Human Computer Interaction

UNIT-5
Lecture 6:
Task modeling and analysis

Mr. Nachiket Sainis/Ms.Reena Saini

B K Birla Institute of Engineering & Technology, Pilani

Dialog Design

Petri Nets

Objective

- In the previous lecture, we discussed about StateCharts, a formalism suitable for dialog design, which is potentially more expressive than STNs.
- In this lecture, we shall learn about another powerful formalism for dialog design, namely the (classical) Petri Nets.

(Classical) Petri Net

- The formalism was first proposed by Carl Adam Petri (1962).
- It is a simple model of dynamic behavior
  - Just four elements are used to represent behavior: **places**, **transitions, arcs and tokens**
  - Graphical and mathematical description for easy understanding
  - Formal semantics allow for analysis of the behavior

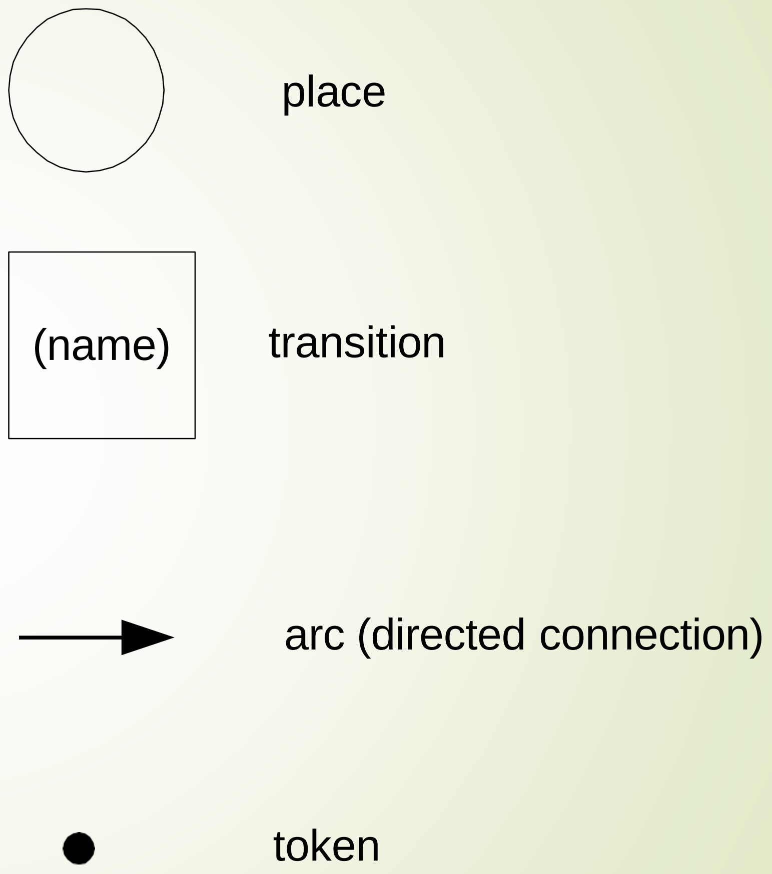
Elements of PN

- Place: used to represent passive elements of the reactive system(possible state of the system).
- Transition: used to represent active elements of the reactive system(event or action).
- Arc: used to represent causal relations
- Token: elements subject to change

The state (space) of a process/system is modeled by places and tokens and state transitions are modeled by transitions

Elements of PN: Notation

- A place is represented by a circle
- Transitions are represented by squares/rectangles
- Arc are represented by arrows
- Tokens are represented by small filled circles



Role of a Token

- **Tokens can play the following roles:**
  - A **physical object**, for example a product, a part, a drug, a person
  - An **information object**, for example a message, a signal, a report
  - A **collection of objects**, for example a truck with products, a warehouse with parts, or an address file
  - An **indicator of a state**, for example the indicator of the state in which a process is, or the state of an object
  - An **indicator of a condition**: the presence of a token indicates whether a certain condition is fulfilled

