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

MATHEMATICAL MODELING (CO2012)

Assignment

PETRI NETWORKS

Advisor: Nguyen Tien Thinh

Students: Nguyen Cong Dat - 2052942
Nguyen Minh Hung - 2052504.
Nguyen Quoc Minh Thu -2052736.
Tran Trang Ky Phong - 2053325.
Trinh Vo Hoang Nam - 2052606.
La Ky Phuong - 2014203.

HO CHI MINH CITY, SEPTEMBER 2020

Contents

1 Member list & Workload

No.	Full name	Student ID	Problems	Percentage
1	Nguyen Cong Dat	2052942	- Preliminary Knowledge: 4.Application. - Assignment Problem 6,Practise Problem. - Code: 30% - Report: 30%	20%
2	Tran Trang Ky Phong	2053325	- Preliminary Knowledge: 4.Application. - Assignment Problem 5, Practise Problem. - Code: 35% - Report: 15%	20%
3	Nguyen Quoc Minh Thu	2052736	- Preliminary Knowledge: 3.Concepts - Assignment Problem 1, Practise Problem. - Code: 10% - Report: 30%, .Log file.	20%
4	Nguyen Minh Hung	2052504	- Preliminary Knowledge: 2.Introduction. - Assignment Problem 4. - Code: 5% - Report: 10%	14%
5	La Ky Phuong	2014203	- Preliminary Knowledge: 4.Application. - Assignment Problem 2. - Code: 5% - Report: 10%	13%
6	Trinh Vo Hoang Nam	2052606	- Preliminary Knowledge: 2.Introduction. - Assignment Problem 3. - Code: 5% - Report: 5%	13%

2 Introduction to Process Mining

2.1 Overview of Process Mining

Process mining emerged as a new research field that focuses on the analysis of process using event data since 1990s. Nowadays, process mining is applicable to the wide range of field. For example, more and more business process are being conducted under the supervision of enterprise information system. There are three stages in process mining:

- **Process Discovery:** based on event logs of practical process, using certain techniques some model is constructed.
- **Process Conformance:** is a process mining techniques to compare a process model founded in discovery stage with event logs. It is used to check if the actual process conforms to the model and vice versa. The model should balance four competing quality criteria: *simplicity*, *fitness*, *precision*, and *generalization*.
- **Process Enhancement:** the goal of this step is to enrich the model.

2.2 Transition System

A system can stay in the same state for a short or a long time, but it normally change from one state to another after a certain of time. For example, when an elevator go up or down, the state of it changes. To handle that changes in real model, we need a tool to represent the system and its changes. That is a **Transition System**.

To deeply understand the concept of *transition system*, firstly we need to learn how transition is defined in system. In practical application, we abstract away the time being needed for a transition. During a transition, the system changes from one state to other state, we are not interested in what exactly happens during the time change, we only focus on the old state and the new state. For that reason, the transition can be written as a **ordered pair**. In case of the elevator, we are interested in the starting floor and the ending floor. The ordered state pair **(1^{st} floor, 3^{rd} floor)** describe exactly what is going on, the ordered pair represents action the elevator go from the first floor to the third floor.

Definition 1 (Transition) *The transition is an ordered pair (x, y), in which x, y are elements of state space S – that is $x, y \in S$*

In mathematics, the set of ordered pairs is a binary relation. So the set of all transitions in the system is a **transition relation**. The transition relation is unique for each system denoted by identifier TR. By specify the state space S, the transition relation TR, and the initial state, we can describe a system, that is called **Transition System**.

Definition 2 (Transition System) *A Transition System is a triple (S, TR, s_0) where S is a finite set of state, $TR \subseteq S \times S$ is a binary relation containing all possible state changes of the system, and $s_0 \in S$ is the initial state.*

♣ **Example 1** A simple elevator system serving a building with five floor. Define its transition system, and draw a state-transition diagram. Assume that the elevator is initially at the ground level.

The simple elevator system is **dynamic**, because it does not stay in one state but jumps one floor to the next. We have the **transition system**

- The state space $S = \{0, 1, 2, 3, 4\}$, one states for each floor.
- The transition relation $TR = \{(0, 1), (1, 2), (1, 0), (2, 1), (2, 3), (3, 2), (3, 4), (4, 3)\}$
- Assume that the initial state is $s_0 = 0$, figure 1 depicts the state-transition system

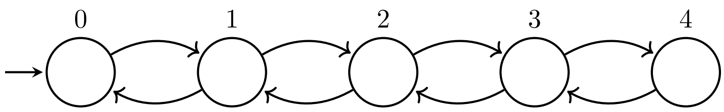


Figure 1: State-transition diagram for the elevator system

2.3 Petri Net

Nowadays, more and more business process are being conducted under the supervision of enterprise information system which can be modeled as a transition system. However, as a modeling tool, the transition systems are not suitable, because it is too hard to describe a complex system in term of state space and transition relation. **Petri Nets** emerge as a more advanced modeling tool, with several advantages over the *transition systems.*

We consider the *elevator system* of previous example, **Figure 1** using two Petri Net. The first Petri Net has a structure very close to the transition system. The second shows why Petri Net is more suitable for process the complex model than a transition system.

2.3.1 The first example of Petri Net

The elevator serves the building with 5 floor, it moves between these floor and can stop at any of these floor. For that reason, this model has five *places*, each place symbols for one floor in this building. Moreover, at each floor (except the first and the last) the elevator can choose two action: go up or down. Each action in practical system, is represented in Petri Net by a *transition.* We consider the model:

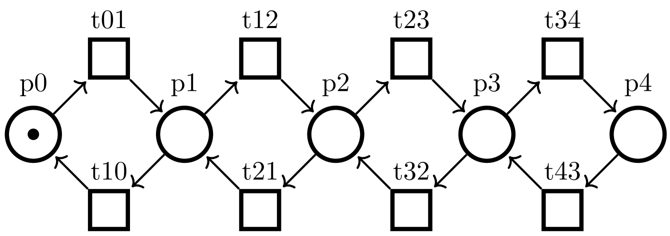


Figure 2: The elevator system modeled by a Petri Net

In **Figure 2**, graphically, a circle represents a *place*, and the *transition* is represented by a square. A transition t01 represents the action go from the ground to the 1^{st} floor, the same for other transitions. A *token* (black dot) indicates which floor the elevator currently in. For example, in **Figure 2** there is one token in place p0, show that the elevator is in the ground floor. In the Petri net, the places, the transitions, the arcs is fixed, but the transition can take the token from one place to another place which is called *firing*. For example, in **Figure 2**, the transition t01 can move the token from place p0 to place p1. In practical, it means that the elevator move from the ground to the first floor. The model after t01 firing in **Figure 3**.

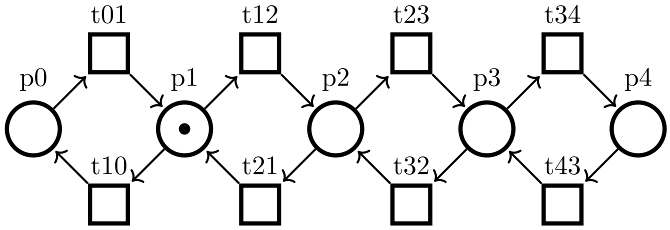


Figure 3: The elevator system after t01 firing

The previous example, the Petri net model has 5 places and eight transitions, whereas the transition system has only 5 states. Does The Petri net look more complicated than the transition system. In this example, the answer is Yes. The reason is in example 1 **Figure 1** each place represent one state in transition system, but there is ways to implicit describe the state of this system compactly. To prove that, we consider the following example.

2.3.2 The second example of Petri Net

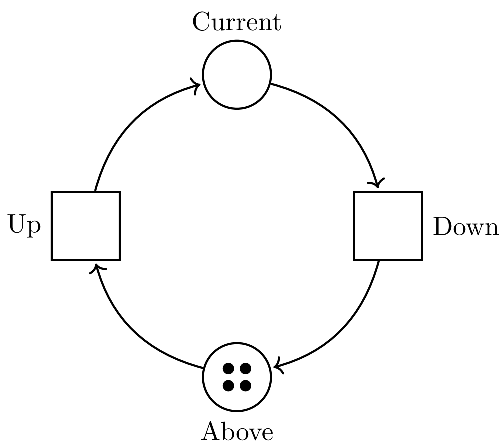


Figure 4: State-transition diagram for the elevator system

In this example, each place is not the floor. We have only two places: Current and Next, and two transitions: Up and Down. In this elevator system, we have to know exactly, which floor the elevator in and each position of elevator is one state of this system. So in **Figure 4**, we need to answer the question what floor the elevator in. So to represent the state of system, in Petri net, we can use the token with different meaning. In the first **example 2**, a place contains token means the elevator is in that floor. In this example, the number tokens in place Current is the floor the elevator in. There is the advantage of Petri net, each place does not need to represent a state of this system because in the complex system, it can have the infinite number of states.

In the model, if the elevator travel one floor up, one token should be consumed from place Above, and one token should be produced in place Current. The number of token in place Current increase one so that the elevator go up one floor. If the elevator travel on floor down, the opposite should happens.

3 Basic Concepts of Petri Nets

3.1 Petri Net - Structure

The Petri net is determined if we know the places, the transition and the arcs. A Petri net can contains one or more places, each place has unique name, which is called label. We can describe the all places of the Petri net as the set of all place label P. We can describe the set of transitions in the same way. The set T is the set of all transition labels in the system. For example, let consider the example in **Figure 2**. We can define the set of places:

$$P = \{p_0, p_1, p_2, p_3, p_4\}$$

and the set of all transitions

$$T = \{t_{01}, t_{10}, t_{12}, t_{21}, t_{32}, t_{23}, t_{34}, t_{43}\}$$

. The set of arcs in the Petri net is defined differently. In Petri net, there are two type of arcs: the arc from a place to a transition and the opposite from a transition to a place. We can define the arc of the Petri net using ordered pair (x, y). The set of all arcs in Petri net is the set of all ordered pair, and as we know in mathematics, it is called the binary relation. The relation $R_1 \subseteq S \times T$ is the set of all arcs from a place $p \in S$ to a transition in T and the relation $R_2 \subseteq T \times S$ contains all arcs connecting a transition and its output place. For the Petri net in **Figure 2**, the two relations is defined as:

$$R_1 = \{(p_0, t_{01}), (p_1, t_{10}), (p_1, t_{12}), (p_2, t_{21}), (p_2, t_{23}), (p_3, t_{32}), (p_3, t_{34}), (p_4, t_{43})\}$$

$$R_2 = \{(t_{01}, p_1), (t_{10}, p_0), (t_{12}, p_2), (t_{21}, p_1), (t_{23}, p_3), (t_{32}, p_2), (t_{34}, p_4), (t_{43}, p_3)\}$$

The union $R_1 \cup R_2$ represents all arcs of the Petri net. This union denoted by F, the flow relation. From set P, T, F we can construct the Petri net, and form the Petri net we can derived these sets. So we have a definition of Petri net.

