets into one node of the merged **Petri net** $N = N_1 \oplus N_2$ if the events act on the same physical token.

b) The superimposed (merged) **Petri net** is determined by

   $N = N_1 \oplus N_2 = (P_1 \cup P_2,\ T,\ F_1 \cup F_2, M_0)$.

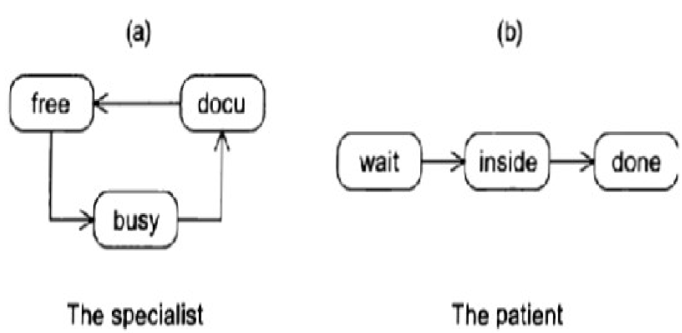
5.6.4 *Practical assignment: Consulting Medical Specialists*

This assignment will make up 20% for the final evaluation of CSE learners, and few questions in the final exam could be taken from this assignment.

SCENARIO: Under a SARS pandemic where a huge lack of ICU beds occurs in city H, patients should consult specialists in the outpatient clinic of a hospital, we describe the course of business around a specialist in this outpatient clinic of hospital X as a process model, formally, we use **Petri Net**.

ASSUMPTION and DATA

- **Specialist**: Each patient has an appointment with a certain specialist. The specialist receives patients. At each moment, the specialist is in one of the following three states:
  (1) the specialist is free and waits for the next patient (state *free*),
  (2) the specialist is busy treating a patient (state *busy*), or
  (3) the specialist is documenting the result of the treatment (state *docu*).
- **Every patient** who visits a specialist is in one of the following three states:
  (1) the patient is waiting (state *wait*, gets value n if there are n patients waiting),
  (2) the patient is treated by the specialist (state *inside*), or
  (3) the patient has been treated by the specialist (state *done*).



The possible states of a specialist and a patient.

Figure 5.20: *The transition systems of a specialist and a patient*

Figure 5.20 shows the states of the specialist and the states of a patient.

The specialist goes through the three states in an iterative way. A patient goes through these states only once (per visit).

- **Event data**: three events are important for this business process.
  First, the specialist starts with the treatment of a patient (event "start").
  Second, the specialist finishes the treatment of a patient and starts documenting the results of the treatment (event "change").
  Third, the specialist stops documenting (event "end").

  OBJECTIVE: We model this business process (step by step) as a Petri net.

1. Given the Petri net N_S modeling the state of the specialist, as in Fig. 5.21.

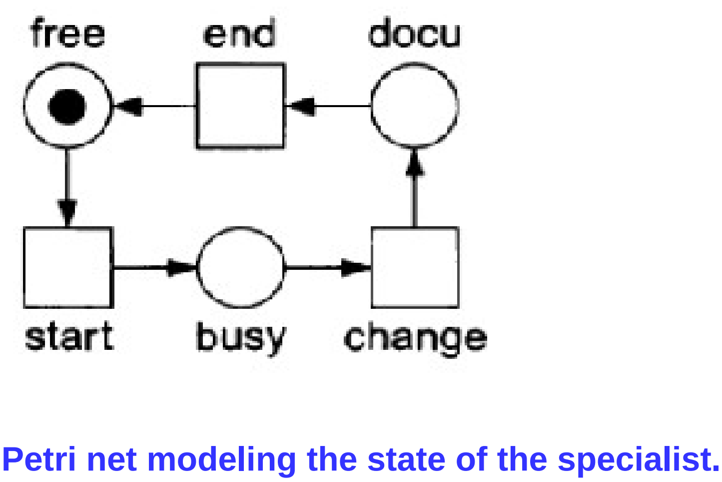


Figure 5.21: *The Petri net of the specialist's state*

In the displayed marking, the specialist is in state `free`.