Role of a Place

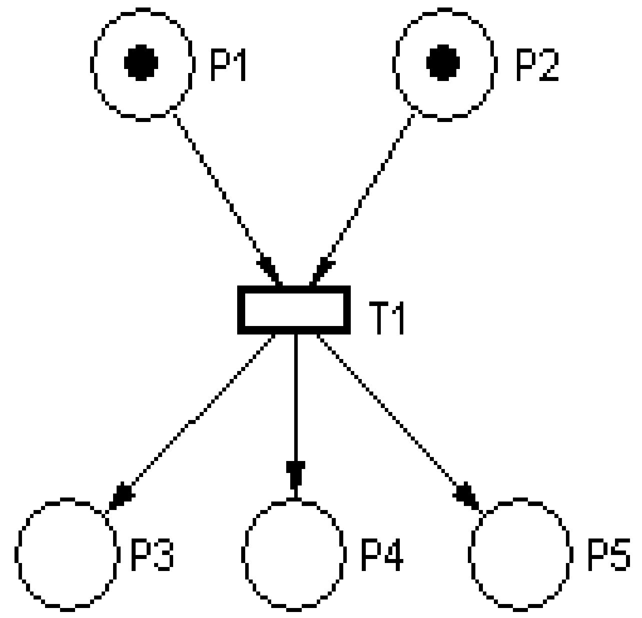
- **A place in a PN can represent the following**
  - A type of **communication medium**, like a telephone line, a middleman, or a communication network
  - A **buffer**: for example, a depot, a queue or a post bin
  - A **geographical location**, like a place in a warehouse, office or hospital
  - A possible **state or state condition**: for example, the floor where an elevator is, or the condition that a specialist is available

Role of a Transition

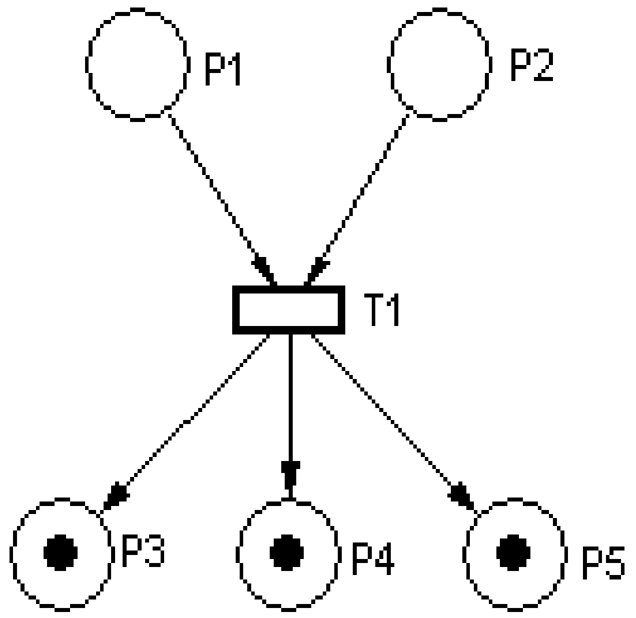
- A transition can be used to represent things such as
  - An **event** (e.g., starting an operation, the switching of a traffic light from red to green)
  - A **transformation of an object**, like adapting a product, updating a database, or updating a document
  - A **transport of an object**: for example, transporting goods, or sending a file

PN Construction Rules

- Arcs have capacity 1 by default; if other than 1, the capacity is marked on the arc.
- Places have infinite capacity by default, and transitions have no capacity, and cannot store tokens at all.
- A transition is enabled when the number of tokens in each of its input places is at least equal to the arc weight going from the place to the transition.
- An enabled transition may fire at any time. When fired, the tokens in the input places are moved to output places, according to arc weights and place capacities.
- The firing represents an occurrence of the event or an action taken.

