The Petri Net Method

By

Dr Chris Ling

School of Computer Science & Software Engineering
Monash University

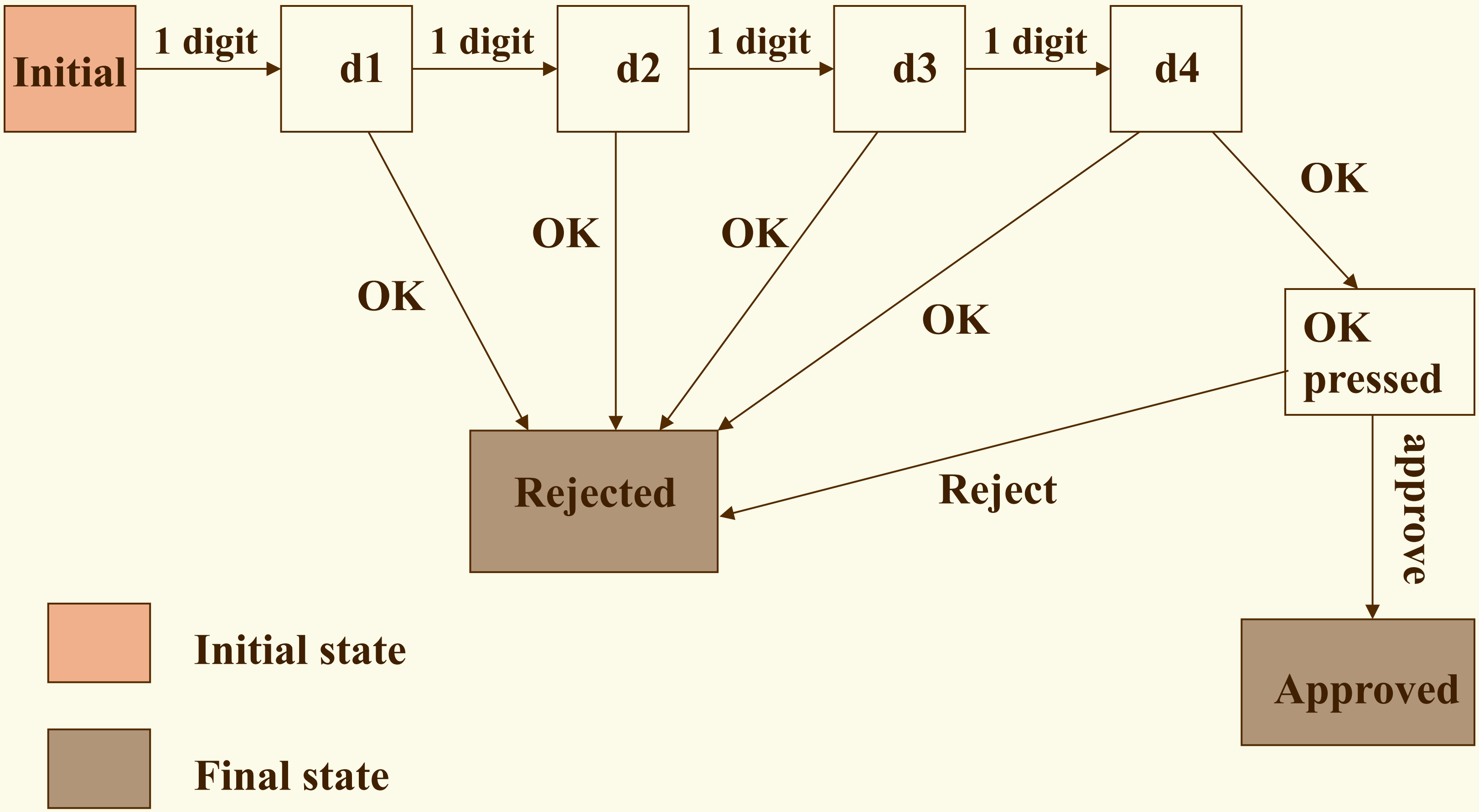
Chris.Ling@csse.monash.edu.au

Introduction

- ✓ First introduced by Carl Adam Petri in 1962.
- ✓ A diagrammatic tool to model concurrency and synchronization in distributed systems.
- ✓ Very similar to State Transition Diagrams.
- ✓ Used as a visual communication aid to model the system behaviour.
- ✓ Based on strong mathematical foundation.

Example: EFTPOS System (STD of an FSM)

(EFTPOS= Electronic Fund Transfer Point of Sale)