Definition 3 (Petri Net) *Petri Net is a bipartite directed graph N of places and transitions, a Petri Net is defined by a triplet $N = (P, T, F)$, where*

- P *is a finite set of places*
- T *is a set of transitions*
- $F \subseteq (T \times P) \cup (P \times T)$ *is a flow relation*

- A **token** is a special transition node, being graphically rendered as a black dot, place can contain tokens, and transition **cannot**
- A **transition** is *enabled* if each of its input places contains a **token**

We have a definition of marked Petri net

Definition 4 (Marked Petri Net) *A marked Petri net is a pair* (N, M)*, where*

1. N *is a Petri net*
2. M *is a multi-set over* P *denoting the marking of the net*

♣ **Example 1** Consider the Petri net for X-ray machine in **Figure 5**, describe the Petri net by a set of places, a set of transitions and a set of arcs.

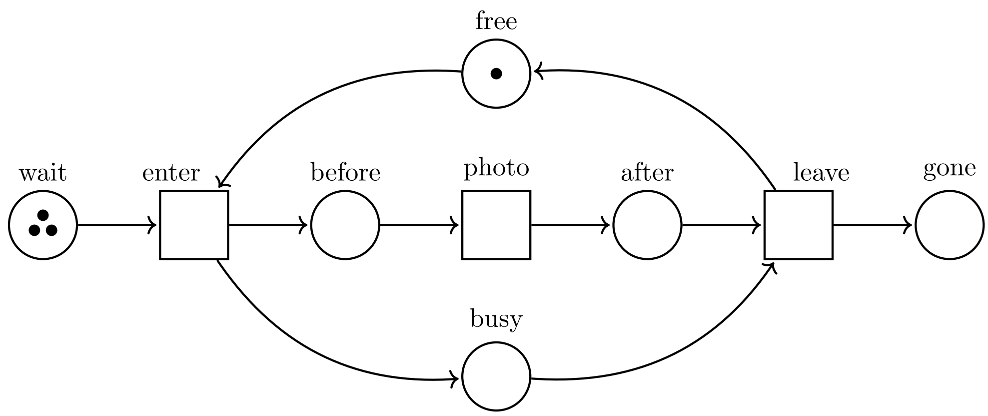


Figure 5: A Petri net for the process of an X-ray machine

The Petri net in **Figure 5** has three transitions (draws as square): T = {enter, photo, leave} and five places (circle): P = {wait, before, free, after, gone, busy}. The set of arcs in the net is F = {(wait, enter), (enter, before), (before, photo), (photo, after), (after, leave), (leave, gone), (leave, free), (free, enter), (enter, busy), (busy, leave)}

3.2 Petri Net - Behaviour

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. In previous section, we quickly introduce token, but we do not have a tool to represent the distribution of tokens in the Petri net. That tool is a multi-set

Definition 5 (Multi-set) *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.*

♣ **Example 2** Consider a set $D = \{a, b, c\}$ consisting of three elements. A possible multi-set over D is, for example, $[a, a, a, b, c, c, c]$. In this multi-set, element a appears three times, element b once, and element c appears three times. A more compact representation of this multi-set is $[3.a, 1.b, 3.c]$ or just $[3.a, b, 3.c]$

The function m assigns to each place $p \in P$ a natural number that specifies the number of tokens in this place p. In **Figure 5**, we can defined m as: **m(wait) = 3, m(free) = 0, m(before) = 0, m(after) = 0, m(gone) = 0, m(busy) = 0**. We can, therefore, represent the marking m as the multi-set **m = [wait, wait, wait, free]** or more compact as **m = [3.wait, free]**. We now summarize the preceding discussion in the following definition.

Definition 6 (Marking) *A marking of a Petri net (P, T, F) 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.*

To define the concept of firing and enableness, we must know the *input* and the *output* places of a transition. A place p is a input place of a transition t if there is a arc from p to t. Likewise, the place p is a output place of a transition t if there is a arc from t to p.

♣ **Example 3** What are the output places and input places of transition t_1 in **Figure 6**

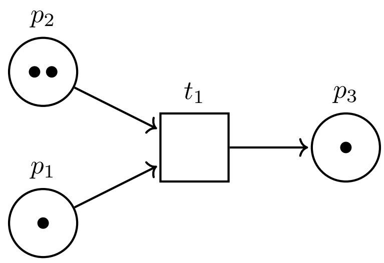


Figure 6: A simple Petri Net

Transition t_1 has two input places, p_1 and p_2, and only one output places, p_3. We can define input places and output places of t_1 in term of **preset** and **postset** of t_1. The preset $^\bullet t = \{p \mid (p, t) \in F\}$ is the set of all input places of a transition t. The postset $t^\bullet = \{p \mid (p, t) \in F\}$ defines all output places of a transition t. Back to this example, we have:

- The postset of t_1: $t_1{}^\bullet = \{p_3\}$
- The preset of t_1: $^\bullet t_1 = \{p_1, p_2\}$

We now can define the concept of **enabling**. In an informal way, a transition $t \in T$ is *enabled* at marking m if every input place of t contains at least one token

Definition 7 (Enabledness) *In a Petri net (P, T, F), a transition $t \in T$ is enabled at marking $m : P \to \mathbb{N}$ if and only if for all $p \in {}^\bullet t$, $m(p) > 0$.*

An enabled transition t can fire, it consumes one token from each input places and produces one token in each of its output places, thereby it changing the marking m to the marking m'. We can calculate the number of token of place p in marking m' from marking m as follows:

- If p is an input place of t, means $(p, t) \in F$ then $m'(p) = m(p) - 1$
- If p is an output place of t then $m'(p) = m(p) + 1$
- If p is not connected to t or p is both output and input place of t then the number of token in p stay remain, it means $m'(p) = m(p)$

Based on this idea, we can formalize the effect of firing of the transition by a function w. Formally, this function is defined by w : if $(x, y) \in F$ then $w(x, y) = 1$ and $w(x, y) = 0$ when $(x, y) \notin F$.

Definition 8 (Transition firing) *For a Petri net (P, T, F), let w be the weight function and $m : P \to \mathbb{N}$ is a current marking. A transition $t \in T$ can fire if and only if it is enabled at m. The firing t yields a new marking $m' : P \to N$ where for all places $p \in P$, $m'(p) = m(p) + w((t, p)) - w((p, t))$*

To determine the behavior of a Petri net, we need to specify the structure (P, T, F) and an initial marking. The Petri net with the structure and an initial marking is a Petri net system.

Definition 9 (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 4** Formalize the Petri net system in **Figure 7** as q quadruplet

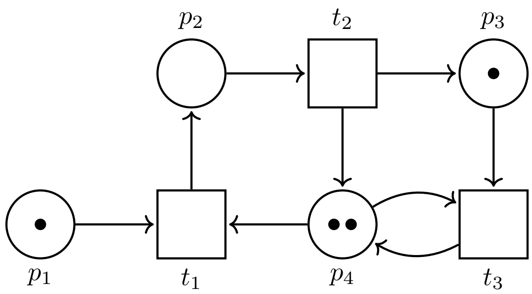


Figure 7: The Petri net system for example

This net can be represent as a quadruplet (P, T, F, m_0)

- $P = \{p_1, p_2, p_3, p_4\}$
- $T = \{t_1, t_2, t_3\}$
- $F = \{(p_1, t_1), (p_4, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (t_2, p_4), (p_4, t_3), (t_3, p_4), (p_3, t_3)\}$
- $m_0 = [1.p_1, 0.p_2, 1.p_3, 2.p_4]$

3.3 Representing Petri Nets as Transition Systems

As described, the behavior of a system can be represented by a transition system (S, TR, s_0). Transition systems are the primal models of process modeling. We can also describe the behavior the a Petri net system as a transition system, by showing how to determine the state space S, the transition relation TR and the initial state m_0 for the system (P, T, F, m_0).

The state of a Petri net system is the distribution of token over its places. For that reason, we can say that, **each state $s \in S$ of a transition system is a marking m of a Petri net system**.

Let consider two arbitrary states in state space S- that is, two marking m and m'. Transition $(m, m') \in TR$ if there is a transition t enabled at marking m, and the firing transition t in the marking m yields marking m'.

Now we can determine the state space S and the transition relation TR in transition system for the Petri net system. s_0 is the initial state of transition system, so it is the initial marking at Petri net. We formalize this process converting from Petri net system to transition system.

Definition 10 (Petri Nets as Transition Systems) *A Petri net system $N = (P, T, F, m_0)$ can be converted into a transition system (S, TR, s_0) with:*

- *S is all reachable markings of net N*
- *$TR = (M, M') \in S \times S$ where $\exists t \in T \mid (N, M)[t\rangle(N, M_1)$*
- *$s_0 = M_0$*

4 Application of Petri Nets

4.1 Essential notion for Modeling with Petri Nets

A Petri net consists of places, transitions, and tokens. When we model a system as a Petri net, we must decide whether a certain aspect or part of the system should be represented as a place, transition, or as a token. To answer such questions, we discuss in this section the roles of that places, token, and transitions can play.

4.1.1 The role of token

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;
- An indicator of a condition: the presence of a token indicates whether a certain condition is fulfilled.

4.1.2 The role of place

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;
- 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.

4.1.3 The role of transition

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;
- 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.

4.2 Modeling by Petri networks- Problem

In practice, the system has many objects interacting with each other - for example, in the road, there is the interaction between pedestrians, traffic lights, and vehicles. So that in the Petri net model, it may have distinct places, tokens and transitions. It should be the superimposition of the smaller.
Question: 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?
To answer this question, we need to know about the concept of superimposition operator.

Definition 11 (The superimposition (or merging) operator) *The Petri net has two agent types, denote* $N_1 = (P_1, T_1, F_1, M_0)$ *and* $N_2 = (P_2, T_2, F_2, M_0)$.

Let $t_1 \in T_1$ *and* $t_2 \in T_2$. *The superimposition operator,* $\bigoplus : T_1 \times T_2 \to T$ *defined as follow:*

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

Else $\bullet t_1 \neq \bullet t_2$ $(t_1, t_2) \mapsto \bigoplus(t_1, t_2) = \{t_1, t_2\} \subseteq T$, *keep both presets.*
Namely we can identify two transitions/events of two nets into one node of the merged Petri Net $N = N_1 \bigoplus N_2$ *if the events act on the same physical token.*

The superimposed (merged) Petri net is determined by
$N = N_1 \bigoplus N_2 = (P_1 \cup P_2, T, F_1 \cup F_2, M_0)$

4.3 Typical Network Structures

Places and transitions in Petri net are connected by arcs. The way in which these nodes connected determines the behavior of the network. Therefore, it is useful to consider several typical network structures.

