# Petri Nets I

# Definition of Petri Net

- **C = ( P, T, I, O)**
  - Places
    $P = \{ p_1, p_2, p_3, \ldots, p_n\}$
  - Transitions
    $T = \{ t_1, t_2, t_3, \ldots, t_n\}$
  - Input
    $I : T \rightarrow P^r$ (r = number of places)
  - Output
    $O : T \rightarrow P^q$ (q = number of places)

- marking μ : assignment of tokens to the places of Petri net $\mu = \mu_1, \mu_2, \mu_3, \ldots \mu_n$

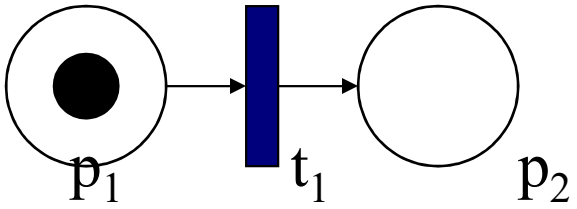
# Applications of Petri Net

- Petri net is primarily used for studying the dynamic concurrent behavior of network-based systems where there is a discrete flow.

- Petri Nets are applied in practice by industry, academia, and other places. -reference

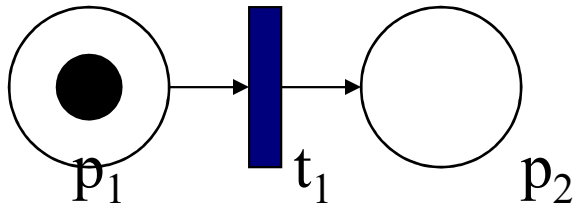
## Basics of Petri Nets

- Petri net consist two types of nodes: *places* and *transitions*. And arc exists only from a place to a transition or from a transition to a place.
- A place may have zero or more *tokens*.
- Graphically, places, transitions, arcs, and tokens are represented respectively by: circles, bars, arrows, and dots.



## Basics of Petri Nets -continued

- Below is an example Petri net with two places and one transaction.
- Transition node is ready to ***fire*** if and only if there is at least one token at each of its input places

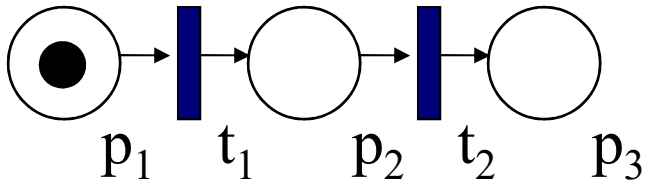



state transition of form (1, 0) → (0, 1)

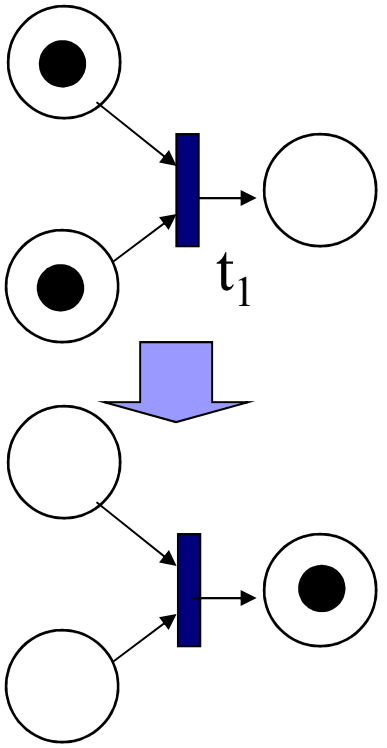
$p_1$ : input place $p_2$: output place

# Properties of Petri Nets

- **Sequential Execution**
  Transition $t_2$ can fire only after the firing of $t_1$. This impose the precedence of constraints "$t_2$ after $t_1$."




- **Synchronization**
  Transition $t_1$ will be enabled only when a token there are at least one token at each of its input places.

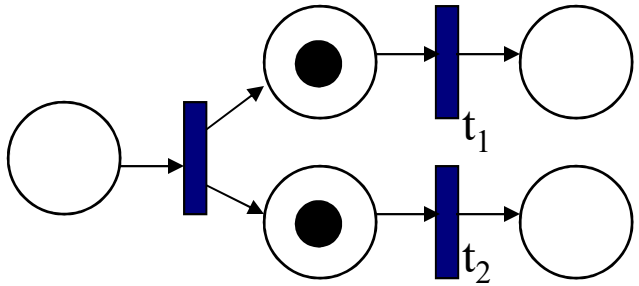



- **Merging**
  Happens when tokens from several places arrive for service at the same transition.