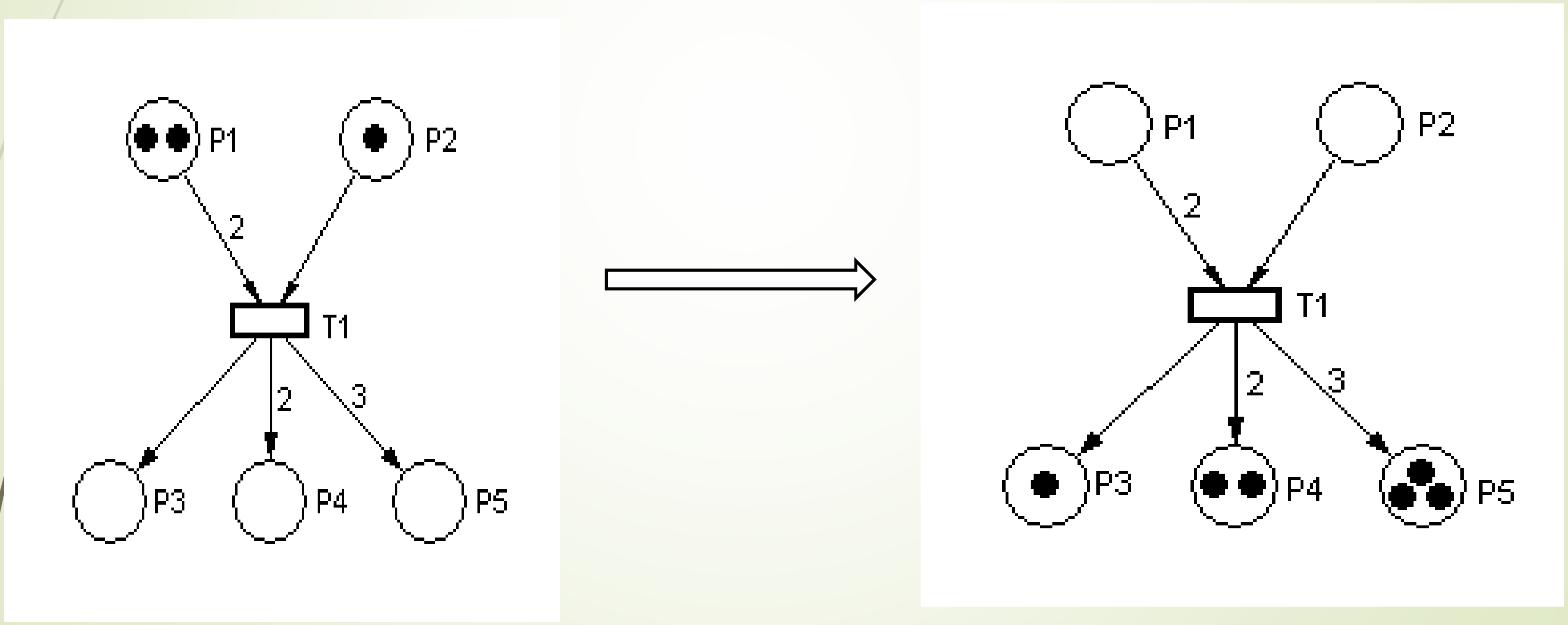


Enabled.ready to file



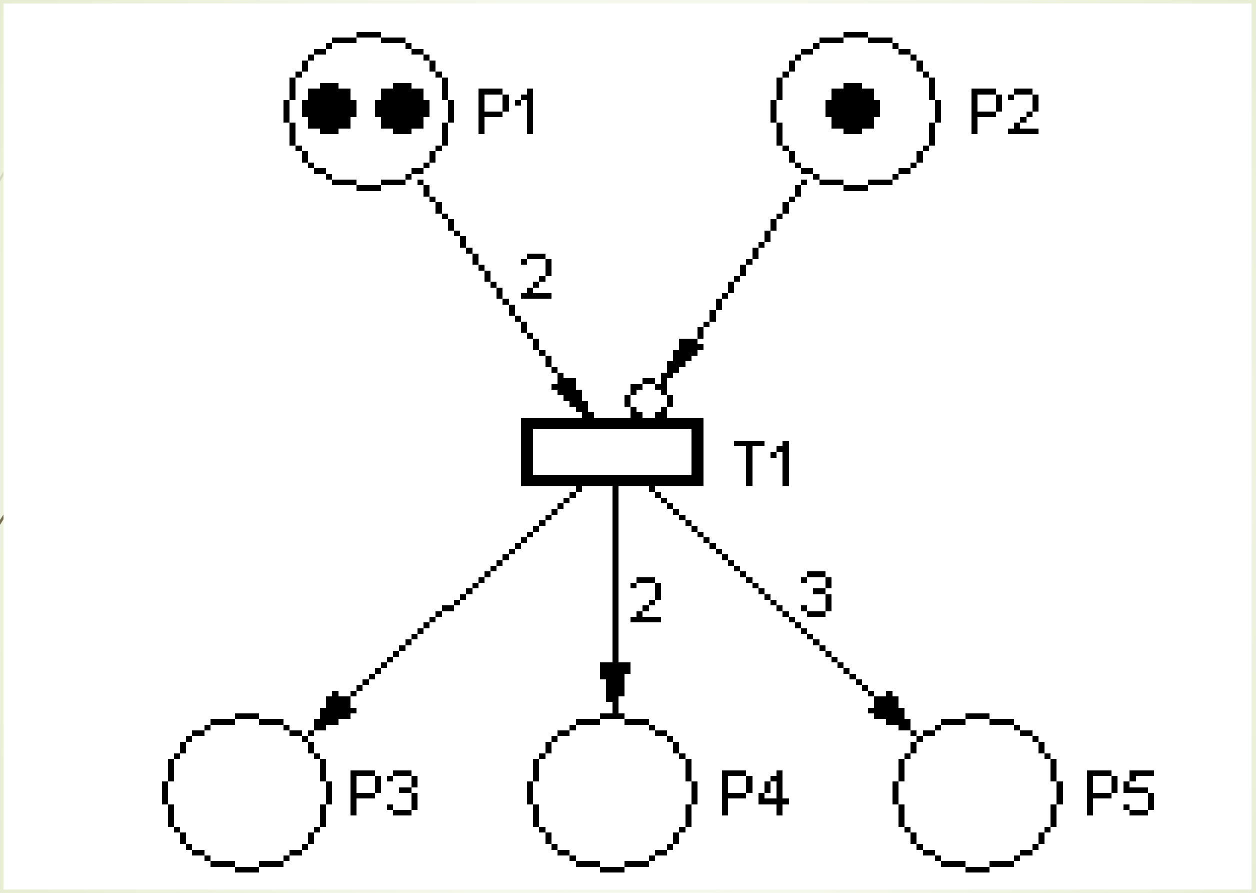
Firing complete

- When arcs have different weights, we have what might at first seem confusing behavior. Here is a similar net, ready to fire: and here it is after firing:



- Change of state is denoted by a movement of *token(s)* (black dots) from place(s) to place(s); and is caused by the *firing* of a transition.
- The firing represents an occurrence of the event or an action taken.
- A transition is *firable* or *enabled* when there are sufficient tokens in its input places.
- After firing, tokens will be transferred from the input places (old state) to the output places, denoting the new state.

- If the arc weights are all the same, it appears that tokens are moved across the transition. If they differ, however, it appears that tokens may disappear or be created. That, in fact, is what happens; think of the transition as removing its enabling tokens and producing output tokens according to arc weight.
