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

MATHEMATICAL MODELING (CO2011)

Assignment

Petri Networks

Advisor: Nguyen An Khuong
Nguyen Van Minh Man
Nguyen Tien Thinh
Students: Nguyen Duy Bao - 2052399
Tran Cong Huy Hoang - 2052482
Hoang Dinh Huy - 2053034
Nguyen Tran Minh Phuoc - 2052662
Nguyen Hoang Anh Thu - 2053478.

HO CHI MINH CITY, NOVEMBER 2021

Contents

1 Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Nguyen Duy Bao	2052399	- Reviewed Problems: Problem 5.2 Assignment: Problem 3, 6.	20%
2	Tran Cong Huy Hoang	2052482	- Reviewed Problems: Problem 5.3 Assignment: Problem 5, 7.	20%
3	Hoang Dinh Huy	2053034	- Reviewed Problems: Problem 5.1 Assignment: Problem 6, 7.	20%
4	Nguyen Tran Minh Phuoc	2052662	- Reviewed Problems: Problem 5.2 Assignment: Problem 2. Code Latex.	20%
5	Nguyen Hoang Anh Thu	2053478	- Reviewed Problems: Problem 5.3 Assignment: Problem 1. Code Latex.	20%

2 Preliminary knowledge

2.1 Background of Petri Net

2.1.1 Definition

Petri Net is a bipartite directed graph N of places and transitions.
A Petri net is a triplet $N = (P, T, F)$ where:

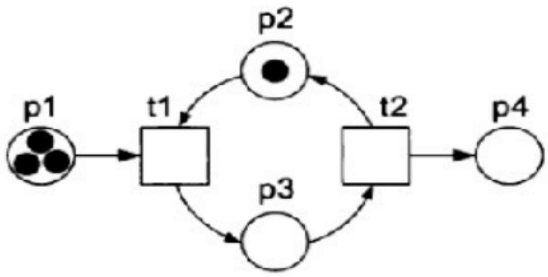
- P is a finite set of **places**. Graphically, a place is represented by a circle or an ellipse. A Place has discrete states, it can store, accumulate or show things (tokens).
- T is a finite set of **transitions** such that $P \cap T = \varnothing$. Graphically, a transition is represented by a square or a rectangle. A transition can produce things (tokens), consume, transport or change them.
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, called **flow relations**.

2.1.2 Concepts about Petri net

1. **Token:** A token is a special transition node, being graphically rendered as a black dot, $\bullet$. The symbolic tokens generally denote elements of the real world. Places can contain tokens, and transitions cannot.
2. **Enable transition:** A transition is enabled if each of its input places contains a token. An enabled transition can fire, thereby consuming one token from each input place and producing at least one token for each output place next and cause the **Marking change**.
3. **Marking:** A marking is a distribution of tokens across places. A marking of net N is a function $m : P \rightarrow \mathbb{N}$, assigning to each place $p \in P$ the number m(p) of tokens at this place. Denoted M, viewed as a multi-set.
4. **Marked Petri net:** A marked Petri net is a pair (N,M), where $N = (P, T, F)$ is a Petri net and where M is a multiset over P denoting the marking of the net.
5. **Operations:** the sum of two multi-sets ($A \uplus B$), the difference($A \setminus B$), the presence of an element in a multi-set ($x \in M$), and the notion of subset ($X \leq Y$)are defined in the classic way of set theory.
6. **Node:** 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 ($(x, y) \in F$). Node x is an output node of node y if and only if $(y, x) \in F$.
   - For any node $x \in P \cup T$, write $\bullet$x = {y|(y,x) $\in$ F} - the preset of x, and x $\bullet$= {y|(x,y) $\in$ F} - the postset of x.
7. **Arcs:** Places and transitions are connected to each other by directed arcs. Graphically, arcs are represented by directed arrows. The set of arcs is a binary relation, as the Petri Net has 2 kinds of arcs, we have 2 binary relations:
   - The binary relation $R_I \subseteq P \times T$ contains all arcs connecting input places and their transitions.
   - The binary relation $R_O \subseteq T \times P$ contains all arcs connecting transitions and their output places.

The union $R_I \cup R_O$ represents all arcs of a Petri net, which is called the **flow relation**, where (p,t) ∈ F define the arc from a place p to a transition t.

Practice 5.1.
Consider the Petri net in figure below:



1. Define the net formally as a triple (P,T,F).
   Consider the Petri net $N = (P, T, F)$ where:
   - P = {p1, p2, p3, p4}
   - T = {t1, t2}
   - F = {(p1,t1)∪(t1,p3), (p2,t1)∪(t1,p3), (p3,t2)∪(t2,p4), (p3,t2)∪(t2,p2)}
2. List presets and postsets for each transition.
   - •t1 = {p1,p2}
   - t1•= {p3}
   - •t2 = {p3}
   - t2•= {p2,p4}
3. Determine the marking of this net.
   Marking M = $\{p1^3, p2\}$
4. Are the transitions t1 and t2 enabled in this net?
   - t1 is enable
   - t2 is not enable

Practical problem 1.
Given a process of a X-ray machine in which we assume the first marking in Figure 5.8.a shows that there are three patients in the queue waiting for an X-ray. Figure 5.8.b depicts the next marking, which occurs after the firing of transition enter.

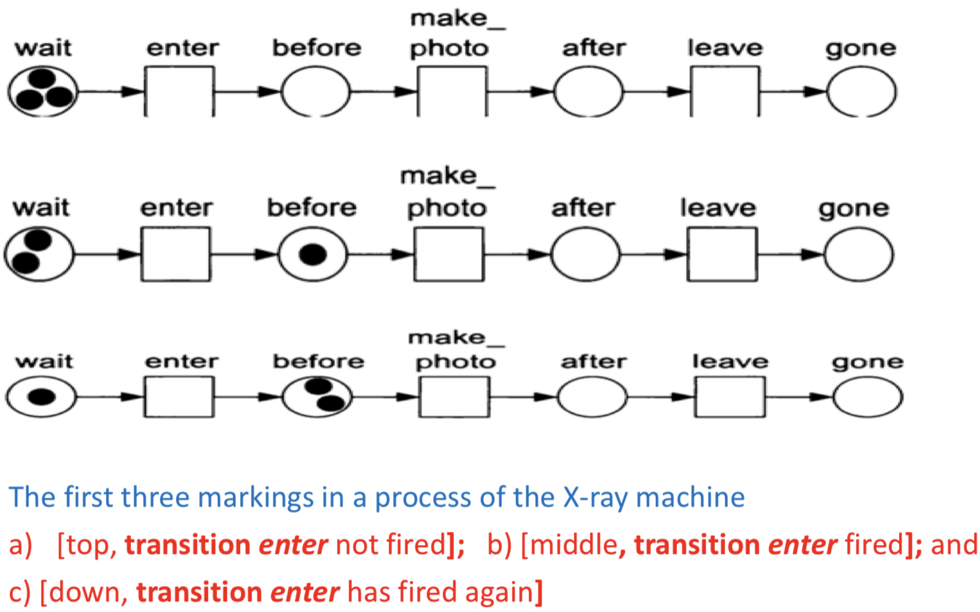
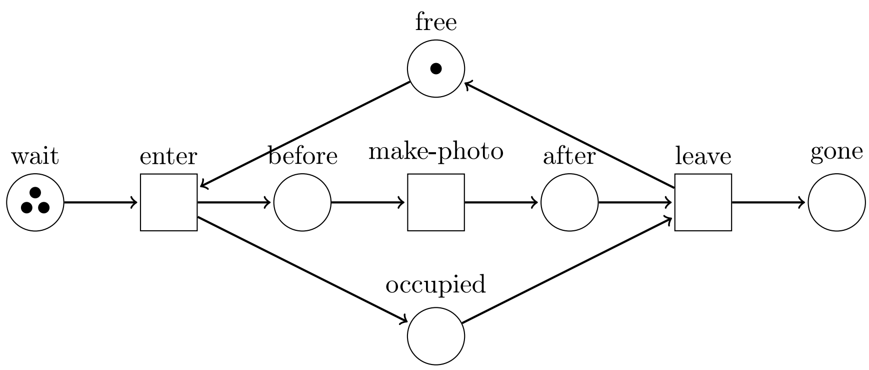


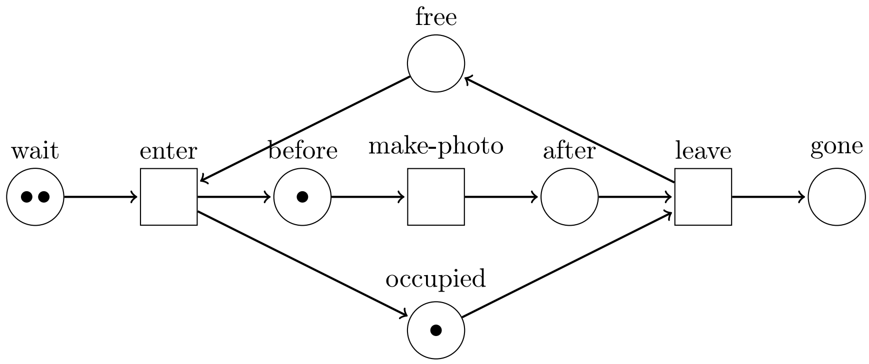
Figure 5.8: *A Petri net model of a business process of an X-ray machine*

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



An improved Petri net for the business process of an X-ray machine

There are two possible states: the room can be **free** or **occupied**. We model this by adding theses two places to the model, to get the improved the Petri net, in Figure 5.9. Now for this Petri net, can place **before** contain more than one token? Why? Rebuild the set P of place labels.