# Properties of Petri Nets -continued

- Concurrency

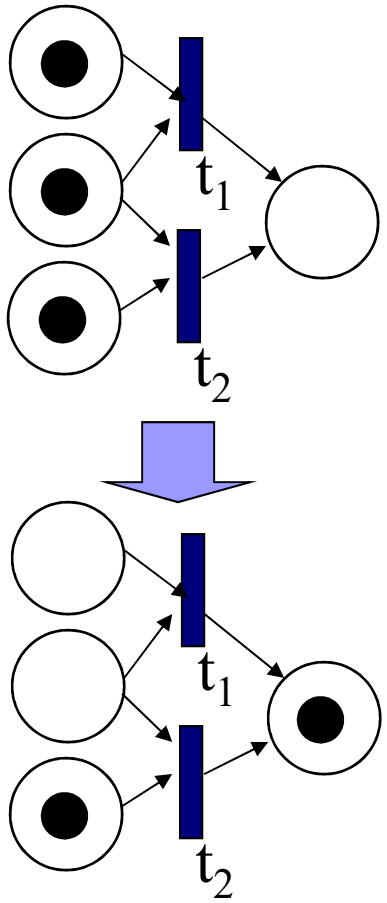
  $t_1$ and $t_2$ are concurrent.
  - with this property, Petri net is able to model systems of distributed control with multiple processes executing concurrently in time.



# Properties of Petri Nets -continued

- Conflict

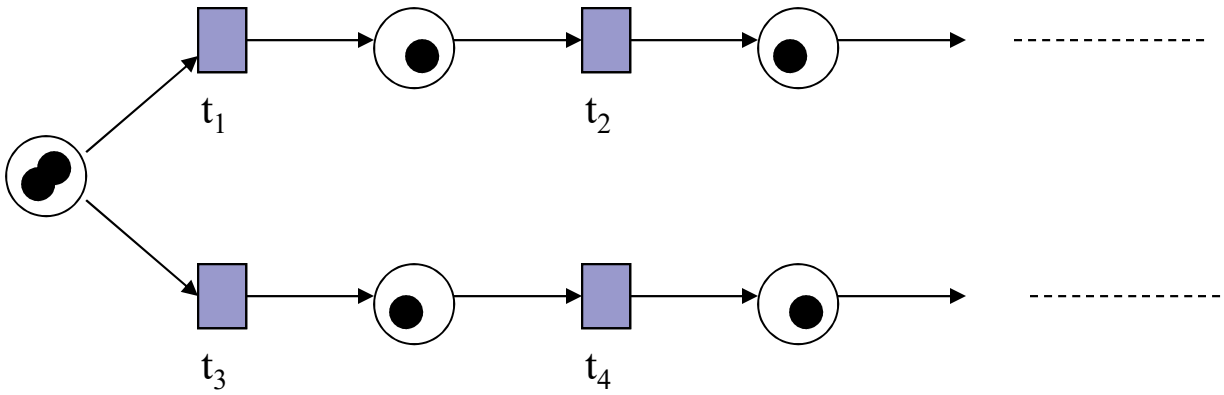
  $t_1$ and $t_2$ are both ready to fire but the firing of any leads to the disabling of the other transitions.