- A special kind of arc, the inhibitor arc, is used to reverse the logic of an input place. With an inhibitor arc, the absence of a token in the input place enables, not the presence:

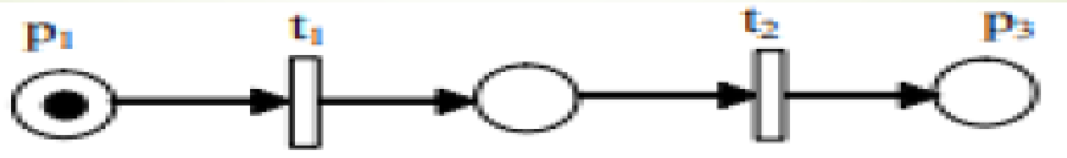

P1
P2
2
T1
2
3
P3
P4
P5

Remarks

- Firing is **atomic** (i.e., it always completes after start)
- Non-determinism: multiple transitions may be enabled, but only one fires at a time
- The **state** of the reactive system is represented by the distribution of tokens over places (also referred to as **marking**)

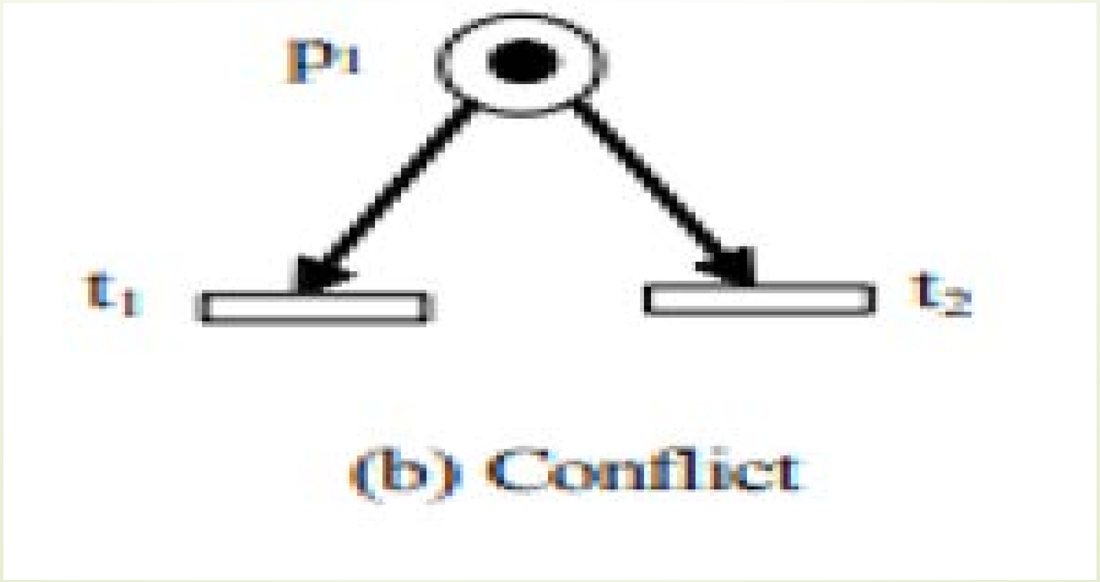
Modeling Power

- The typical characteristics exhibited by the activities in a dynamic event-driven system, such as concurrency, decision making, synchronization and priorities, can be modeled effectively by Petri nets.
- **Sequential Execution.** In Figure (a), transition t2 can fire only after the firing of t1. This imposes the precedence constraint "t2 after t1." Such precedence constraints are typical of the execution of the parts in a dynamic system. Also, this Petri net construct models the causal relationship among activities.