Initial

1 digit

d1

1 digit

d2

1 digit

d3

1 digit

d4

OK

OK

OK

OK

OK

OK pressed

Rejected

Reject

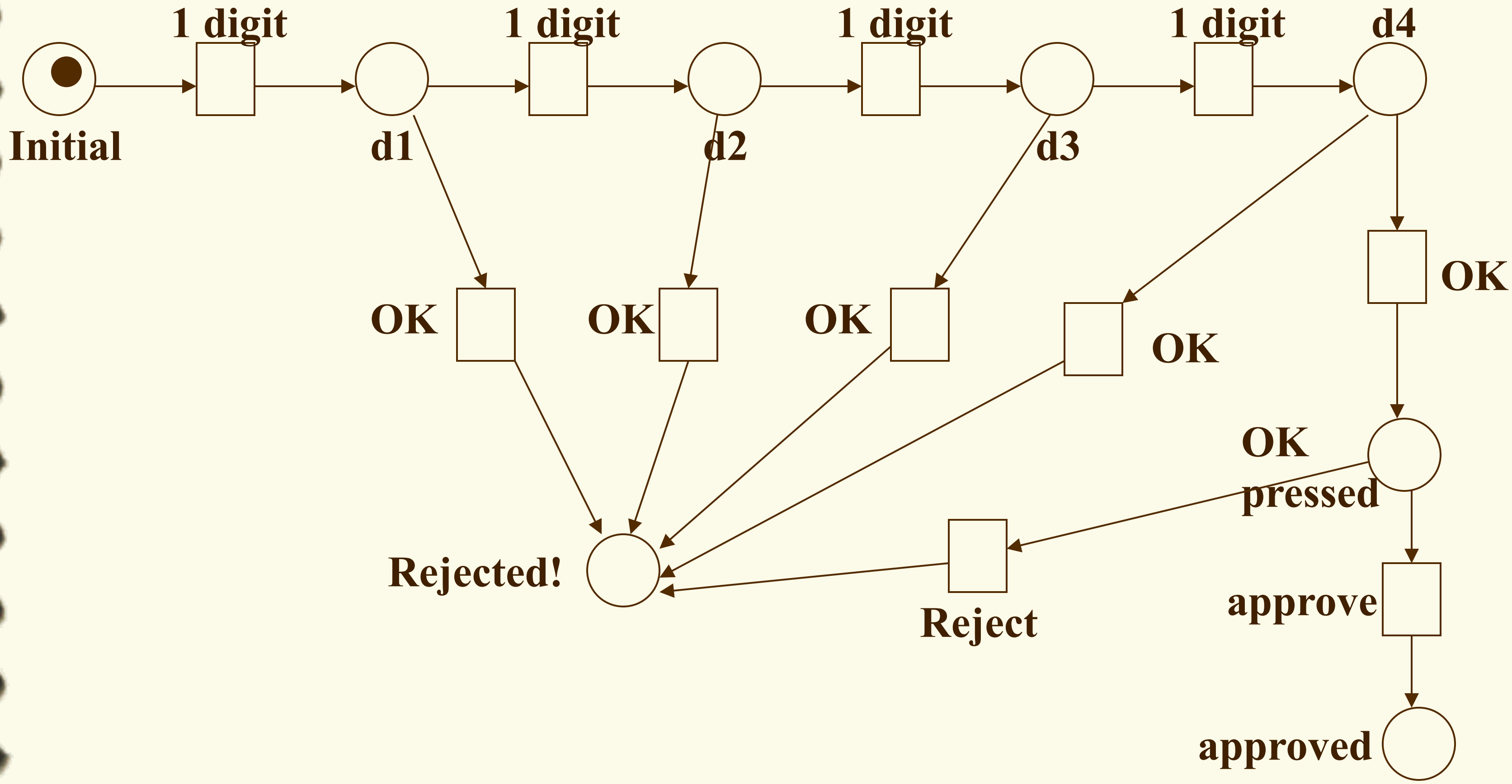
approve

Approved

🟧 Initial state

🟫 Final state

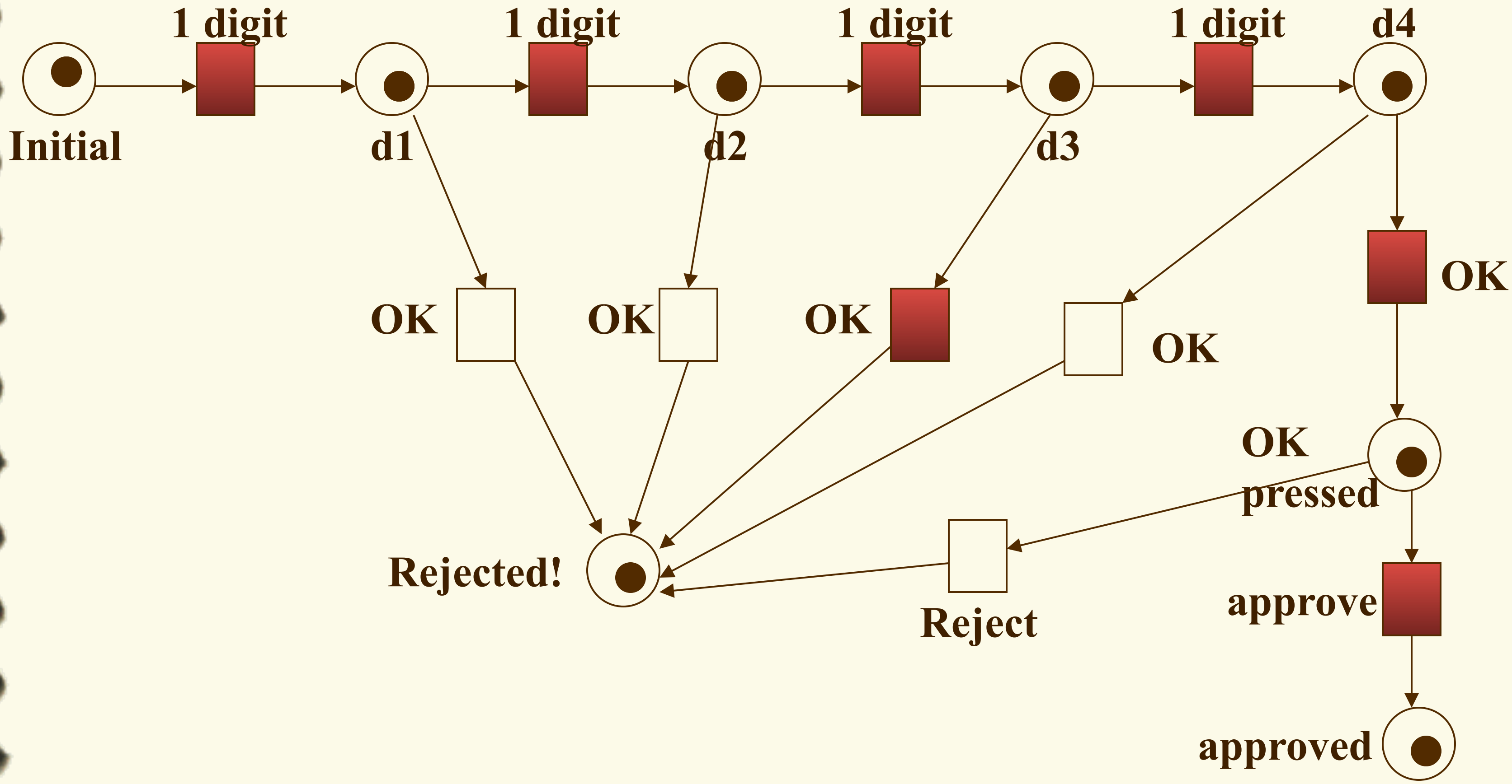
Example: EFTPOS System (A Petri net)



EFTPOS System

- ✓ Scenario 1: Normal
  - – Enters all 4 digits and press OK.
- ✓ Scenario 2: Exceptional
  - – Enters only 3 digits and press OK.

Example: EFTPOS System (Token Games)



A Petri Net Specification ...

✓ consists of three types of components: *places* (circles), *transitions* (rectangles) and *arcs* (arrows):

- Places represent possible states of the system;
- Transitions are events or actions which cause the change of state; And
- Every arc simply connects a place with a transition or a transition with a place.

A 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.
- ✓ The firing is subject to the input conditions, denoted by token availability.

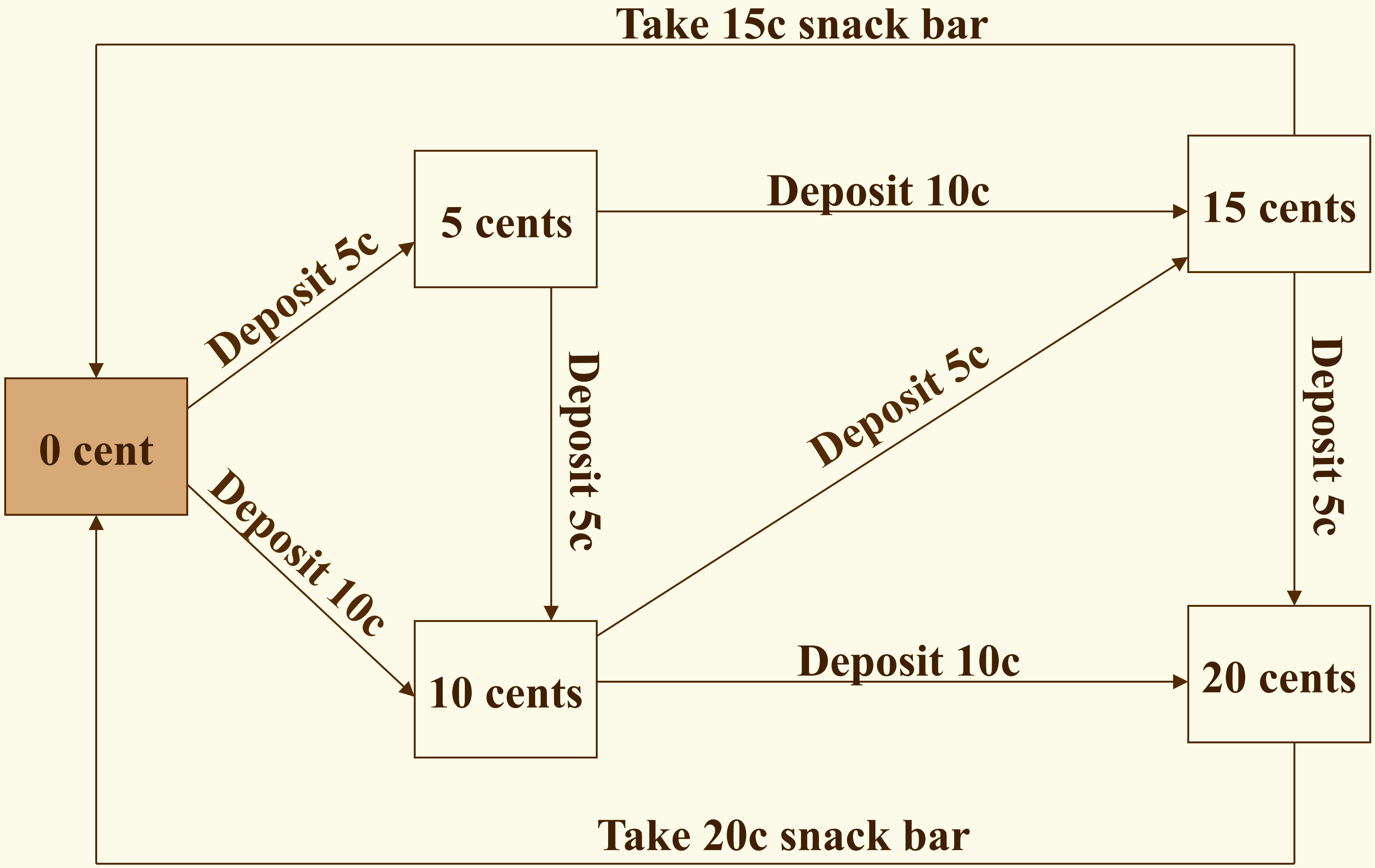
A Change of State

- ✓ 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.
- ✓ Note that the EFTPOS example is a Petri net representation of a finite state machine (FSM).