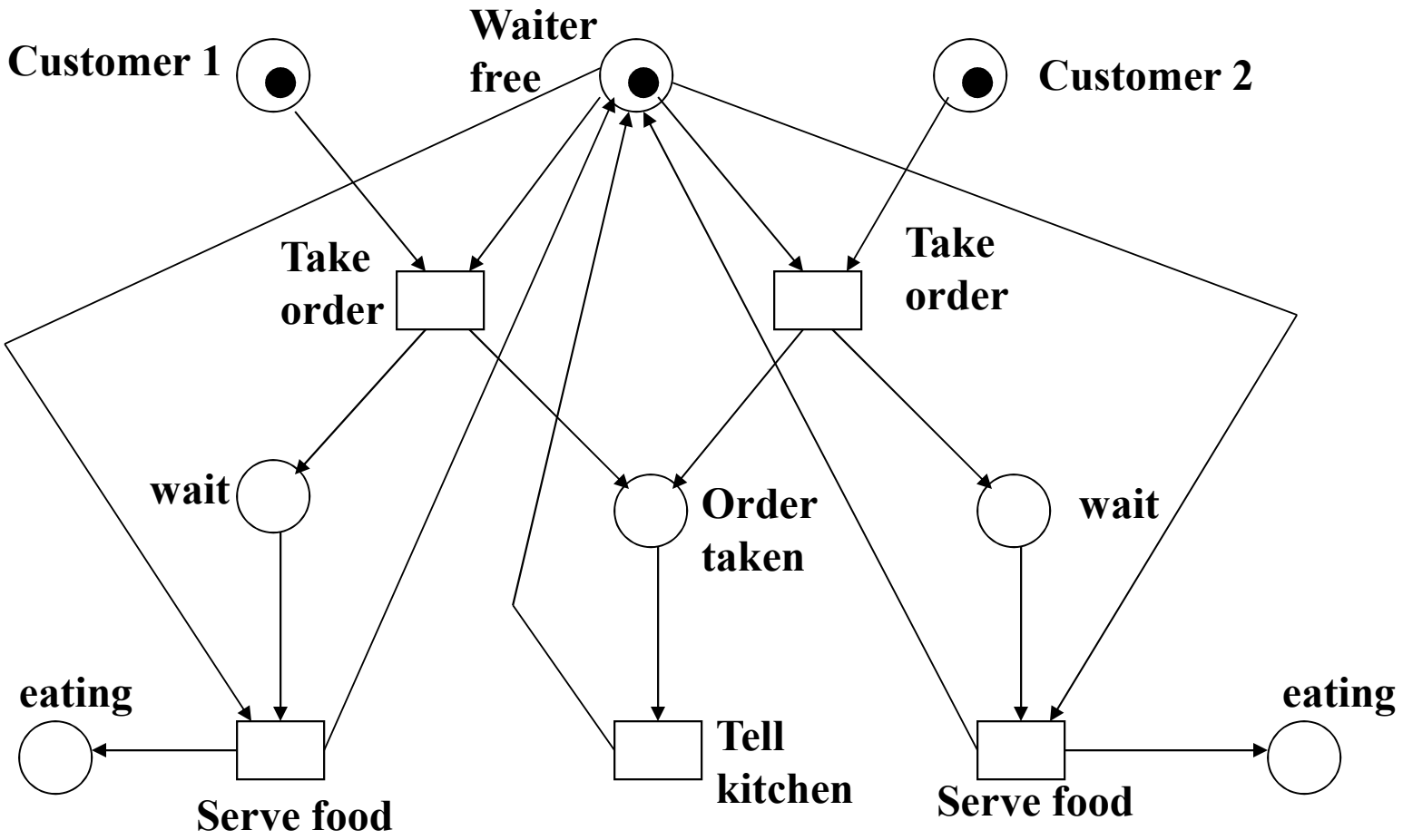
# Properties of Petri Nets -continued

- Conflict - continued
  - the resulting conflict may be resolved in a purely non-deterministic way or in a probabilistic way, by assigning appropriate probabilities to the conflicting transitions.