The marking of the Net after transition **enter** has fired

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

Answer:

1. $R_I = \{$(wait $\times$ enter), (before $\times$ make-photo), (after $\times$ leave)$\}$
   $R_O = \{$(enter $\times$ before), (make-photo $\times$ after), (leave $\times$ gone)$\}$
   $F = R_I \cup R_O = \{$(wait,free $\times$ enter) $\cup$ (enter $\times$ before,occupied), (before $\times$ make-photo) $\cup$ (make-photo $\times$ after), (after,occupied $\times$ leave) $\cup$ (leave $\times$ free, gone)$\}$

2. The place **before** cannot have more than one token, because assuming there is one token in **before**, when one token from **wait** to **before** is produced, a token from **before** to **after** is also produced (which is conflicted with hypothesis).

3. **Definition 5.2** 3: An **enabled transition** can fire, thereby consuming (energy of) one token from each input place.
   Because the place **free** is empty, transition **enter** cannot fire.
   $R_I = \{(wait, enter), (before, make{-}photo), (after, leave), (occupied, leave), (free, enter)\}$
   $R_O = \{(enter,\ before),\ (make-photo,\ after),\ (leave,\ gone),\ (enter,\ occupied)\}$

2.2 The behavior of Petri Nets

2.2.1 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 $\bullet\, t \leq M$.
- The firing rule $\alpha\ \ [t\rangle\ \ \beta \subseteq \mathcal{N} \times T \times \mathcal{N}$ is the smallest relation satisfying

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

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

1. An enabled transition t can fire, thereby changing the marking M to a marking $M_1 = (M \setminus \bullet t) \uplus t\bullet$.
   If an enabled transition fires, then it consumes one token from each of its input places and produces one token in each of its output places.

2. $(N, M)\ [t\rangle$ means that transition t is enabled at marking M.

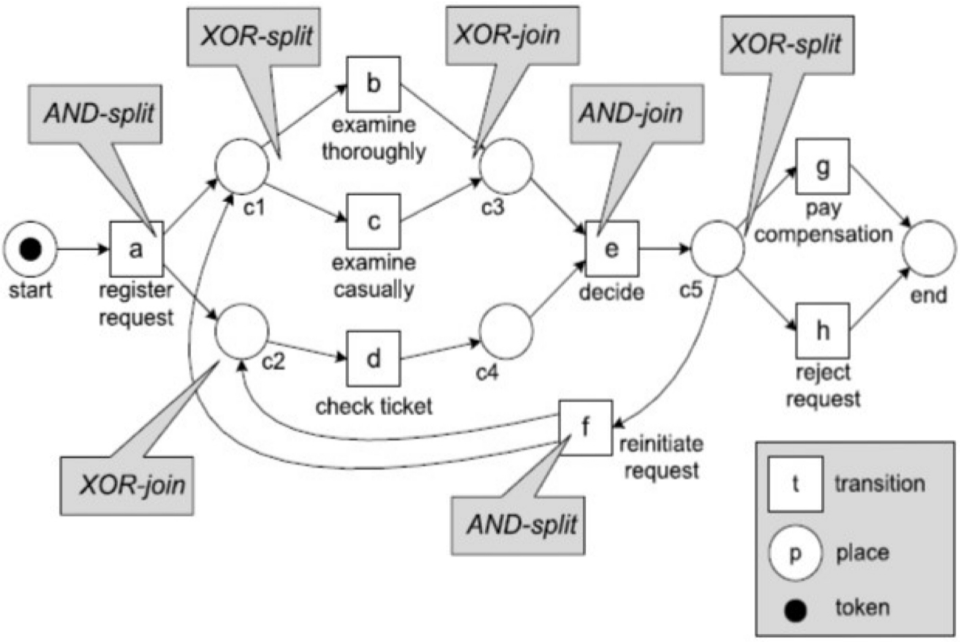


Figure 5.11: *A marked Petri net with one initial token*

For example, in figure 5.11, $(N, [start])[a\rangle$ means the transition a is enabled at marking [start].

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

2.2.2 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 $M_1, M_2, ..., M_n$ and transitions $T_1, T_2, ..., T_n$ such that:
   - $\sigma = (t_1, t_2, ..., t_n) \in T^*$
   - and for all i with $0 \le i \le n$, 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 marking of (N, M_0) is denoted $[N, M_0\rangle$.

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

Example 5.6.
In Fig. 5.11, write marking $M_0 = [start] = [1, 0, 0, 0, 0, 0, 0]$, we get the marked Petri net (N, M_0) and see that

- The empty sequence $\sigma_0 = \langle\rangle$ - being enabled in (N, M_0) - is a firing sequence of (N, M_0).
- The sequence $\sigma_1 = \langle a, b\rangle$ is also enabled in (N, M_0), and firing σ_1 results in marking $[c2, c3]$. We can write $(N, [start])[a, b\rangle(N, [c2, c3])$ or $(N, M_0)[\sigma_1\rangle(N, [c2, c3])$.

2.2.3 Labeled Petri net

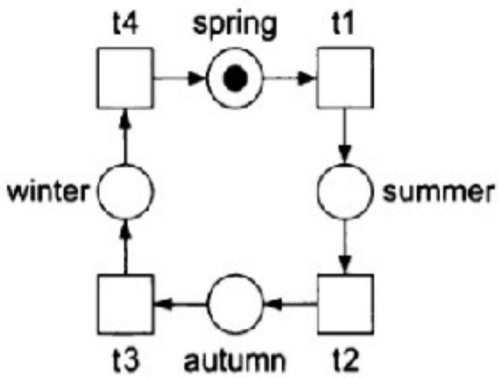
Often transitions are identified by a single letter, but also have a longer label describing the corresponding activity.

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

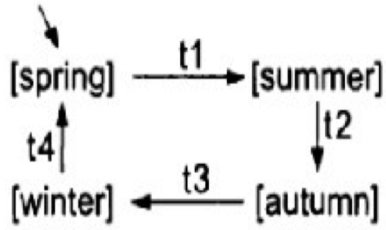
Question 5.1.
On markings in nets, when we have modeled a system as a **Petri net** system (N, M_0) then some matter occur, including

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.
Example 5.7
Consider the **Petri net** system in figure below modeling the four seasons.



(a) The four seasons



(b) Reachability graph

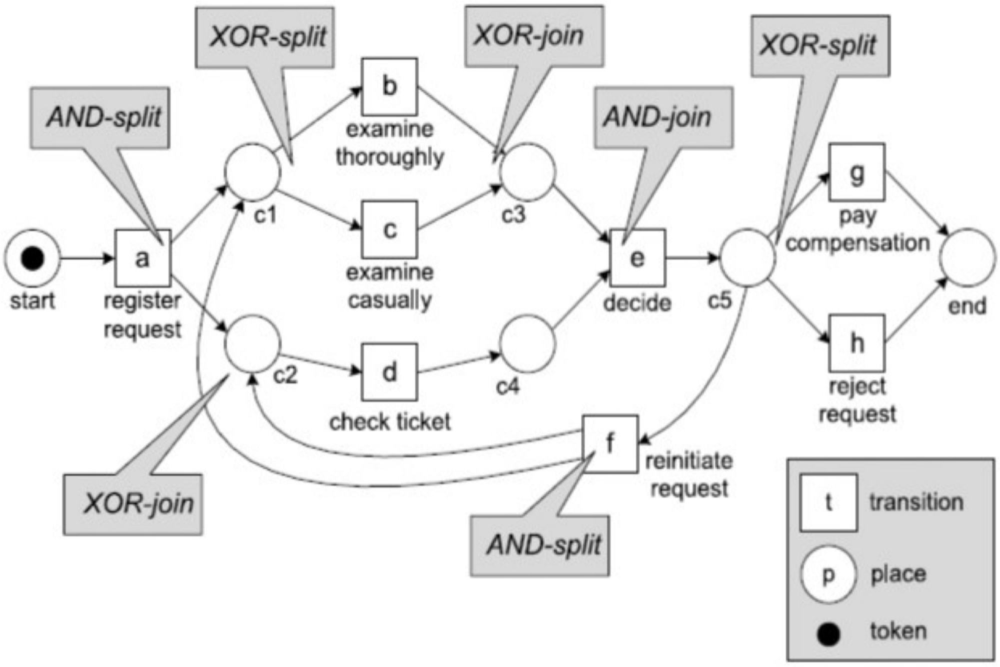
Recall that, each of the reachable markings is represented as a *multiset* (where the same element may appear multiple times). Multiset [`spring`] thus represents the marking in figure (a).

