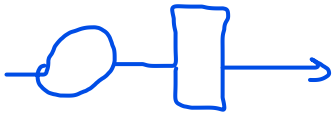
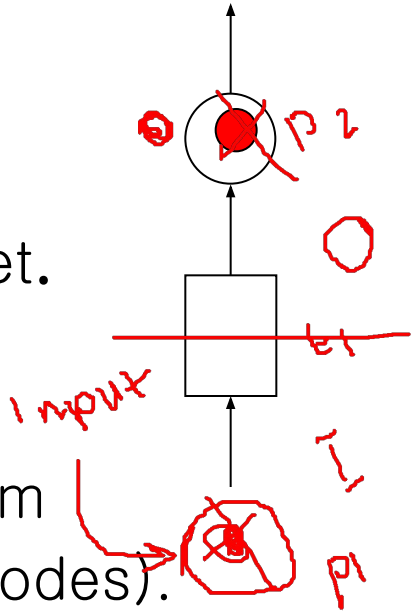
Petri Nets

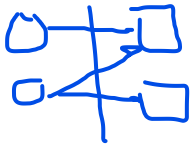
PTF

Introduction

- Introduced in 1962 by Carl Adam Petri in his PhD thesis.
- Focus on modeling causal dependencies;
- no global synchronization assumed (message passing only).
- Key elements:
  - **Conditions**
    Either met or not met.
  - **Events**
    May take place if certain conditions are met.
  - **Flow relation**
    Relates conditions and events.
- Conditions, events and the flow relation form
- a **bipartite graph** (graph with two kinds of nodes).







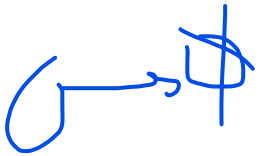
Definition of Petri Net

- **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)

Places symbolise states, conditions, or resources that need to be met/be available before an action can be carried out.

the place to be active, it must contain a token.

Transitions symbolise actions.



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

filling an empty place with a token to satisfy its action.

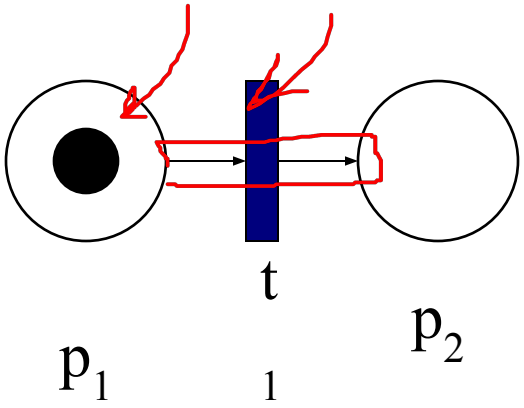
Applications of Petri Net

- 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

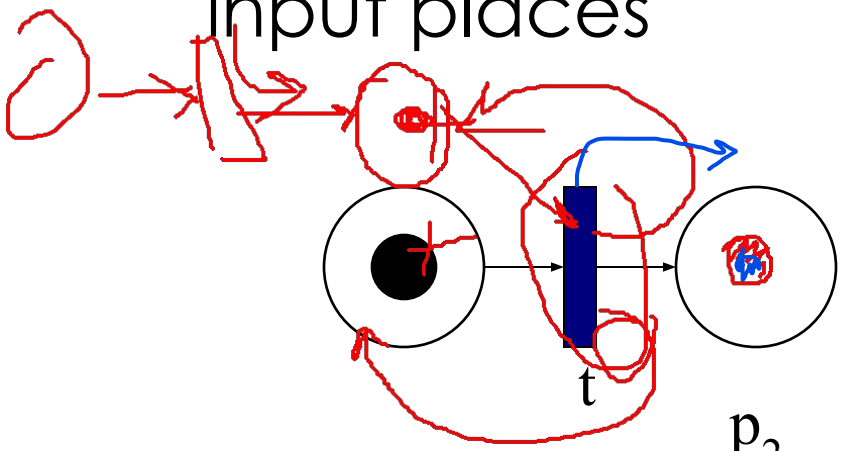
Basics of Petri Nets

- 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.

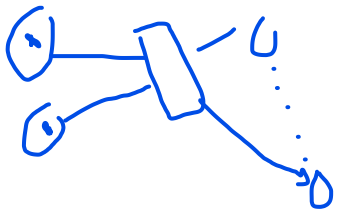


Basics of Petri Nets -continued

- 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



transitions can fire if it has enough tokens,
after it fires, it fills all the places that depend on it.
and it consumes one token from each place.