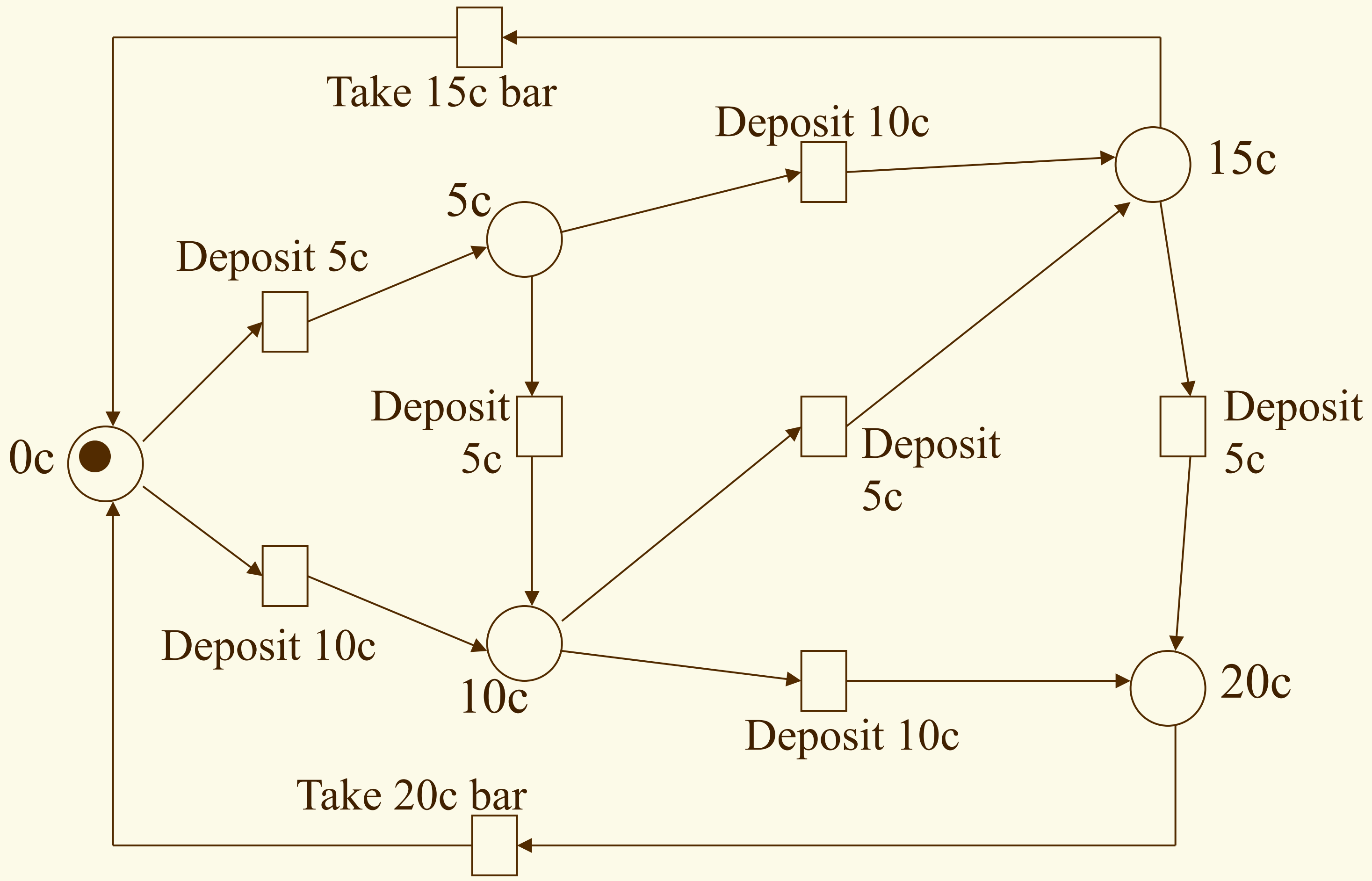
Example: Vending Machine

- ✓ 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.

Example: Vending Machine (STD of an FSM)



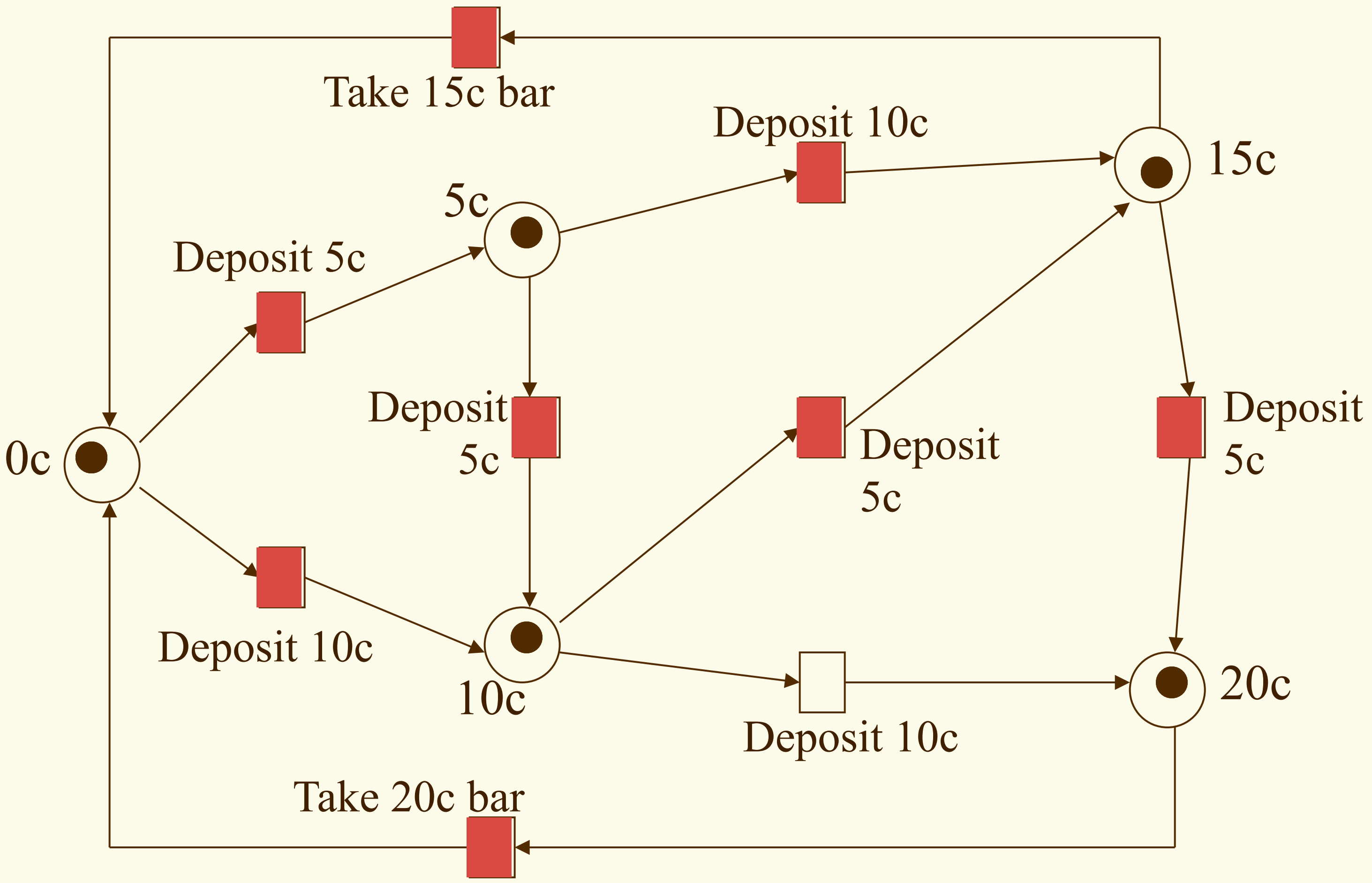
Example: Vending Machine (A Petri net)



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.

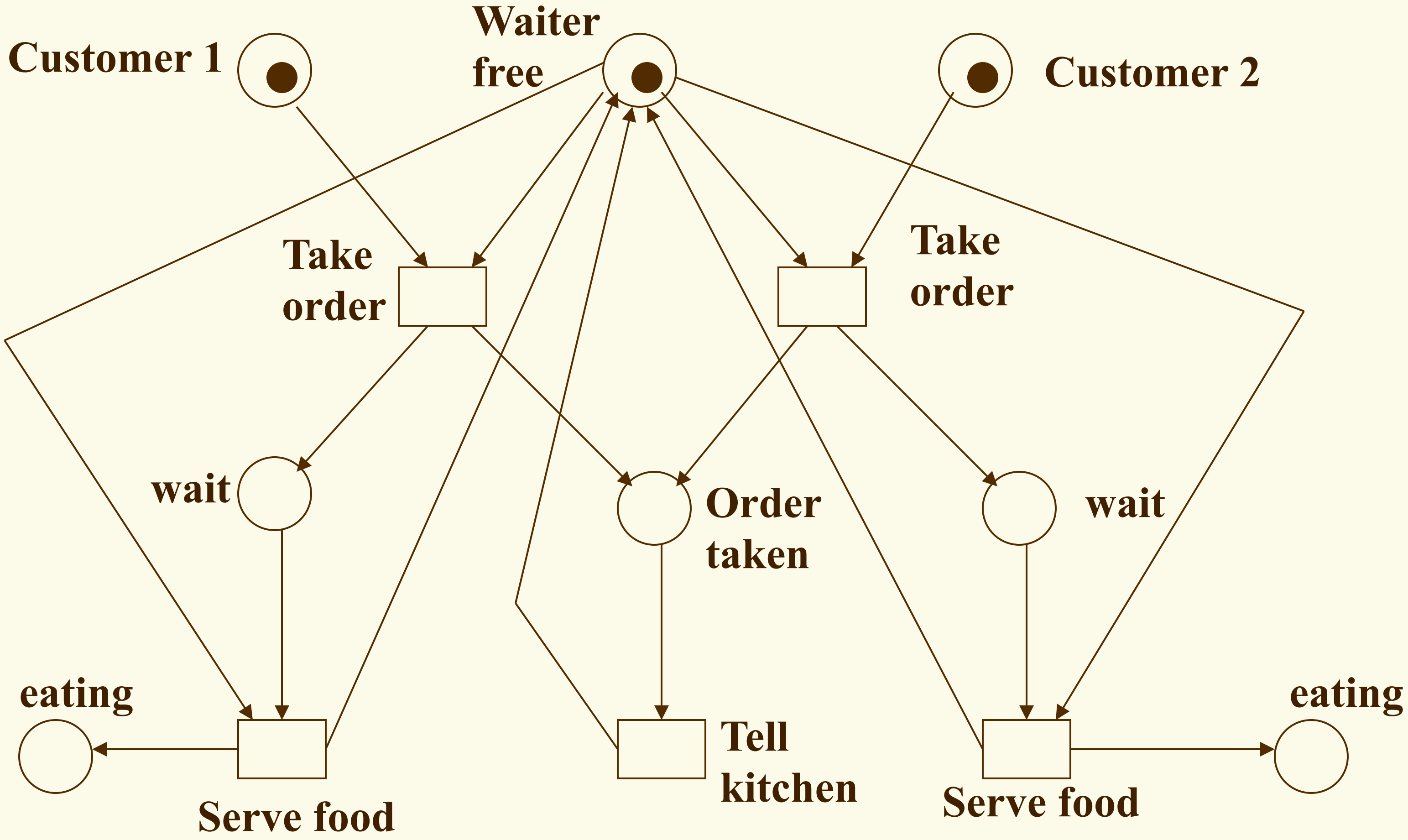
Example: Vending Machine (Token Games)