2.2.4 Reachability graph

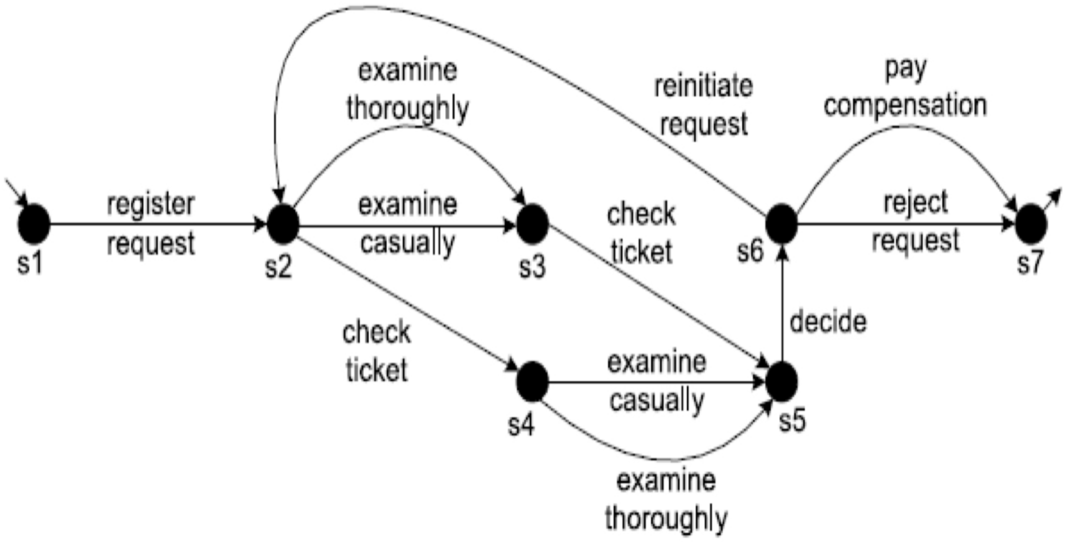
Let (N, M_0) with $N = (P, T, F, A, l)$ be a marked labeled Petri net.
(N, M_0) defines a transition system $TS = (S, A_1, TR)$ with
$S = [N, M_0\rangle$, $S^{start} = M_0$, $A_1 = A$, and
$TR = \{(M, M_1) \in S \times S \mid \exists t \in T \quad (N, M)[t\rangle \ (N, M_1)\}$, or with label $l(t)$:

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

TS is often referred to as the **reachability graph** of (N, M_0).
Practice 5.2.



Build up the transition system TS generated from the labeled marked Petri net shown above.
Answer:
States of TS correspond to reachable markings, i.e., multi-sets of tokens.
Note that $S^{start} = \{[start]\}$ is a singleton containing the initial marking of the Petri net. The Petri net does not explicitly define a set of final markings S^{end}.
However, in this case it is obvious to take $S^{end} = \{[end]\}$.
The outcome should be as follows



2.2.5 Causality, Concurrency and Synchronization

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

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

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

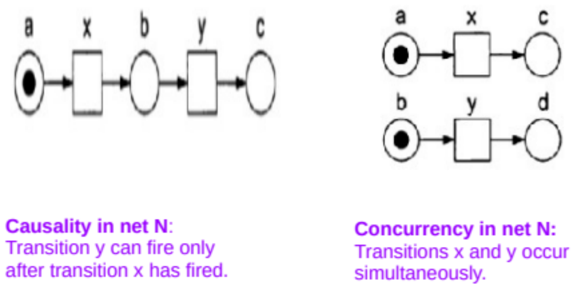


Figure 5.15: *Causality and Concurrency in a* **Petri net**

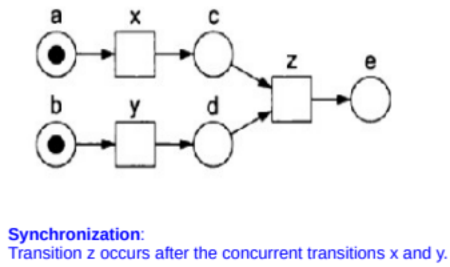
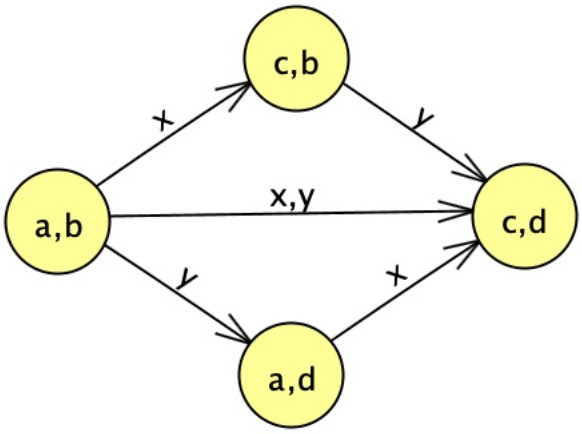


Figure 5.16: *Synchronization in a* **Petri net**

Question: Can we find the reachability graph of the net N in Figure 5.15 (right)?
Answer: Yes, we can. Here is the reachability graph of the net N:



Question 5.2. *Could we quantify the concurrency for a given process or* **Petri net**?
The answer is yes, and the tool is transition system. Remind that, a transition system is formally a triplet $TS = (S, A, T)$ where S is the set of *states*, $A \subseteq \mathcal{A}$ is the set of *activities* (often referred to as *actions*), and $T \subseteq S \times A \times S$ is the set of *transitions*.
For more information: 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.

3 Reviewed Problems

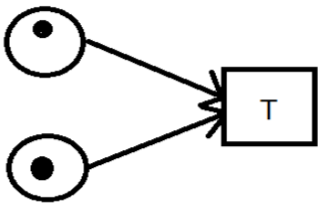
3.1 Problem 5.1

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

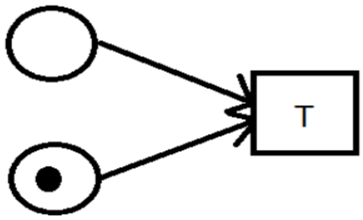
1. "enabled transition”
2. "firing of a transition"
3. "reachable marking"
4. "terminal marking"
5. "non-deterministic choice"

Answer:

1. “enable transitions”: A transition is enabled if each of its input places contains at least one token.

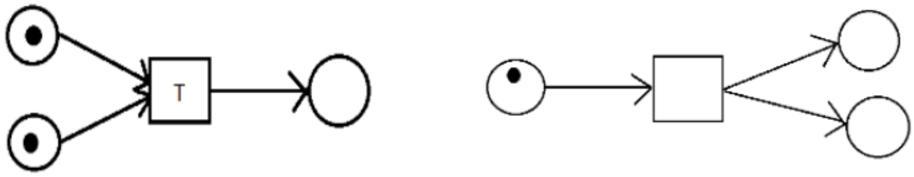


Transition T is enabled

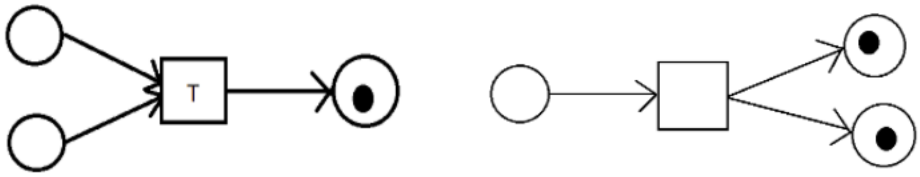


Transition T is not enabled.

2. “firing of transitions”: a transition of a Petri net may fire if it is enabled; when the transition fires, it consumes the required input tokens, and creates tokens in its output places. A firing is atomic.



Before firing



After firing

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

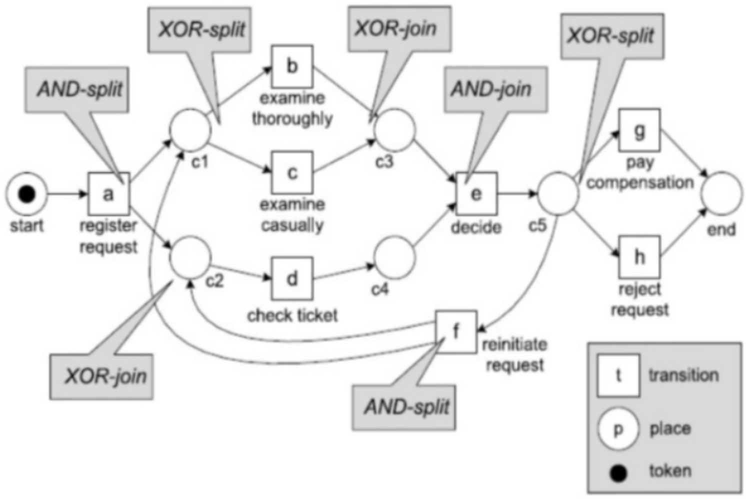
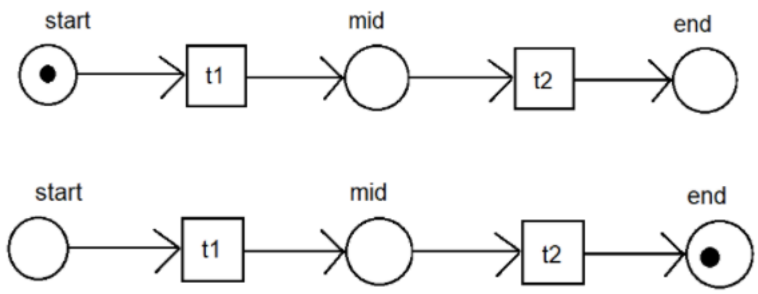


Figure 5.11: *A marked Petri net with one initial token*

There are 7 reachable markings:

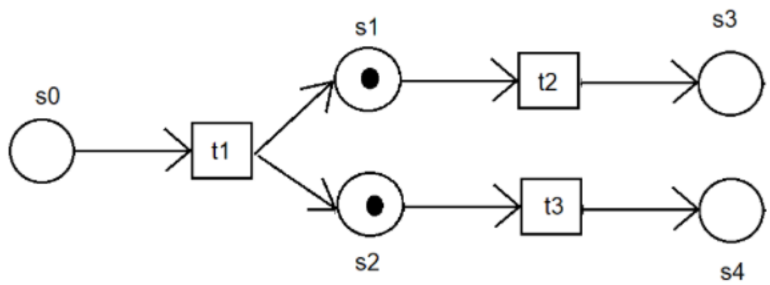
- $M_0 = \{start\}$
- $M_= \{c1, c2\}$
- $M_= \{c1, c4\}$
- $M_= \{c2, c3\}$
- $M_= \{c3, c4\}$
- $M_= \{c5\}$
- $M_= \{c1, c2, end\}$

4. "terminal marking": the transitions keep firing until the net reaches a marking that does not enable any transition, this marking is a terminal marking.