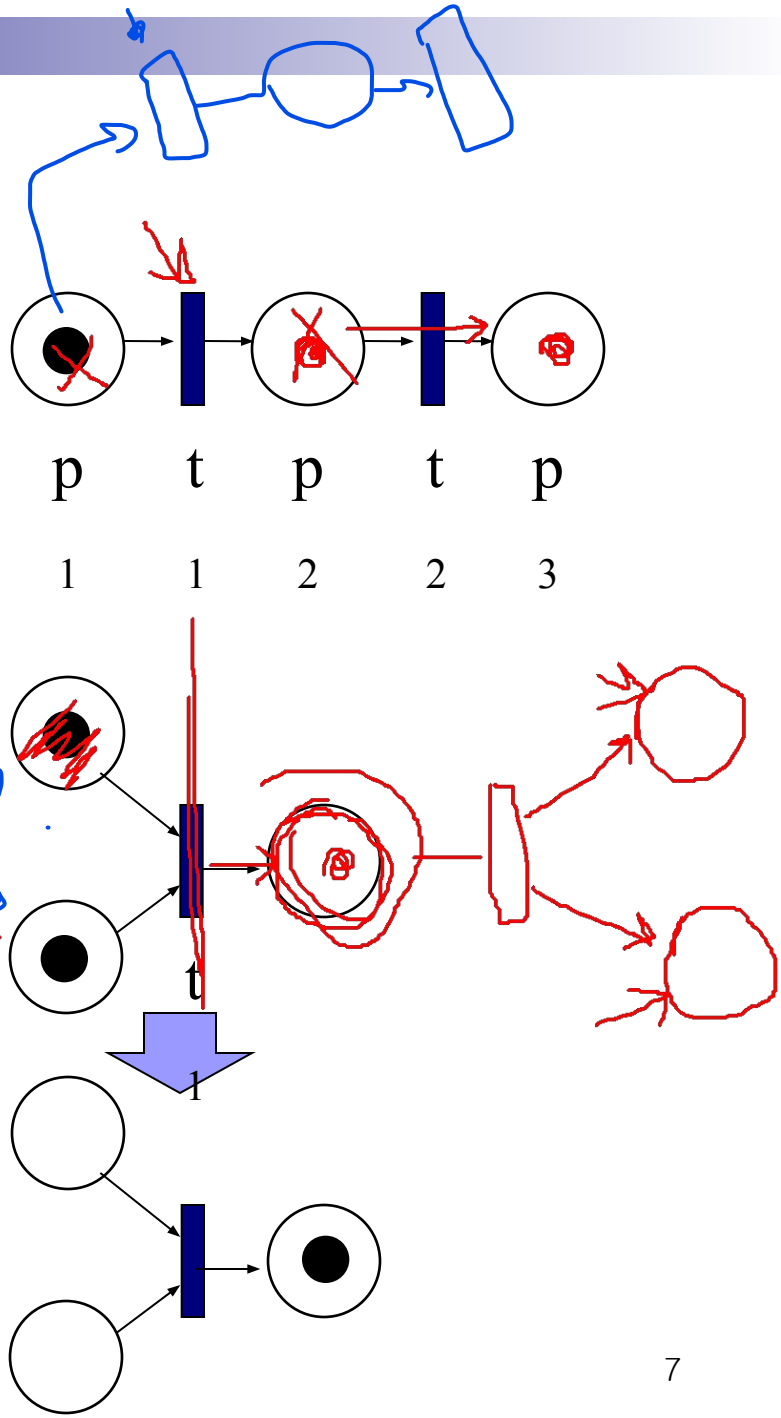


state transition of form (1, 0) □ (0, 1)

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.

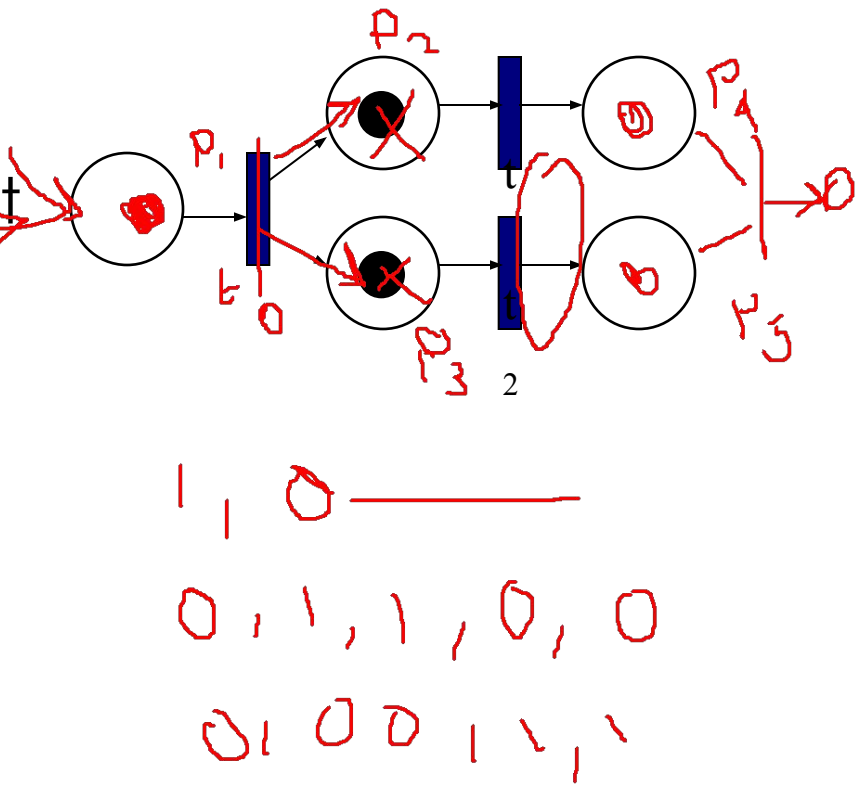


Properties of Petri Nets –continued

here you have one transition resulting into two places.