1a) Write down states and transitions of the Petri net N_S. [**1 point**]

1b) Could you represent it as a transition system assuming that

(i) Each place **cannot** contain more than one token in any marking; and

(ii) Each place may contain any natural number of tokens in any marking. [**1 point**]

HINT: Describe a marking of the net as a triple (x, y, z) with x specifying the number of tokens in place `free`, y in place `busy`, and z in place `docu`.

2. Figure 5.21 of net N_S is made by information of **Specialist** and **Event data** above. Define N_{Pa} as the Petri net modeling the state of patients. By the similar ideas,
   a) explain the possible meaning of a token in state `inside` of the net N_{Pa}; [**1 point**]
   b) construct the Petri net N_{Pa}, assuming that there are five patients in state `wait`, no patient in state `inside`, and one patient is in state `done`. [**1 point**]
   HINT: The tokens in place `wait` represent the waiting patients. Transitions `start`, and `change` represent two possible events, for both nets. But in net N_S for the Specialist the event `start` leads to state or place `busy`, while in net N_{Pa} for patients the event `start` leads to state/place `inside`.

3. Determine the superimposed (merged) **Petri net** model $N = N_S \oplus N_{Pa}$ allowing a specialist treating patients, assuming there are four patients are waiting to see the specialist/ doctor, one patient is in state `done`, and the doctor is in state `free`. (The model then describes the whole course of business around the specialist). [**1 point**]

4. Consider an initial marking $M_0 = [3.wait, done, free]$ in the grand net $N = N_S \oplus N_{Pa}$. Which markings are reachable from M_0 by firing one transition once? Why? [**1 point**]

5. Is the superimposed Petri net N deadlock free? Explain properly. [**1 point**]

6. Propose a similar **Petri net** with two specialists already for treating patients, with explicitly explained construction. [**1 point**]
   HINT: possibly the concept of colored **Petri net** [39] would help.

7. Write a computational package to realize (implement) Items 1,2,3 and 4 above. You could employ any programming language (mathematical- statistical like R, Matlab, Maple, Singular, or multi-purposes language like C or Python), that you are familiar with, to write computational programs. The programs of your package should allow input with max 10 patients in place `wait` [**2 points**]

REMARK: To smoothly work out solutions learners should fully comprehend basic concepts given from Section 5.2 on PETRI NETWORKS- Background.

5.7 INSTRUCTIONS

Students must work closely with the other members in their own group. All of the aspects related to this assignment will be quizzed (about 10 - 12 of about 25 multiple-choice questions) in the final exam of the course. Therefore, team members must work together so that all of you understand all of the aspects of the assignment. The team leader should organize the group so that this requirement will be met. During the work, if you have any question about the assignment, please post it on the forum for the assignment on BK-eLearning or write an email to your instructor.

5.7.1 *Requests*

- Deadline for submission: **November 28, 2021**. Students have to answer each question in a clear and coherent way.
- Write a report by using LaTeX in accordance with **the layout as in the template file attached with this assignment**.
- Each group when submitting the report **needs to submit also a log file (working diary)** in which clearly state: **weekly working progress from now to the deadline**, working tasks of each member, content of opinions exchanged within the group, ...
- There is no restriction on programming languages that can be used to do this assignment.

5.7.2 *Submission*

- Students have to submit their own group report via BK-eLearning system (to be opened in the coming weeks): compress all necessary files (.pdf file, .tex file, coding files, ...) into a single file named "*Assignment-CO2011-CSE211-{All member ID numbers separated by hyphens}.zip*" and submit it in the assignment section on the site of the course on BK-eLearning.
- Noting that for each group, **only the leader will submit the report of the group.**

5.8 EVALUATION AND CHEATING JUDGEMENT

5.8.1 *Evaluation*

Each assignment will be evaluated as follows.