4.3.1 Causality

Causality is a relationship between two events in a system that must take place in certain order. In a Petri net, we may represent this relationship by two transitions connected through an intermediate place, as shown in **figure 8**.

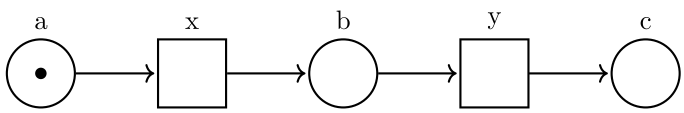


Figure 8: Transition y can fire only after transition x has fired

4.3.2 Concurrency and Synchronization

Concurrency (i.e., parallelism) is an important feature of information system. In a concurrency system, many events can occur simultaneously. For example, computer can run several tasks at the same time. Concurrency happens when there is two transition in a Petri net not directly connected with each other, such as in **figure 9**.

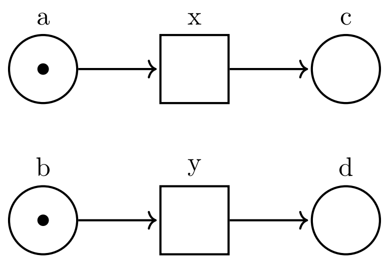


Figure 9: Transition x and y can fire simultaneously

This explicit modeling of concurrency is the advantage of using Petri nets over transition system. Transition system cannot explicitly model concurrency. Let consider the example to clearly understand that.

♣ **Example 1** Consider the Petri net in **figure 9**.How many states in equivalent transition system

As we discuss before, the each state in a transition system is one reachable marking in the Petri net. Now we count the number of reachable marking. In **figure 9**, at initial marking m_0 there is three cases:

- Transition x is fired, then we have marking $[b, c]$
- Transition y is fired, the same we have marking $[a, d]$
- Transition x and y can fired at the same time (concurrency case), we have marking $[c, d]$

Now let formalize this example, assume there are k concurrency transitions (in **figure 9** denotes the case $k = 2$). How many markings are reachable for an arbitrary k? We have k concurrency transitions. At the specific marking, each transition has two options: firing or not firing, and each option leads to different marking. For that reason, the number of state in transition system is 2^k. For example, if $k = 10$, then there are $2^{10} = 1024$ reachable state and $10! = 3628800$ possible transition sequences. This number show that the Petri net model can compactly represent concurrency system without directly enumerating all possible combinations of transition sequences.

4.3.3 Mutual Exclusion

In many system, there are safety requirements of the following kind: after event x, either event y or event z can happens. This is known as **mutual exclusion**. In **figure 10**, after transition x fires, both transition y and z are enabled. At that moment, a *choice* has to be made (nondeterministic choice)

♣ **Example 2** Mutual exclusion can be used to model the situation of a light connects to two switches, either of two switches can toggle the light. In **figure 11**

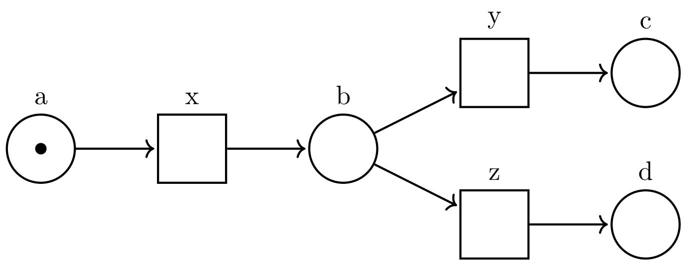


Figure 10: After x fires, transition y and z are enabled

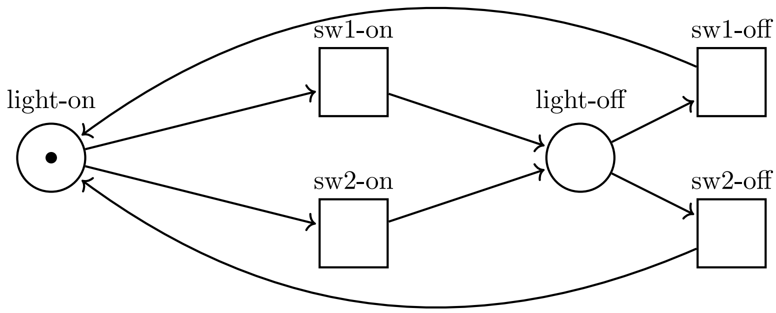


Figure 11: After x fires, transition y and z are enabled

4.4 Special properties of the Petri net

Now we can construct the Petri net system, however to evaluate and get the information from this model, this is vital to grasp the concepts of some special properties of the Petri net. Here are some generic properties in Petri net.

1. K-bounded: A marked Petri net (N, M_0) is k-bounded if no place ever contains more than k tokens. Formally, for any $p \in P$ and any $M \in [N, M_0\rangle : M(p) \leq k$. In **figure 12** , the

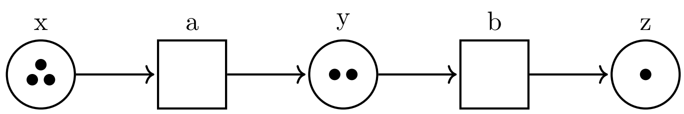


Figure 12: 6-bounded Petri net

Petri net is 6-bounded because in none of the reachable markings there is a place with more than 6 tokens. It is not 5-bounded, because in final marking place z contains 6 tokens.

2. Safe: A marked Petri net is safe if and only if it is 1-bounded.
The marked Petri net shown in **figure 10** is safe because in each of the reachable markings there is no place holding multiple tokens.

3. Bounded: 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 12** is bounded with $k = 6$.

The Petri net in **Figure 13** below is unbounded, because the number of token in place green can be infinity.

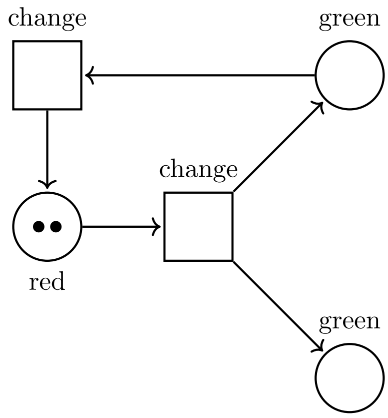


Figure 13: Unbounded Petri net

4. Deadlock free: A marked Petri net (N, M_0) is deadlock free if at every reachable marking at least one transition is enabled. In **Figure 12** shows a net that is not deadlock free because at marking $[z.6]$ no transition is enabled.

5. Live: A marked Petri net is live if each of its transitions is live. Transition is live if from every reachable marking it is possible to enable t.

5 Practice Problem

5.1 Practical Problem 1

Given a process of a **X-ray machine** in which we assume the first marking in **Figure 14** shows that there are three patients in the queue waiting for an X-ray. Figure 8 depicts the next marking, which occurs after the firing of transition enter. Determine the two relations R_1 and R_0 , and

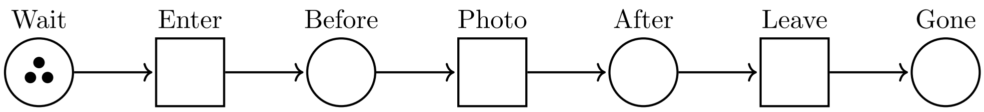


Figure 14: **transition enter** not fired

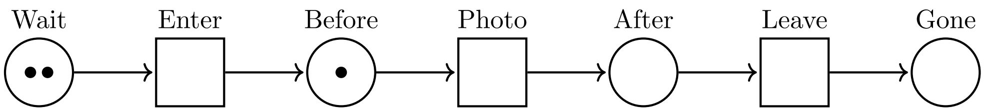


Figure 15: transition **enter**

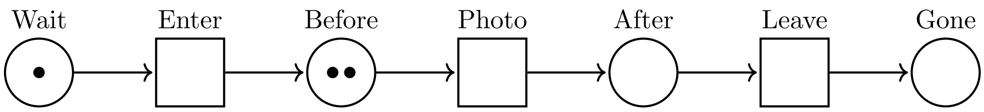


Figure 16: **transition enter** has fired again

the flow relation $F = R_1 \cup R_2$.

- $R_1 = \{$(wait, enter), (before, make-photo), (after, leave)$\}$
- $R_0 = \{$(enter, before), (make-photo, after), (leave, gone)$\}$
- $F = \{$(wait, before), (before, after), (after, gone)$\}$

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 below.

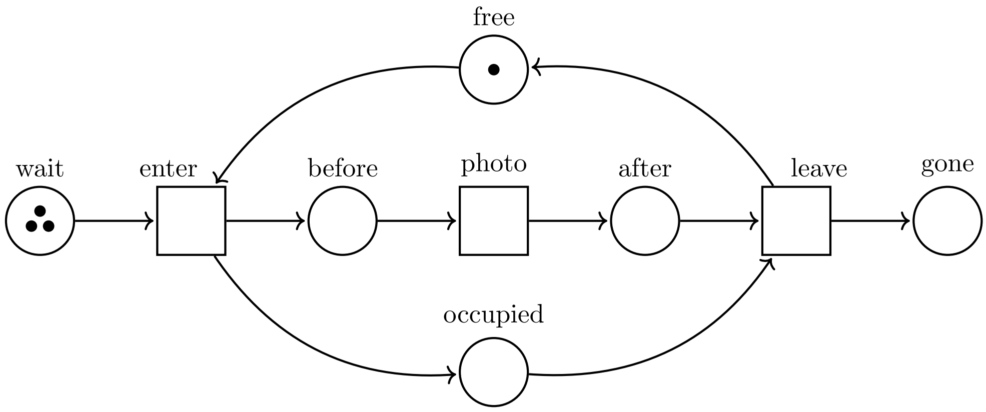


Figure 17: A improved Petri net for the business process of an X-ray machine

Now for this Petri net, can place **before** contain more than one token? Why?
Answer: The place before can not have more than one token because when there is one token in before, if the next token in wait is consumed to produce one in before, the token in before also be consumed to produce one in after.

As long as there is no token in place **free** [Figure 2.2.4], can transition **enter** fire again?

