VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

MATHEMATICAL MODELING (CO2011)

Assignment

Process Mining
PETRI NETWORKS

Advisor:	Nguyễn Tiến Thịnh.
Team name:	Team Aesthetic.
Students:	Phạm Huy Thanh - 1952977 - CC02 (Leader).
	Vũ Tiến Giang - 1952240 - CC02.
	Nguyễn Trọng Nghĩa - 1951175 - CC01.
	Phan Quang Thiện - 1952997 - CC02.
	Phạm Bá Trọng - 1953045 - CC06.
	Đoàn Việt Tú - 1952521 - CC02.

HO CHI MINH CITY, NOVEMBER 2021

Contents

1 Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Phạm Huy Thanh	1952977	- Prepare theory - Write Latex: part 3, 4, 5	16%
2	Vũ Tiến Giang	1952240	- Prepare theory - Write Latex: part 2, 5, 6	17%
3	Nguyễn Trọng Nghĩa	1951175	- Assignment part 6 (Supported by Giang and Thanh)	17%
4	Phan Quang Thiện	1952997	- Exercises and examples	16%
5	Phạm Bá Trọng	1953045	- Summary part 5	17%
6	Đoàn Việt Tú	1952521	- Exercise and problems codes - Review the report (Supported by Thiện and Trọng)	17%

2 PETRI NETWORKS- Background

2.1 The Art of Modeling- motivated from Operation Research

2.1.1 Definition of Operation Research

Operations Research (OpRe) is a branch of management science heavily relying on modeling, also called Decision Science or Operations Analysis, is the study of applying mathematics to business questions. As a sub-field of Applied Mathematics, it has a very interesting position alongside other fields as Data Science and Machine Learning.

2.1.2 Some examples of OpRe:

- Linear Programming
- Waiting line theory or queuing theory
- Inventory control systems
- Replacement problems
- Network Analysis
- Sequencing Problems

2.2 Petri net

A **Petri net**, also known as a place/transition (PT) net, is one of several mathematical modeling languages for the description of distributed systems. It is a class of discrete event dynamic system.
We can define it is a directed bipartite graph that has two types of elements, places and transitions, depicted as white circles and rectangles, respectively. A place can contain any number of tokens, depicted as black circles. A transition is enabled if all places connected to it as inputs contain at least one token.
Some sources state that **Petri nets** were invented in August 1939 by Carl Adam Petri — at the age of 13 — for the purpose of describing chemical processes.

2.3 Formal definitions

Petri nets are state-transition systems that extend a class of nets called elementary nets.

2.3.1 Definition 1 (Transition system or State transition system)

Formally, a **transition system** is a pair $(S, \rightarrow)$, where

- S is a set of states.
- $\rightarrow$ is a relation of state transitions (a subset of $S \times S$).

A **labelled transition system** is a tuple $(S, \Lambda, \rightarrow)$, where

- S is a set of states.
- Λ is a set of labels.
- $\rightarrow$ is a relation of labelled transitions (S, Λ, S).

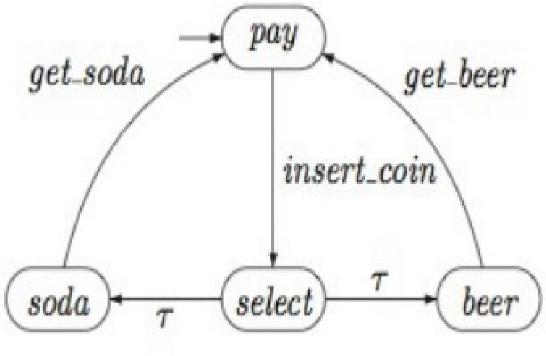
It also can be present as $TS = (S, A, T)$, where

- S is a set of states.
- $A \subseteq \Lambda$ is set of activities (also called actions).
- $T \subseteq (S \times A \times S)$.

The following subsets are defined implicitly,

- $S^{start} \subseteq S$ is the set of initial states (sometimes referred to as 'start' states).
  $S^{end} \subseteq S$ is the set of final states (sometimes referred to as 'accept' states).
  For most practical applications the state space S is finite.
- Activities can be executed sequentially, activities can be optional or concurrent, and the repeated execution of the same activity may be possible. Using transition systems in a process model will help arrange the *order* of activities and determine *which activities* need to be executed.
- The net structure and dynamic of a transition system may be used to study and express its *behavior*. One of the first states is where the transition begins. A conceivable *execution sequence* corresponds to any path in the graph that starts in this state.

A transition system of a vending machine:



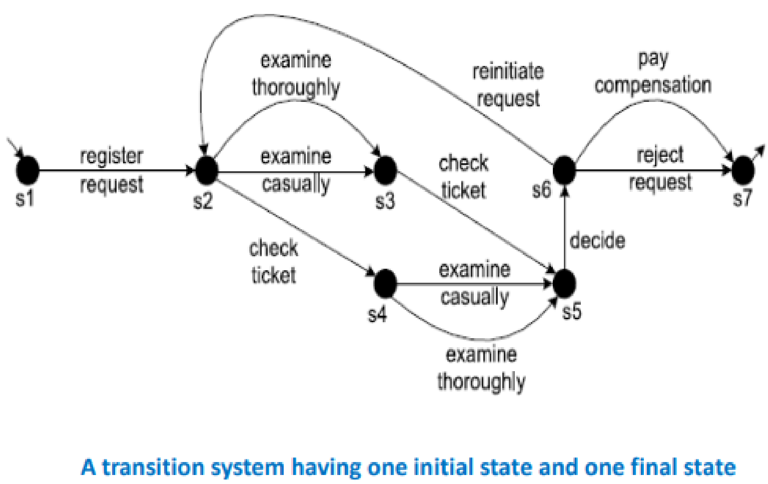


Figure 5.2: *A small size transition system*

Example 5.1. Observe a transition system in Figure 5.2
$A = \{registerrequest, examinethoroughly, examinecasually, checkticket,$
$reinitiaterequest, decide, rejectrequest, paycompensation\}$

$T = \{(s1, registerrequest, s2), (s2, examinecasually, s3), (s2, checkticket, s4),$
$(s2, examinethoroughly, s3), (s3, checkticket, s5), (s4, examinecasually, s5),$
$(s4, examinethoroughly, s5), (s5, decide, s6), (s6, reinitiaterequest, s2),$
$(s6, rejectrequest, s7), (s6, paycompensation, s7)\}$

Example 5.2. A multi-set example
$M = [a, b, b, c, c, c, d, d, e] = \{a, b^2, c^3, d^2, e\} = \{e, c^3, b^2, e, d^2\} = [1, 3, 2, 1, 2]$

2.3.2 Definition 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 also a finite set of transitions, which P and T are disjoint.
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of (directed) arcs, called the flow relation.