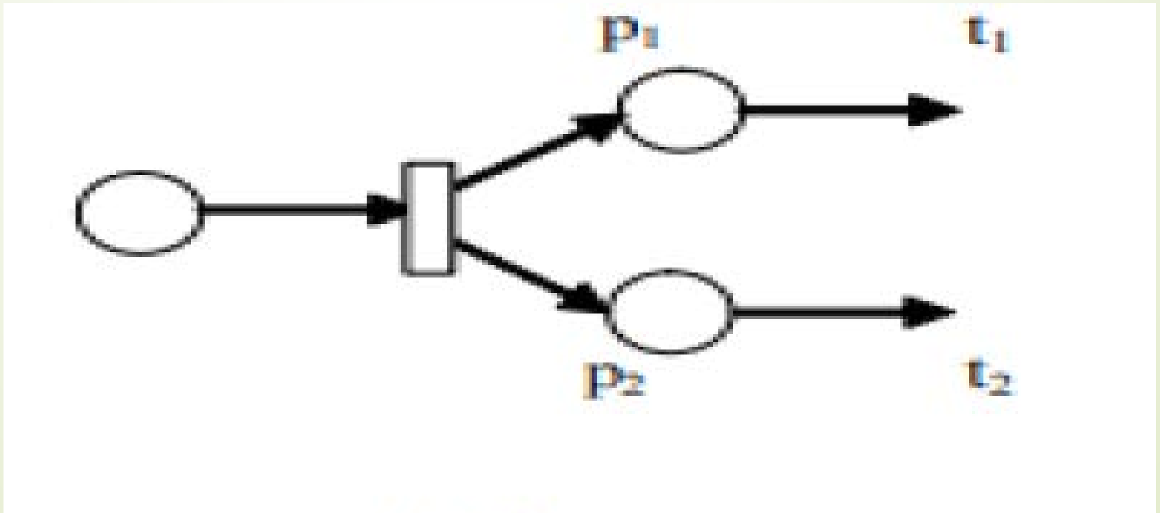


(a) Sequential

Conflict. Transitions t1 and t2 are in conflict in Figure (b). Both are enabled but the firing of any transition leads to the disabling of the other transition. Such a situation will arise, for example, when a machine has to choose among part types or a part has to choose among several machines.

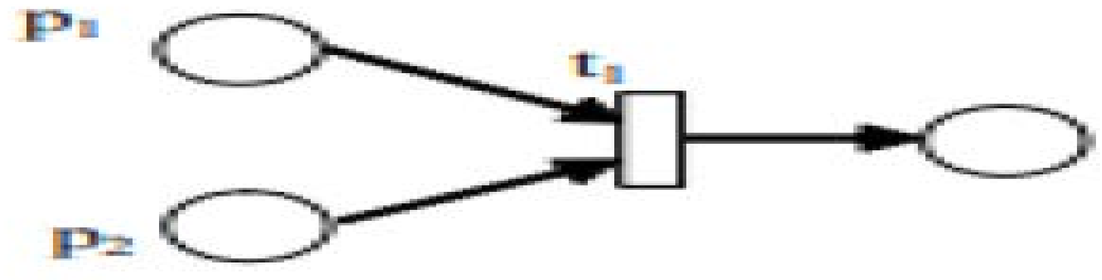


Concurrency. In Figure (c), the transitions t1, and t2 are concurrent. Concurrency is an important attribute of system interactions. Note that a necessary condition for transitions to be concurrent is the existence of a forking transition that deposits a token in two or more output places.