M_end = [end] is terminal marking.

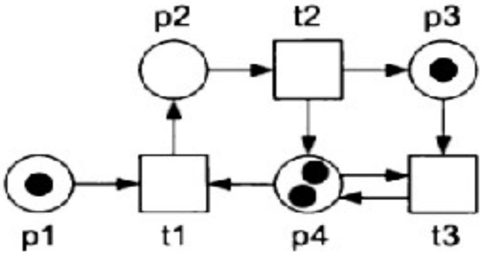
5. "nondeterministic choice": when several transitions are enabled at the same moment, it is not determined which of them will fire. This situation is a nondeterministic choice.



Transitions t2 and t3 are enable at the same time
=> nondeterministic choice

3.2 Problem 5.2

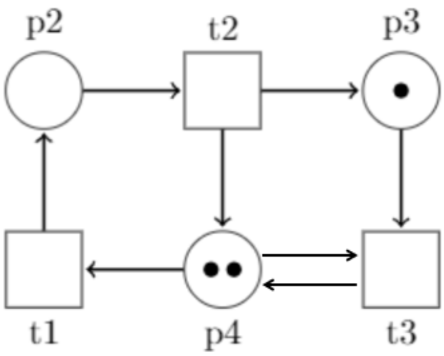
Consider the Petri net system in figure below



1. Formalize this net as a quadruplet (P, T, F, M_0)
2. Give the preset and the postset of each transition.
3. Which transitions are enabled at M_0?
4. Give all reachable markings. What are the reachable terminal markings?
5. Is there a reachable marking in which we have a nondeterministic choice?
6. Does the number of reachable markings increase or decrease if we remove
   (a) place p1 and its adjacent arcs and
   (b) place p3 and its adjacent arcs?

Answer:

1. $(P, T, F, M_0) = (\{p1, p2, p3, p4\}, \{t1, t2, t3\}, \{(p1, p4 \times t1) \cup (t1 \times p2), (p2 \times t2) \cup (t2 \times p3) \cup (p3, p4 \times t3) \cup (t3 \times p4)\}, [p1, p3, p4^2])$
2. The preset and postset of each transitions:
   - $\bullet t1 = \{p1, p4\}$
   - $t1\bullet = \{p2\}$
   - $\bullet t2 = \{p2\}$
   - $t2\bullet = \{p3, p4\}$
   - $\bullet t3 = \{p3, p4\}$
   - $t3\bullet = \{p4\}$
3. Transitions are enabled at M_0: $t1$, $t3$.
4. $[N, M_0\rangle = \{[p1, p3, p4^2], [p2, p3, p4], [p3^2, p4^2], [p3, p4^2], [p4^2], [p2, p4], [p1, p4^2]\}$
   Reachable terminal markings are: $[p4^2]$
5. Yes, there are: $[p1, p3, p4^2]$, $[p2, p3, p4]$.
6. (a) Increase
   Justification:
   After removing place p_1 and its arc adjacent, the Petri net will become:

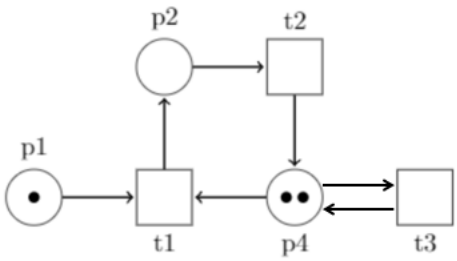


At marking $M_0 = \{p4^2, p3\}$, we have 2 enabled transitions t1,t3. If we fire t1, the marking will be $M_1 = \{p4, p2, p3\}$ and $(N, M_1)[t2\rangle$. Then if we fire t2, the marking will be $M_2 = \{p4^2, p3^2\}$ and $(N, M_2)[t1\rangle$. Notice that after the sequence $\sigma =< t1, t2 >$, the number of tokens of p4 remains the same, p3 has 1 more token and t1 can be fire again, which mean we can follow the sequence $\sigma =< t1, t2 >$ again. And after each sequence σ, we will have 2 more reachable marking and we still can repeat the sequence. Therefore we will have infinite markings according to this net, so the number of marking will increase.

(b) Decrease

Justification:

After removing place $p3$ and its arcs adjacents, the Petri net will become:



The number of reacheable marking is 3:

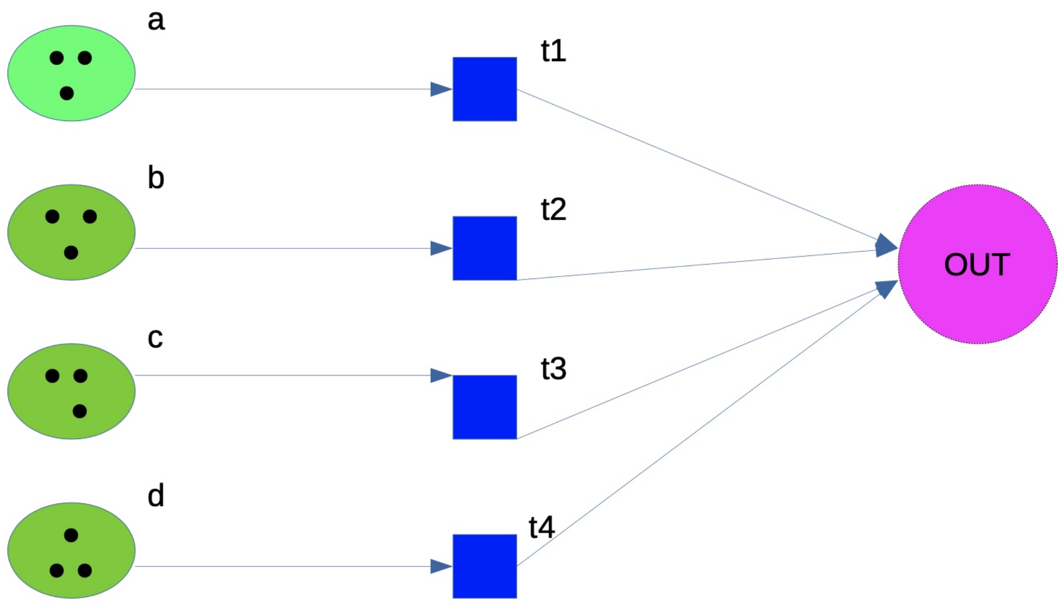
- $M_0 = p1, p4^2$ $(N, M_0)[t1, t3\rangle$, but if we fire t3, the marking will remain the same, so we fire t1 and have M_1
- $M_1 = p2, p4$ $(N, M_1)[t2, t3\rangle >$, fire t2 and have M_2
- $M_2 = p4^2$ $(N, M_1)[t3\rangle >$, but fire t3 will not change the marking.

The number of marking of the complete Petri Net at (1) is 7, so the number of reachable markings decreased.

3.3 Problem 5.3.

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

$\mid T \mid = 4$, and the initial marking is M_0, as seen in figure below.

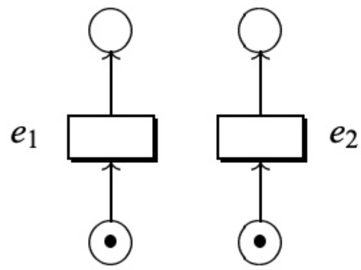


Denote TS for the whole transition system made by the Petri net $N = (P, T, F)$.

1. Write down M_0, P, T of N.
2. If **not allow concurrency** in this process (marked Petri net) then how many states of the transition system TS can be created? How many transitions are there?
   NOTE: The states of TS are in fact the *reachable markings* of the Petri net N.
   HINT:
   Extend the ideas of Hamming distance in the hypercube $H_n = (\{0, 1\}^n, E)$ to "quaternary cube" $K_n = (\{0, 1, 2, 3\}^n, E)$, and modify the concept of edge in K_n to capture the transitions in TS.