Remarks:

- If the diagram were of an elementary net, then those places in a configuration would be conventionally depicted as circles, where each circle encompasses a single dot called a *token*. It is a special *transition node*, being graphically rendered as a black dot. **Note:** *Places* can contain tokens but the *transitions* **cannot**.
- A transition is *enabled* if each of its input places contains a token.
- The configuration of tokens distributed over an entire Petri net diagram is called a *marking*. A marking of net N is a function $m : P \to N$: assigning to each place $p \in P$ the number $m(p)$ of tokens at this place.
  Denote $M = m(P)$, the range of map m, viewed as a multiset.
- 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* (*bag*) over P denoting the *marking* of the net.

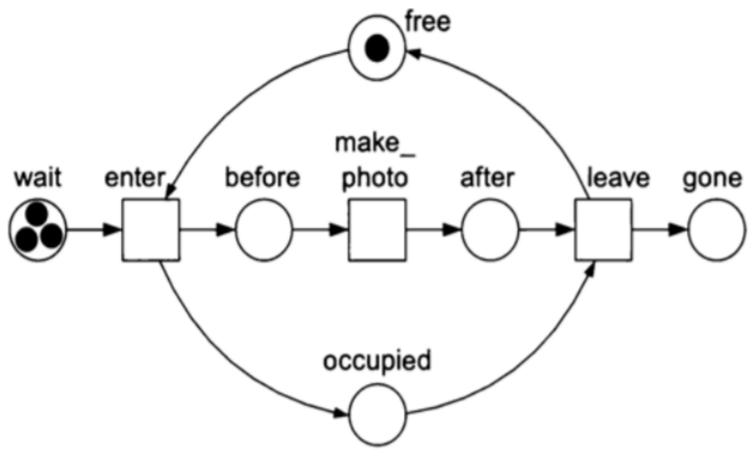


Figure 5.3: *A Petri net for the process of an X-ray machine*

Example 5.3.
The Petri net in figure 5.3 has three transitions: $T = \{enter, make - photo, leave\}$.
Solution:
$P = \{wait, free, before, occupied, after, gone\}$

$M = \{wait^3, free^1, before^0, occupied^0, after^0, gone^0\} = [3, 1, 0, 0, 0, 0]$
$F = \{(wait, enter), (enter, before), (before, make-photo), (make-photo, after), (after, leave), (leave, gone), (enter, occupied), (occupied, leave), (leave, free), (free, enter)\}$

ELUCIDATION

- PLACES: a place $p \in P$ is represented by a circle or eclipse. p can store, accumulate or show things. A place has discrete states.
- TRANSITIONS: a transition $t \in T$ is represented by a square or rectangle. t can produce things/tokens, consume, transport or change them.
- 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.
- OPERATIONS: Operations in Petri net are the mathematical functions of multi-set such as: the sum of two multi-sets or the difference between them.

2.3.3 Definition 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. Node x is an output node of y if and only if $(y, x) \in F$.
- For $x \in P \cup T$, the preset of x: $\bullet x = \{y|(y, x) \in F\}$, the postset of x: $x\bullet = \{y|(x, y) \in F\}$.

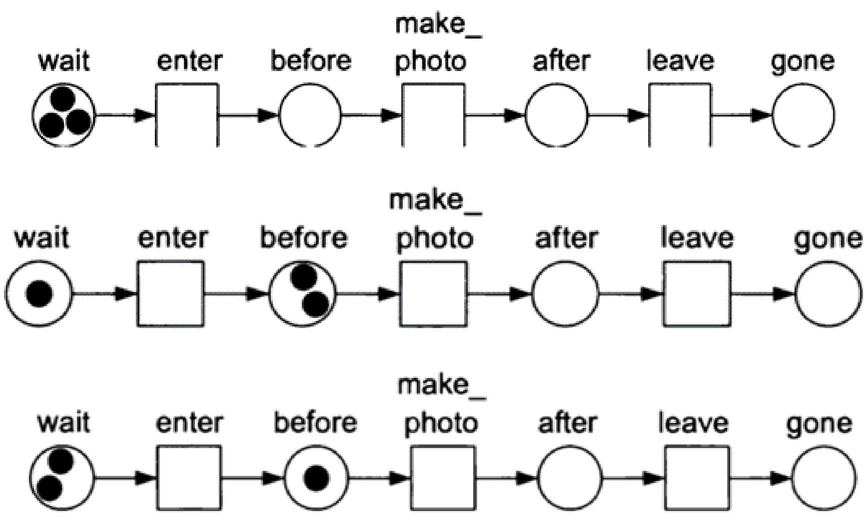


Figure 5.4: *Three different markings on a Petri net, modeling of a process of an X-ray machine*

Example 5.4.

Figure 5.4 shows a Petri net with three different markings.
Find the places P, and give the transitions T of net.
Write down completely three different markings in format of lists or tables.
Solution:
$T = \{enter, make - photo, leave\}$

P	M_0	M_1	M_2
Wait	3	2	1
Before	0	1	2
After	0	0	0
Gone	0	0	0

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

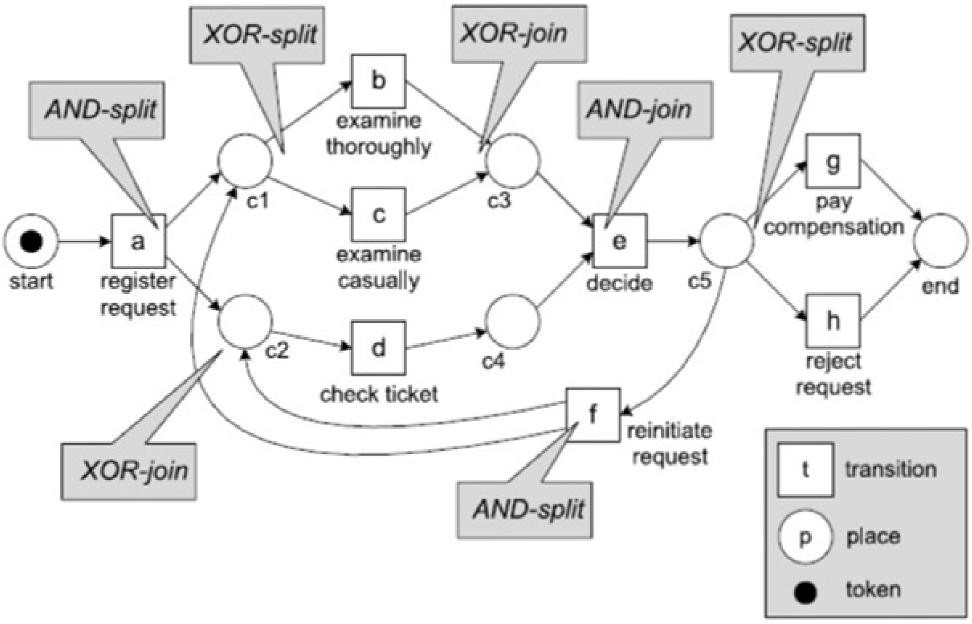


Figure 5.5: *A marked Petri net with one initial token*

Preset:

→	C1	C2	C3	C4	C5
a	1	1	0	0	0
b	0	0	1	0	0
c	0	0	1	0	0
d	0	0	0	1	0
e	0	0	0	0	1
f	1	1	0	0	0
g	0	0	0	0	0
b	0	0	0	0	0

Postset:

←	C1	C2	C3	C4	C5
a	0	0	0	0	0
b	1	0	0	0	0
c	1	0	0	0	0
d	0	1	0	0	0
e	0	0	1	0	0
f	0	0	0	1	1
g	0	0	0	0	1
b	0	0	0	0	1

Preset: $\bullet c1 = \{a, f\}$
Postset: $c5\bullet = \{f, g, h\}$

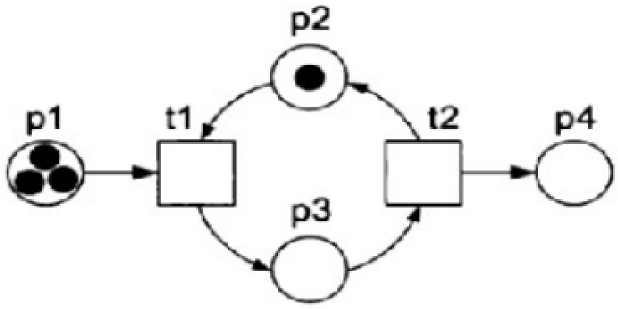
2.4 On Enabled transition and Marking changes

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.
- 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: [start5].
  Firing a now results in the marking [start4, c1, c2]. At this marking a is still enabled.
  Firing a again results in marking [start3, c12, c22].Transition a can fire five times in a
  row resulting in marking [c15, c25].
- Note that after the first occurrence of a, also b, c, and d are enabled and can fire concurrently.

PRACTICE 5.1.



A Petri net showing transitions t1 and t2.

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

Solution:

1. This Petri net has two transitions: $T = \{t1, t2\}$.
$P = \{p1, p2, p3, p4\}$
$F = \{(p1, t1), (t1, p3), (p3, t2), (t2, p2), (p2, t1), (t2, p4)\}$

2.
Preset:

→	p1	p2	p3	p4
t1	0	0	1	0
t2	0	1	0	1

Postset:

←	p1	p2	p3	p4
t1	1	1	0	0
t2	0	0	1	0

3. The marking of this net: $M = \{p1^3, p2^1, p3^0, p4^0\} = [3, 1, 0, 0]$

4. At this marking, t1 is enabled but t2 is not yet enabled.

2.5 Important usages of Petri net via explaining Figure 5.5

2.5.1 Business process modeling (BPM)

- 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.

2.5.2 Information systems (IS)

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.

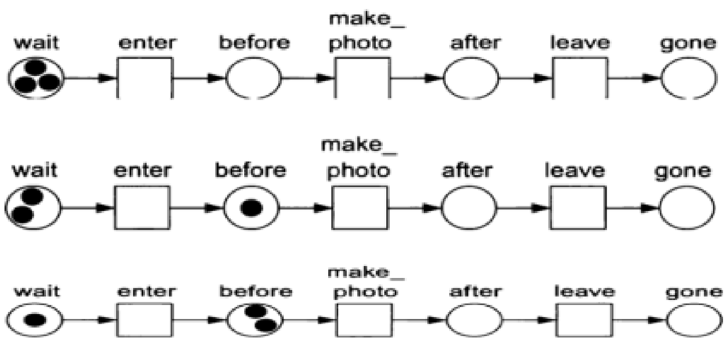
2.5.3 Definition 4 (To study Petri net we first formalize the above concepts)

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.

2.6 Summary

PRACTICAL PROBLEM 1.



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*

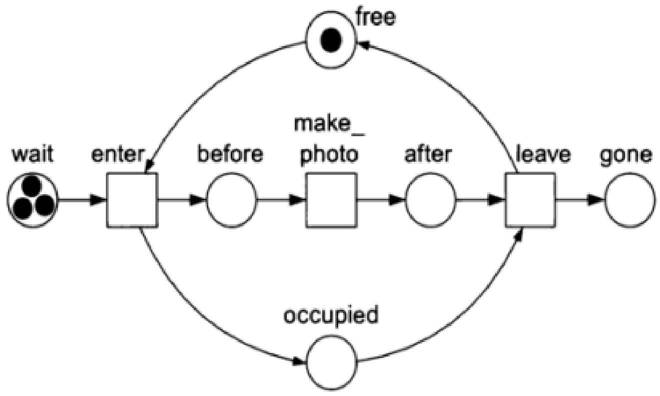


Figure 5.9: *An improved Petri net for the business process of an X-ray machine*

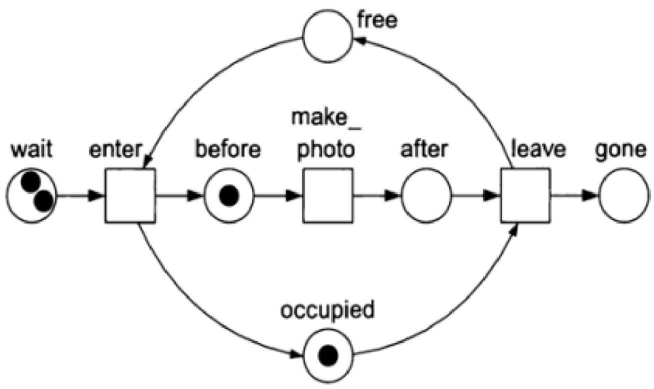


Figure 5.10: *The marking of the improved Petri net for the working process of an X-ray room after transition* ***enter*** *has fired.*

1. In Figure 5.8:
$R_I = \{(wait, enter), (before, make - photo), (after, leave)\}$
$R_O = \{(enter, before), (make - photo, after), (leave, gone)\}$
$F = \{(wait, enter), (enter, before), (before, make - photo), (make - photo, after), (after, leave), (leave, gone)\}$

2. No. Because in Figure 5.9, transition **enter** is only enabled when place **wait** have at least one token and place **free** also has one token. If the condition is satisfied, it then consumes and produces a new token to place **before** and another to place **occupied** (AND-split). This process makes sure only one patient can enter at once. $P_{\text{new}} = \{wait, free, occupied, before, after, gone\}$

3. No. Transition **enter** can not fire when there is no token in place **free**, as explained above.

3 PETRI NETWORKS - Behaviors

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 $t \in T$.