there is a choice of either $t_1$ and $t_2$, or $t_3$ and $t_4$



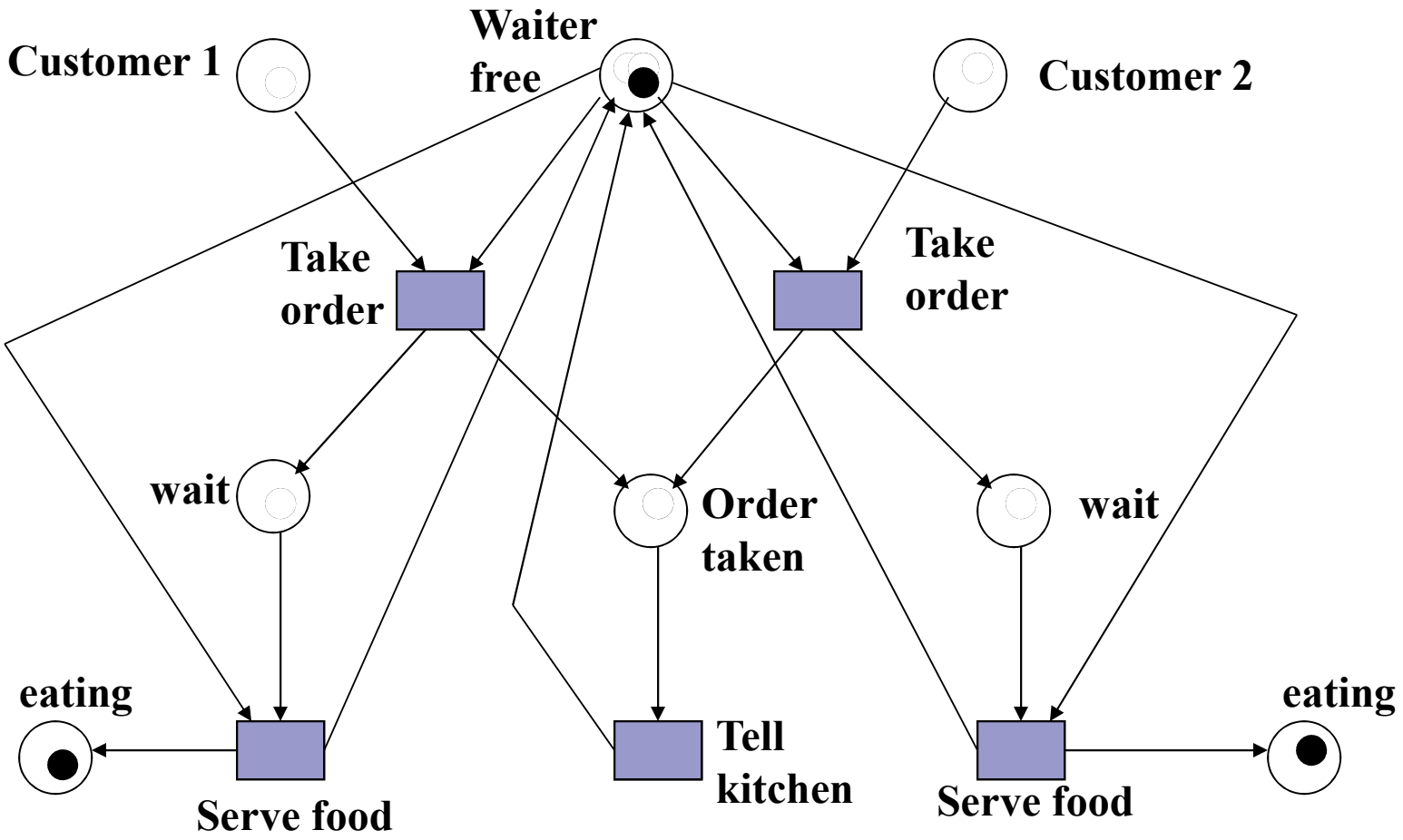
# Example: In a Restaurant (A Petri Net)