- Concurrency

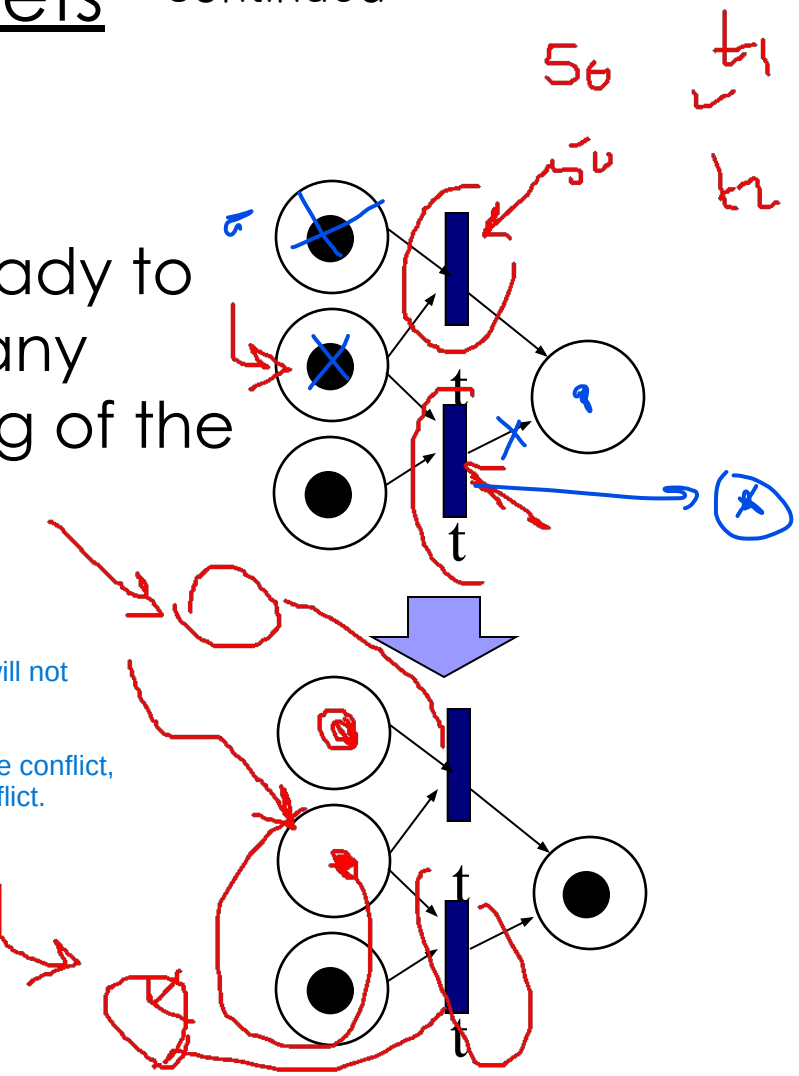
  t_1 and t_2 are concurrent.
  - with this property, Petri net is able to model systems of distributed control with multiple processes executing concurrently in time.



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.



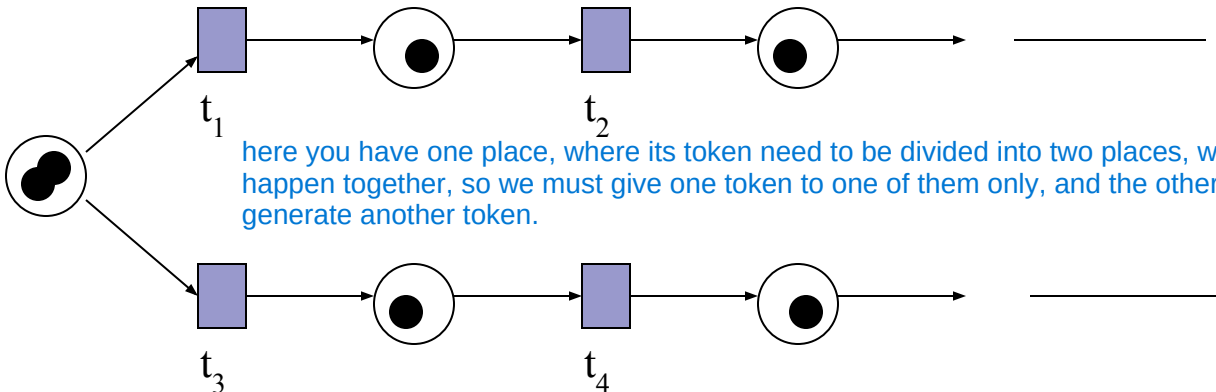
after consuming the two upper tokens, the lower transition will not have enough tokens to be fired.

if they came again as three, it depends on how you serve the conflict, so you may use FIFO, or RR, or alternation to solve the conflict.