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 $0 \leq i < n$, 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.
$\sigma_1 = \langle a, b\rangle \rightarrow (N, M_0)\ [\sigma_1\rangle\ (N, [c_2, c_3])$
$\sigma_2 = \langle a, b, d, e\rangle \rightarrow (N, M_0)\ [\sigma_2\rangle\ (N, [c_5])$
$\sigma = \langle a, c, d, e, f, b, d, e, g\rangle \rightarrow (N, M_0)\ [\sigma\rangle\ (N, [end])$
The set $[N, M_0\rangle$ has seven reachable markings.

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

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

Answer: 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**.

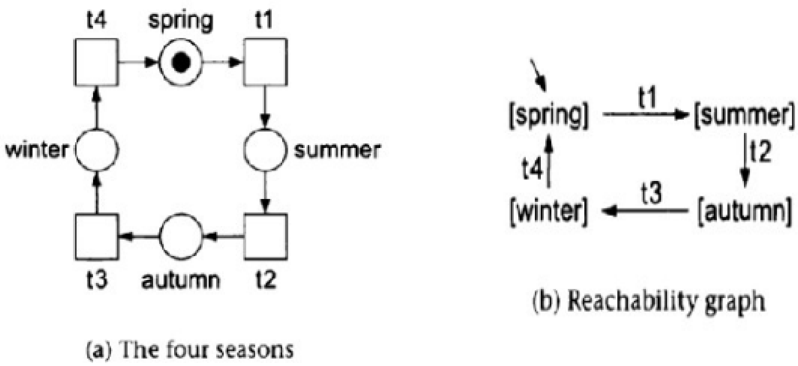


Figure 5.12: *A Petri net system and its reachability graph*

Example 5.7.
Consider the Petri net system in Figure 5.12 modeling the four seasons:

- 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**.
- The path from marking $[spring]$ to marking $[winter]$ is a finite run $(t1, t2, t3)$ of the net in figure 5.12(a). However, it **can have infinite run**, as the path can be reduplicated (for example: $(t1, t2, t3, t4, t1, t2, t3), (t1, t2, t3, t4, t1, t2, t3, t4, t1, t2, t3), ...$).

3.2 Representing Petri Nets as Special Transition Systems

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

(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

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

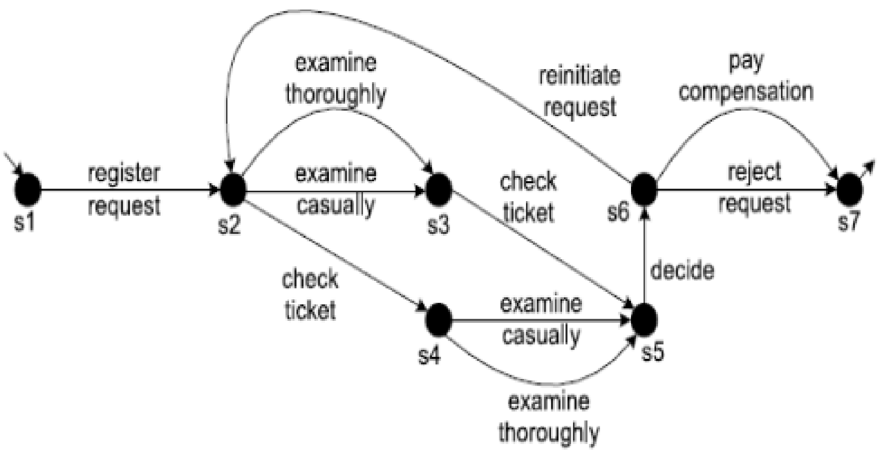
$$TR = \{(M, l(t), M_1) \in S \times A \times S \mid \exists t \in T \quad (N, M)[\, t\rangle \quad (N, M_1)\}. \qquad (5.4)$$

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

After reading ELUCIDATION carefully, we summarize that:

1. In this part, the document brings us into the problem of dealing with reachable markings: Sometimes, in the bad situation, we use state transition systems to represent some complex Petri nets in order to categorize places, transitions and avoid forgetting markings.
2. Moreover, this part provides us some clear materials on how to use the transition systems effectively by the knowledge of multi-set, labeling, sub-graph,...
3. We also understand more clearly the given examples, thanks to this part as well as some reference books.

PRACTICE 5.2.



4 PETRI NETWORKS - Structures and Basic Problems

In this part:

We know the connecting between token, places and transition. And the document provides materials for us to research the status of token, we try to understand that there are two kind of relationship - Dependence and independence.

Next, we understand where the token goes and events happen in each cases.
In each kinds, there will be a sub-kind, dependence include causality and synchronization.

Moreover, the system that we design must include case in which various events can access or happen at the same time. Next, students need to design the system that allow more threads in order to control the usage of the users and enhance the effective of the system for multiple token.

For the picture below:

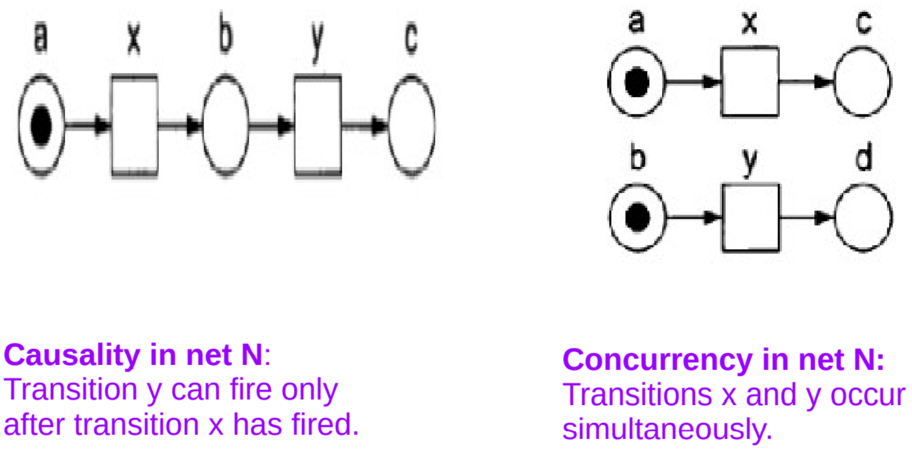


Figure 5.15: *Causality and Concurrency in a* **Petri net**

It helps us to imagine how the Petri net works in 2 different cases: Causality and Concurrency.

Moreover, the fact that 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.