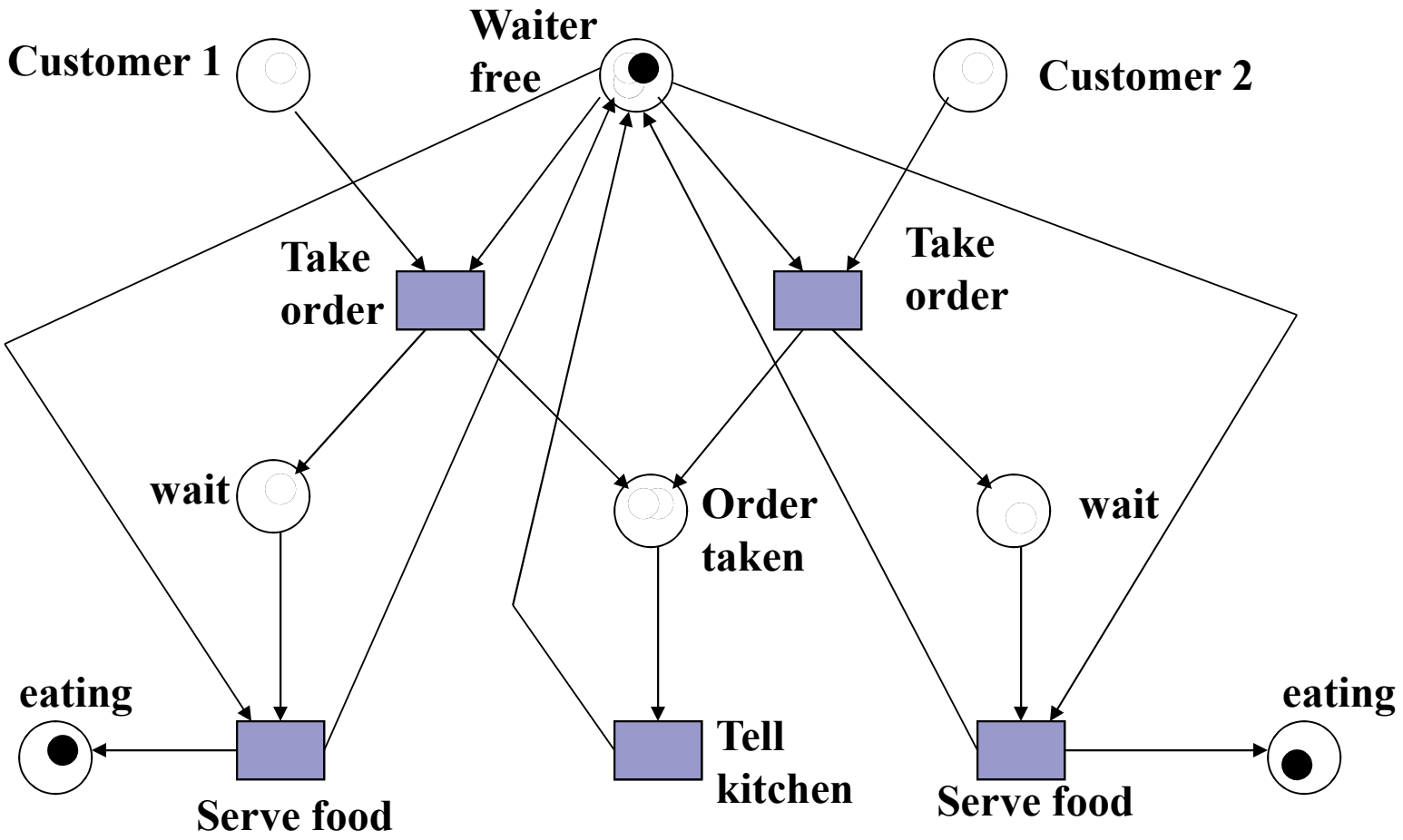
# Example: In a Restaurant (Two Scenarios)

- Scenario 1:
  - Waiter takes order from customer 1; serves customer 1; takes order from customer 2; serves customer 2.
- Scenario 2:
  - Waiter takes order from customer 1; takes order from customer 2; serves customer 2; serves customer 1.