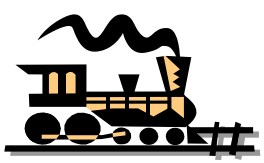
Properties of Petri Nets –continued

- 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

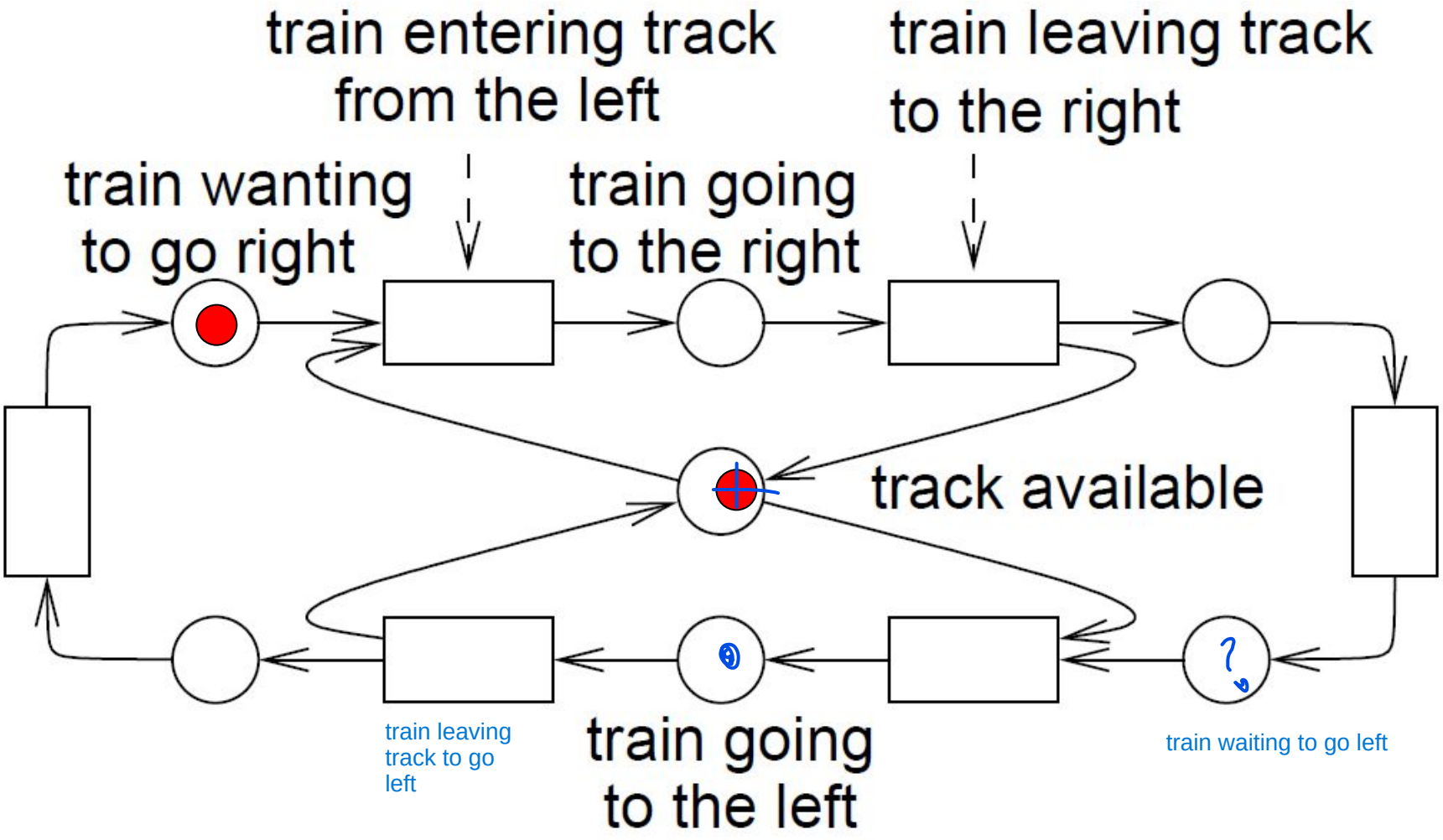


Example:
Synchronization at single track rail segment



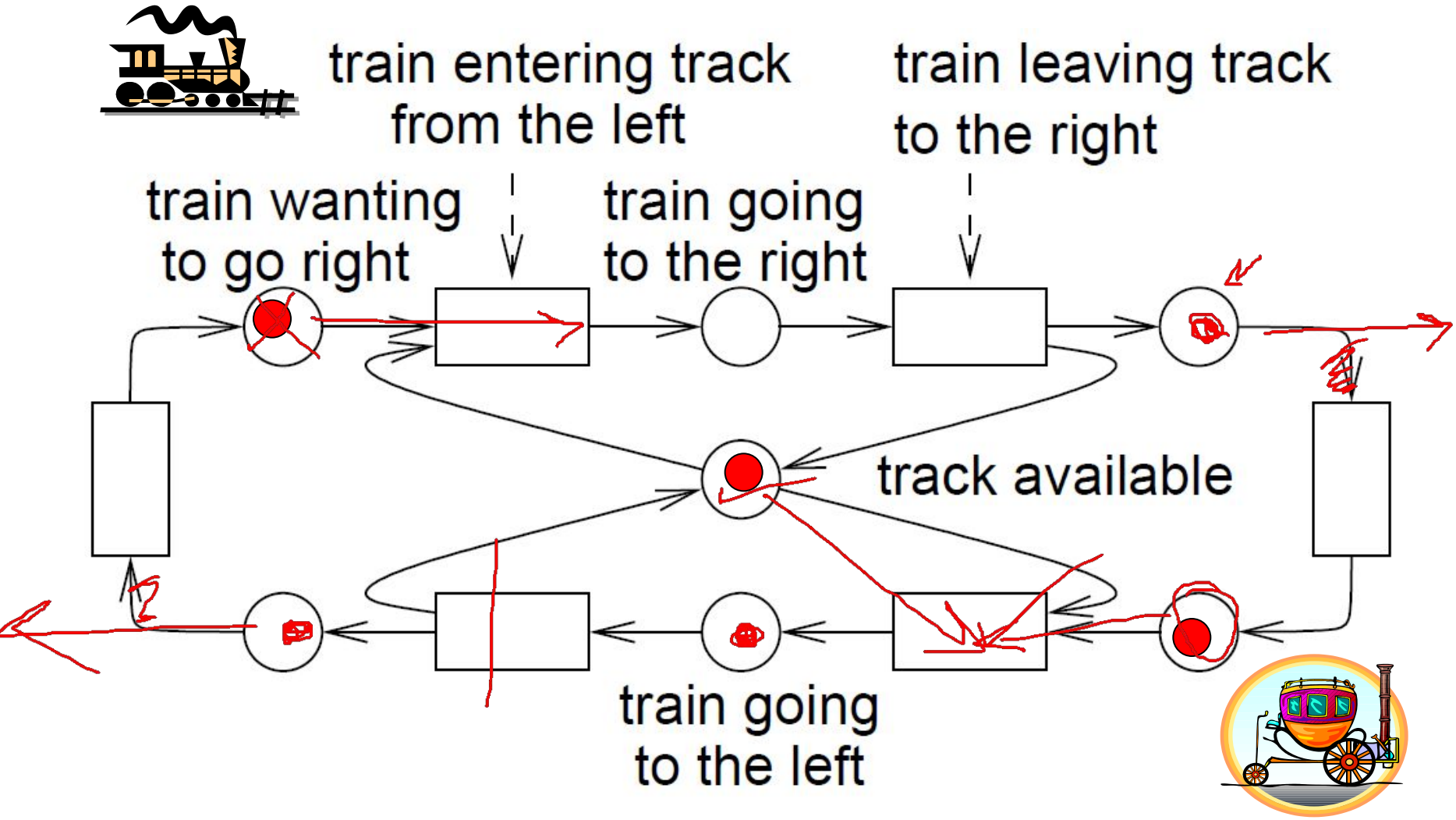
- "Precondi tions"

