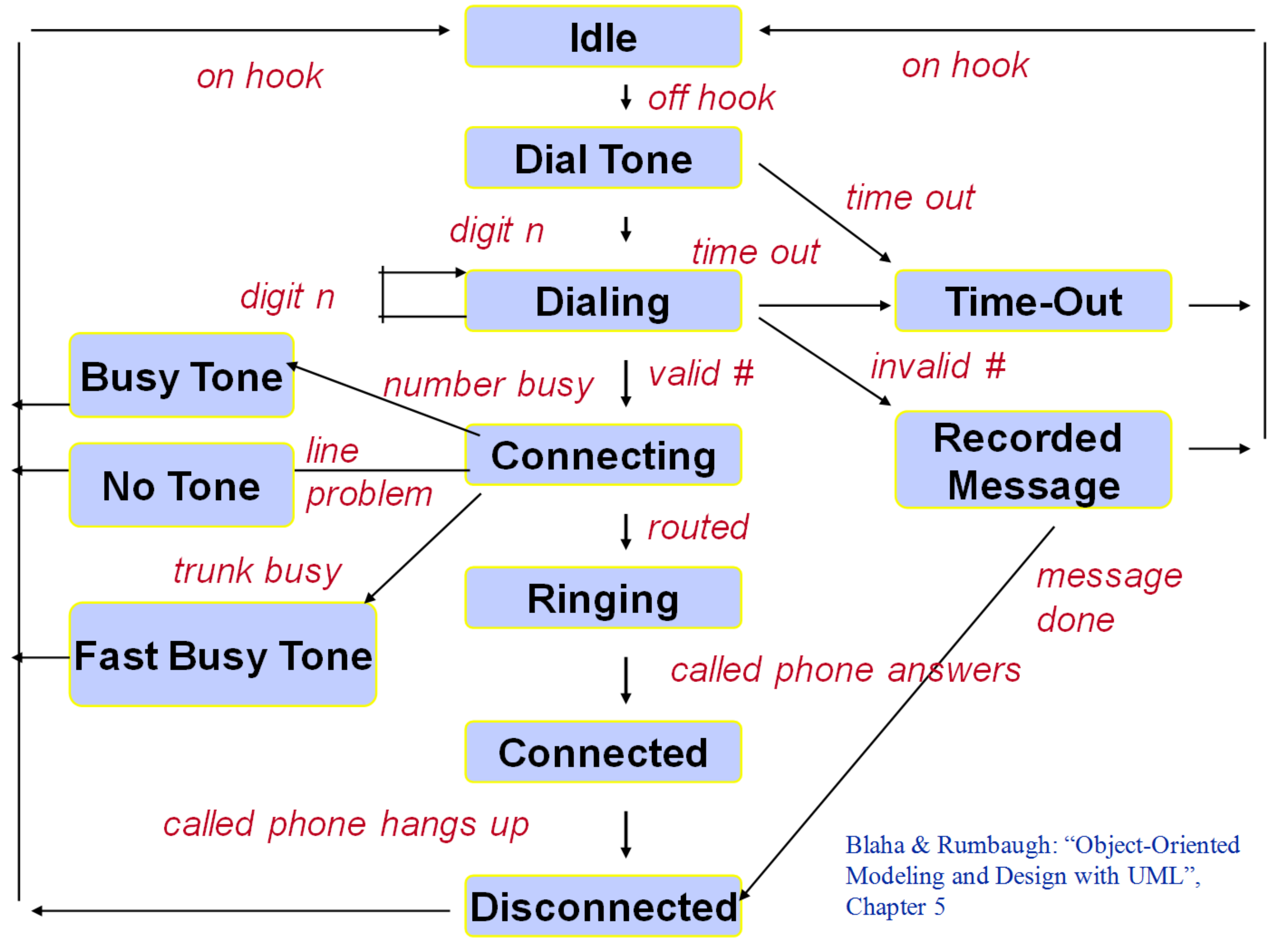
Concurrent Systems Modeling using Petri Nets

Fabrizio Maria Maggi

Based on lecture material by Marlon Dumas (University of Tartu, Estonia) and Wil van der Aalst

(Eindhoven University of Technology, The Netherlands http://www,workflowcourse.com)