Multiple Local States

- In the real world, events happen at the same time.
- A system may have many local states to form a global state.
- There is a need to model concurrency and synchronization.

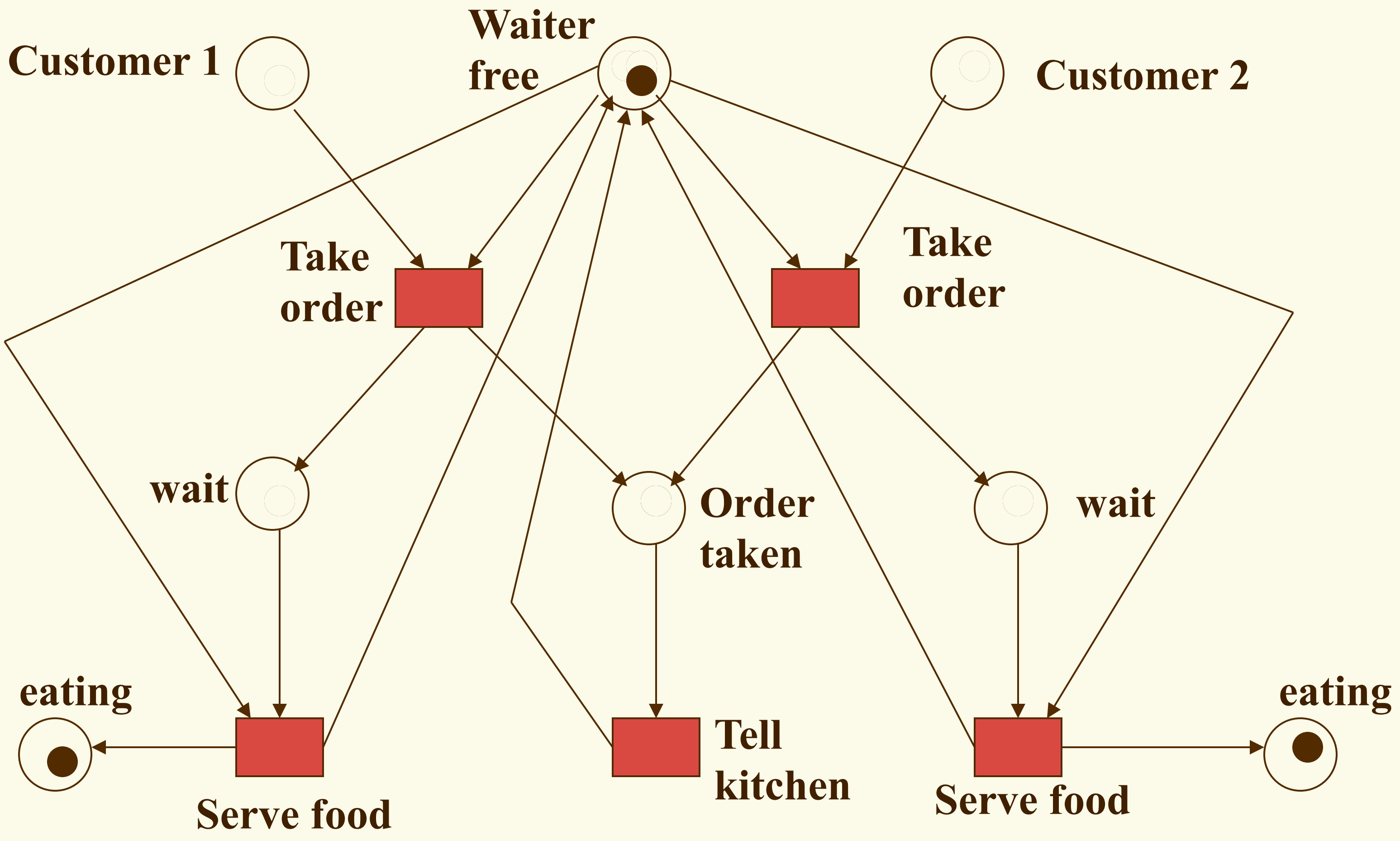
Example: In a Restaurant (A Petri Net)



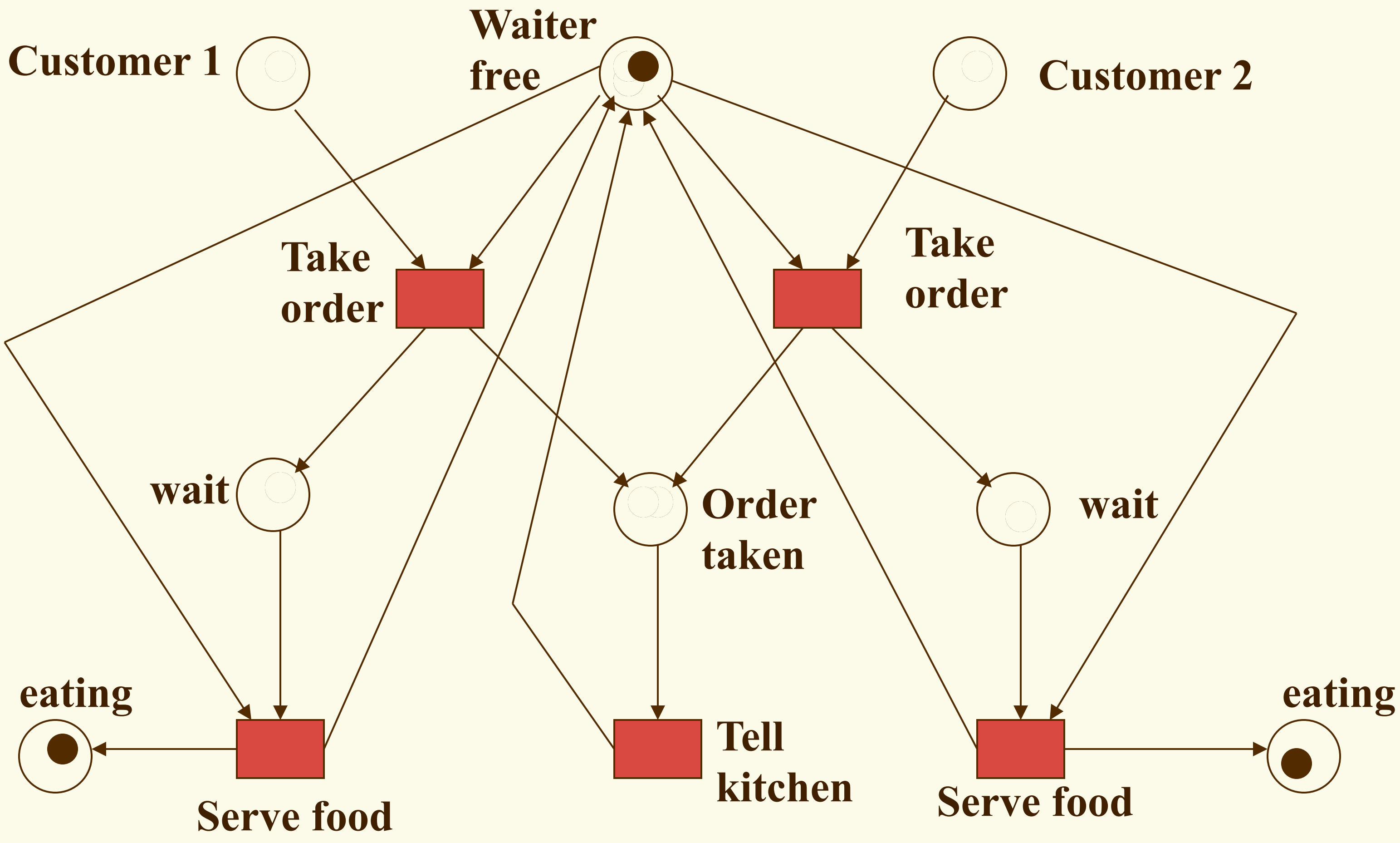
Example: In a Restaurant (Two Scenarios)