Playing the “token game“



use normal view

Conflict for resource “track“

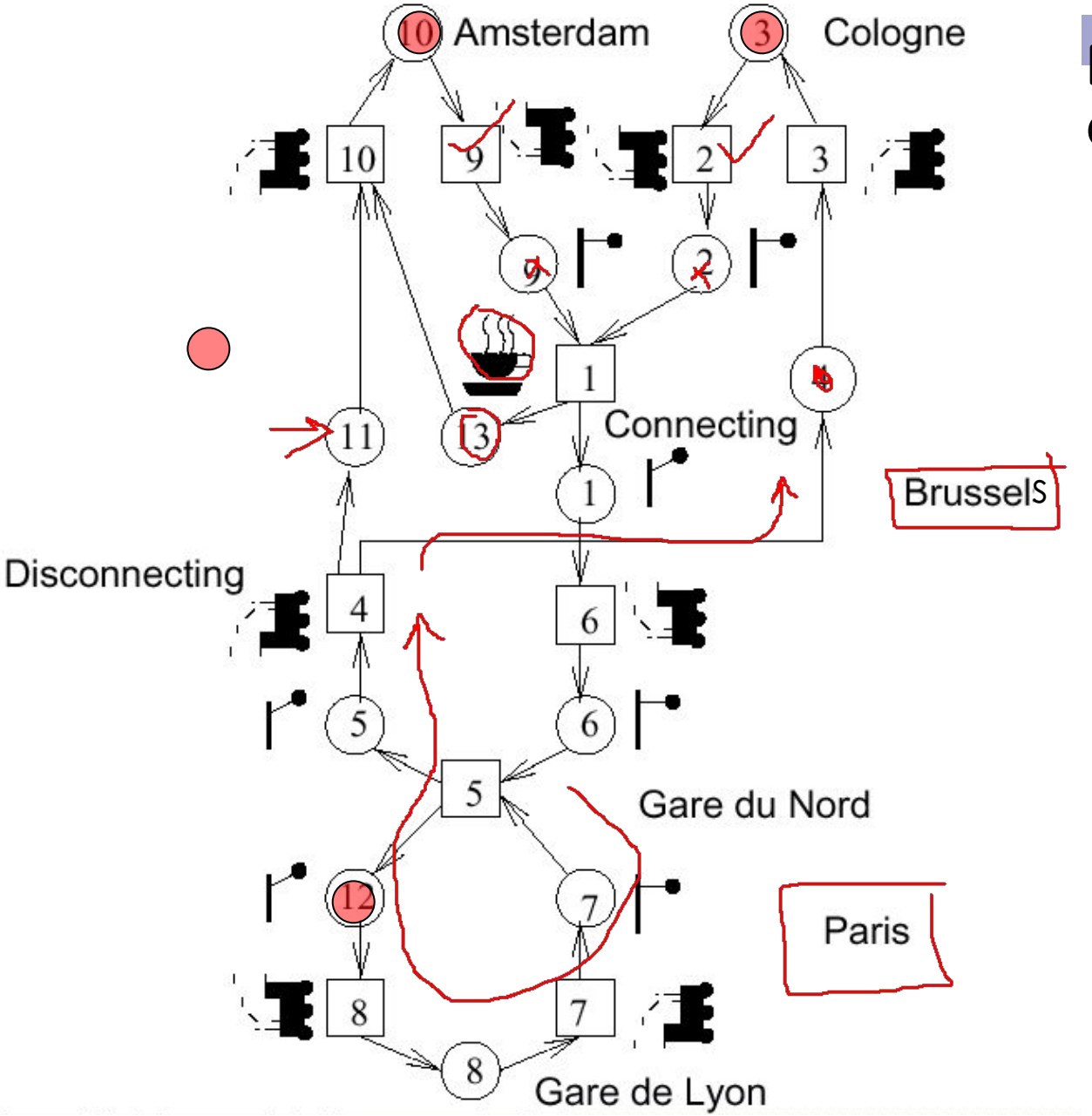


More complex example (1)

- Thalys trains between Cologne, Amsterdam, Brussels and Paris.

[http://www.thalys.com/be/en]

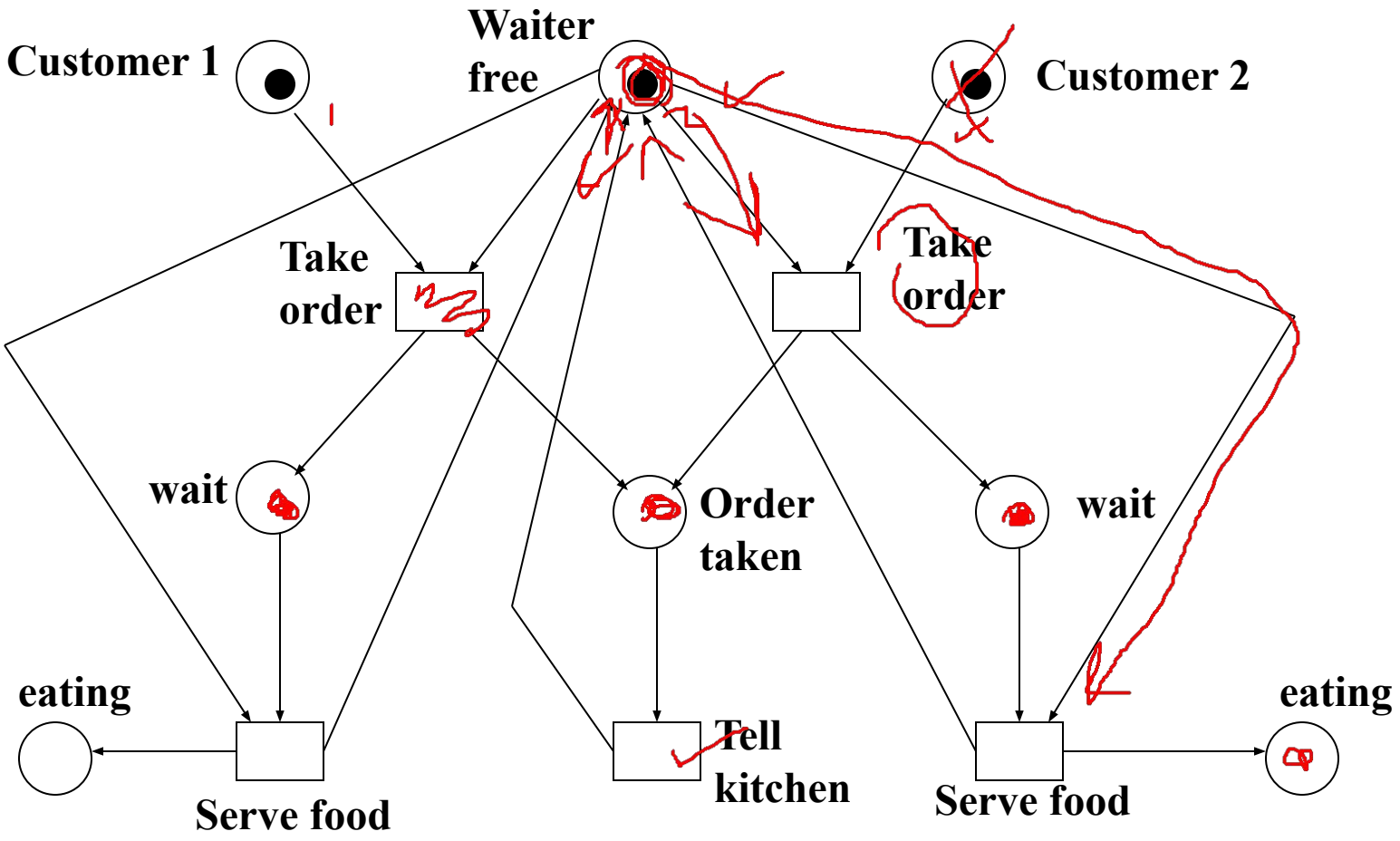
More complex example (2)