Content	**Score (%)**
- Analyze, answer coherently, systematically, focus on the goals of the questions and requests	30%
- The programs are neatly written and executable	30%
- Correct, clear, and intuitive graphs & diagrams	20%
- Background section is well written, correct and appropriate	15%
- Well written report and correct	5%

5.8.2 *Cheating*

The assignment has to be done by each group separately. Students in a group will be considered as cheating if:

- There is an abnormal similarity among the reports (especially in the background section). In this case, ALL submissions that are similar are considered as cheating. Therefore, the students of a group have to defend the works of their own group.
- They do not understand the works written by themselves. You can consult from any source, but make sure that you understand the meaning of everything you will have written.

If the article is found as cheating, students will be judged according to the university's regulations.

THE END

Man VM. Nguyen, Khuong An Nguyen and Thinh Tien Nguyen

This page is left blank intentionally.

Bibliography

[1] ALEXANDER HOLMES, Introductory Business Statistics, OpenStax, Rice University, 2017

[2] Anna Mikusheva, course materials for 14.384 Time Series Analysis, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology, nology. Downloaded on [01 June 2021]
Annette J. Dobson and Adrian G. Barnett,
An Introduction to Generalized Linear Models, Third Edition, CRC (2008)

[3] Annette J. Dobson and Adrian G. Barnett,
An Introduction to Generalized Linear Models, Third Edition, CRC (2008)

[4] Antal Kozak, Robert A. Kozak, Christina L. Staudhammer, Susan B. Watts *Introductory Probability and Statistics Applications for Forestry and Natural Sciences*, CAB International (2008)

[5] *Canvas paintings* by Australian artists of ethnic minorities, Australian National Museum

[6] **Practical Optimization: a Gentle Introduction**, John W. Chinneck, 2000

[7] **Introduction to Linear Optimization**, Dimitris Bertsimas and John N. Tsitsiklis, Athena Scientific, 1997

[8] U. N. Bhat. *A controlled transportation queueing process*. Management Science, 16(7): 446-452, 1970.

[9] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*, Second Ed. Springer, 2006.

[10] David S. Moore, George P. McCabe and Bruce A. Craig.

Introduction to the Practice of Statistics, 6th edition, (2009) W. Freeman Company, New York

[11] Man VM Nguyen. *Data Analytics- Foundation: Inference, Regression and Stochastic Processes*. Saarbrücken Germany: LAP LAMBERT Academic Publishing (2020) ISBN 978-620-2-79791-7

[12] Douglas C. Montgomery, George C. Runger, *Applied Statistics and Probability for Engineers*, Sixth Edition, (2014) John Wiley & Sons

[13] Jay L. Devore and Kenneth N. Berk, *Modern Mathematical Statistics with Applications*, 2nd Edition, Springer (2012)

[14] Glonek G.F.V. and Solomon P.J. *Factorial and time course designs for cDNA microarray experiments, Biostatistics* **5**, 89-111, 2004.

[15] Hedayat, A. S., Sloane, N. J. A. and Stufken, J. *Orthogonal Arrays, Springer-Verlag*, 1999.

[16] Robert V. Hogg, Joseph W. McKean, Allen T. Craig *Introduction to Mathematical Statistics,* Seventh Edition *Pearson*, 2013.

[17] David Ruppert and David S. Matteson. *Statistics and Data Analysis for Financial Engineering with R examples*, Second Edition. Springer (2015)

[18] Härdle, Wolfgang, and Léopold Simar. Applied multivariate statistical analysis. 2nd. Springer, 2007.

[19] Inada, T., Shimamura, Y., Todoroki, A., Kobayashi, H., and Nakamura, H., Damage Identification Method for Smart Composite Cantilever Beams with Piezoelectric Materials, Structural Health Monitoring 2000, Stanford University, Palo Alto, California, 1999,pp. 986-994.