5 SUMMARY and REVIEWED PROBLEMS

5.1 Problem 5.1

A Petri net is a particular kind of bipartite directed graphs populated by three types of objects. These objects are places, transitions, and directed arcs. Directed arcs connect places to transitions or transitions to places. In its simplest form, a Petri net can be represented by a transition together with an input place and an output place. This elementary net may be used to represent various aspects of the modeled systems. For example, a transition and its input place and output place can be used to represent a data processing event, its input data and output data, respectively, in a data processing system. In order to study the dynamic behavior of a Petri net modeled system in terms of its states and state changes, each place may potentially hold either none or a positive number of tokens. Tokens are a primitive concept for Petri nets in addition to places and transitions. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false, for instance.

1. "enabled transition": A transition in a Petri net is enabled if each of its input places holds at least one token.
e.g: In Figure 5.9, if each of the places $wait$ and $free$ holds at least one token, then transition *enter* is enabled.

2. "firing of a transition": An enabled transition can fire, thereby consuming (energy of) one token from each input place and producing at least one token for each output place next.
e.g: In Figure 5.10, the enabled transition *enter* can consume tokens from place $wait$ and $free$ to produce one token to place $before$ and another to place *occupied*.

3. "reachable marking": 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.
e.g: In Figure 5.5, the marked Petri net has seven reachable markings.

4. "terminal marking": The transitions can keep firing until the net reaches a marking that does not enable any transition. Like the terminal state in transition systems, this marking is a terminal marking.
e.g: In Figure 5.5, the terminal marking is [end].

5. "non-deterministic choice": When several transitions are enabled at the same moment, it is not determined which of them will fire. This situation is a non-deterministic choice. Even though we do not know in this case which transition will fire, we know that one of them will be fired.
e.g: In Figure 5.5, this situation happens at state $c1$, when there are 2 transitions b and c to fire concurrently or in a certain order.

5.2 Problem 5.2

5.2.1

$P = \{p1, p2, p3, p4\}$
$T = \{t1, t2, t3\}$
$F \subseteq (P \times T) \cup (T \times P)$
$M_0 = [1, 0, 1, 2]$

5.2.2

Preset:

→	p1	p2	p3	p4
t1	0	1	0	0
t2	0	0	1	1
t3	0	0	0	1

Postset:

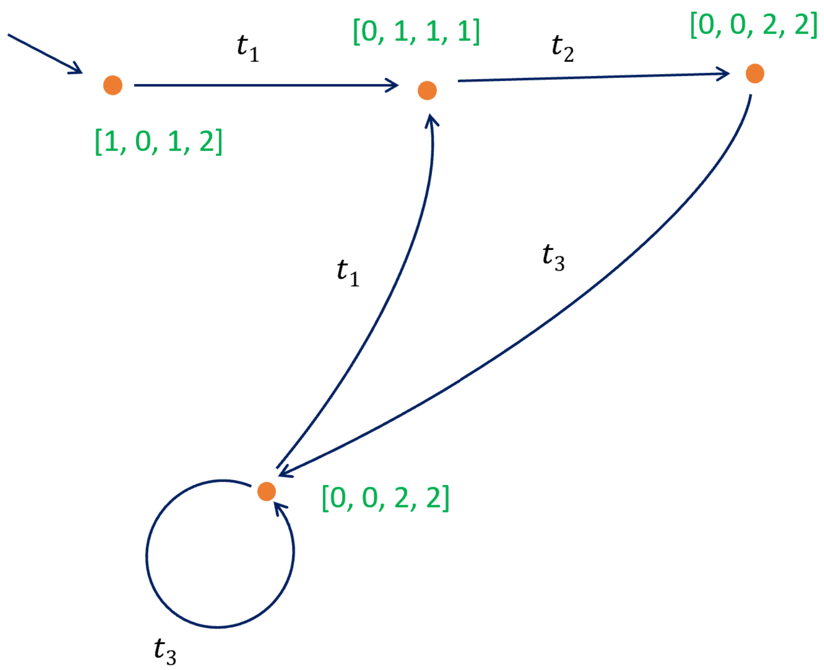
←	p1	p2	p3	p4
t1	1	0	0	1
t2	0	1	0	0
t3	0	0	1	1

5.2.3

At M_0, transition $t1$ and $t3$ are enabled.

5.2.4

There is no reachable terminal marking.



5.2.5

Non-deterministic choice: $[p4]$

5.2.6

P1	P2	P3	P4
1	0	1	2
	$\downarrow$	t1	
0	1	1	1
	$\downarrow$	t2	
0	0	2	2
	$\downarrow$	t3	
0	0	0	2

5.3 Problem 5.3

5.3.1

$P = \{a, b, c, d, out\}$ $T = \{t1, t2, t3, t4\}$ $M_0 = \{3, 3, 3, 3, 0\}$

5.3.2

Interleaving semantics.
Based on the theory of k-ary n-cube and some widely proven concepts, we have the equation:
The number of vertices: k^n vertices
The number of edges:

- nk^{n-1} (with k = 2) edges;
- nk^n (with $k \geq 3$) edges;

Therefore, in this problem, as k = 4 and n = 4 (quaternary cube, seen from the graph), we can conclude:

- The number of states: 256 (4^4)
- The number of transitions: 1024 (4×4^4)

5.3.3

If we allow concurrency $\Rightarrow$ TS = 15.
$P = \{a, b, c, d, out\}$

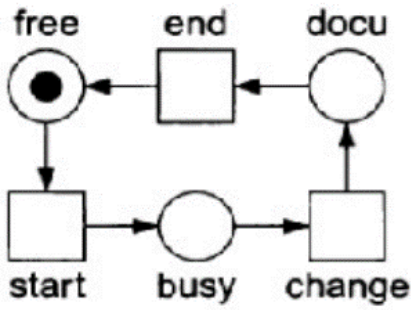
$$\{3, 3, 3, 3, 0\} \xrightarrow[\text{TS = 5}]{\text{t1, t2, t3, t4}} \{2, 2, 2, 2, 4\} \xrightarrow[\text{TS = 5}]{\text{t1, t2, t3, t4}} \{1, 1, 1, 1, 8\} \xrightarrow[\text{TS = 5}]{\text{t1, t2, t3, t4}} \{0, 0, 0, 0, 12\}$$

6 PETRI NETS- ASSIGNMENT on MODELING

6.1 Problem 1

6.1.1 Write down states and transitions of the Petri net

Petri net: $N_S = (P, T, F)$

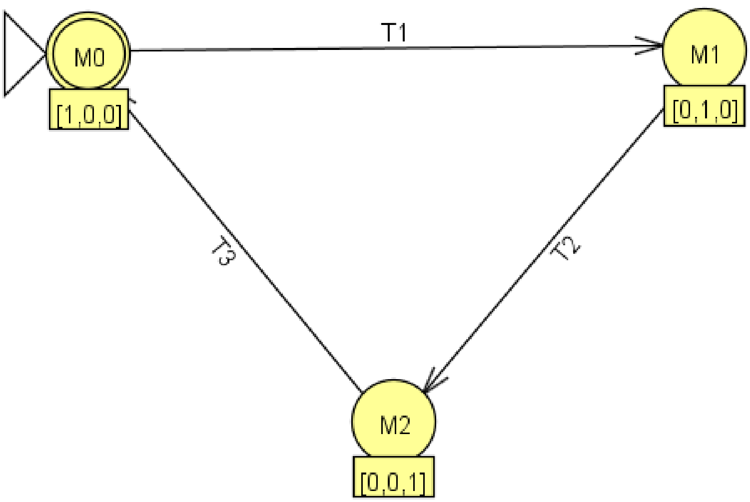


N_S has three states $P = \{free, busy, docu\}$ and transitions $T = \{start, change, end\}$.