- Slightly simplified: Synchronization at Brussels and Paris, using stations “Gare du Nord” and “Gare de Lyon” at Paris

use normal view mode!

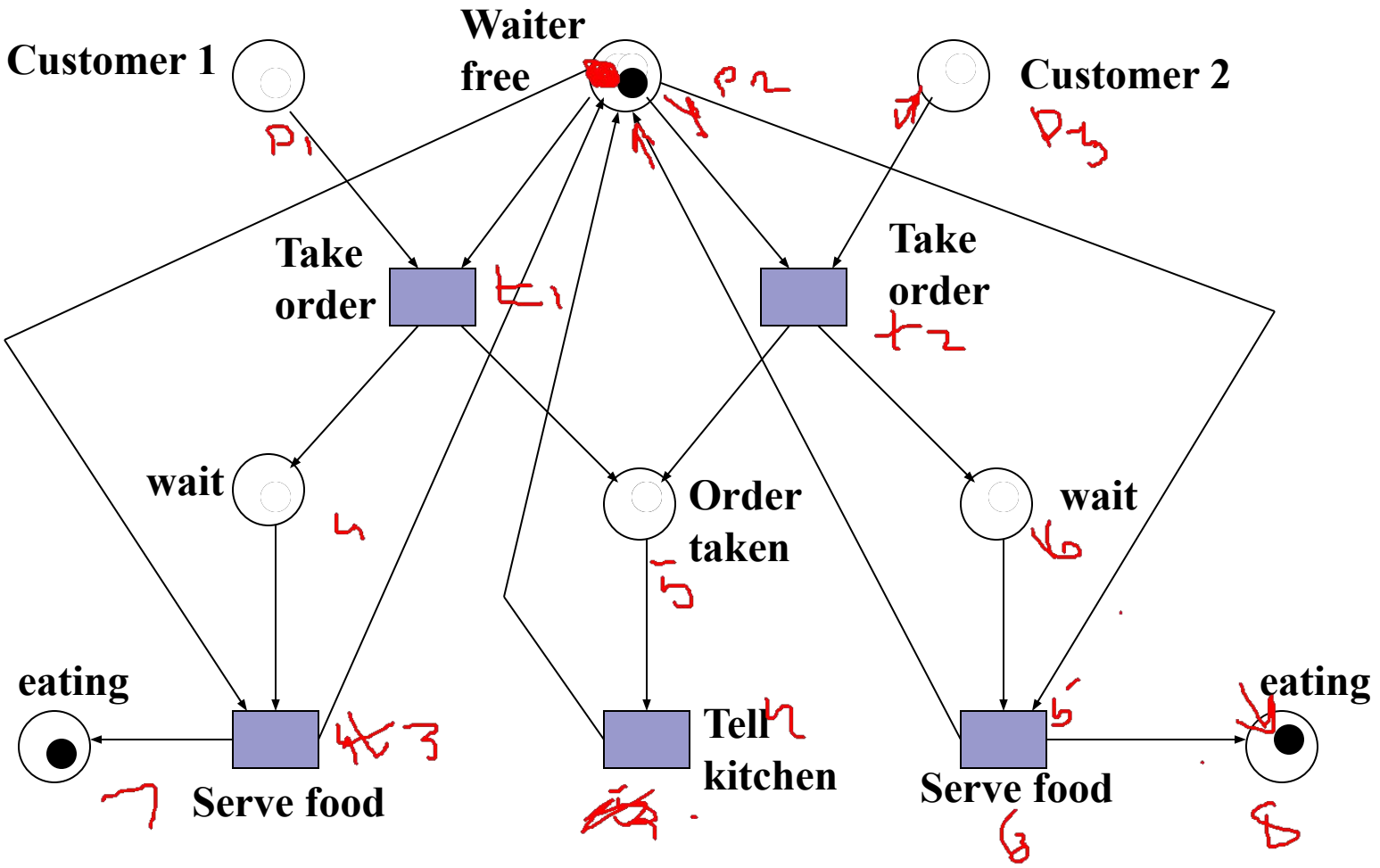
Example: In a Restaurant (A Petri Net)