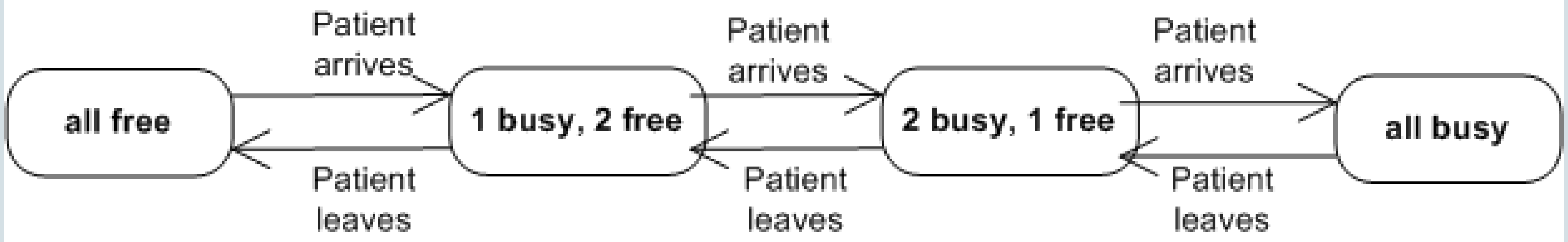
Behavior Modeling: State Machines



Blaha & Rumbaugh: "Object-Oriented Modeling and Design with UML", Chapter 5

Limitations of state machines

- Three doctors in a medical centre



- What if there are 6 doctors?
- What if a patient arrives and all doctors are busy?
- What if doctors can arrive and leave (so long as they are not busy)?
- State explosion…

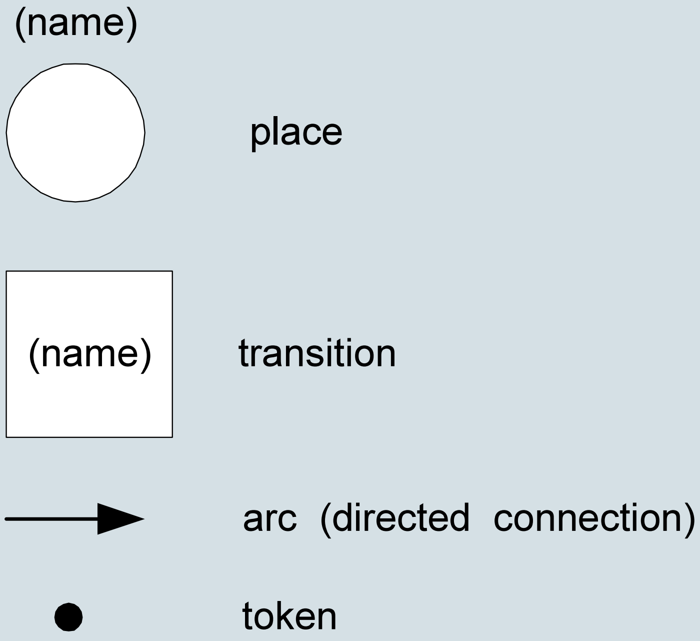
Concurrent systems modeling

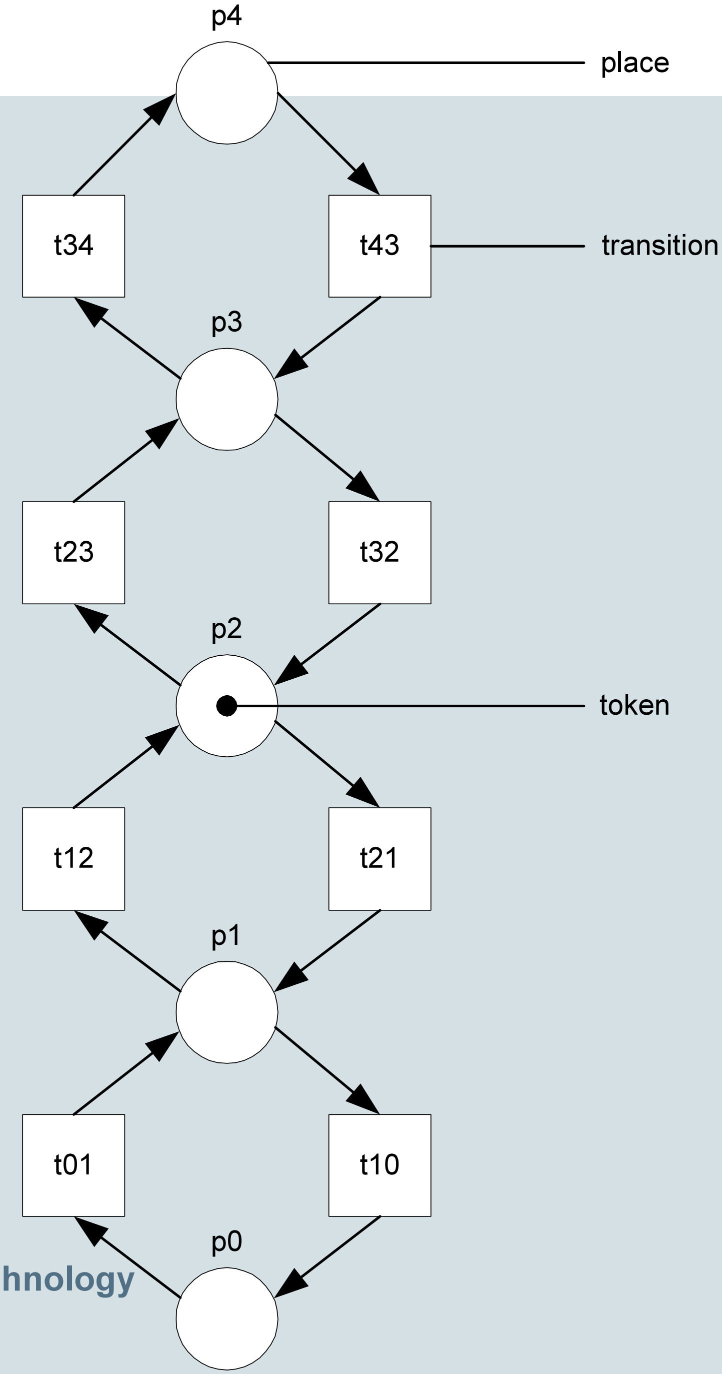
- State machines are useful to model behaviour of sequential systems
- But many systems are concurrent by nature
- Petri nets are a family of techniques for modeling systems with concurrency, communication and synchronization