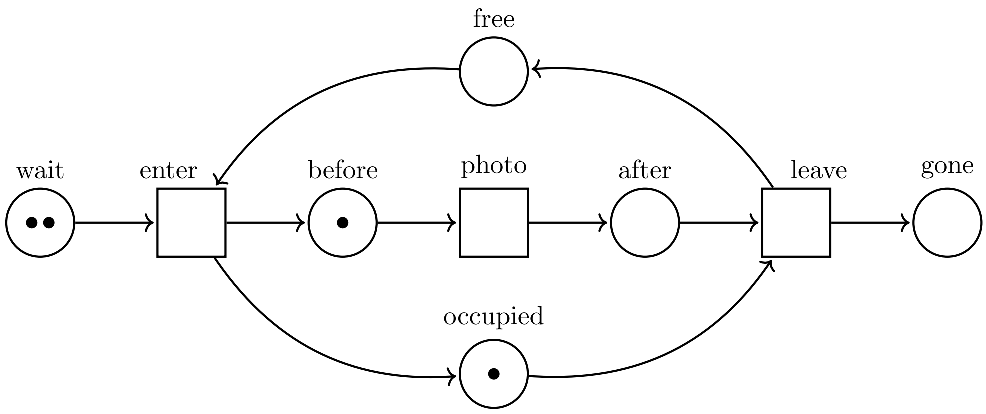


Figure 18: The marking of the improved Petri net for the working process of an X-ray room after transition **enter** has fired.

Explain why or why not. Remake the two relations *R1* and R_0.

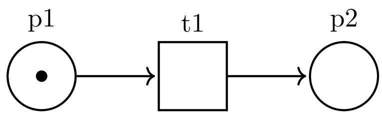
Answer: enter can not fire because it is not enabled.

- R_1 = {(wait, enter), (before, make-photo), (after, leave), (occupied, leave), (free, enter) }
- R_0 = { (enter, before), (make-photo, after), (leave, gone), (enter, occupied) }

5.2 Practical Problem 2

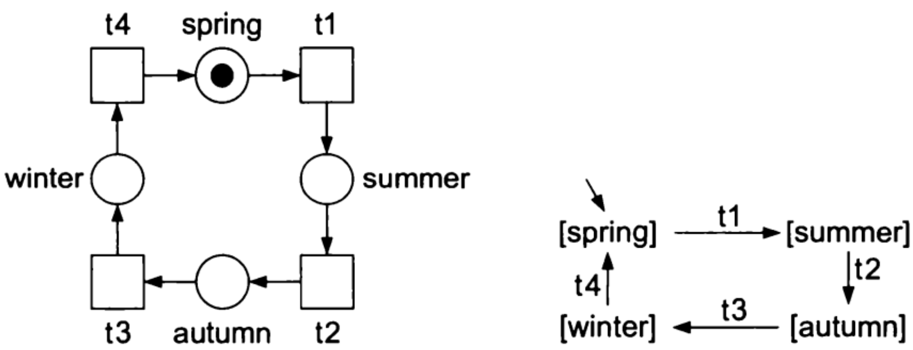
Explain the following terms for Petri nets, and provide a specific example for each term.

1. "Enabled transition" 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: t1 is enabled in this figure below.



2. "Firing of a transition" can be characterised by subtracting a number of tokens from its input places equal to the multiplicity of the respective input arcs and accumulating a new number of tokens at the output places equal to the multiplicity of the respective output arcs:

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. From

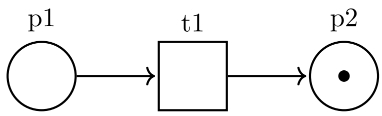


(a) The four seasons (b) Reachability graph

the reachability graph in figure (b), we can conclude that the net in figure (a) has four reachable markings. 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.

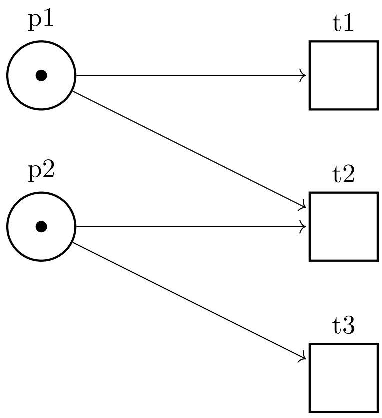
We refer to this transition sequence as a *run* (as an execution in *finite* automaton). A run is finite if the path and hence the transition sequence is finite. Otherwise, the run is *infinite*.
The path from marking [spring] to marking [winter] is a finite run ⟨t1, t2, t3⟩ of the net.
The net also has one infinite run:⟨t1, t2, t3, t4, t1, ...⟩

4. "Terminal marking," The transitions keep firing until the net reaches a marking that does not enable any transition.

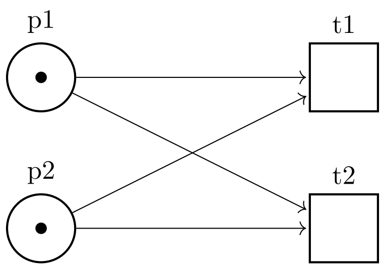


In this simple case the Terminal Marking is M={0,1}.
We also consider the above example, Four-season net. The figure (b) does not have terminal markings; that is, at each of the reachable marking, a transition is enabled. The reachability graph for this simple example does not provide new insights, however, for industrial systems with billions of states, reachability graphs are helpful.

5. "Non deterministic choice" When several transitions are enabled at the same moment, it is not determined which of them will fire.



In this case, t1, t2 and t3 are enable at the same time. This kind of case is called Behaviorally Free choice.



In this case, t1, t2 are enable at the same time. This kind of case is called Extended Free choice

5.3 Practical Problem 3

Consider the Petri net system in figure below.

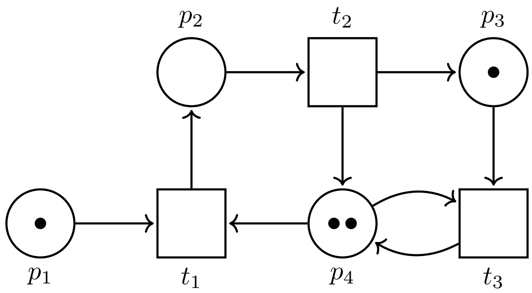


Figure 19: Problem 3.

1. Formalize this net as quadruplet $(\mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{M_0})$.
This Petri net can be represented as a quadruplet (P, T, F, M_0) with:

- $P = \{p_1, p_2, p_3, p_4\}$
- $T = \{t_1, t_2, t_3, t_4\}$
- $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_3), (t_3, p_4), (p_4, t_3), (p_4, t_1\}$

2. Give the preset and the postset of each transition.

- The postset and preset of t_1: $t_1\bullet = \{p_2\}$, $^\bullet t_1 = \{p_1, p_4\}$
- The postset and preset of t_2: $t_2\bullet = \{p_3, p_4\}$, $^\bullet t_2 = \{p_2\}$
- The postset and preset of t_3: $t_3\bullet = \{p_4\}$, $^\bullet t_3 = \{p_3, p_4\}$

3. Which transitions are enabled at M_0.
At initial marking, transition t_1 and t_3 are enabled.

4. Give all reachable markings. What are the reachable terminal markings

- The Petri net has **7** reachable markings: $[p_1, p_3, 2.p_4]$, $[p_2, p_3, p_4]$, $[2.p_3, 2.p_4]$, $[p_3, 2.p_4]$, $[2.p_4]$, $[p_2, p_4]$, $[p1, 2.p_4]$.
- Reachable terminal marking is $[2.p_4]$

5. Is there a reachable marking in which we have a non deterministic choice?
Yes, there are $[p_1, p_3, 2.p_4]$ and $[p_2, p_3, p_4]$

6. Does the number of reachable markings increase or decrease if we remove
a) Place p_1 and its adjacent arcs?

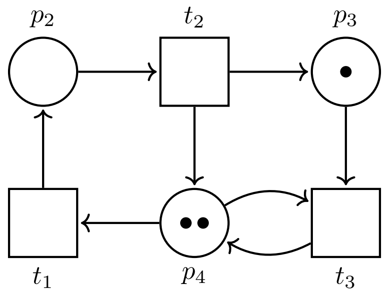


Figure 20: Petri net after remove p_1

We have a Petri net after remove p_1 and its adjacent arcs in **Figure 20**. Clearly, the number of reachable markings increase, because, The number of reachable markings increase when remove p_1 and its adjacent arcs. Because, in the Petri net (**figure 19**), we have the sequence $< t_1, t_2 >$. When firing this sequence, the number of tokens in p_4 stays remain and tokens in p_3 increase 1. For that reason, the number of token in the Petri net can be infinite, and we will have infinite reachable markings.

b) Place p_3 and its adjacent arcs?

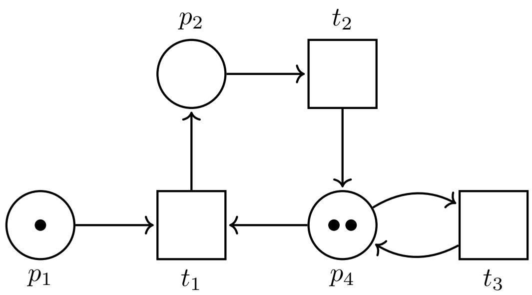
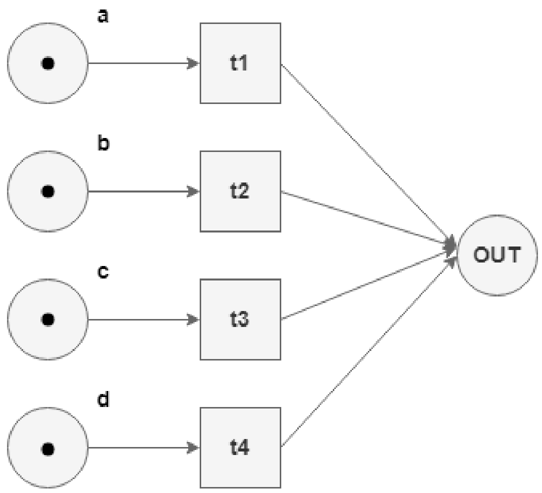


Figure 21: The Petri net after remove p_3

We have the Petri net in **figure 21**. The number of token in this Petri net decrease from 7 reachable markings to 3 markings. These markings are $[1.p_1, 2.p_4]$. $[1.p_2, 1.p_4]$ and $2.p_4$.

5.4 Practical Problem 4

From small marked Petri net to bigger transition system.

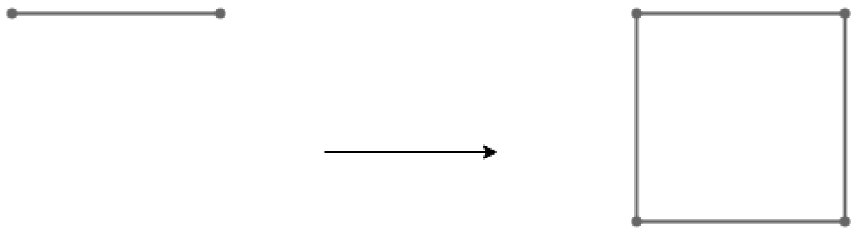


Write down M_0, P, T.