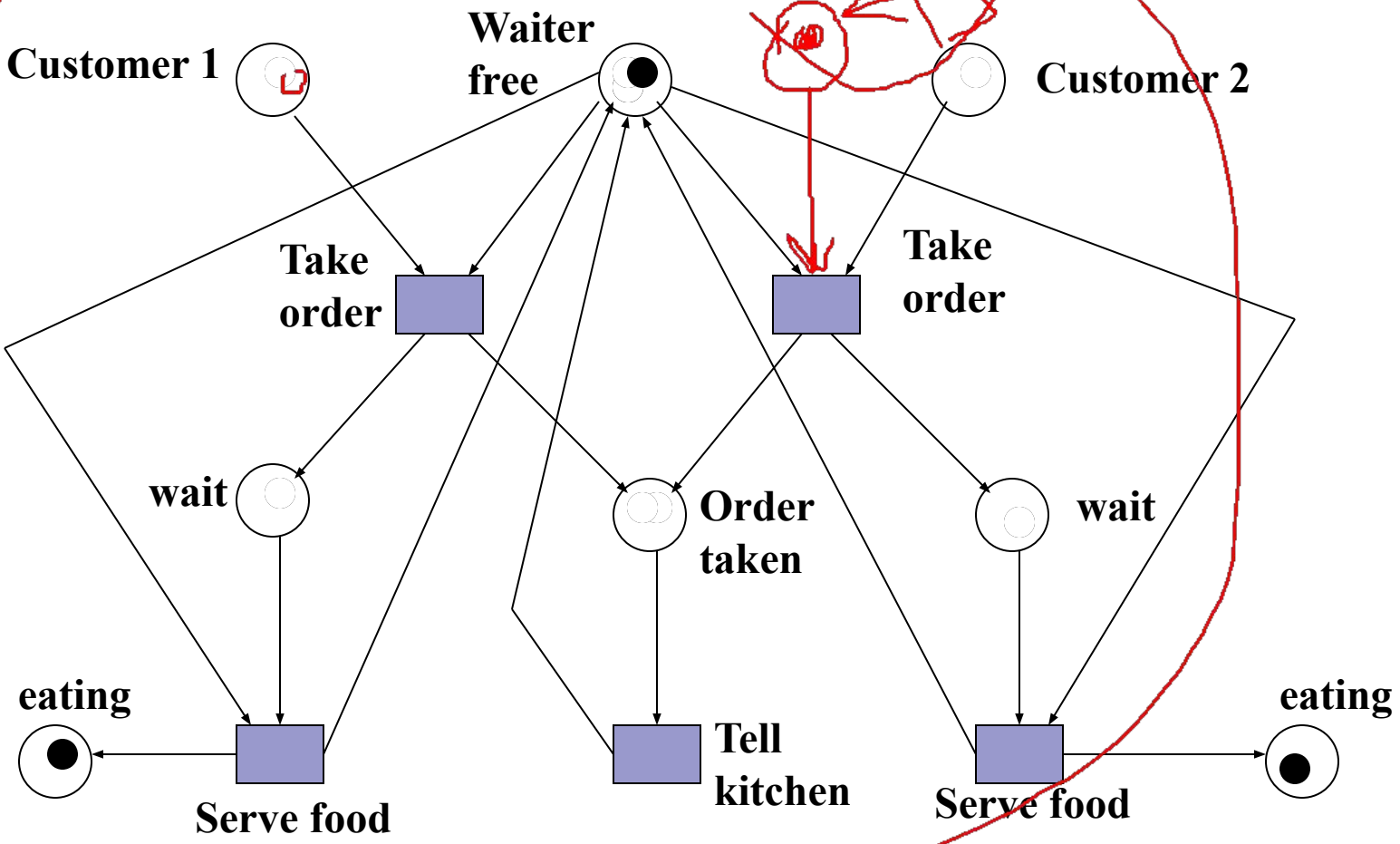
Example: In a Restaurant (Two Scenarios)

- 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.

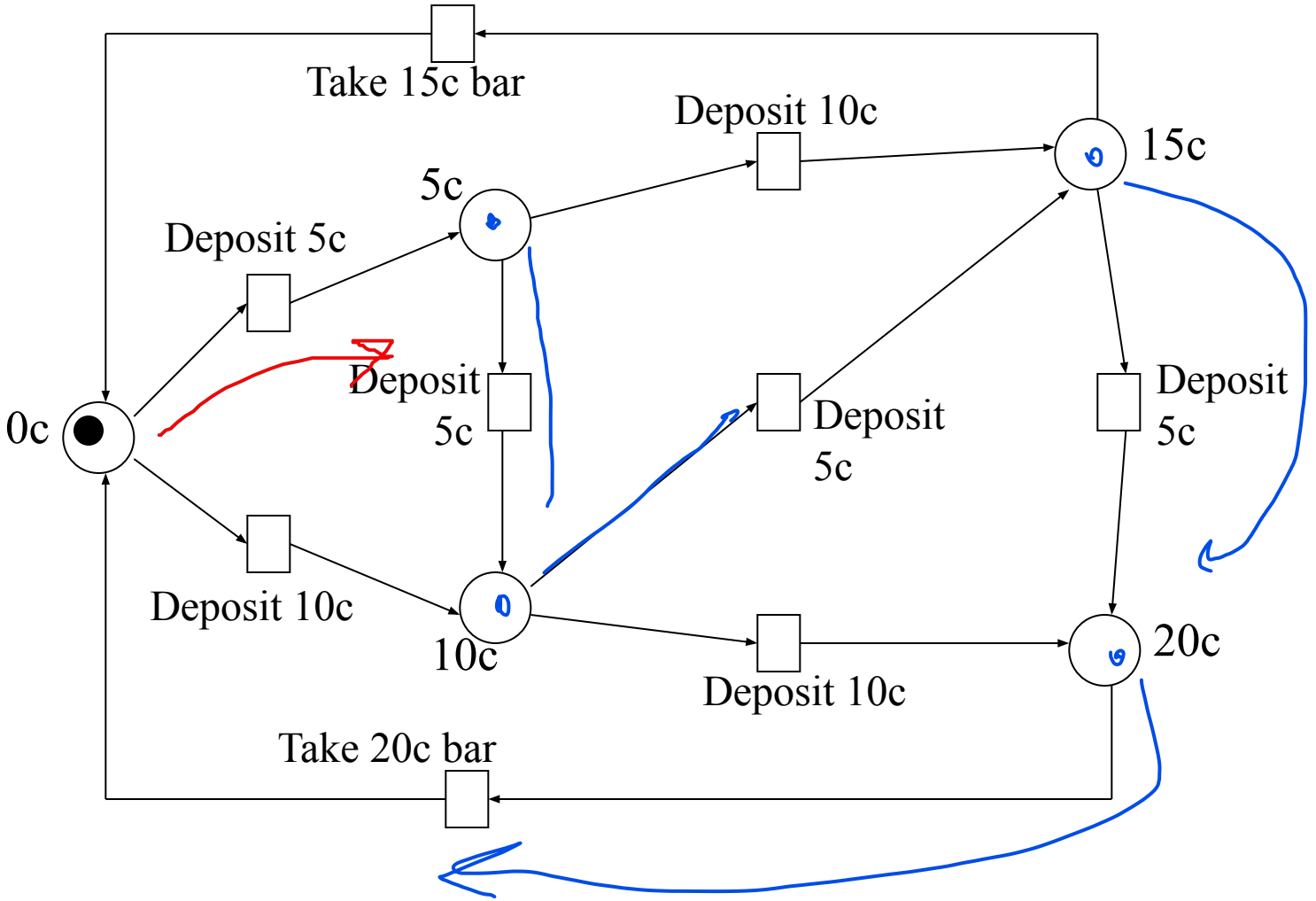
Example: In a Restaurant (Scenario 1)