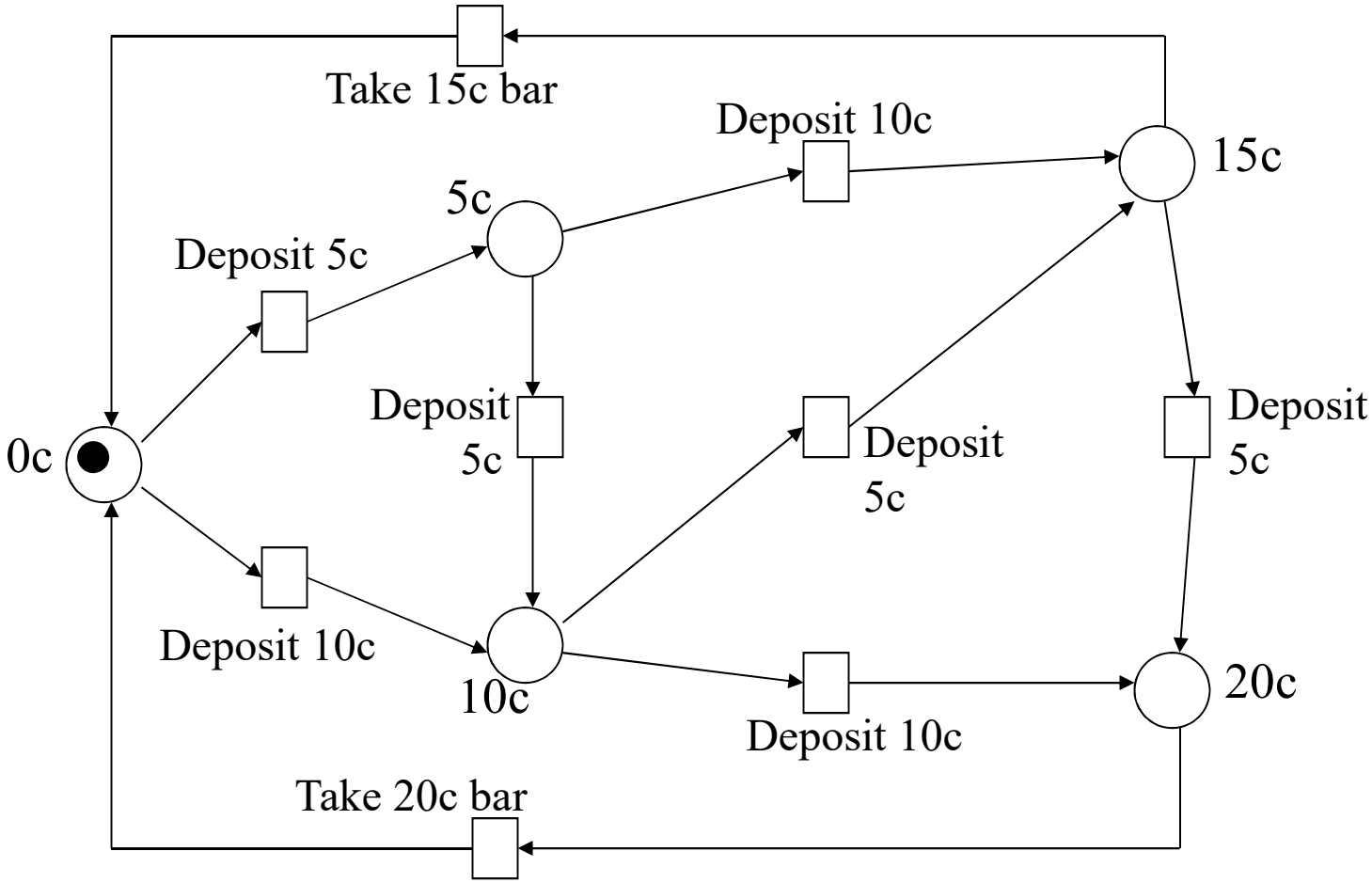
# Example: In a Restaurant (Scenario 1)