- M_0 = [3, 3, 3, 3, 0]
- P = {a, b, c, d, OUT}
- T = {t1, t2, t3, t4}

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?

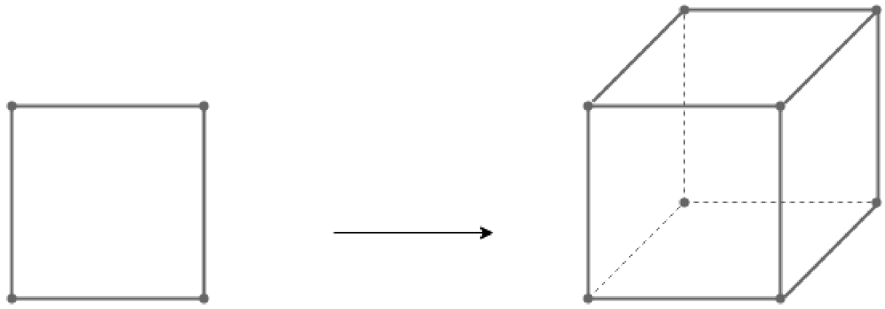
- Extend Idea **Hamming distance**



$$\begin{cases} V_1 = 2 \\ V_{n+1} = 2 \times V_n \end{cases}$$

- Transform the sequence to general formula using recursion

$$V_n = 2 \times V_{n-1} = 2^2 \times V_{n-2} = ... \Rightarrow V_= 2^{n-1} \times V_1 = 2^{n-1} \times 2 \Rightarrow V_n = 2^n$$



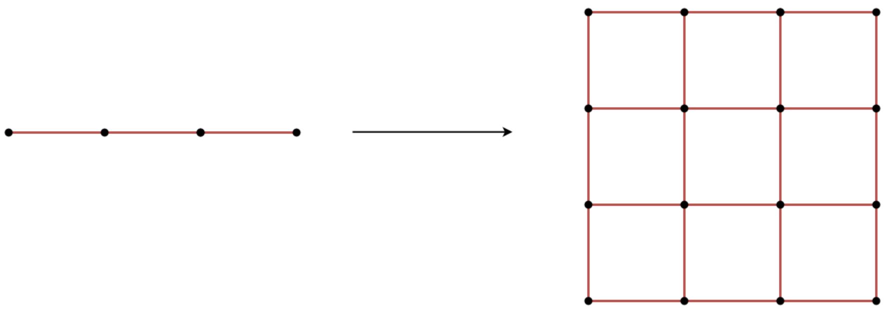
- Define a sequence of numbers illustrates edges of a hypercube (En) as:

$$\begin{cases} E_1 = 1 \\ E_2 = 4 \\ E_{n+1} 2 \times E_n + V_n \end{cases}$$

- We already have $V_n = 2^n$ in the previous proving. Transform the sequence to general formula using recursion:

$$E_n = 2 \times E_{n-1} + 2_{n-1} = 2 \times (2 \times E_{n-2}) + 2^{n-1} = ...$$
$$E_n = 2^{n-1} \times E_1 + (n-1) \times 2^{n-1} \times 1 + (n-1) \times 2^{n-1}$$
$$E_n = n \times 2^{n-1}$$

- The ideas is **Hamming distance** in the **Quaternary-cube** in 4 parallel state,



- We expressed geometry **Hamming distance** in the **Quaternary-cube** in 2D
Define a sequence of numbers illustrates vertices of a "quaternary cube" (V_n) . Transform the sequence to general formula using recursion

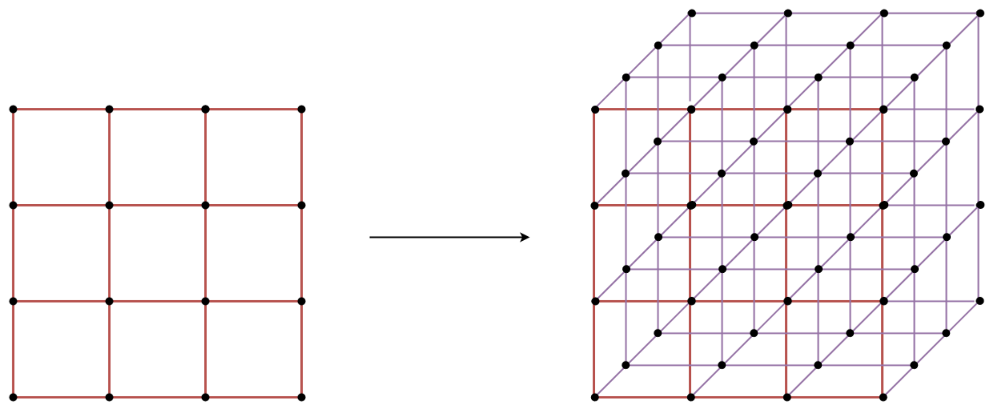
$$\begin{cases} V_1 = 4 \\ V_{n+1} = 4 \times n \end{cases}$$

- Transform the sequence to general formula using recursion:

$$V_n = 4 \times V_{n-1} = 4^2 \times V_{n-2} = ...$$

$$\rightarrow V_n = 4^{n-1} \times V_1 = 4^{n-1} \times 4$$

$$\rightarrow V_n = 4^n$$



$$\begin{cases} E_1 = 3 \\ E_2 = 24 \\ E_{n+1} = 4 \times E_n + 3 \times V_n \end{cases}$$

- We already have $V_n = 4^n$ in the previous proving. Transform the sequence to general formula using recursion:
  $E_n = 4 \times E_{n-1} + 3 \times 4_{n-1} = 4 \times (4 \times E_{n-2} + 3 \times 4^{n-2}) + 3 \times 4^{n-1} = ...$
  $\Rightarrow E_n = 4^{n-1} \times E_1 + 3 \times (n-1) \times 4^{n-1} = 4^{n-1} \times 3 + 3 \times (n-1) \times 4^{n-1}$
  $\Rightarrow E_n = 3 \times n \times 4^{n-1}$

- All possible cases are $4^4 = 256$ or 256 number of the states of TS
  $= 4 \times 4 \times 4 \times (3 + 3 + 3 + 3) = 768$ number of transition of TS.

6 ASSIGNMENT on MODELING EXERCISES

6.1 Problem 1:

Given the Petri net N_s modeling the state of the specialist, as in **Figure 22**.

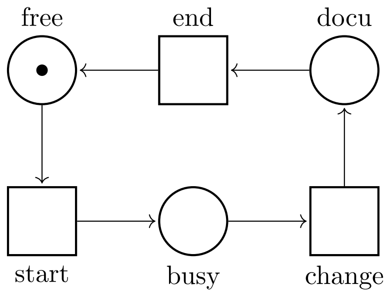


Figure 22: The Petri net of the specialist's state

a) Write down states and transitions of the Petri net N_s.

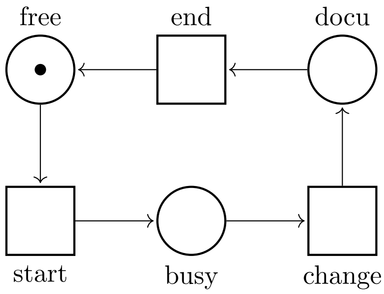
- States: $P = \{\text{free, busy, docu}\}$
- Transition: $T = \{\text{start, end, change}\}$

b) Representing the Petri net as a transition system.

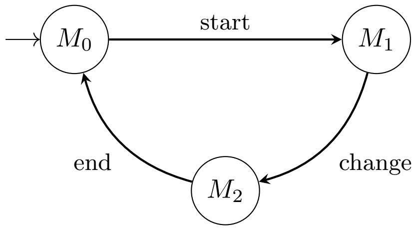
(i) Each place **cannot** contain more than one token in any marking. In that case, the net has only three reachable marking: $M_0 = \{1, 0, 0\} \rightarrow M_1 = \{0, 1, 0\} \rightarrow M_2 = \{0, 0, 1\}$. As we discuss in previous section about how to convert from the Petri net to a transition system, the converting follows three rules:

1. Each state in the transition system is a reachable marking in the Petri net
2. The initial state of the transition system is the initial marking M_0 in the net.
3. The transition relation in the transition system, connect two reachable marking M_0 and M_1 in which from firing one transition t, the Petri net changes from marking M_0 to M_1.

After following these rules, we get a transition system in **Figure 23b**. In figure 23a, we assume that, there is a token in place *free* at initial marking. Therefore, M_0 is an initial state in a transition system in figure 23b. If we change the position of token in the Petri net to place *busy* or *docu*, a initial state of a transition system will change to M_1 and M_2, respectively.



(a) The Petri net



(b) The equivalent transition system

Figure 23: Representing Petri net as a transition system

(ii) Each place may contain any natural number of tokens in any marking.
In that case, we formalize the first case, with the arbitrary number of tokens in place *free, busy, docu.* Assume that place *free* has x tokens, *busy* contains y tokens, and z tokens in place *docu.* We can easily realize that, the number of tokens in places off the Petri net from 0 to $x+y+z$. And the total tokens in the Petri net is $x+y+z$ also. Therefore, the number of reachable markings in the Petri net is a solution of equation

$$x_1 + x_2 + x_3 = x + y + z$$
$$x_1, x_2, x_3 \geq 0$$

Solve this equation, we get the solution

$$\binom{x+y+z+2}{2}$$

Let consider the simple case, to apply this formula. Suppose the initial marking of the Petri net is $(1,1,0)$ in **Figure 24**

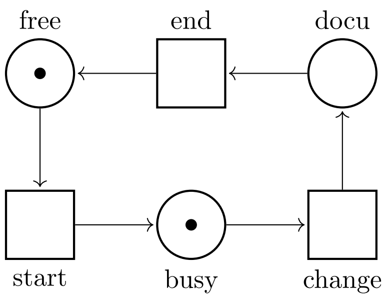


Figure 24: Example initial marking

Applying this formula, the number of reachable markings in this Petri net is $\binom{4}{2} = 6$. Actually, the Petri net in **Figure 24** has 6 reachable markings. The equivalent transition system is shown in **Figure 25**

$$M_0 = \{1,1,0\} \rightarrow M_1 = \{0,2,0\} \rightarrow M_2 = \{1,0,1\}$$
$$\rightarrow M_3 = \{0,1,1\} \rightarrow M_4 = \{2,0,0\} \rightarrow M_5 = \{0,0,2\}$$

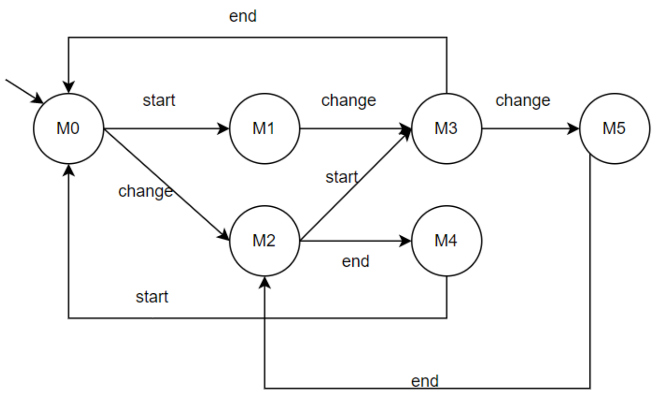
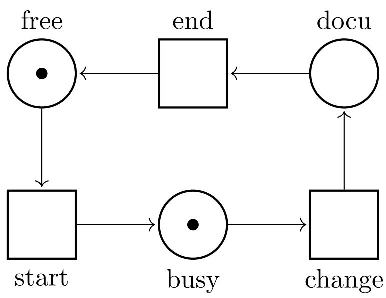


Figure 25: Transition system representation

6.2 Problem 2

Figure below of net N_S made by information of **Specialist** and **Event data** above.Define N_{Pa} 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**]
Answer: A token in state **inside** of the net N_{Pa} means that the patient is being treated by the specialist.