Example: In a Restaurant (Scenario 2)



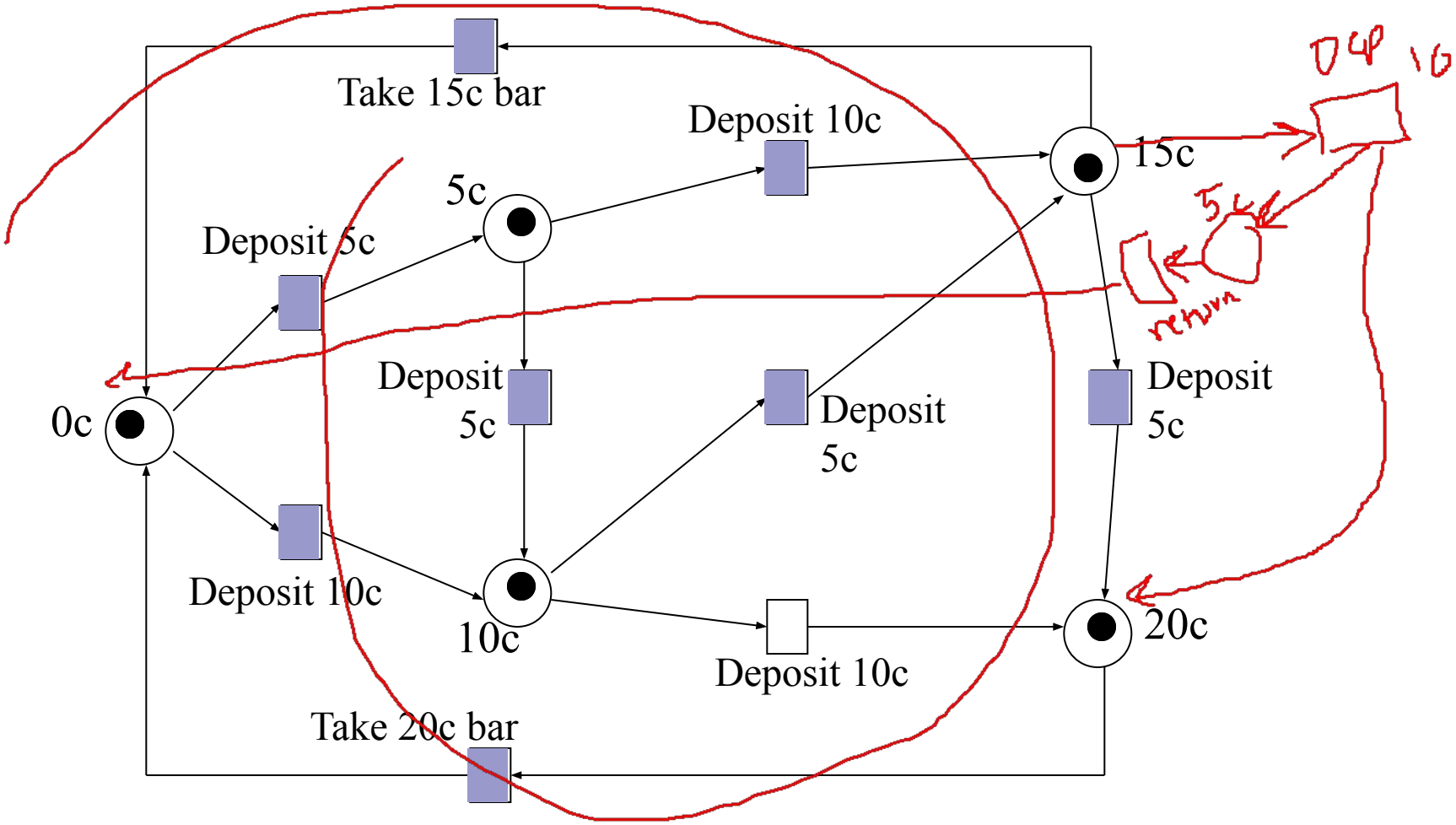
Example: Vending Machine (A Petri net)



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 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

- Peter Marwdwel Chapter 2.6 Petri-nets
- 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