# Example: In a Restaurant (Scenario 2)



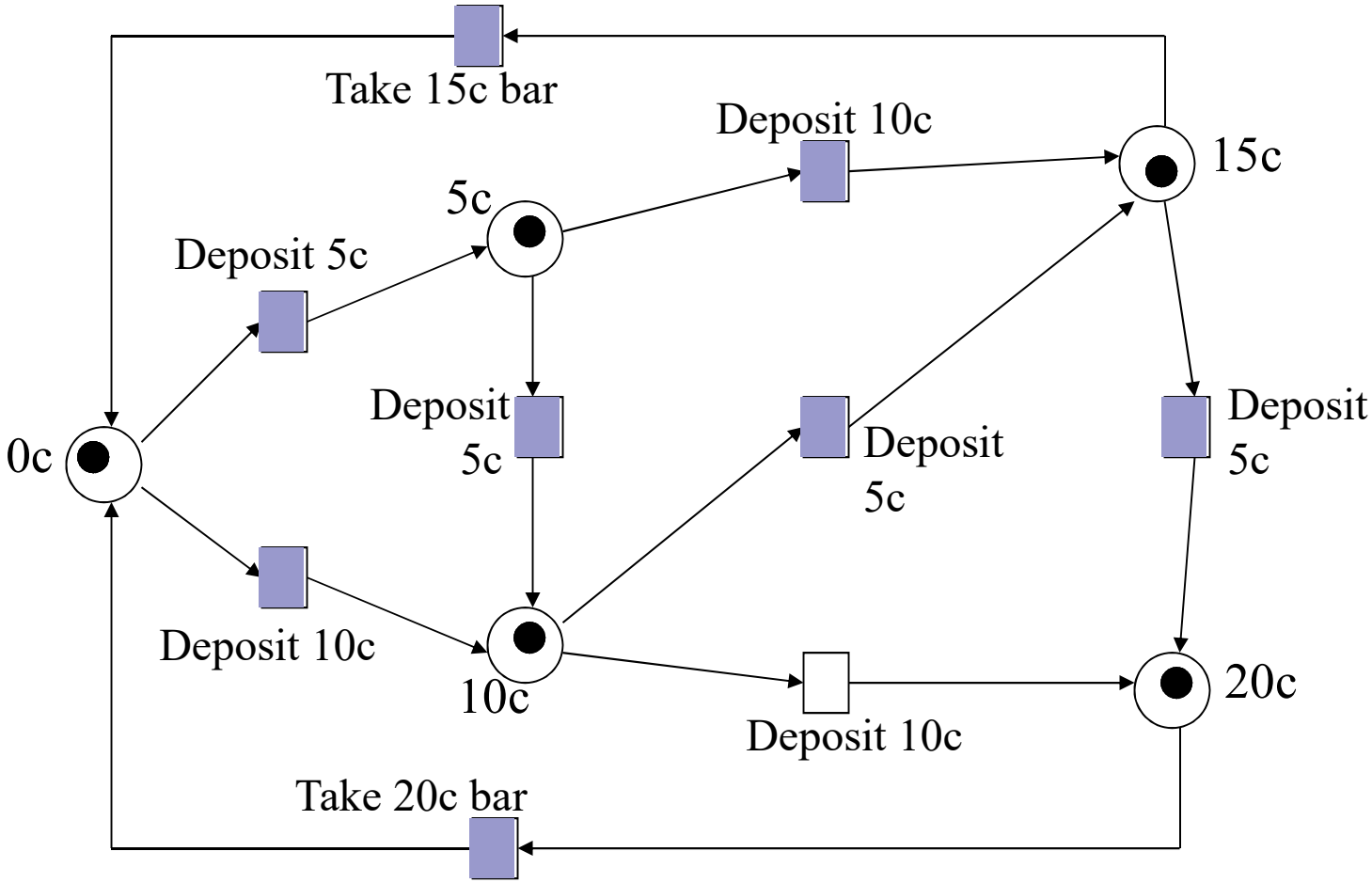
# Example: Vending Machine (A Petri net)