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**]
Answer:

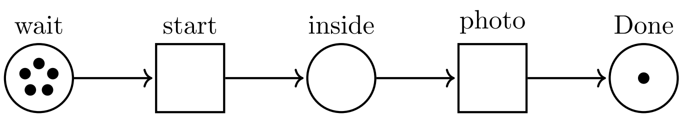
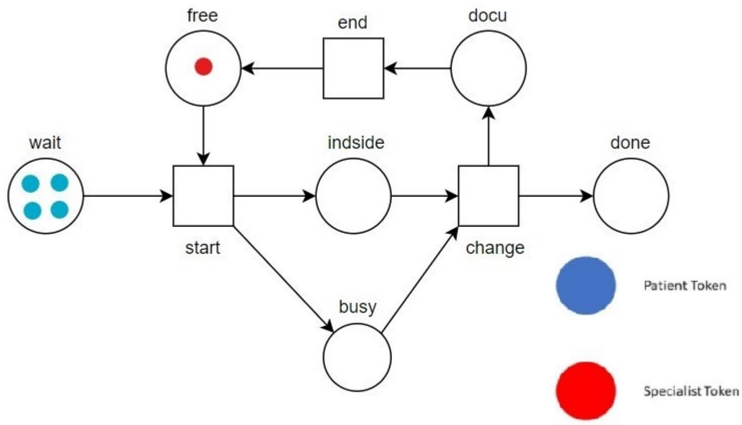


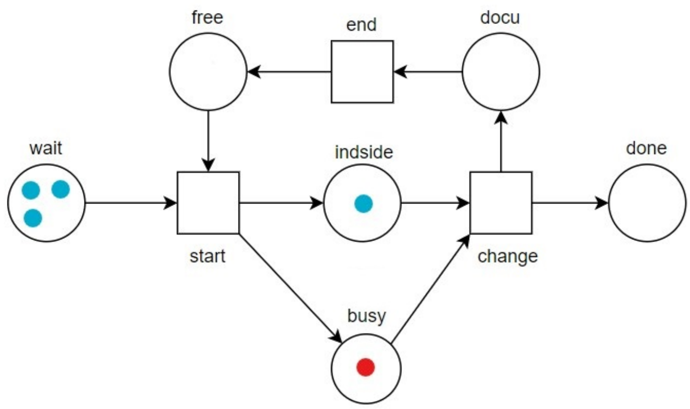
Figure 26: **transition enter** not fired

6.3 Problem 3

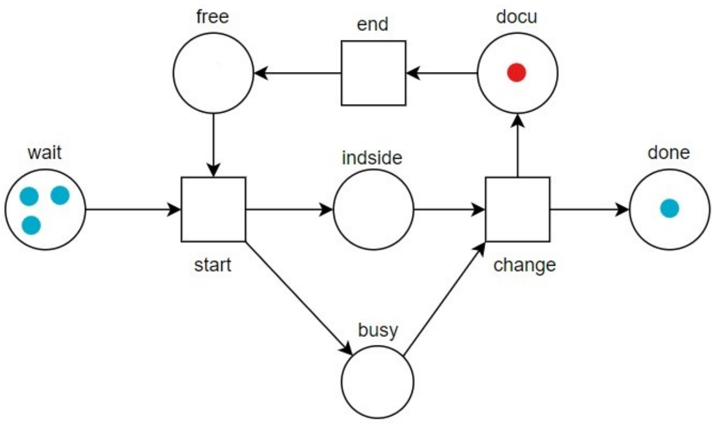
Determine the superimposed (merged) Petri net model $N = N_s \oplus N_{Pa}$ flowing a specialist treating patients, assuming there are four patients are waiting to see the specialist/ doctor, one patient is in state done, and the doctor is in state free. [1 point]
Answer: The approach we propose for constructing the Petri Net model of the superimposed (merged) According to the model, the red token is symbolized for the specialist/doctor, the blue token is symbolized for the patient



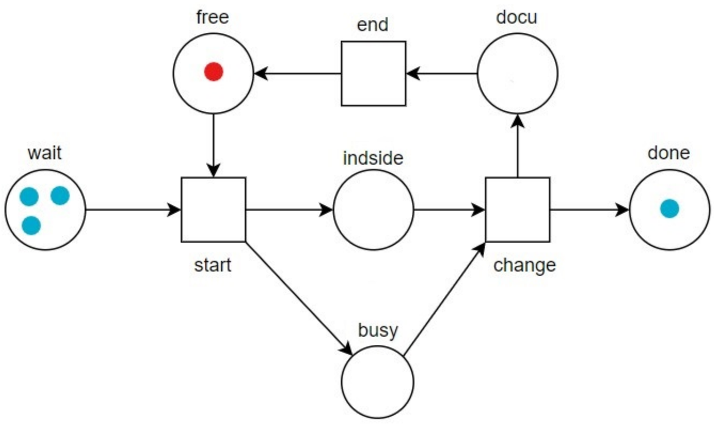
According to the model, the red token is symbolized for the specialist/doctor, the blue token is symbolized for the patient
There are 4 patient tokens in the "wait" place and 1 specialist token in the "free" place.
Transition "start" only fire if there is at least 1 token in "free" and 1 token in "wait"
The initial marking M_0={4.wait,1.free,0.busy,0.inside,0.docu,0.done}={4,1,0,0,0,0}



After firing, the 1 patient token in "wait" is moved to "inside", while specialist token is moved to "busy". Transition "change" only fire if there is at least 1 specialist token in "busy" and "inside".



After firing transition “change” , the 1 patient token in “busy” is moved to “done”, while specialist token is moved to “docu”.

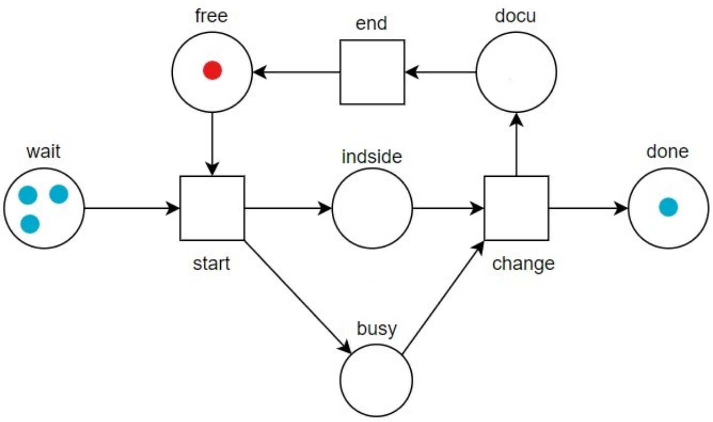


After firing, the Specialist token return to “free”. And the cycle repeats for the remaining patient tokens

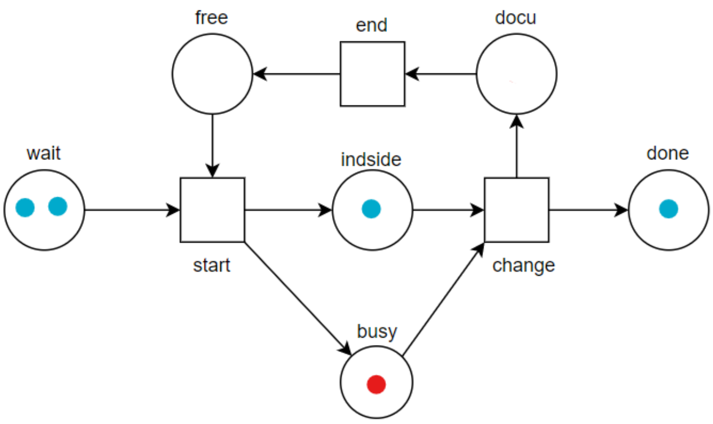
6.4 Problem 4

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]

The according initial Marking of our grand net N is M_0 ={3.wait, 1.free , 0.busy, 0.inside, 0.docu, 1.done} = {3,1,0,0,0,1}

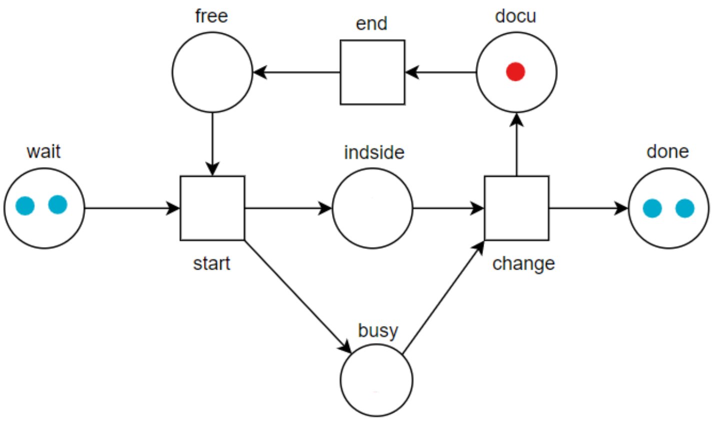


Step 1: Fire transition start
The first reachable marking from M_0 by firing one transition once is: M_1 = {2,0,1,1,0,1}

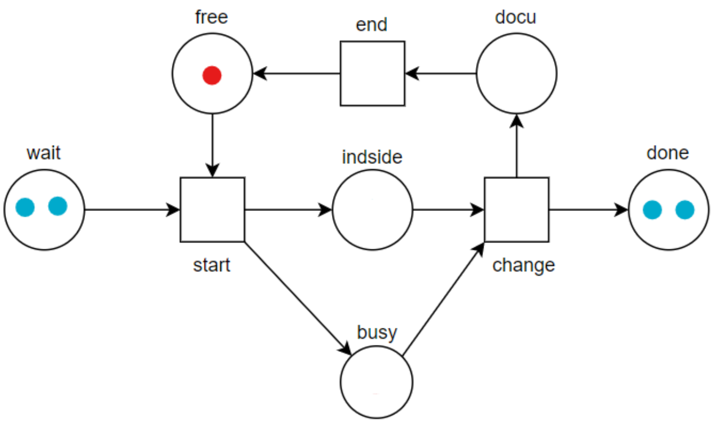


Because initially, transition start is the only enabled transition so only it can fire. So one token from each of start's input is consumed to produce one token in the next place: one token from wait goes to inside, one token from free goes to occupied. The remain token stays the same

Step 2: Fire transition change
After one token goes to busy, the transition change now enabled so it can fire. The token in busy goes to docu: $M_2 = \{2,0,0,0,1,2\}$



Step 3: Fire transition change
Now both input of transition end have token, it can fire, the token from inside goes to done, the token in docu goes to free: $M_3 = \{2,1,0,0,0,2\}$



This loop is finished and Petri Net can begin the new loop.