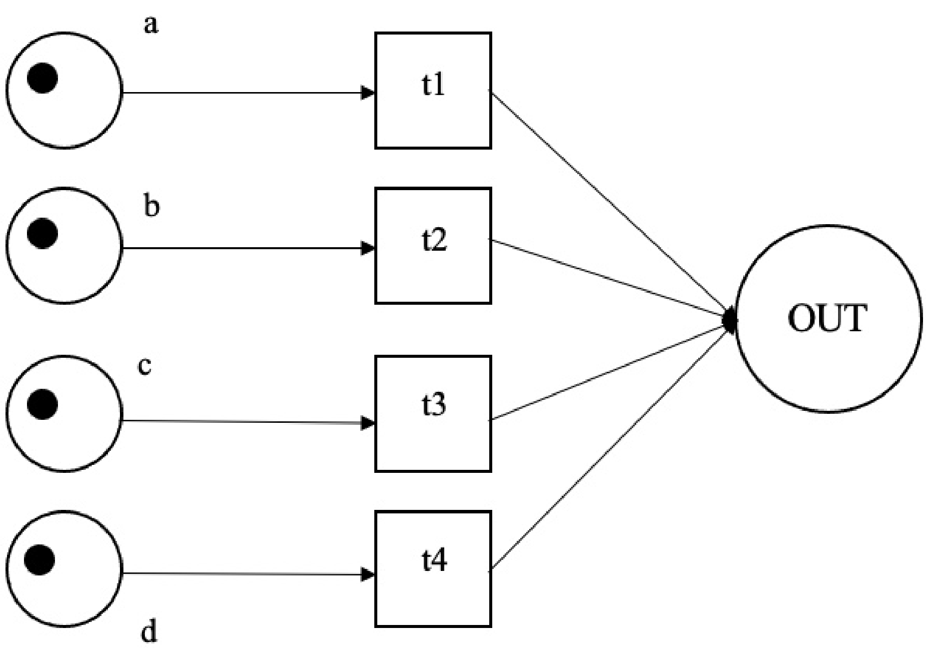
Answer:

1. $M_0 = \{3, 3, 3, 3, 0\}$
   $P = \{\text{a, b, c, d, OUT}\}$
   $T = \{t1, t2, t3, t4\}$
2. To do this question, we have to understand what is **not allow concurrency**.
   At the "Preliminary knowledge" section, **concurrency** is defined as ability that allows several events can occur simultaneously.



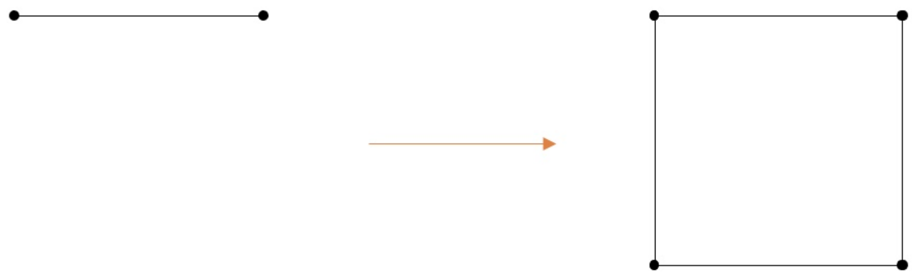
In the figure above, the events $e1$ and $e2$ can occur concurrently; they are both independent in not having any pre or post conditions.
Therefore, **not allow concurrency** is an ability that disallow events from occurring simultaneously.
Which means, in the figure above, the events $e1$ and $e2$ cannot occur concurrently; if $e1$ is enabled, $e2$ must be disabled, and vice versa.
We figure out how many states and transitions can be created from the transition system TS if not allow concurrency.

(a) *Predict the numbers of vertices and edges in the hypercube* $H_n = (\{0,1\}^n, E)$
Simplify the question as there are only one token in each state a, b, c, d:



This is the idea of Hamming distance in the hypercube $H_n = (\{0,1\}^n, E)$
It is helpful to think of cubes as generated by lower-dimensional cubes in motion. A point in motion generates a segment; a segment in motion generates a square; a square in motion generates a cube; and so on.

Each time we move a cube to generate a cube in the next higher dimension, the number of vertices doubles.



Define a sequence of numbers illustrates vertices of a hypercube (V_n) as:

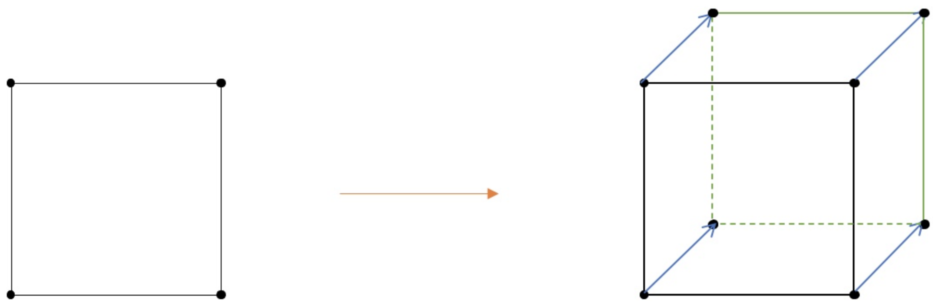
$$\begin{cases} V_1 = 2 \\ V_{n+1} = 2 \times V_n \end{cases}$$

Transform the sequence to general formula using recursion:

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

Therefore, the number of vertices is $V_n = 2^n$.

A square has 4 edges, and as it moves from one position to the other, each of its 4 vertices traces out an edge. Thus, we have 4 edges on the initial square, 4 on the final square, and 4 traced out by the moving vertices for a total of 12. That pattern repeats itself. If we move a figure in a straight line, the number of edges in the new figure is twice the original number of edges plus the number of moving vertices.



Define a sequence of numbers illustrates edges of a hypercube (E_n) 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:

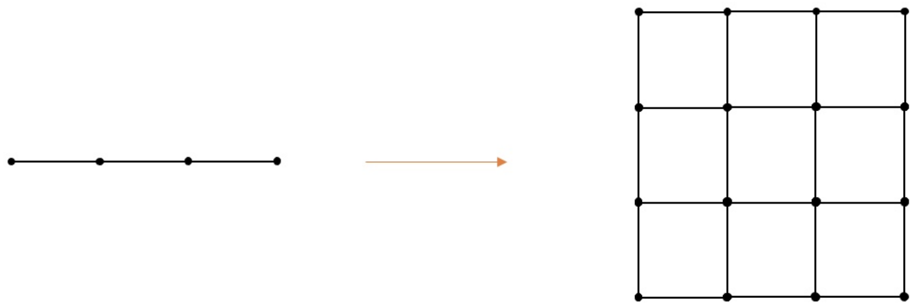
$$E_n = 2 \times E_{n-1} + 2_{n-1} = 2 \times (2 \times E_{n-2} + 2^{n-2}) + 2^{n-1} = ...$$

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

$$\implies E_n = n \times 2^{n-1}$$

Therefore, the number of edges is $E_n = n \times 2^{n-1}$.

(b) *Extend (a) to "quaternary cube"* $K_n = (\{0, 1, 2, 3\}^n, E)$
Each time we move a cube to generate a cube in the next higher dimension, the number of vertices is 4 times more than the original.



Define a sequence of numbers illustrates vertices of a "quaternary cube" (V_n) as:

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

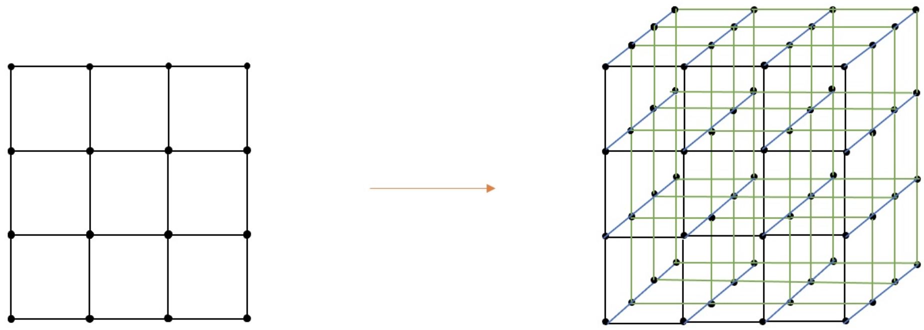
Transform the sequence to general formula using recursion:

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

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

$$\implies V_n = 4^n$$

Therefore, the number of vertices is $V_n = 4^n$.
As it moves from one position to the other, each of its vertices traces out an edge. Thus, we have 24 edges on the initial square, 24 on the 3 final squares, and 48 traced out by the moving vertices for a total of 144. That pattern repeats itself. If we move a figure in a straight line, the number of edges in the new figure is 4 times the original number of edges plus the number of moving vertices.



Define a sequence of numbers illustrates edges of a "quaternary cube" (E_n) as:

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

We already have $V_n = 4^n$ in the previous proving.
Transform the sequence to general formula using recursion:

$$E_n = 4 \times E_{n-1} + 3 \times 4_{n-1} = 4 \times (4 \times E_{n-2} + 3 \times 4^{n-2}) + 3 \times 4^{n-1} = ...$$

$$\Longrightarrow E_n = 4^{n-1} \times E_1 + 3 \times (n-1) \times 4^{n-1} = 4^{n-1} \times 3 + 3 \times (n-1) \times 4^{n-1}$$

$$\Longrightarrow E_n = 3 \times n \times 4^{n-1}$$

Therefore, the number of edges is $E_n = 3 \times n \times 4^{n-1}$.

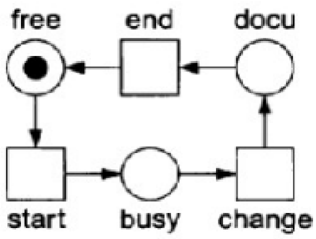
We conclude that:

- There are $4^4 = 256$ states.
- There are $3 \times 4 \times 4^3 = 768$ transitions.

4 Assignment

4.1 Problem 1

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



A Petri net modeling the state of the specialist.