6.1.2 Represent it as a transition system assuming that

6.1.2.a Each place *cannot* contain more than one token in any marking

We represent the above Petri net N_S as a transition system that a triplet $TS = (S, A, T)$ where $S = \{M_0, M_1, M_2\}$, $A = \{start, change, end\}$, $T = \{T_1, T_2, T_3\}$.
We denote M_0, M_1, M_2 are the marking of Petri net N_S. Because each state has only one token so there are possibly three markings where $M_0 = [1, 0, 0]$, $M_1 = [0, 1, 0]$, $M_2 = [0, 0, 1]$. Because state *free* are both initial and final states since then M_0 is also both initial and final states. Then T is set of transition where $T_1\{M_0, free, M_1\}$, $T_2\{M_1, change, M_2\}$, $T_3\{M_2, end, M_0\}$.



6.1.2.b Each place may contain any natural number of tokens in any marking

We can formalize the states as $S = \{(x, y, z)|x, y, z \in N\}$. The initial state s_0 is equal to the initial marking $m_0 = (1, 0, 0)$. The transition relation can be specified by the union of the following three sets:

$$TR = \{((x+1,y,z),(x,y+1,z))|x,y,z \in N\}$$
$$\cup\{((x,y+1,z),(x,y,z+1))|x,y,z \in N\}$$
$$\cup\{((x,y,z+1),(x+1,y,z))|x,y,z \in N\}$$

the first, the second, and the third set contains all possible states that can be reached by firing transition start, change, and end, respectively.

6.2 Problem 2

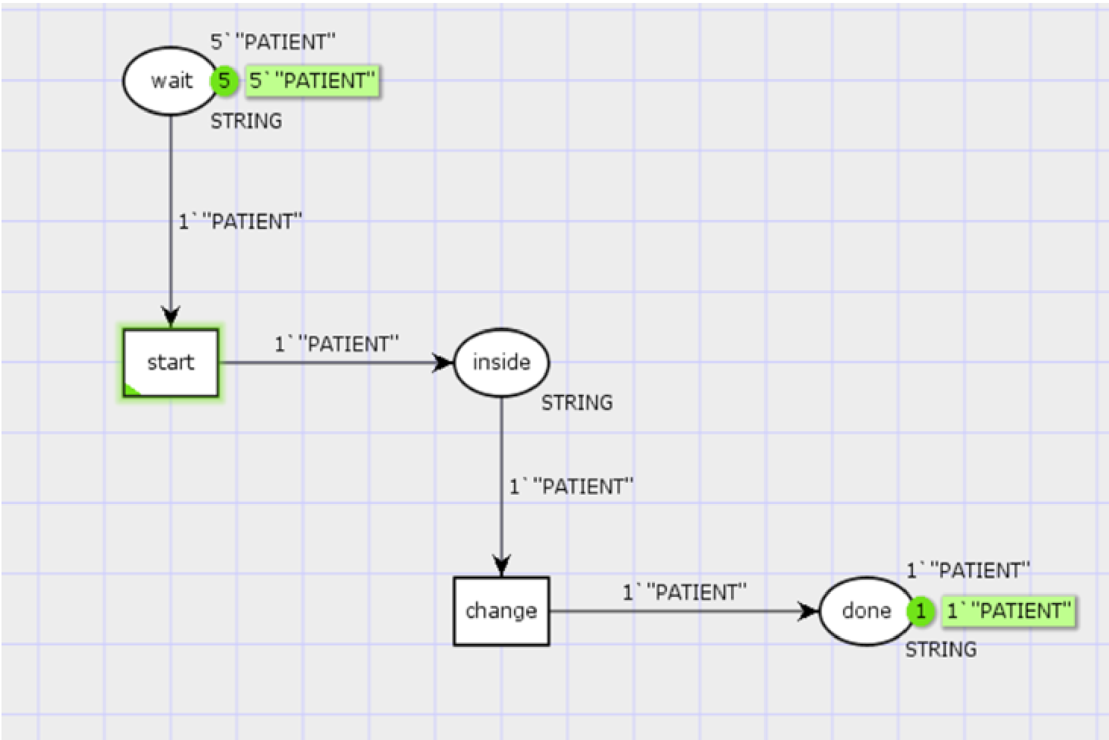
We define Petri net $N_{Pa} = (P, T, F)$ where $P = \{wait, inside, done\}$, $T = \{start, change\}$.

6.2.1 Explain the possible meaning of a token in state inside of the net

We define a token in state *inside* as a patient being treated by the specialist. In other words, we evaluate the state *inside* similar to the state *busy* of the N_s Petri net model.

6.2.2 Construct the Petri net, assuming that there are five patients in state wait, no patient in state inside, and one patient is in state done

The tokens in place *wait* represent the waiting patients. Transition *start* and *change* represent two possible events. the transition *start* firing that transfer tokens from state *wait* into state *inside* and then transition *change* firing that transfer tokens from states *inside* into states *done*.



6.3 Problem 3

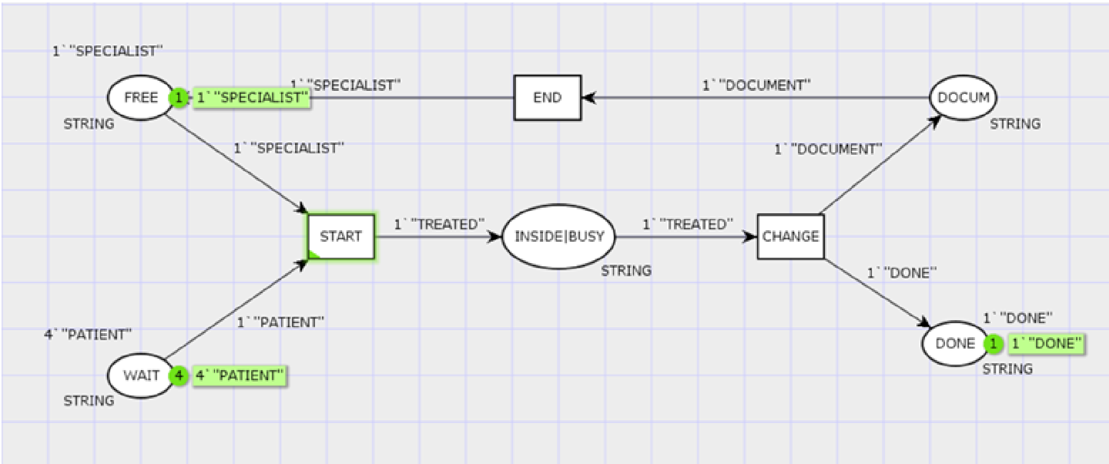
We construct $N = N_S \oplus N_{Pa} = (P, T, F)$, and $P_N = P_S \cup P_{Pa}$, $T_N = T_S \cup T_{Pa}$, $F_N = F_S \cup F_{Pa}$. Since we define *busy* and *inside* as the concurrency states so