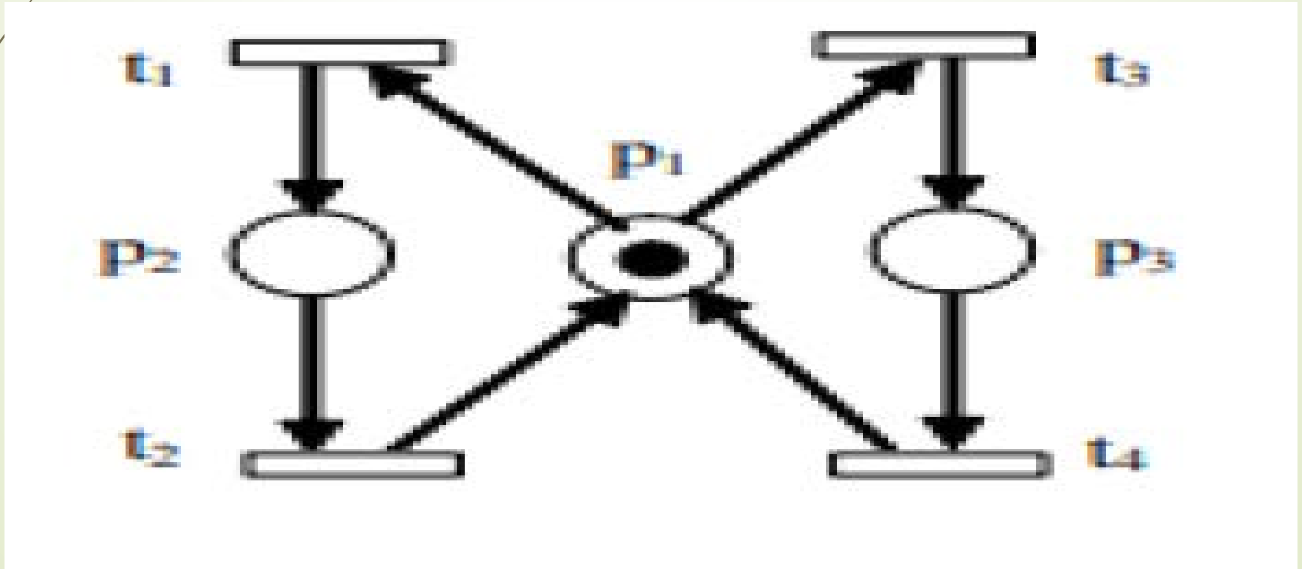
(c) Concurrent

Synchronization. It is quite normal in a dynamic system that an event requires multiple resources. The resulting synchronization of resources can be captured by transitions of the type shown in Figure (d). Here, t1 is enabled only when each of p1 and p2 receives a token. The arrival of a token into each of the two places could be the result a possibly complex sequence of operations elsewhere in the rest of the Petri net model. Essentially, transition t1 models the joining operation.



(d) Synchronization

Mutually exclusive. Two processes are mutually exclusive if they cannot be performed at the same time due to constraints on the usage of shared resources. Figure (e) shows this structure. For example, a robot may be shared by two machines for loading and unloading. Two such structures are parallel mutual exclusion and sequential mutual exclusion.

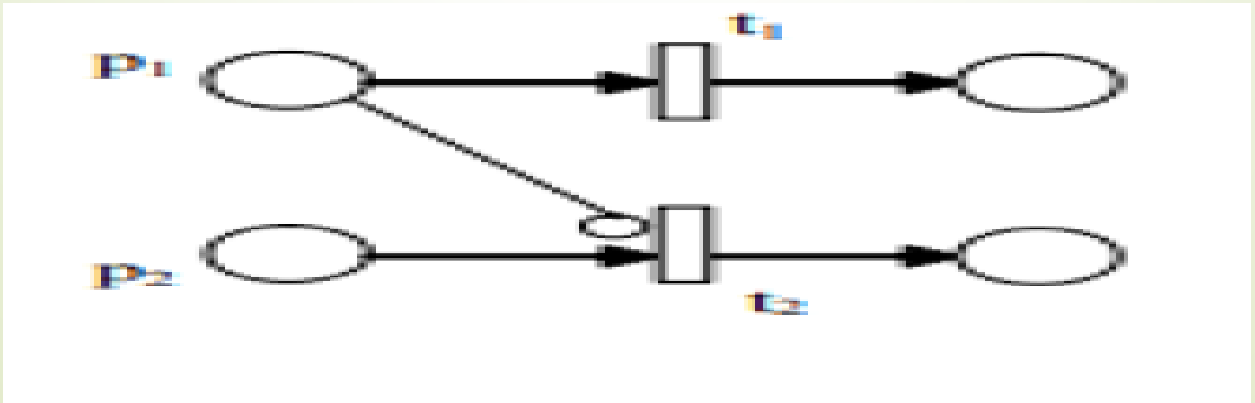


(e) Mutual exclusive

Priorities.

- The classical Petri nets discussed so far have no mechanism to represent priorities. Such a modeling power can be achieved by introducing an inhibitor arc.
- The inhibitor arc connects an input place to a transition, and is pictorially represented by an arc terminated with a small circle.
- The presence of an inhibitor arc connecting an input place to a transition changes the transition enabling conditions.