[20] Jolliffe, I. T. Principal component analysis. 2nd. Springer, 2002.

[21] Lapin, L.L. , Probability and Statistics for Modern Engineering, PWS-Kent Publishing, 2nd Edition, Boston, Massachusetts,1990.

[22] Ljung, L. System identification: theory for the user, Prentice Hall, Englewood Cliffs, NJ, 1987

[23] Mahmut Parlar, *Interactive operations research with Maple: methods and models*, (2000) Springer

[24] Paul Mac Berthouex, Linfield C. Brown, *Statistics for Environmental Engineers*, 2nd Edition, LEWIS PUBLISHERS, CRC Press, 2002

[25] Madhav, S. P., *Quality Engineering using robust design, Prentice Hall*, 1989.

[26] Michael Baron, *Probability and Statistics for Computer Scientists*, 2nd Edition (2014), CRC Press, Taylor & Francis Group

[27] R. H. Myers, Douglas C. Montgomery and Christine M. Anderson-Cook *Response Surface Methodology : Process and Product Optimization Using Designed Experiments,* Wiley, 2009.

[28] Man Nguyen, Tran Vinh Tan and Phan Phuc Doan, *Statistical Clustering and Time Series Analysis for Bridge Monitoring Data*, Recent Progress in Data Engineering and Internet Technology, Lecture Notes in Electrical Engineering 156, (2013) pp. 61 - 72, Springer-Verlag

[29] Nguyen Van Minh Man, *Computer-Algebraic Methods for the Construction of Designs of Experiments*, Ph.D. thesis, 2005

[30] Nguyen, Man V. M. *Some New Constructions of strength 3 Orthogonal Arrays*, the Memphis 2005 Design Conference Special Issue of the **Journal of Statistical Planning and Inference**, Vol 138, Issue 1 (Jan 2008) pp. 220-233.

[31] Nathabandu T. Kottegoda, Renzo Rosso. *Applied Statistics for Civil and Environmental Engineers*, 2nd edition (2008), Blackwell Publishing Ltd and The McGraw-Hill Inc

[32] Man Nguyen (2005) *Computer-algebraic Methods for the Construction of Design of Experiments*, Ph.D thesis, Eindhoven Univ. Press, 2005

[33] Paul Mac Berthouex. L. C. Brown. *Statistics for Environmental Engineers*; 2nd edition (2002), CRC Press

[34] A. Ravi Ravindran (editor). *Operations Research and Management Science Handbook*, CRC, 2008

[35] Alvin C. Rencher and William F. Christensen, *Methods of Multivariate Analysis*, Wiley, 2012

[36] Ron S. Kenett, Shelemyahu Zacks.

Modern Industrial Statistics with Applications in R, MINITAB, 2nd edition, (2014), Wiley

[37] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*, **Springer Texts in Statistics**, 3rd Edition, (2011)

[38] W.M.P. Aalst, van der PROCESS MINING, 2nd edition, 2016, Springer

[39] Kurt Jensen, Wil M.P. van der Aalst, Gianfranco Balbo, Maciej Koutny, Karsten Wolf (Eds.). *Transactions on Petri Nets and Other Models of Concurrency VII* LNCS 7480, Springer, 2013 .

[40] Sheldon M. Ross. *Introduction to probability models*, 10th edition, (2010), Elsevier Inc.

[41] Sloane N.J.A., `http://neilsloane.com/hadamard/index.html/`

[42] Google Earth, Digital Globe, 2014- 2019

[43] Larry Wasserman, *All of Statistics- A Concise Course in Statistical Inference*, **Springer**, (2003)

[44] C.F. Jeff Wu, Michael Hamada *Experiments: Planning, Analysis and Parameter Design Optimization,* Wiley, 2000.

[45] Wendy L. Martinez and Angel R. Martinez, *Computational Statistics Handbook with MATLAB*, CHAPMAN & HALL/CRC, 2002