$$P_N = \{free, wait, (busy|inside), docu, done\}$$

and

$$T_N = \{start, change, end\}$$

Then we can have the model below:



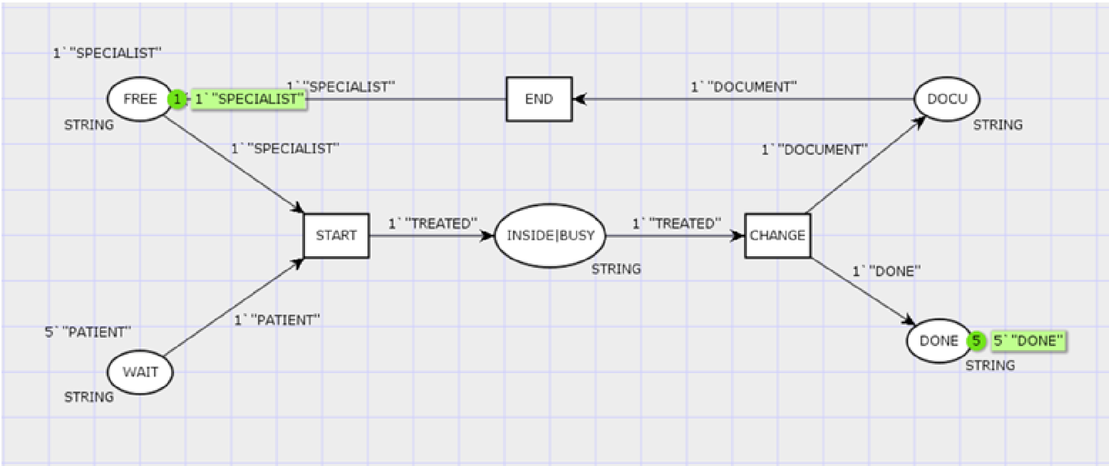
6.4 Problem 4

The reachable markings from M_0 by firing one transition once are:

$$M_0 \to M_1[2.wait, (busy|inside), done] \to M_2[2.wait, docu, 2.done] \to M_3[2.wait, free, 2.done] \to M_4[wait, (busy|inside), 2.done] \to M_5[wait, docu, 3.done] \to M_6[wait, free, 3.done] \to M_7[(busy|inside), 3.done] \to M_8[docu, 4.done] \to M_9[free, 4.done]$$

6.5 Problem 5

The Petri net N is not deadlock free because there exists a terminal marking which no longer enables any transition. For example, if we construct an initial marking M_0 $[1.free, 5.done]$ so this initial making is deadlock and then Petri net N is not deadlock free.



6.6 Problem 6

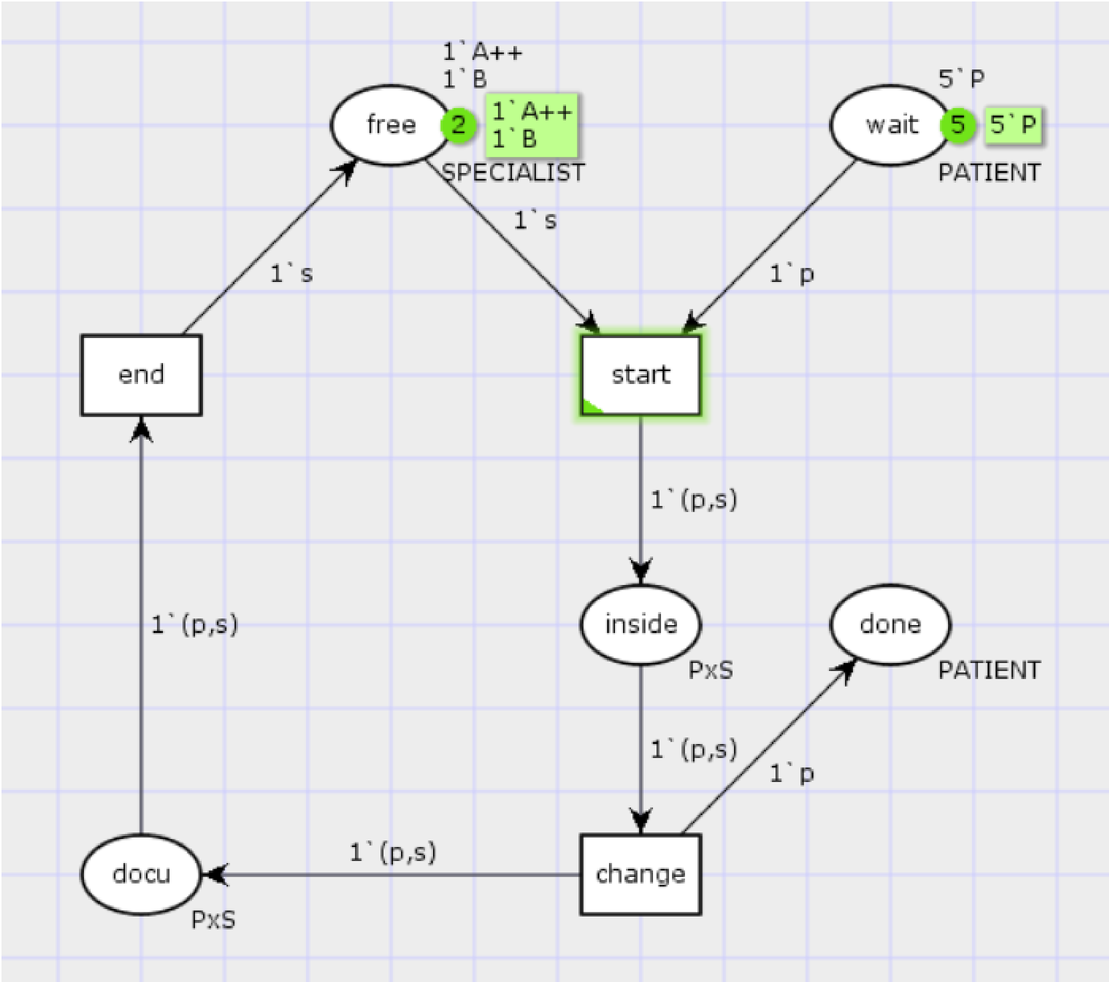
We have 2 ideas about Petri net model for 2 specialists.
First Petri net we define 2 kinds of tokens in one state free that representing 2 specialists.