- The transition firing rule is the same for normally connected places. The firing, however, does not change the marking in the inhibitor arc connected places.
- A Petri net with an inhibitor arc is shown in Figure (f). t1 is enabled if p1 contains a token, while t2 is enabled if p2 contains a token and p1 has no token. This gives priority to t1 over t2.

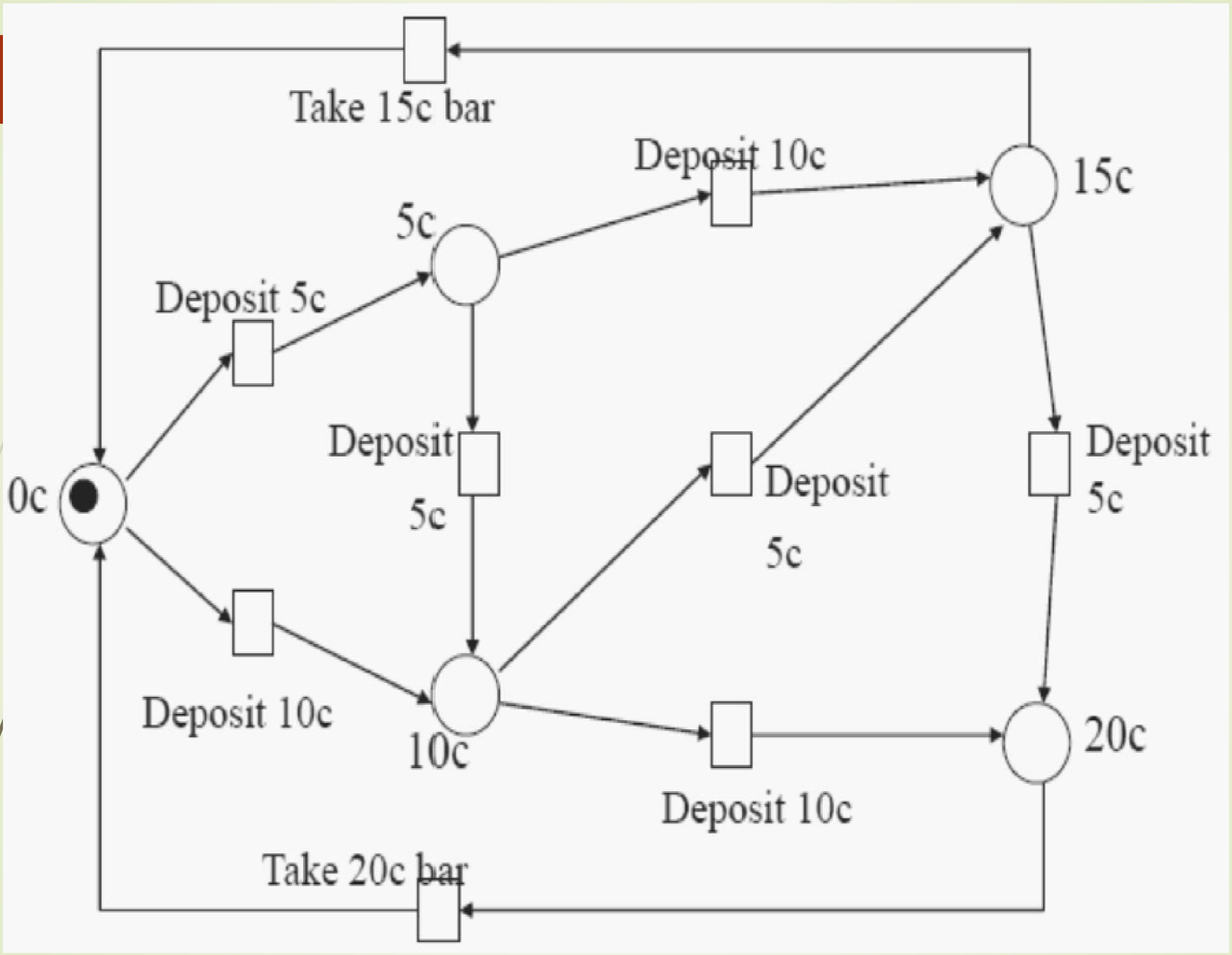


(f) Priority

Example- Vending Machine

- The machine dispenses two kinds of snack bars –20c and 15c.
- Only two types of coins can be used –10c coins and 5c coins.
- The machine does not return any change.