- ✓ 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)

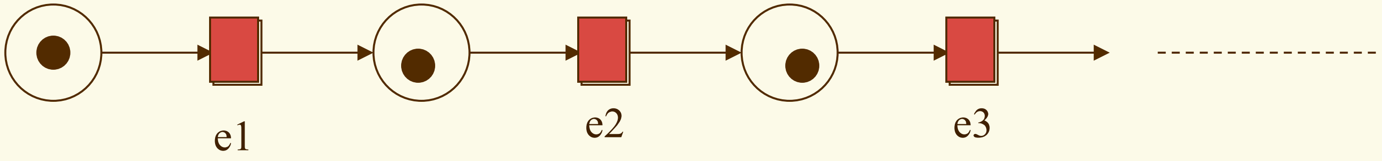


Example: In a Restaurant (Scenario 2)

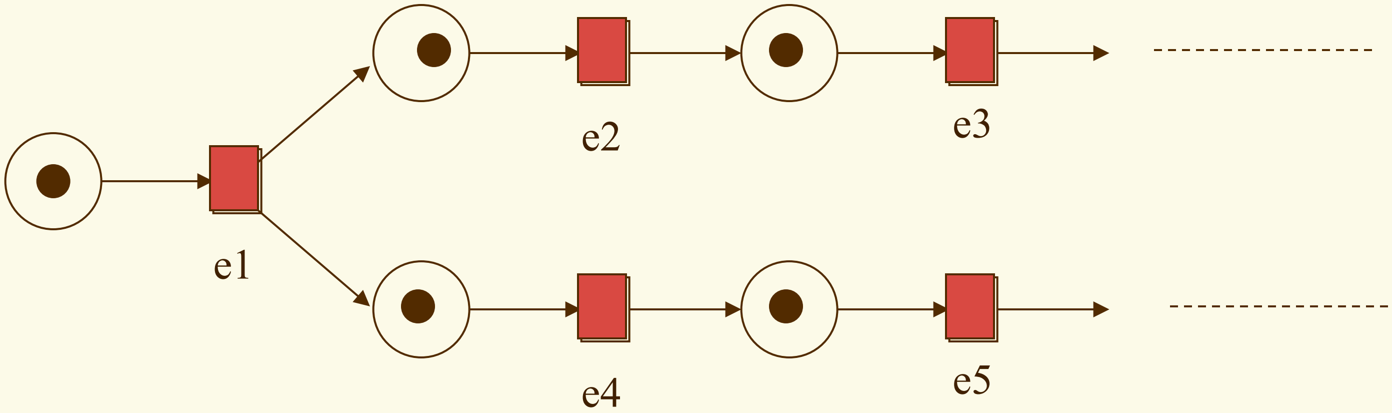


Net Structures

- ✓ A sequence of events/actions:

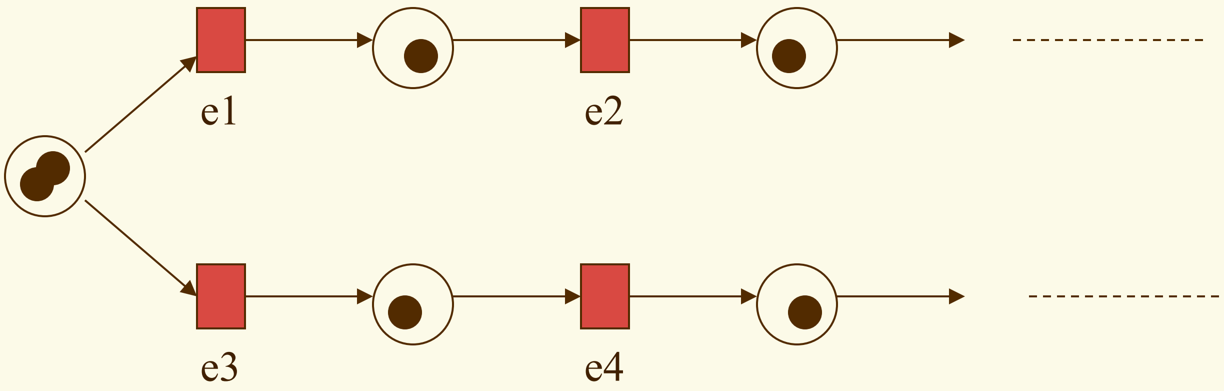


- ✓ Concurrent executions:



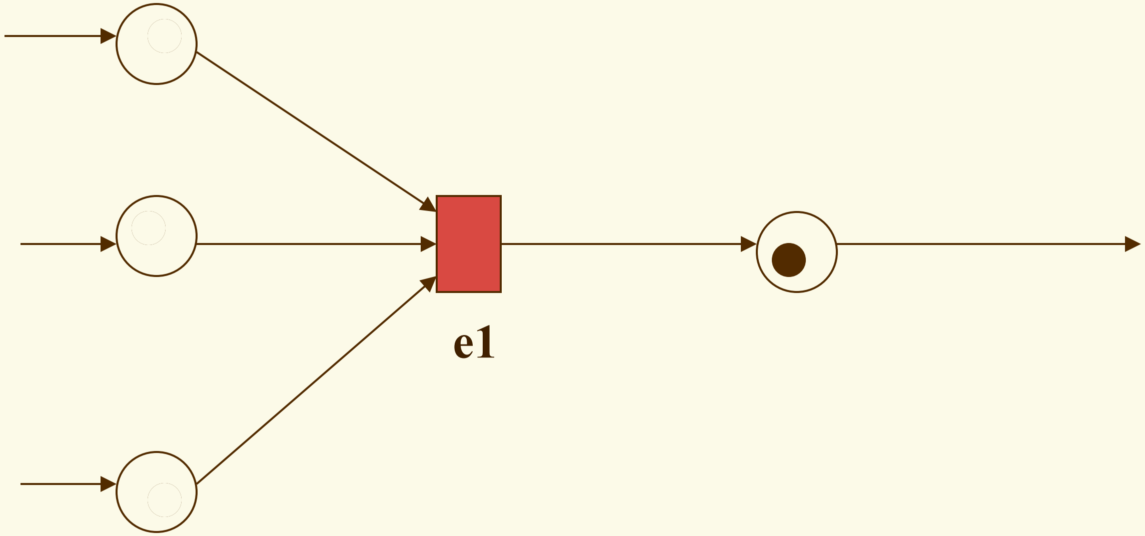
Net Structures

- ✓ Non-deterministic events - conflict, choice or decision: A choice of either e1, e2 … or e3, e4 ...