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

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

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

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

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

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

Answer:

1a) $P = \{\text{free},\ \text{busy},\ \text{docu}\}$
$T = \{\text{start},\ \text{change},\ \text{end}\}$

1b)

(i) *Each place cannot contain more than one token in any marking.*
We create a transition system $TS = (S, TR, M)$ with:

- S is the set of states follows the condition.

$$S = \{(0,0,0),\ (1,0,0),\ (0,1,0),\ (0,0,1),\ (1,1,0),\ (1,0,1),\ (0,1,1),\ (1,1,1)\}$$

- According to the Petri net of specialist, the initial marking $M = (1,0,0)$
- TR is the set of transition relation.
  We have:
  * $(0,0,0)$ is not possible because in this case, all transitions are not enabled.
  * $(1,0,0) \longrightarrow (0,1,0) \longrightarrow (0,0,1) \longrightarrow (1,0,0) \longrightarrow ...$
  * $(1,1,0) \longrightarrow (1,0,1) \longrightarrow (0,1,1) \longrightarrow (1,1,0)...$
  * $(1,1,1)$ is not possible because of the condition.

$\Longrightarrow$ TR $= \{ \ ((1,0,0),\ (0,1,0)) \cup ((0,1,0),\ (0,0,1)) \cup ((0,0,1),\ (1,0,0)) \cup ((1,1,0),\ (1,0,1)) \cup ((1,0,1),\ (0,1,1)) \cup ((0,1,1),\ (1,1,0)) \ \}$

(ii) *Each place may contain any natural number of tokens in any marking.*
We create a transition system $TS = (S, TR, M)$ with:

- S is the set of states follows the condition. Assume $a + b + c = constant$ where a, b, c are in the set of natural numbers:

$$S = \{(a, b, c) \mid a, b, c \in \mathbb{N}\}$$

- According to the Petri net of specialist, the initial marking $M = (1, 0, 0)$
- TR is the set of transition relation.
TR $= \{ \ ((x+1, y, z),\ (x, y+1, z)) \cup ((x, y+1, z),\ (x, y, z+1)) \cup ((x, y, z+1),\ (x+1, y, z) | x, y, z \in \mathbb{N},\ x + y + z + 1 = a + b + c) \ \}$.

4.2 Problem 2

Define N_{Pa} as the Petri net modeling the state of patients. By the similar ideas,

a) explain the possible meaning of a token in state **inside** of the net N_{Pa} **[1 point]**

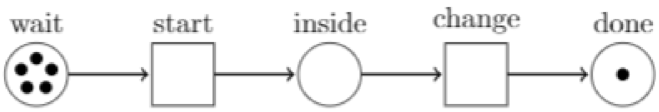
b) construct the Petri net N_{Pa}, assuming that there are five patients in state **wait**, no patient in state **inside**, and one patient is in state **done**. **[1 point]**

Answer:

a) Meaning: The patient is being treated by specialist

b) Consider Petri net $N_{Pa} = P, T, F, M_0$ where:

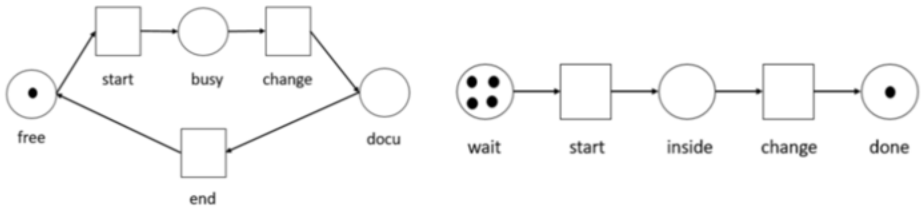
- $P = \{wait, inside, done\}$
- $T = \{start, change\}$
- $F = \{(wait \times start) \cup (start \times inside), (inside \times change) \cup (change \times done)\}$
- $M_0 = \{wait^5, done\}$



4.3 Problem 3

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

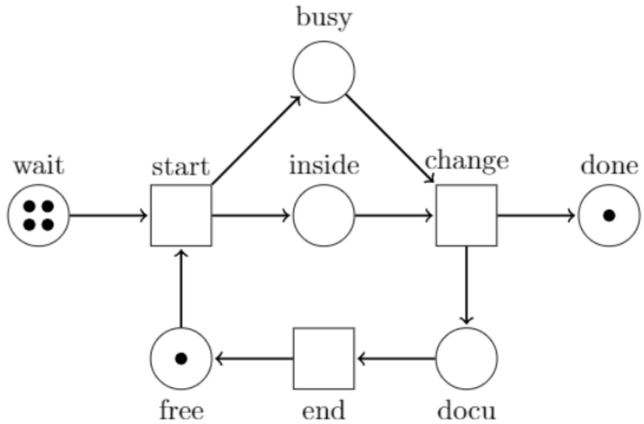
According to 2 Petri net N_S and N_{Pa}, transitions **start**(N_S) and **start**(N_{Pa}) act on the same physical token, we identify them into 1 transition **start** in N. In a similar way, transitions **change**(N_S) and **change**(N_{Pa}) act on the same physical token, we identify them into 1 transition **change** in N.



Petri net N_S Petri Net N_{Pa}

Now, the combined Petri net N will be:
$N = N_S \oplus N_{Pa} = \{P_S \cup P_{Pa}, T, F_S \cup F_{Pa}, M_0\}$ where:

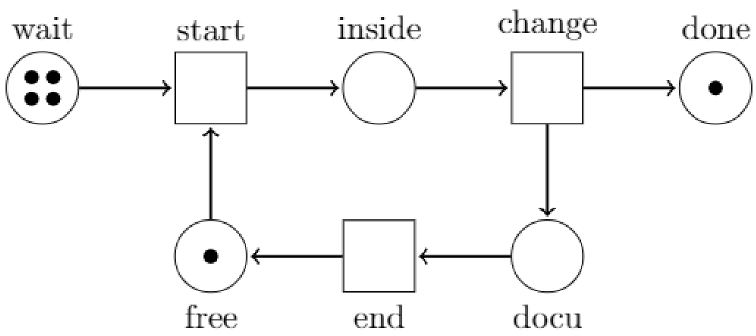
- $P_S \cup P_{Pa} = \{wait, inside, done, free, busy, docu\}$
- $T = \{start, change, end\}$
- $F_S \cup F_{Pa} = \{(wait, free \times start) \cup (start \times inside, busy), (inside, busy \times change) \cup (change \times done, docu), (docu \times end) \cup (end \times free)\}$
- $M_0 = \{wait^4, done, free\}$



We can see that place **inside** and **busy** have the same input and output transition, then we merge them into 1 place, calling **inside**. Now the Petri net N will be:
$N = N_S \oplus N_{Pa} = \{P_S \cup P_{Pa}, T, F_S \cup F_{Pa}, M_0\}$ where:

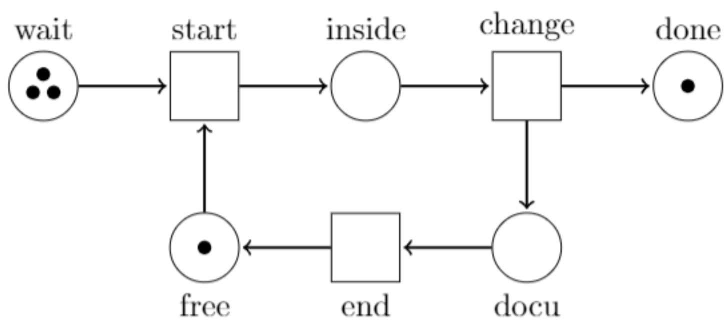
- $P_S \cup P_{Pa} = \{wait, inside, done, free, docu\}$
- $T = \{start, change, end\}$

- $F_S \cup F_{Pa} = \{(wait, free \times start) \cup (start \times inside), (inside \times change) \cup (change \times done, docu), (docu \times end) \cup (end \times free)\}$
- $M_0 = \{wait^4, done, free\}$



4.4 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]**
Answer:
We have the petri net N from Problem 3 with the initial marking $M_0 = \{wait^3, free, done\}$



There are 10 reachable marking from M_0:

- $M_0 = \{wait^3, free, done\}$ can be reached via the sequence $\sigma_0 =<>$
- $M_1 = \{wait^2, inside, done\}$ via $\sigma_1 =< start >$
- $M_2 = \{wait^2, docu, done^2\}$ via $\sigma_2 =< start, change >$
- $M_3 = \{wait^2, free, done^2\}$ via $\sigma_3 =< start, change, end >$
- $M_4 = \{wait, inside, done^2\}$ via $\sigma_4 =< start, change, end, start >$
- $M_5 = \{wait, docu, done^3\}$ via $\sigma_5 =< start, change, end, start, change >$

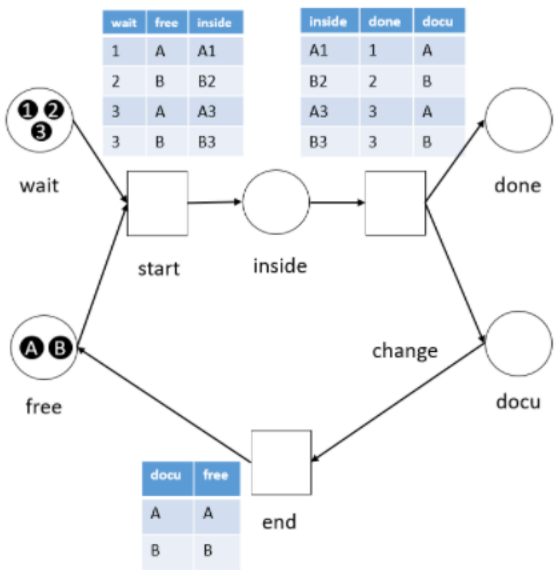
- $M_6 = \{wait, free, done^3\}$ via $\sigma_6 =< start, change, end, start, change, end >$
- $M_7 = \{inside, done^3\}$ via $\sigma_7 =< start, change, end, start, change, end, start >$
- $M_8 = \{docu, done^4\}$ via $\sigma_8 =< start, change, end, start, change, end, start, change >$
- $M_9 = \{free, done^4\}$ via $\sigma_9 =< start, change, end, start, change, end, start, change, end >$

4.5 Problem 5

Is the superimposed Petri net N deadlock free? Explain properly. **[1 point]**
Answer: The answer is No, Petri Net N is not a deadlock free.
Justification:
A marked Petri net is deadlock free if at every reachable marking at least one transition is enabled. Since at marking $M_9 = \{done^4, free\}$, there don't exist any transition $t \in T$ such that $(N, M_9)[t\rangle$

4.6 Problem 6

Propose a similar Petri net with two specialists already for treating patients, with explicitly explained construction. **[1 point]**
Answer: In Colored Petri net, we can control the firing rule at each transition in order to allow a much more concise model of a system (than Classic Petri net).
In the course of business around two specialists, we can control the firing rule to implement some appointments made by patients.