# 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.

# Example: Vending Machine (Token Games)



# Petri Net examples

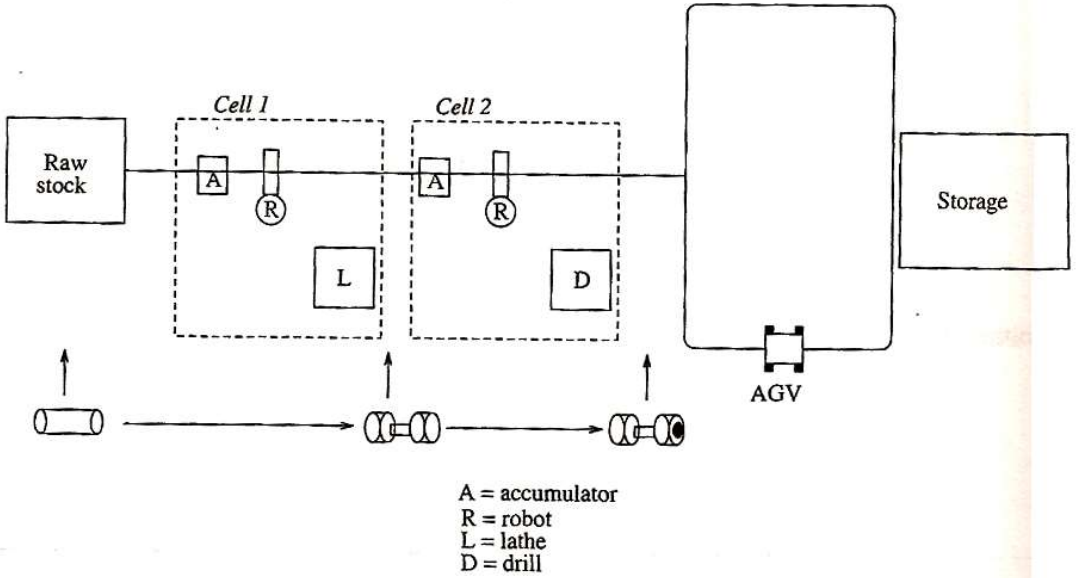



**Figure 4.40** Manufacturing line with two robots and two machines.

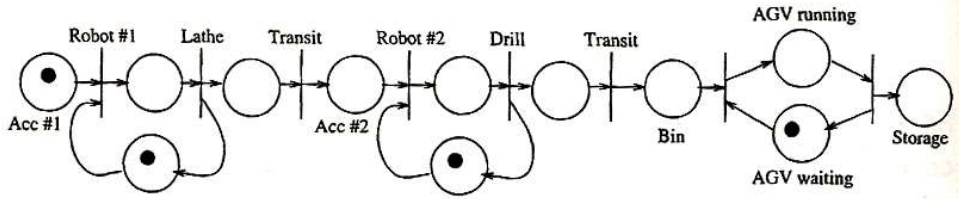



**Figure 4.41** Petri net for manufacturing line.

Dining Philosophers Problem States that there are 5 Philosophers who are engaged in two activities Thinking and Eating. Meals are taken communall y in a table with five plates and five forks in a cyclic manner as shown in th e figure.

**Constraints and Condition for the problem :**

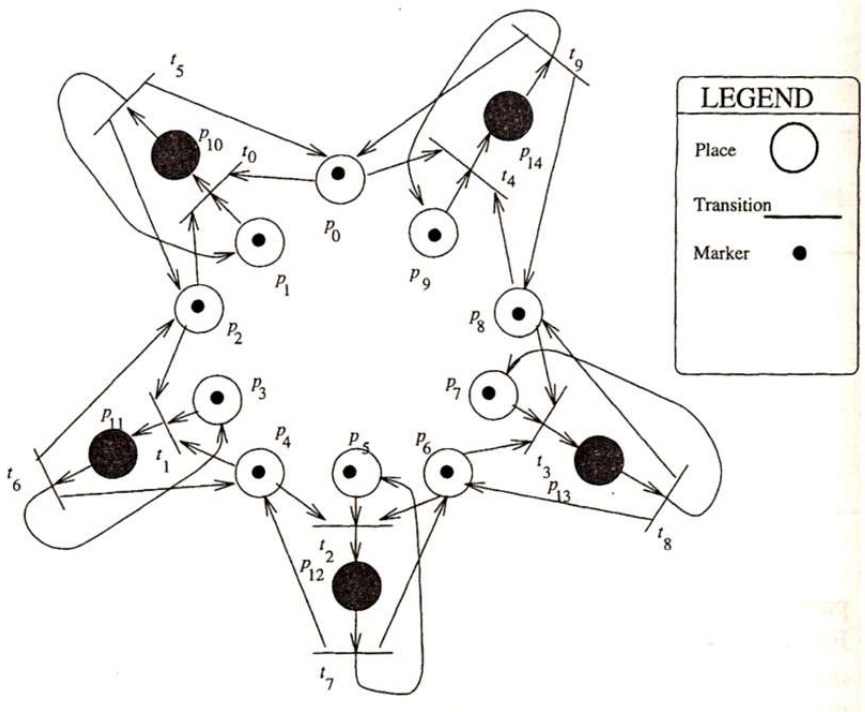
1.Every Philosopher needs two forks in order to eat.

2.Every Philosopher may pick up the forks on the left or right but only one f ork at once.

3.Philosophers only eat when they had two forks. We have to design such a protocol i.e. pre and post protocol which ensures that a philosopher only eats if he or she had two forks.

4.Each fork is either clean or dirty.

# Petri Net examples (Dining Philosophers)




- Five philosophers alternatively think and eating
- Chopsticks:
  $p_0$, $p_2$, $p_4$, $p_6$, $p_8$
- Philosophers eating:
  $p_{10}$, $p_{11}$, $p_{12}$, $p_{13}$, $p_{14}$
- Philosophers thinking/meditating:
  $p_1$, $p_3$, $p_5$, $p_7$, $p_9$

# Petri Net with Time

- 1962 - Carl Adam Petri originally proposed Petri without any notion of time. Concept of time was intentionally avoided because addition of time restricts the behavior of the net.
- 1970s ~ - Addition of time has been discussed in order to analyze the performance of the modeled system.
- Many properties are still undecided for Petri nets extended with data and time.

# References


- Fishwick, Paul(1995) – Simulation Model Design and Execution
- Petri Nets World
- Ling,Chris(2001) – Lecture on Petri Nets Method
- Chapman, Nick(1997) – Surprise97 journal on Petri Nets Models