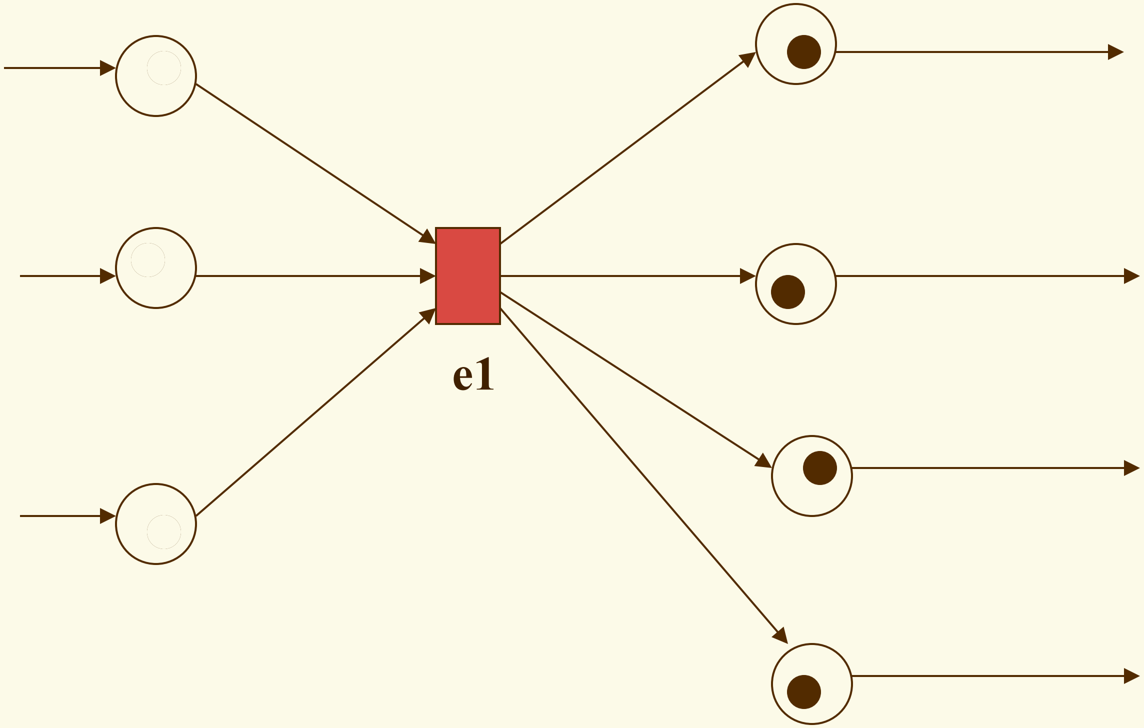
Net Structures

- Synchronization



Net Structures

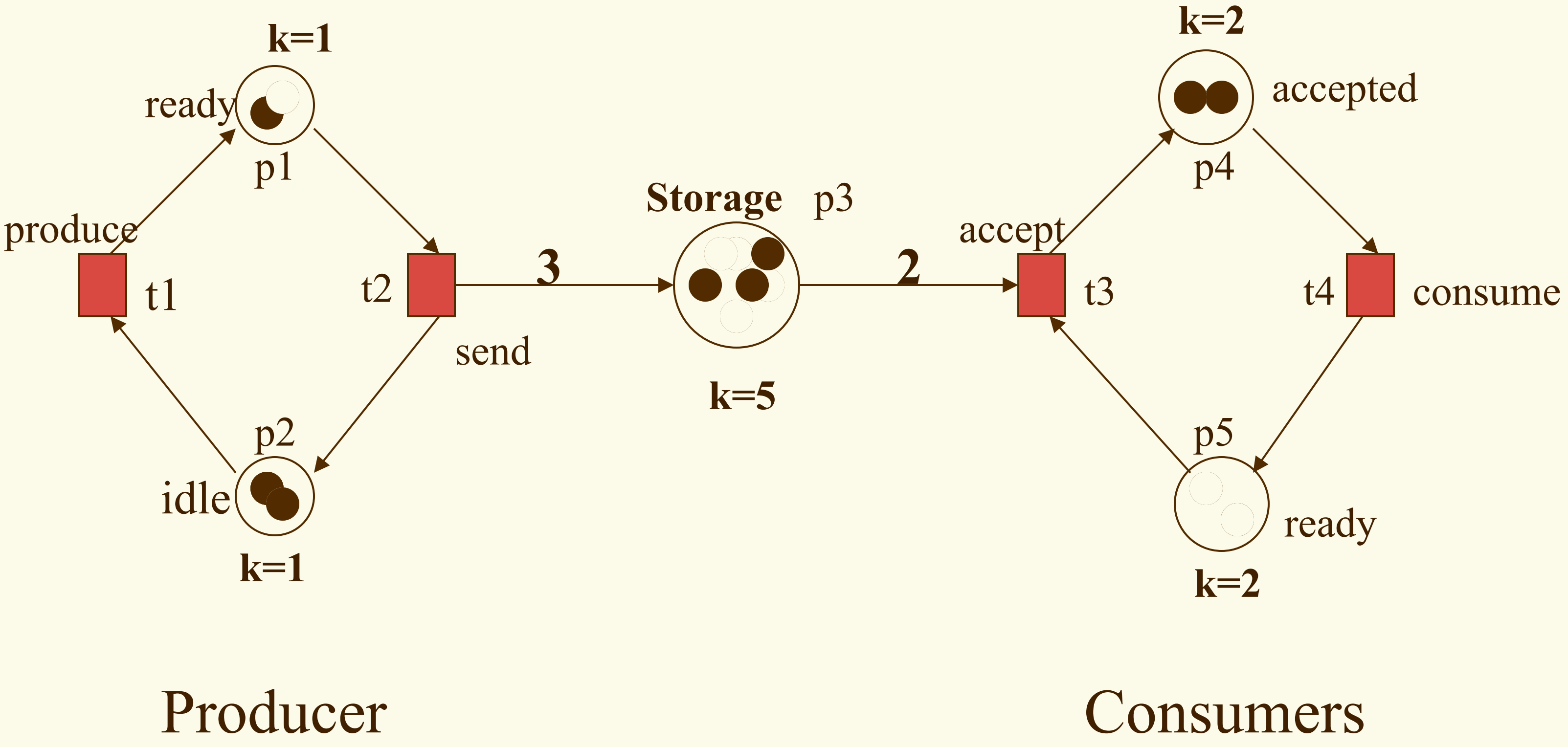
- ✓ Synchronization and Concurrency



Another Example

- A producer-consumer system, consist of one producer, two consumers and one storage buffer with the following conditions:
  - The storage buffer may contain at most 5 items;
  - The producer sends 3 items in each production;
  - At most one consumer is able to access the storage buffer at one time;
  - Each consumer removes two items when accessing the storage buffer

A Producer-Consumer System



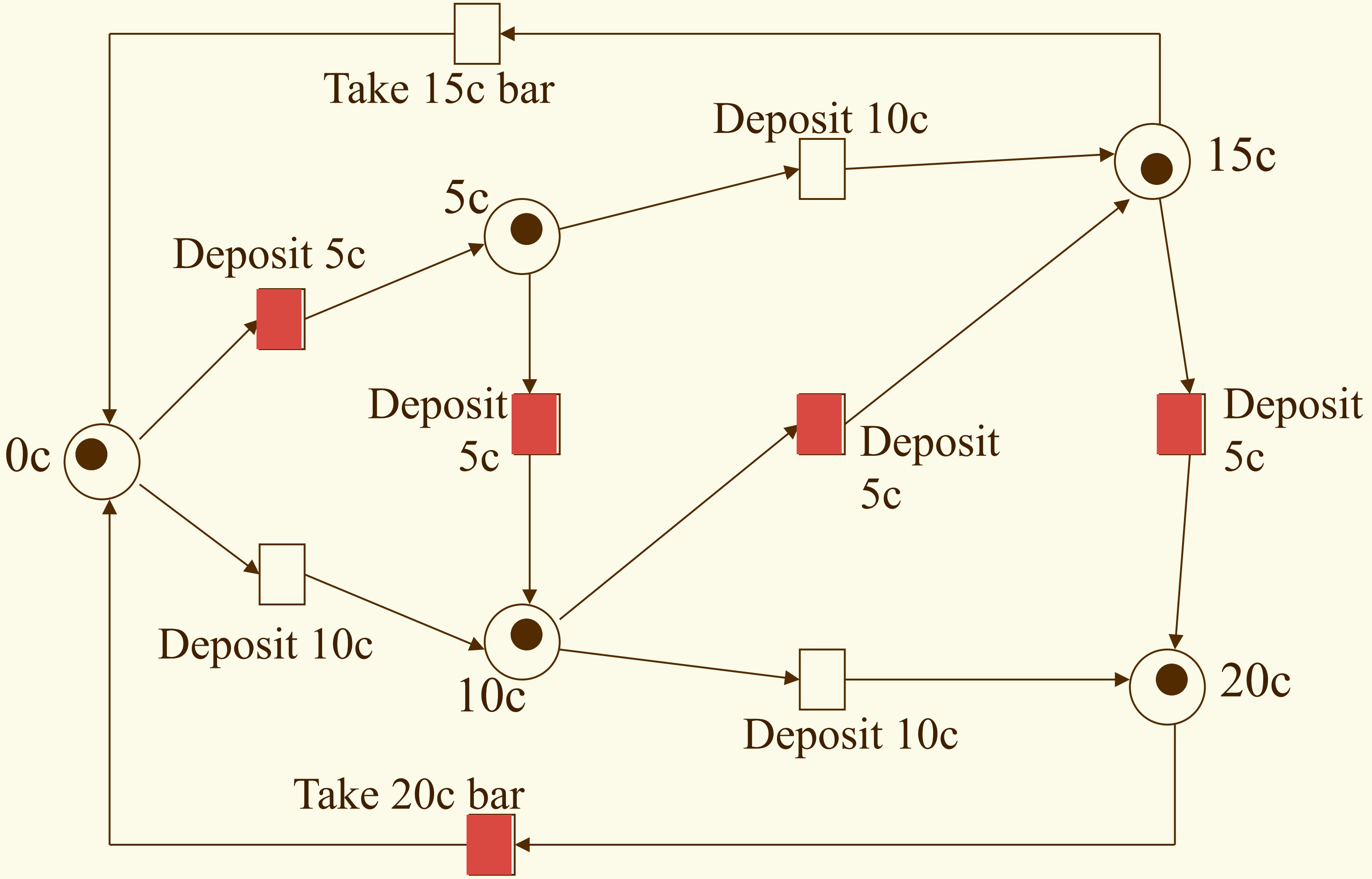
A Producer-Consumer Example

- In this Petri net, every place has a *capacity* and every arc has a *weight*.
- This allows multiple tokens to reside in a place to model more complex behaviour.