Color set declaration:

- wait: $\{1, 2, 3\}$

- inside: $\{A1, A3, B2, B3\}$
- done: $\{1, 2, 3\}$
- docu: $\{A, B\}$
- free: $\{A, B\}$

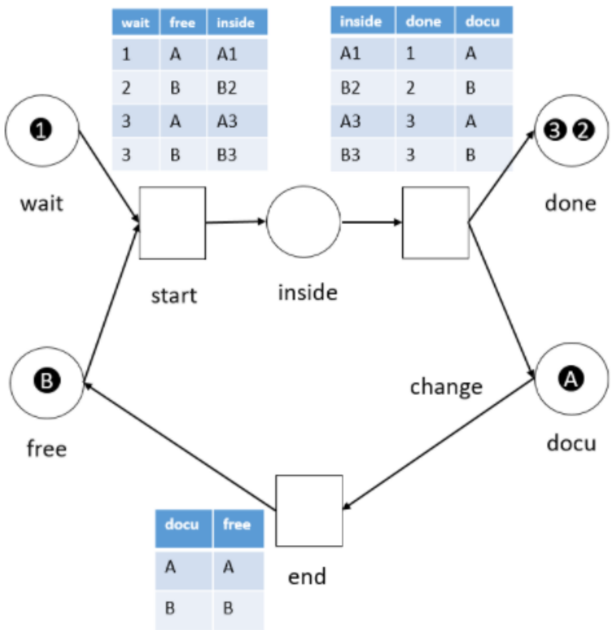
In the case above, at transition **start:**

- patient 1 made an appointment with specialist A,
- patient 2 made an appointment with specialist B,
- patient 3 mode no appointment and agree to be treated by any specialists.

So, transition **start** only fires when:

- place **wait** has token 1, place **free** has token A, or
- Place **wait** has token 2, place **free** has token B, or
- Place **wait** has token 3, place **free** has token A or B.

After firing, place **inside** has a corresponding token (in the firing rule table above). By using the color Petri net, some case call happen because of the firing rule that we set:



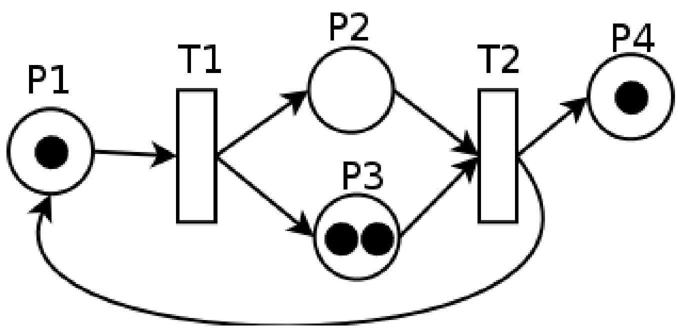
Although both places **wait** and **free** have token, transition start can't fire because firing rule do not allow this situation. It means patient 1 don't make an appointment with specialist B, he/she has to wait for specialist A being free (after transition end fires).

4.7 Problem 7

Write a computational package to realize (implement) Items 1,2,3 and 4 above. You could employ any programming language (mathematical- statistical like R, Matlab, Maple, Singular, or multi-purposes language like C or Python), that you are familiar with, to write computational programs. The programs of your package should allow input with max 10 patients in place wait.
[2 points]
Answer
In this assignment,we decided to use IDE Rad Studio for the UI and to implement the computationnal program using C++.
We come up with the idea of building a matrix for Petri net. For example, we denote a quadruplet of Petri net $N = (P, T, F, M_0)$ where P is a set of places, T is a set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of flow relations, and M_0 is the initial marking.
Create a matrix where the left vertical axis is elements of set P and the above horizontal axis is elements of set T.

- If place p_i is a preset of transition t_i, marked -1.
- If place p_i is a postset of transition t_i, marked 1.
- If there is no relation between place p_i and transition t_i (it means place p_i is not a preset or postset of transition t_i), marked 0.

For example, from a Petri net in the picture below

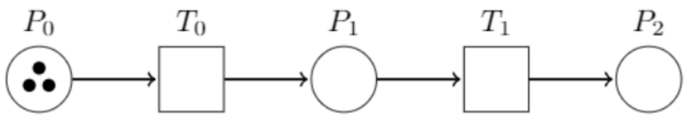


A matrix 4×2 is created according to:

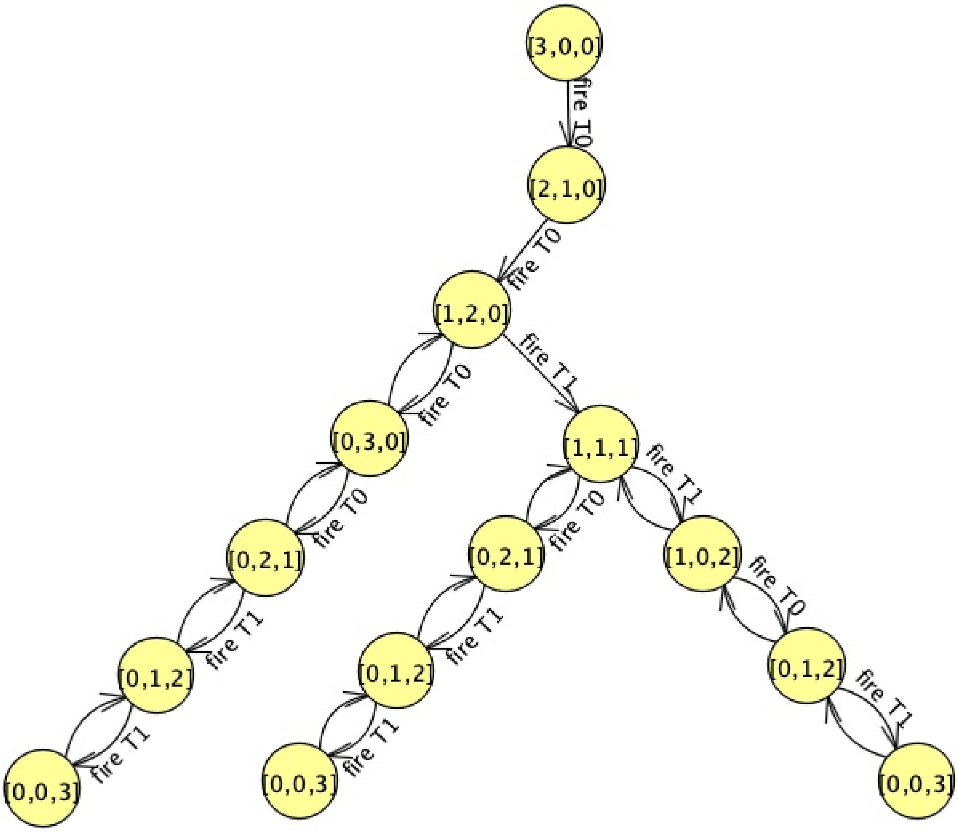
- Place p_1 is a preset of transition t_1 and a postset of transition t_2.
- Places p_2, p_3 are postsets of transition t_1 and presets of transition t_2.
- Place p_4 has no relation with transition t_1, and is a postset of transition t_2.

$$\begin{array}{c|ccc} * & t1 & t2 \\ \hline p1 & -1 & 1 \\ p2 & 1 & -1 \\ p3 & 1 & -1 \\ p4 & 0 & 1 \end{array}$$

Creating matrix saves the connection (relation) between places and transitions.
The initial marking $M_0 = [1\ 0\ 2\ 1]$
The next step, we create a Petri net class with basic variables: places, transitions, matrices store values $-1, 0, 1$ as description.
We come up with the Backtracking algorithm to find the reachable markings with the given input.
The backtracking algorithm enumerates a set of partial candidates that, in principle, could be completed in various ways to give all the possible solutions to the given problem. The completion is done incrementally, by a sequence of candidate extension steps.
Conceptually, the partial candidates are represented as the nodes of a tree structure, the potential search tree. Each partial candidate is the parent of the candidates that differ from it by a single extension step; the leaves of the tree are the partial candidates that cannot be extended any further.
We will check which transition is enable, then find a reachable marking in that flow; if the deadlock is found, use Backtracking algorithm to find another reachable marking.
For example, according to this Petri net:



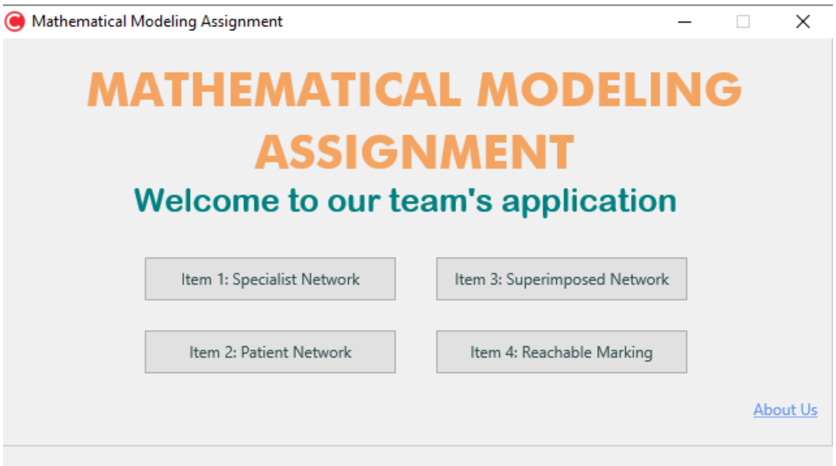
We represent the reachable markings as a tree (this illustration below is a small example, not all reachable markings are graphed here):



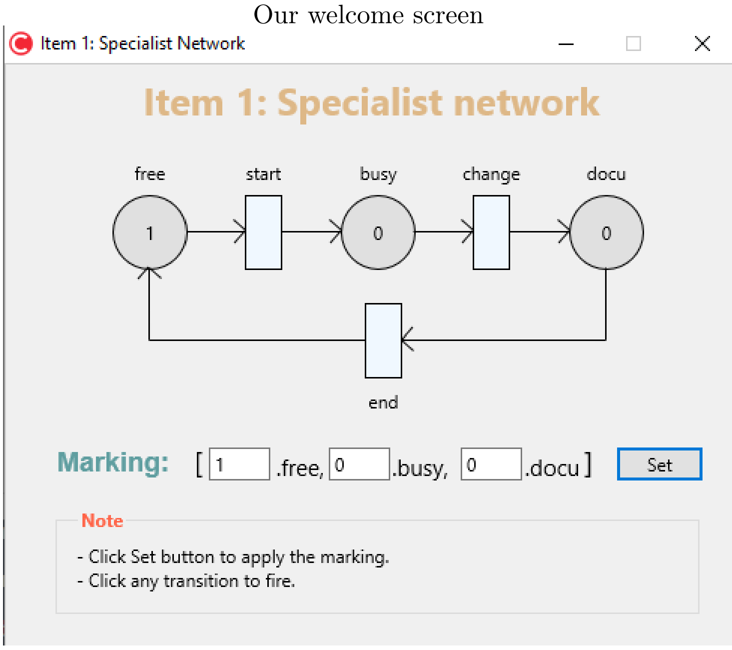
At the end, when it comes to deadlock, we Backtracking it to the "root" node of the subtree. From this useful algorithm, we can find all reachable markings with the given input.

RESULT:

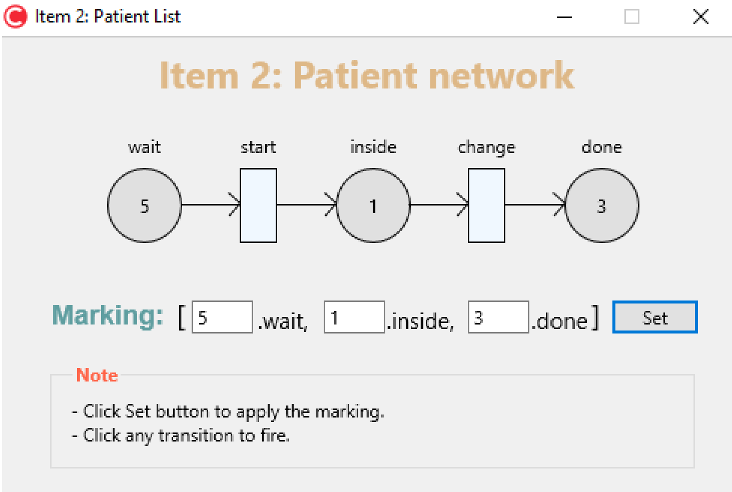
Here is some picture about our application:



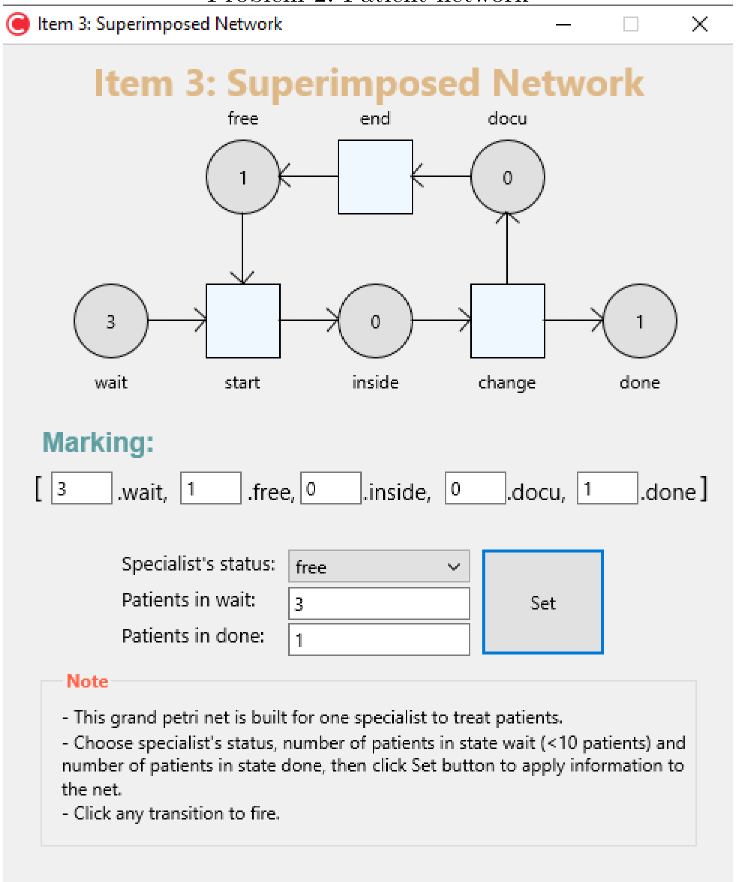
Our welcome screen



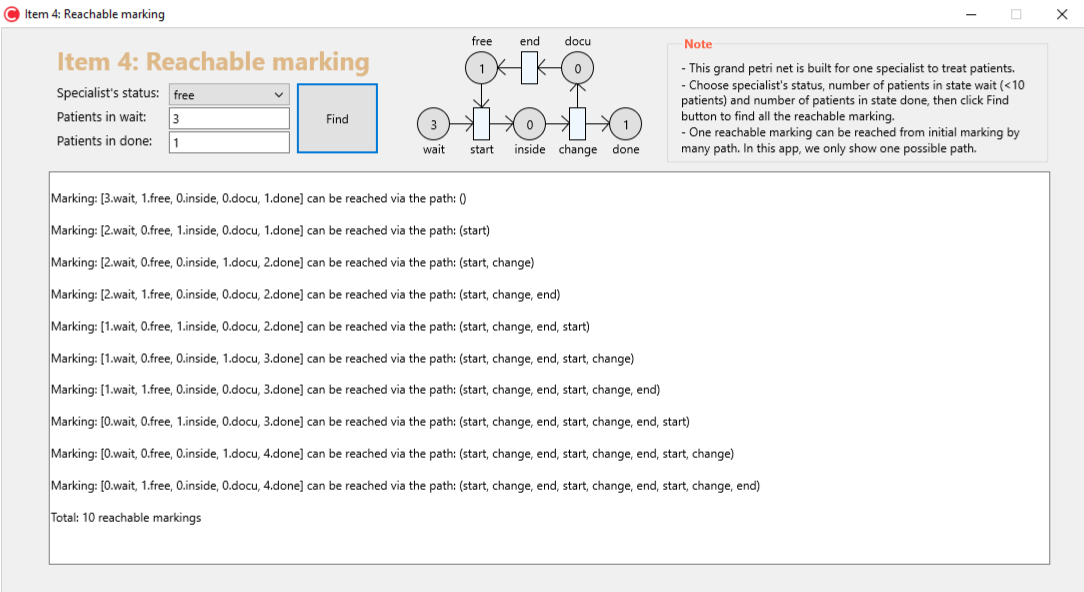
Problem 1: Specialist network



Problem 2: Patient network



Problem 3: Superimposition network



Problem 4: Reachable marking

References

[1] Part I of PROCESS MINING, 2nd edition, 2016, Springer, by W.M.P. Aalst, van der1

[2] Process mining techniques and applications - A systematic mapping study, in Expert Systems With Applications, Vol 133 (2019) 260- 295, Elsevier, by Cleiton dos Santos Garcia et. al.

[3] Chapter 4 - 9 of the text DATA MINING - Concepts, Models, Methods, and Algorithms, 3rd Edition, 2020, IEEE press and Wiley, by Mehmed Kantardzic.

[4] Modeling Business Processes: A Petri Net-Oriented Approach, 2011 Massachusetts Institute of Technology, by Wil van der Aalst and Christian Stahl.

[5] REF for MM Chapter 5: PetriNet, 2021, by Dr Man VM. Nguyen

[6] Chap 4 of Beyond the Third Dimension, by Thomas F. Banchoff