(C= cent)


Take 15c bar
Deposit 10c
15c
5c
Deposit 5c
Deposit
5c
Deposit
5c
Deposit
5c
0c
Deposit 10c
10c
Deposit 10c
20c
Take 20c bar

Example: Vending Machine (3 Scenarios)

Scenario 1:

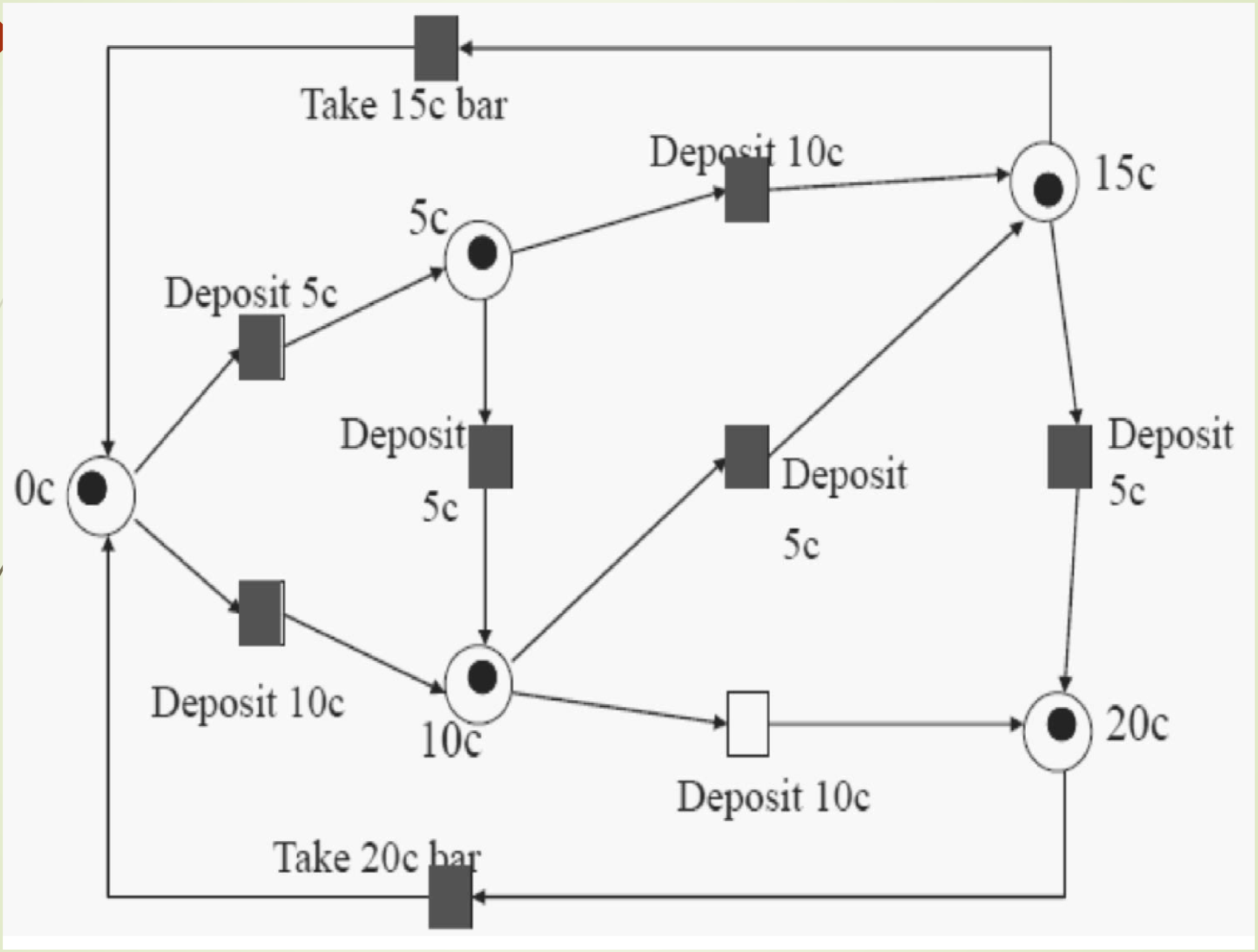
–Deposit 5c, deposit 5c, deposit 5c, deposit 5c, take 20c snack bar.

Scenario 2:

–Deposit 10c, deposit 5c, take 15c snack bar.

Scenario 3:

–Deposit 5c, deposit 10c, deposit 5c, take 20c snack bar.


Take 15c bar
Deposit 10c
15c
5c
Deposit 5c
Deposit
5c
Deposit
5c
Deposit
5c
0c
Deposit 10c
10c
Deposit 10c
20c
Take 20c bar