6.5 Problem 5

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

A marked Petri net (N, M_0) deadlock free if at every reachable marking at least one transition is enabled. Here,

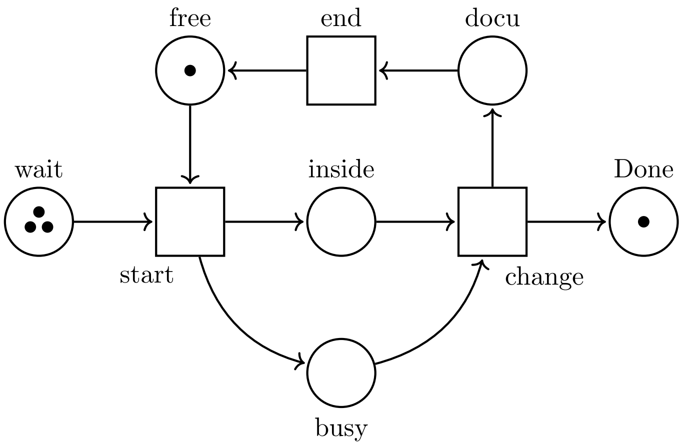


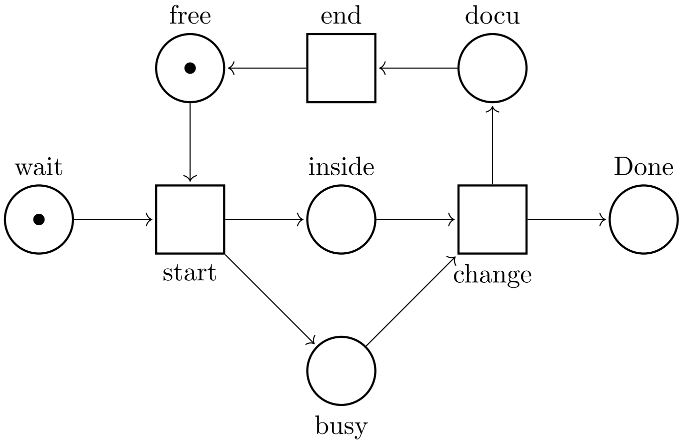
Figure 27: The Petri net of the SPECIALIST'S STATE

The superimposed Petri net N is NOT deadlock free, because at marking [1.free,4.done] there is no transition enabled.

6.6 Problem 6

Propose a similar **Petri net** with **two** specialists already for treating patients, with explicitly explained construction. [1 point].

In the previous exercise, we constructed the simple Petri net model of 1 specialist and 1 patients



There is no problem in this model with 1 special list and 1 patients, but in case of there are more than 1 specialists and patients, there are some problems with it:

- The first problem is that we can not distinguish between specialist tokens and patients tokens (all of them is drawn by a black dot). However, in practical problem, each specialist has their own ability, they takes responsibility for particular health problems. So which specialist enter to help patient is not arbitrary.
- The next problem is the order of patients entering. The patients entering the room has to follow the order.

Hence, we need to have a new tool to solve these problems. And it is called **Colored Petri net** (*CPN*). CPN is an extension of mathematical model Petri net. Instead of using black dot to represent a token, each token in CPN has attached data (token color).

Although token color is an **abstraction concept**, but places in **CPN only contain 1 type of token color**. To simplify, we build the model with 2 specialists and 4 patients. We have 2 specialists, so that at marking M0, in places free, there is 2 color tokens.
This place shows that each token attach an number data called NO and currently there are 2 tokens in place **free** with data attached is 1 and 2. The same for 4 patients at marking M0.
There are 4 patients waiting. All of tokens have type **NOxDATA**, where **NO** is the order of patient, the first patient go will have number 1 and the next is 2. . . .. **DATA** is represent the information of patients, that can be name of problems they need to treat.

We now consider a slightly **more complex CPN**. It is based on the Petri net model which was investigated in the previous sections, but now overtaking the **order of patients entering.** There is a new places called **reception desk**, it places at the beginning have a token with value 1. At marking M0, only transition start can enable. When transition start is firing, it takes

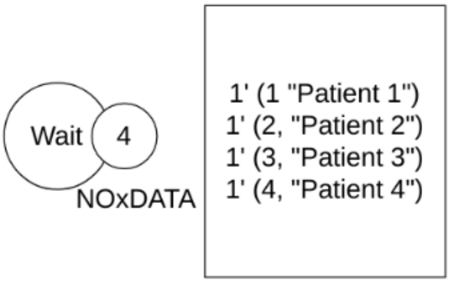


Figure 28: Places with 4 colored tokens

tokens from reception desk, and enable binding n = 1, token (1, "Patient 1") in place wait and token 1 in place free.

When the transition **start** occurs with an enabled binding, tokens are removed from the **Reception desk** according to the result of evaluating the arc expression, but they are immediately replaced by new tokens with the same token colours and the value of this token is added by 1. It means that after reception desk call number 1 to enter the room, the data of token color is added by 1, and expect to call the second patient in the next round.

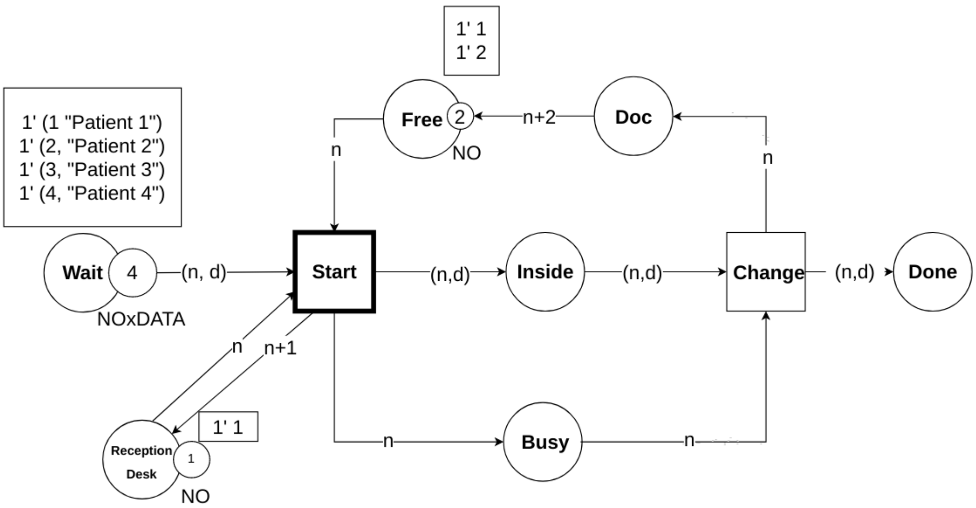


Figure 29: The Colored Petri net model

The other different in this model from the previous one is that after transition end is firing, it will return the token to place **free** with value n + 2. Number 2 is the number of specialist, after help patients the order of specialist is added by 2. It makes sure that 2 specialist is sequentially enter to help patients.

6.7 Problem 7:

Write a computational package to realize (implement) Items 1,2,3 and 4 above. The programs of your package should allow input with max 10 patients in place wait [**2 points**]

In this problem, our team employed **C++ programming language** to write computational programs on implementation of those problems above. Besides, **RAD Studio** was also applied on making the visualization and the more user-friendly interface.

♣ **Strategy:**

The plan is to create a class *Place* represents places in Petri Net which have a number of tokens and name (and it has its own format *pName* that we could distinguish from *tName* mean the name of transition) of that place, class *Transition* has its own name and two vector which contain all place input to that transition and all place that transition can firing.

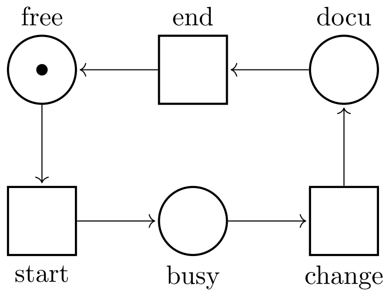
Class *Net* represents a whole Petri Net which contain a number of place and transition, also two vector of all transition and place in the net. To add place we call *addPlace* Function to add place to our vector, similarly to *Transition*, we add arc by call method *addArc*) that will add place into class transition (we can understand that each place we add into each transition is an arc).

By adding some functions to the Class Net, the program now can realize the tasks done on the models in each problem. Nonetheless, the visualization is represented by RAD Studio which brings a more sensitive and astonishing experiment.

♣ **Demo:**

Figure 30: Main page.

Problem 1: The Petri net of the specialist's state



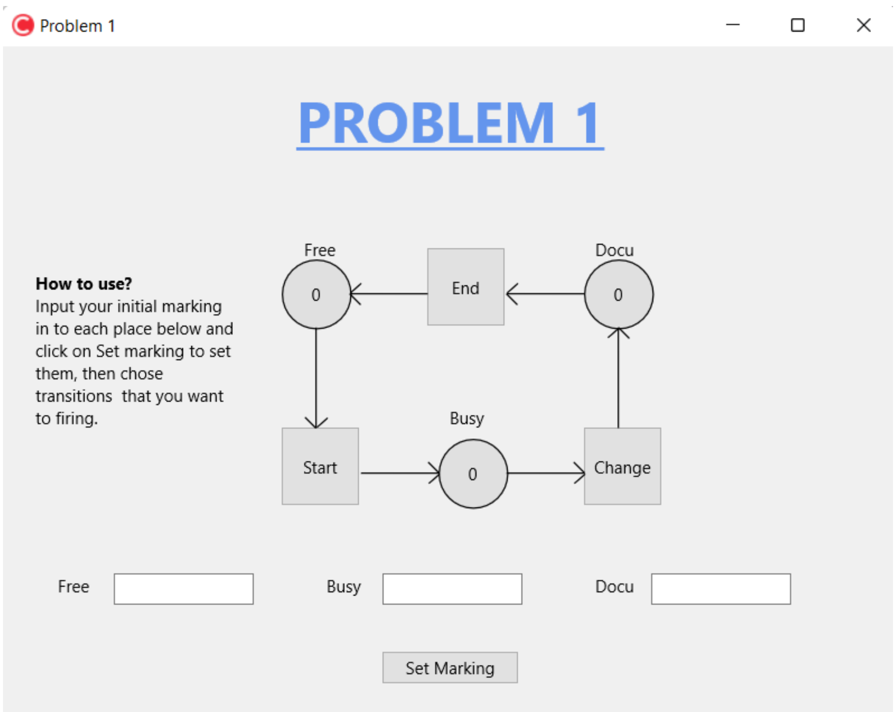
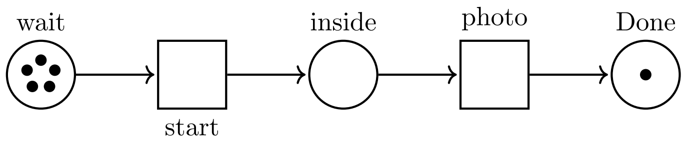


Figure 31: Problem 1.