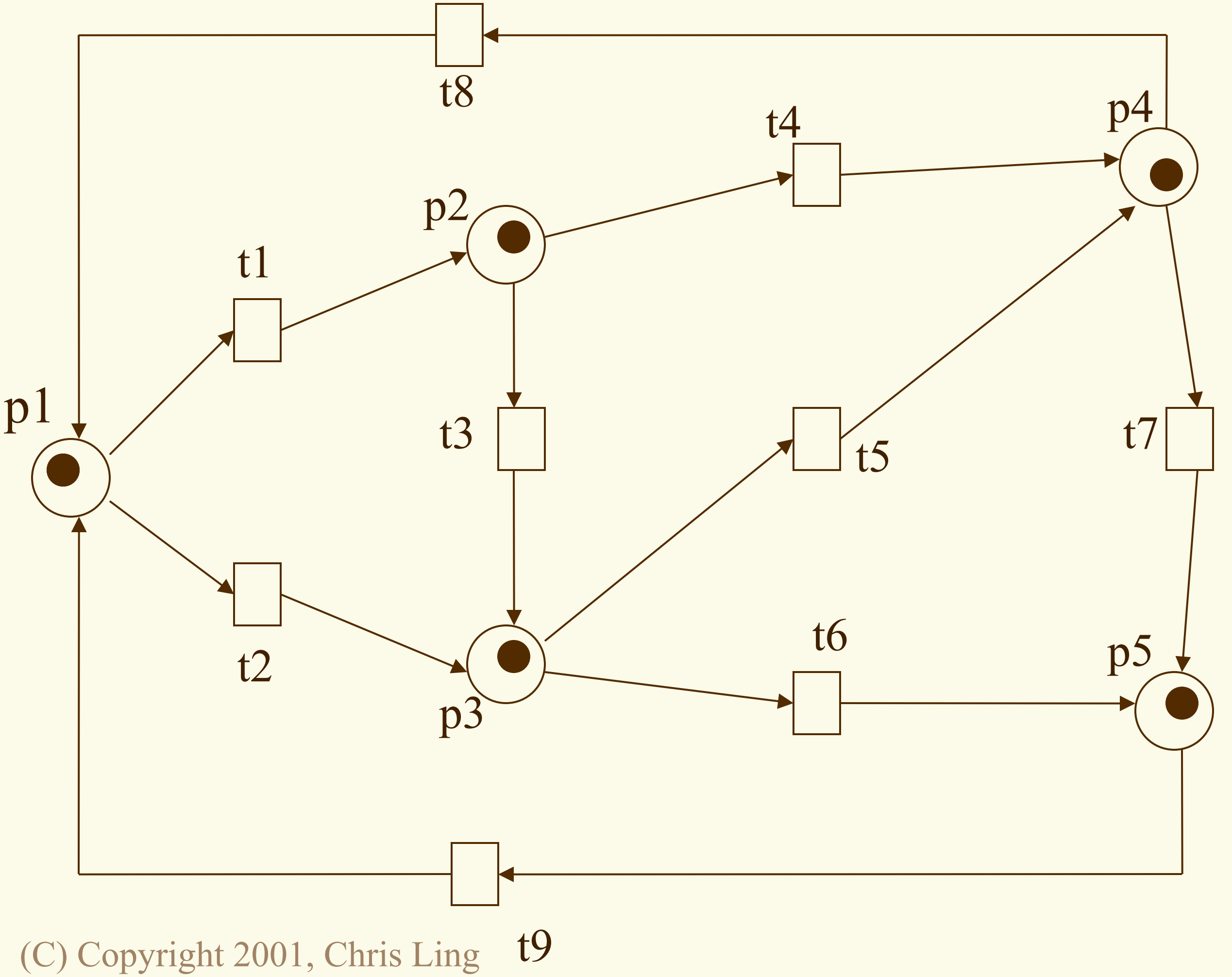
Behavioural Properties

- Reachability
  - “Can we reach one particular state from another?”
- Boundedness
  - “Will a storage place overflow?”
- Liveness
  - “Will the system die in a particular state?”

Recalling the Vending Machine (Token Game)



A *marking* is a state ...



M0 = (1,0,0,0,0)

M1 = (0,1,0,0,0)

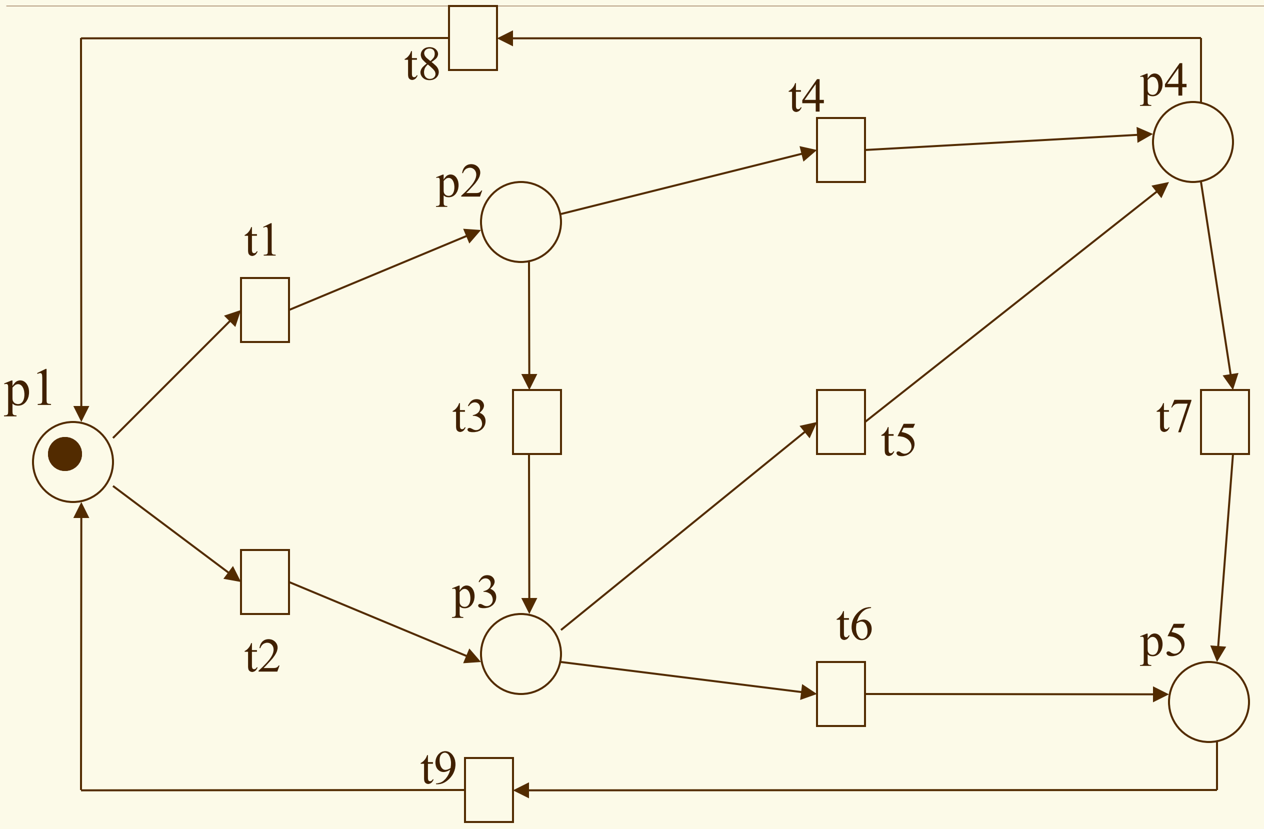
M2 = (0,0,1,0,0)

M3 = (0,0,0,1,0)

M4 = (0,0,0,0,1)

Initial marking:M0

Reachability



M0 = (1,0,0,0,0)

M1 = (0,1,0,0,0)

M2 = (0,0,1,0,0)

M3 = (0,0,0,1,0)

M4 = (0,0,0,0,1)

Initial marking:M0

$$M0 \xrightarrow{t1} M1 \xrightarrow{t3} M2 \xrightarrow{t5} M3 \xrightarrow{t8} M0 \xrightarrow{t2} M2 \xrightarrow{t6} M4$$

Reachability

A firing or occurrence sequence:

$$M0 \xrightarrow{t1} M1 \xrightarrow{t3} M2 \xrightarrow{t5} M3 \xrightarrow{t8} M0 \xrightarrow{t2} M2 \xrightarrow{t6} M4$$

- "M2 is *reachable* from M1 and M4 is *reachable* from M0."
- In fact, in the vending machine example, all markings are reachable from every marking.

Boundedness

- A Petri net is said to be *k-bounded* or simply *bounded* if the number of tokens in each place does not exceed a finite number *k* for any marking reachable from M0.
- The Petri net for vending machine is 1 -bounded.
- A 1-bounded Petri net is also *safe*.