$$N_1 = (P, T, F)$$

where

$$P = \{free, wait, inside, docu, done\},\ T = \{start, change, end\}$$

The tokens in state *free* represent the specialist is free and waits for the next patient. The tokens in state *wait* represent the patient is waiting, one in state *inside* represent the specialist is busy treating the doctor is treating as well as the patient being treated by the doctor. And then state *docu* represent the specialist is documenting the result of the treatment and the end *done* states represent the patient has been treated by the specialist.
Then construction N_1 below the initial marking $M_0 = [2.free, 5.wait]$:



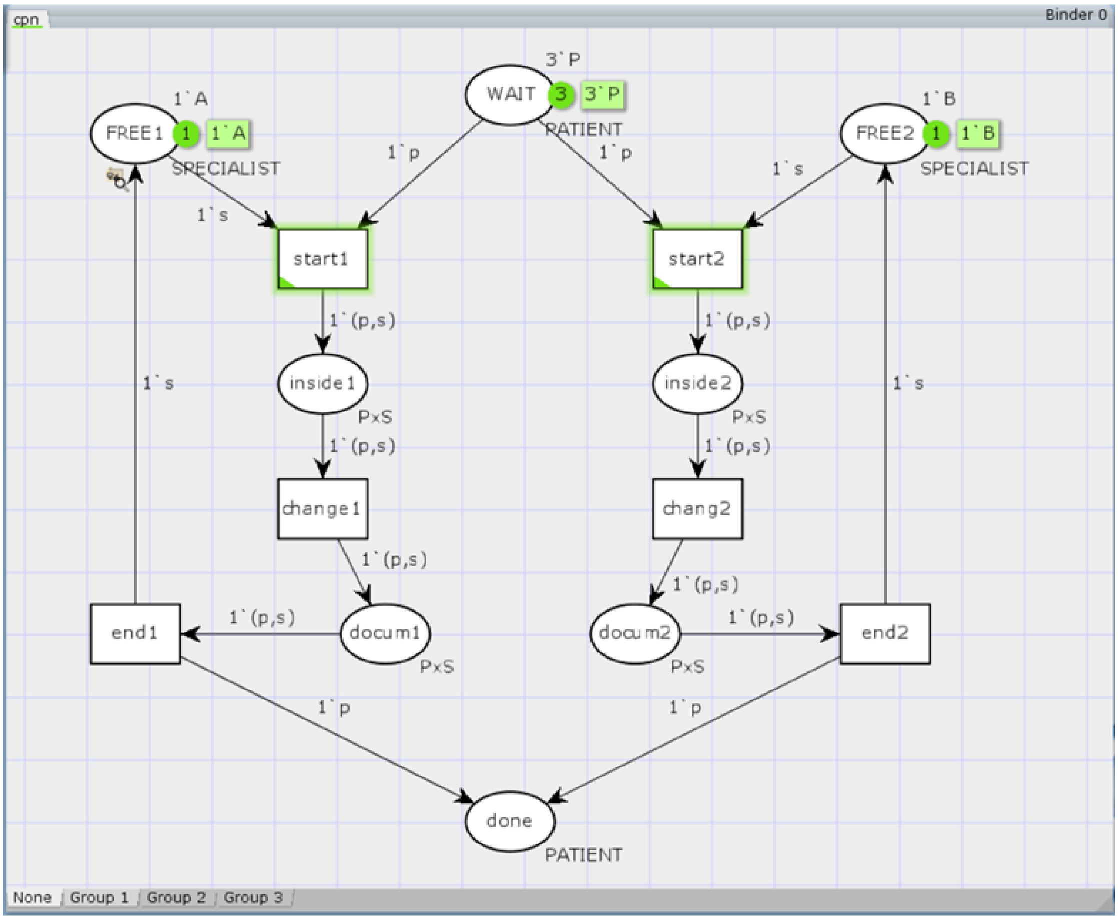
The second idea

$$N_2 = (P, T, F)$$

where

$$P = \{free1, free2, wait, inside1, inside2, docu1, docu2, done\},$$
$$T = \{start1, start2, change1, change2, end1, end2\}$$

With the tokens in states *free1* and *free2* represent the *specialist1* and *specialist2* are free respectively and waits for the next patient. The tokens in state *wait* represent the patient is waiting, one in state *inside1* and *inside2* represent the *specialist1* and *specialist2* are busy treating the doctor is treating as well as the patient being treated by the doctor respectively. And then state *docu1* and *docu2* represent the *specialist1* and *specialist2* are documenting the result of the treatment respectively and the end *done* states represent the patient has been treated by the specialist. The construction N_2 below with initial marking $M_0[free1, free2, 3.wait]$:



The tokens inside states *free*, *free1*, *free2* is no more than one token. State *wait*, gets value n if there are n patients waiting.

6.7 Problem 7

6.7.1 [Package description]

Language used: R

[ITEM 1] PATIENT NETWORK
This program allows:
1) MAX: 10 patients in place wait.
2) Only one patient being treated at a time.
3) The process will run until there is no patient left in the waiting room.

[ITEM 2] SPECIALIST NETWORK
This program allows:
1) MAX: 1 specialist on duty.
2) Only one patient being treated at a time.
3) The number of firing times will define the final display.

[ITEM 3] SUPERIMPOSED NETWORK
This program allows:
1) MAX: 10 patients in place wait and 1 specialist on duty.
2) Only one patient being treated at a time.
3) The process will run until there is no patient left in the waiting room.

[ITEM 4] CALCULATOR

6.7.2 [Source code]:

Click here

References

[1] https://en.wikipedia.org/wiki/Petri_net

[2] https://www.sciencedirect.com/topics/physics-and-astronomy/petri-nets

[3] http://www.scholarpedia.org/article/Petri_net

[4] Girault C, Valk R (2003) Petri Nets for systems engineering. Springer, New York

[5] Peterson JL (1981) Petri Net theory and the modeling of systems. Prentice-Hall, Englewood Cliffs

[6] Murata T (Apr 1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4):541–580

[7] Modeling business processes a Petri net-oriented approach by Aalst, Wil van der Stahl, Christian

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

[9] DATA MINING- Concepts, Models, Methods, and Algorithms, 3rd Edition, 2020, IEEE press and Wiley, by Mehmed Kantardzic