Problem 2: The Petri net of the patient's state



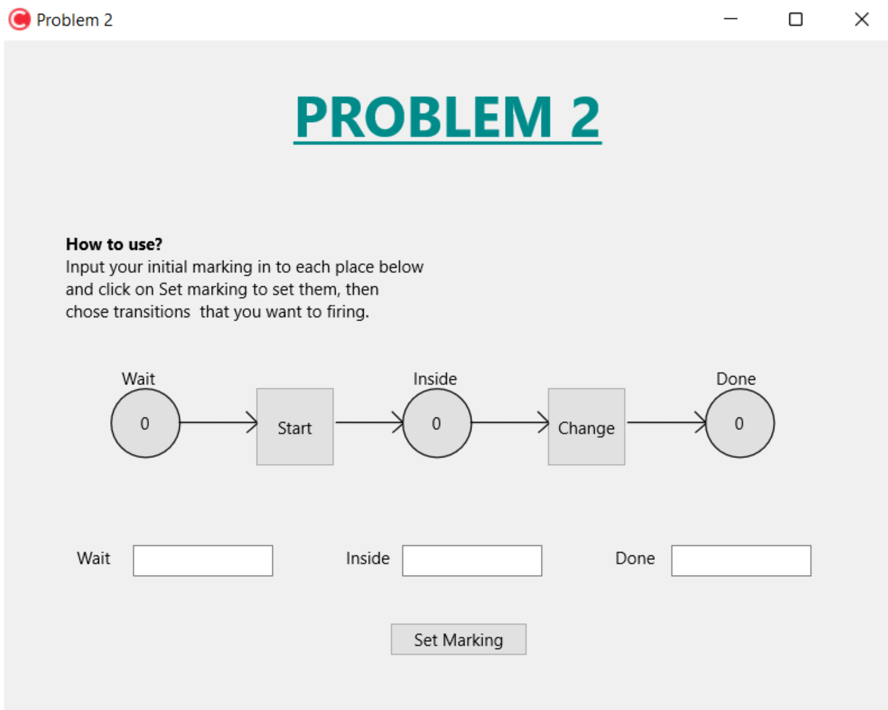
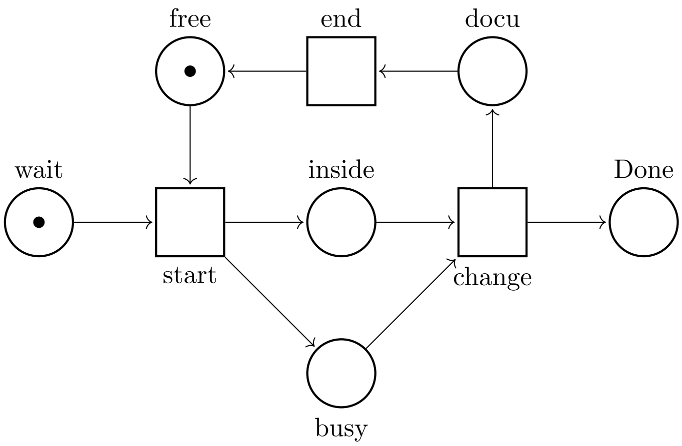


Figure 32: Problem 2.

Problem 3: The superimposed (merged) Petri net of the Patient-Specialist's state



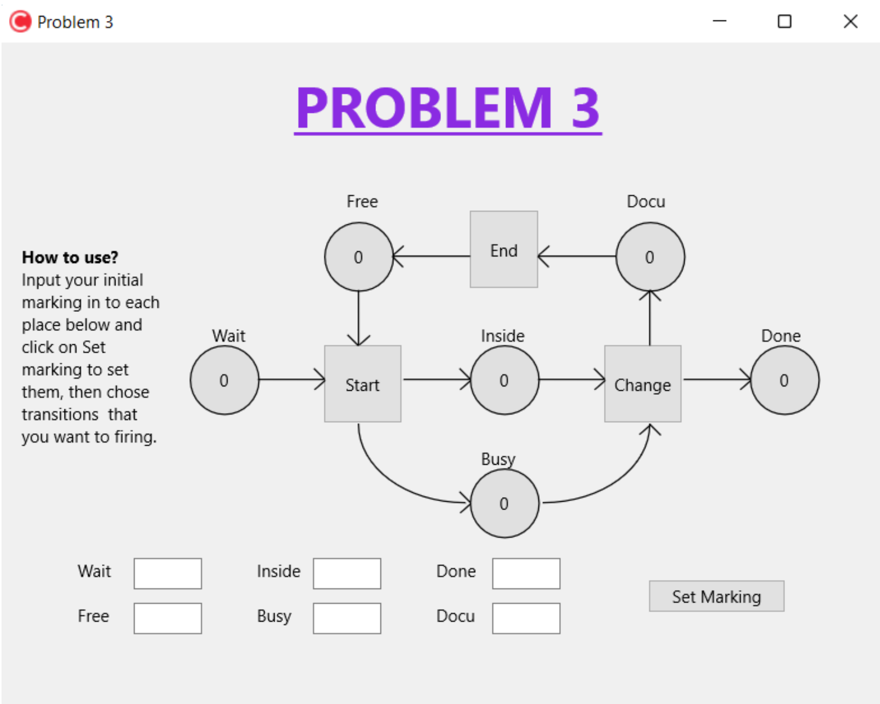
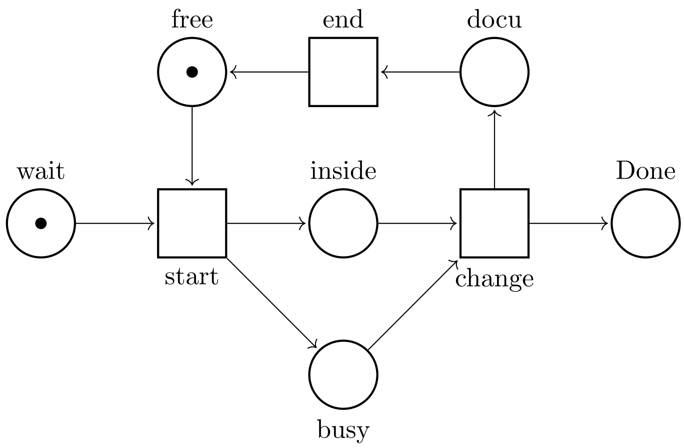


Figure 33: Problem 3.

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



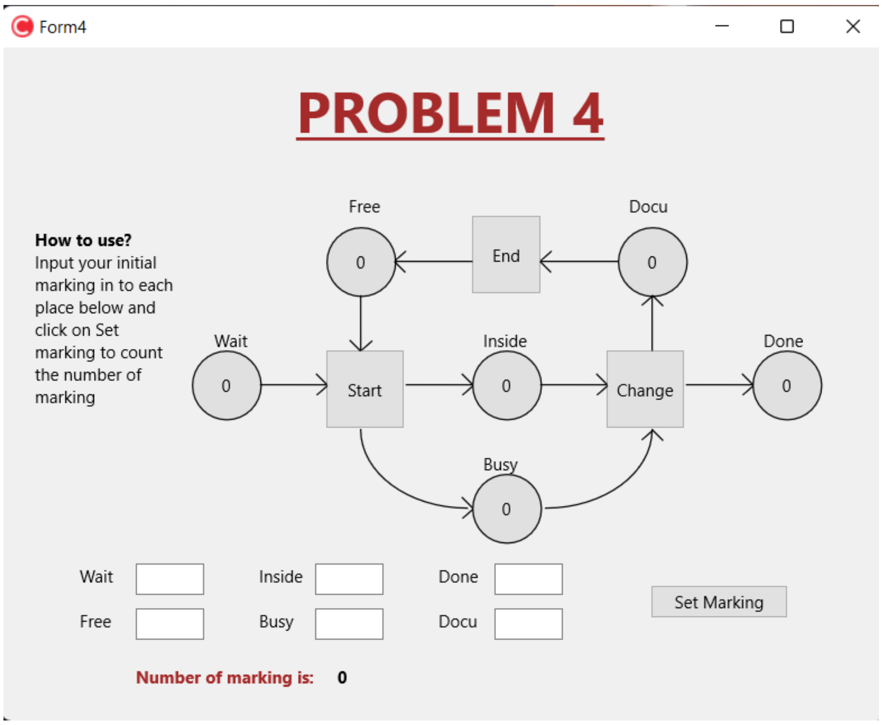


Figure 34: Problem 4.

References

[1] Wil van de Aalst and Christian Stahl, "Modeling business processes, a Petri net-oriented approach ", The MIT Press, Cambridge Massachusetts, (2011).

[2] Kurt Jensen · Lars M. Kristensen, "Coloured Petri Nets Modelling and Validation of Concurrent Systems", Springer-Verlag Berlin Heidelberg, (2009)

[3] A. Venkateswarlu ,"A Study of Petri Nets Modeling, Analysis and Simulation",West Bengal, INDIA, August 2006.

[4] Lei Bu, Petri Nets Presentations, Nanjing University (Uploaded on Jul 19, 2014). (https://www.slideserve.com/aldis/petri-nets)

[5] Westbrook Sterling, Petri Nets Presentations, Mar 12, 2019. (https://www.slideserve.com/westbrook-sterling/petri-nets-powerpoint-ppt-presentation)

[6] Man VM. Nguyen, Khuong An Nguyen and Thinh Tien Nguyen, MATHEMATICAL MODELING AND RISK ANALYSIS, 2021

[7] Kenneth H.Rosen, Discrete Mathematics and Its Applications, Eighth Edition