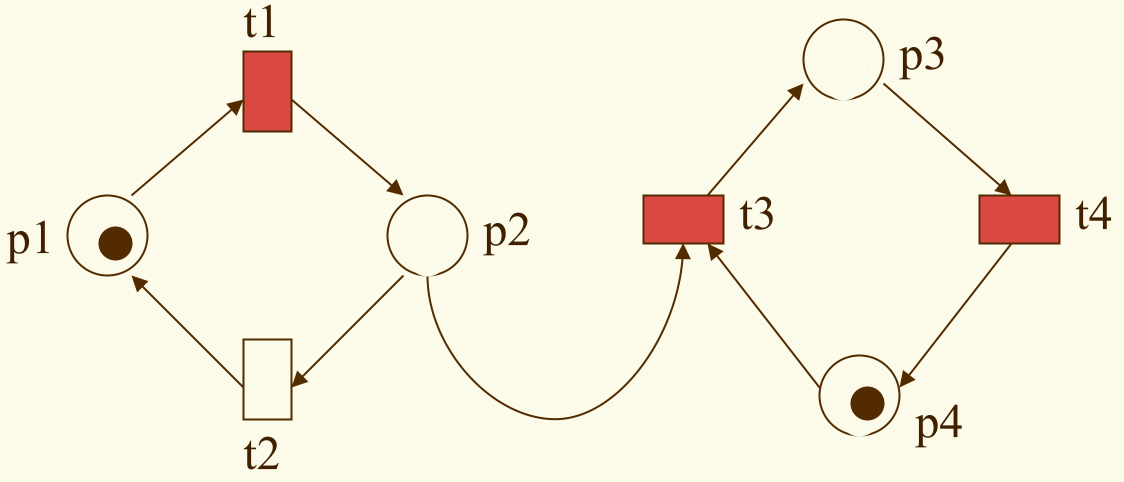
Liveness

- A Petri net with initial marking M0 is *live* if, no matter what marking has been reached from M0, it is possible to ultimately fire *any* transition by progressing through some further firing sequence.
- A live Petri net guarantees *deadlock-free* operation, no matter what firing sequence is chosen.

Liveness

- The vending machine is live and the producer-consumer system is also live.
- A transition is *dead* if it can never be fired in any firing sequence.

An Example



M0 = (1,0,0,1)

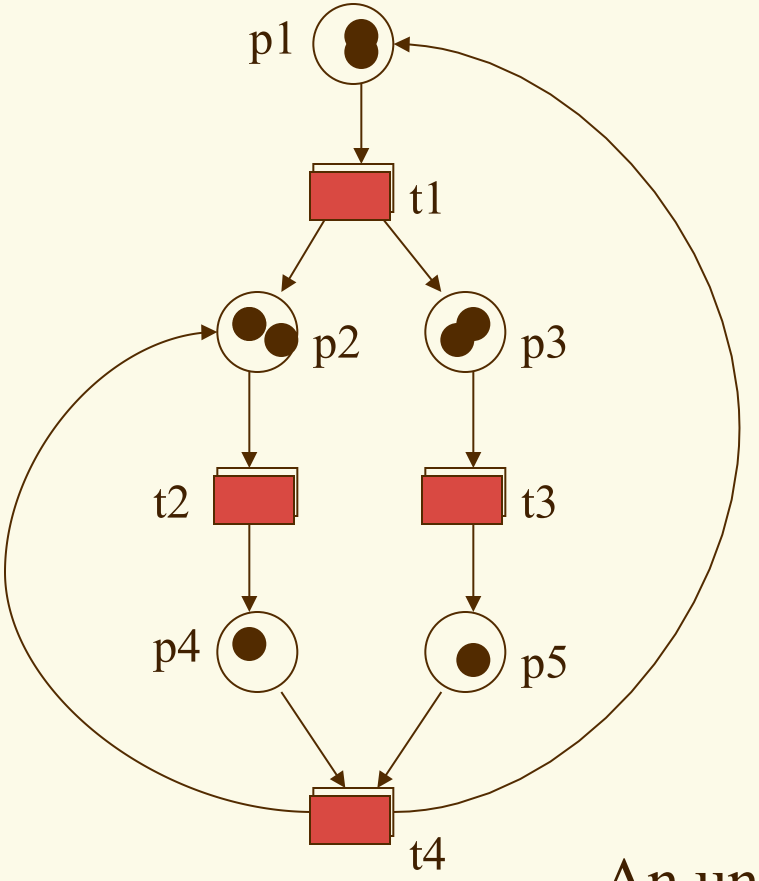
M1 = (0,1,0,1)

M2 = (0,0,1,0)

M3 = (0,0,0,1)

A bounded but non-live Petri net

Another Example



M0 = (1, 0, 0, 0, 0)

M1 = (0, 1, 1, 0, 0)

M2 = (0, 0, 0, 1, 1)

M3 = (1, 1, 0, 0, 0)

M4 = (0, 2, 1, 0, 0)

An unbounded but live Petri net

Analysis Methods

- Reachability Analysis:
  - Reachability or coverability tree.
  - State explosion problem.
- Incidence Matrix and State Equations.
- Structural Analysis
  - Based on net structures.

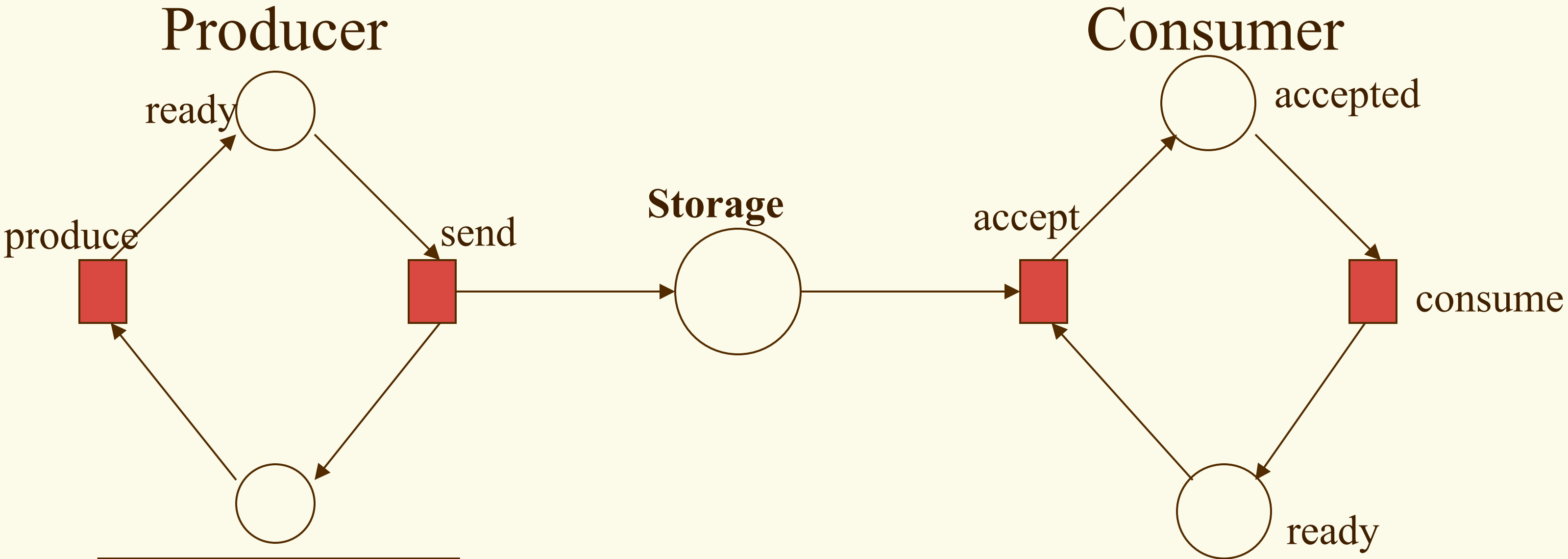
Other Types of Petri Nets

- High-level Petri nets
  - Tokens have “colours”, holding complex information.
- Timed Petri nets
  - Time delays associated with transitions and/or places.
  - Fixed delays or interval delays.
  - Stochastic Petri nets: exponentially distributed random variables as delays.

Other Types of Petri Nets

- Object-Oriented Petri nets
  - Tokens are instances of classes, moving from one place to another, calling methods and changing attributes.
  - Net structure models the inner behaviour of objects.
  - The purpose is to use object-oriented constructs to structure and build the system.

An O-O Petri Net



Producer

data: ITEM

ITEM produce( )

void send(ITEM)

Consumer

data: ITEM

ITEM accept( )

void consume(ITEM)

Petri Net References

- Murata, T. (1989, April). Petri nets: properties, analysis and applications. Proceedings of the IEEE, 77(4), 541-80.
- Peterson, J.L. (1981). Petri Net Theory and the Modeling of Systems. Prentice-Hall.
- Reisig, W and G. Rozenberg (eds) (1998). Lectures on Petri Nets 1: Basic Models. Springer-Verlag.
- The World of Petri nets:

  http://www.daimi.au.dk/PetriNets/