Petri nets

- Simple technique for concurrent systems modeling
  - Four elements: **places**, **transitions**, **arcs** and **tokens**.
  - Graphical and mathematical description.
  - Formal semantics suitable for static analysis.
- Supported by verification and simulation tools (e.g. CPN Tools, ProM, LoLa, Woped).
- Once you understand Petri nets, you will be better equipped to understand other techniques for modeling systems with concurrency (e.g. process modeling notations)

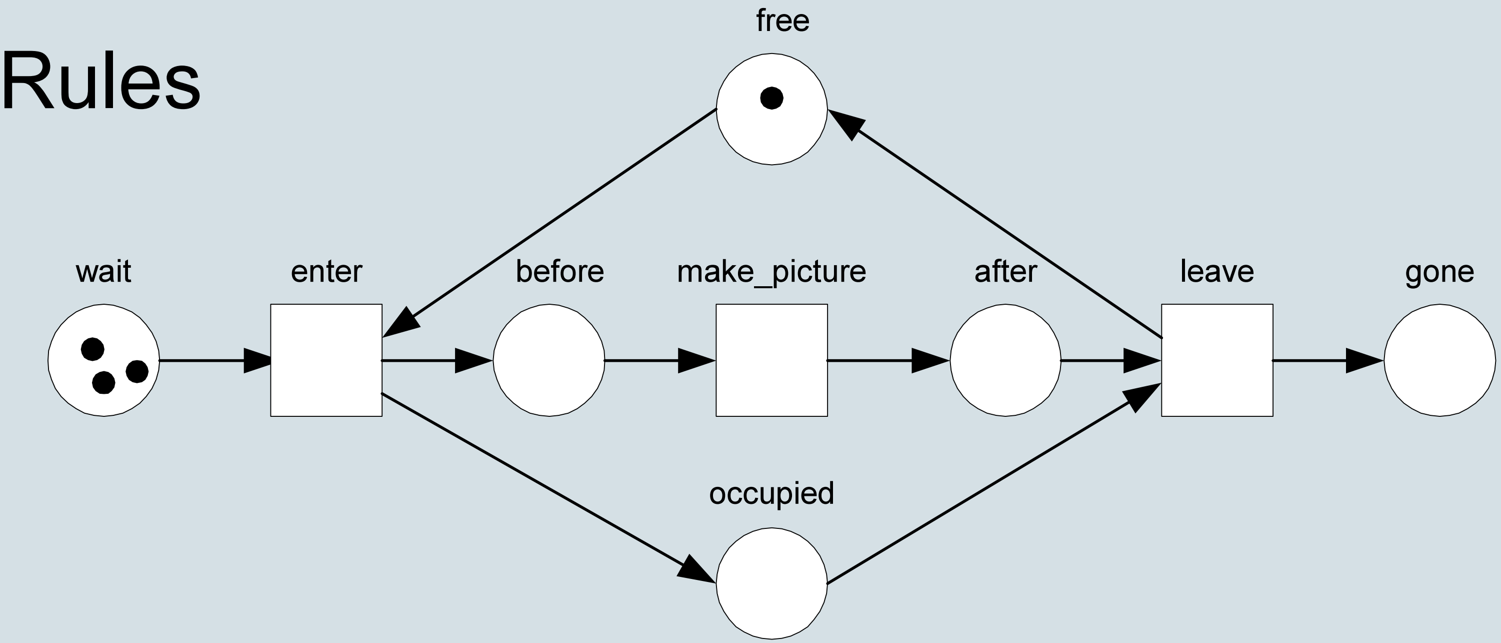
Elements





(c) Wil van der Aalst, Eindhoven University of Technology

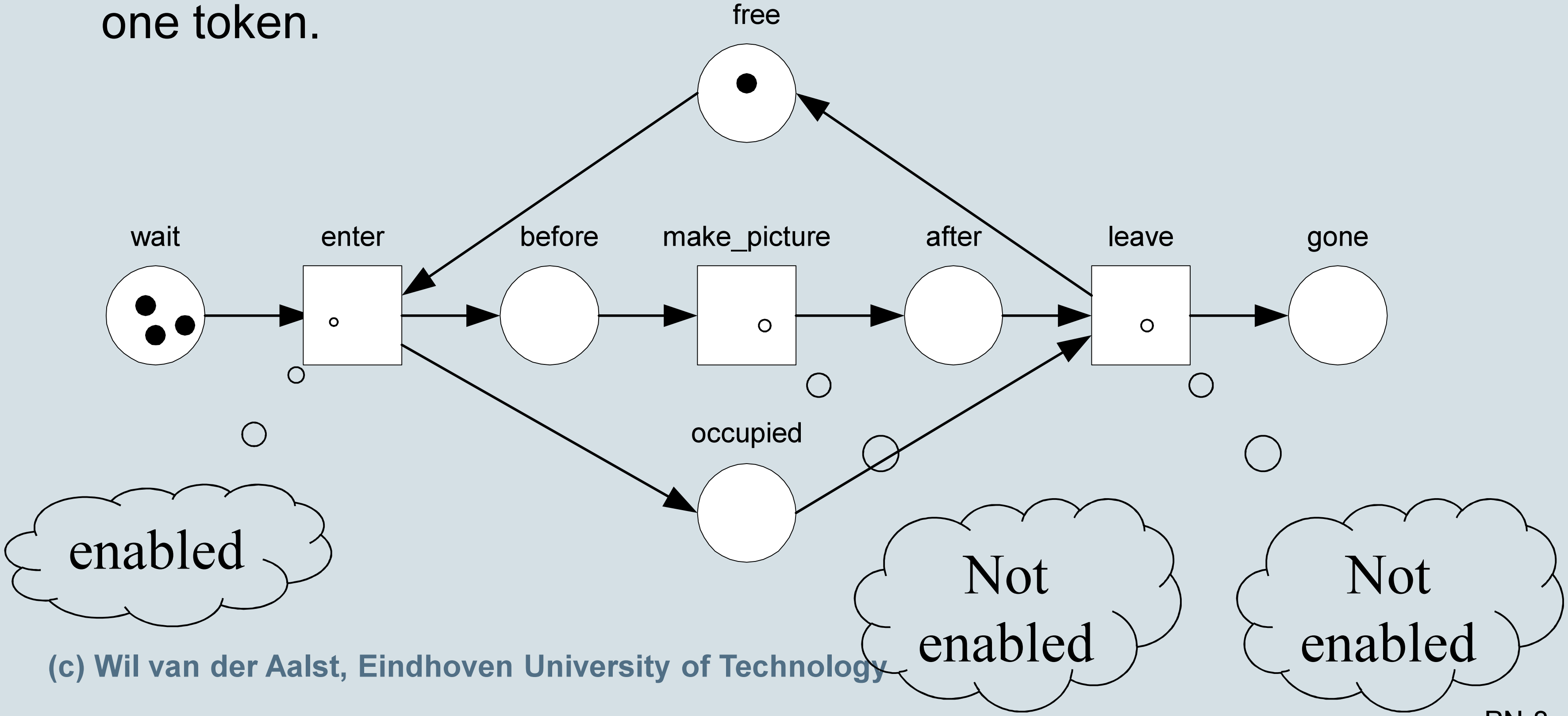
Rules



- Connections are directed.
- No connections between two places or two transitions.
- Places may hold zero or more tokens.
- First, we consider the case of at most one arc between two nodes.

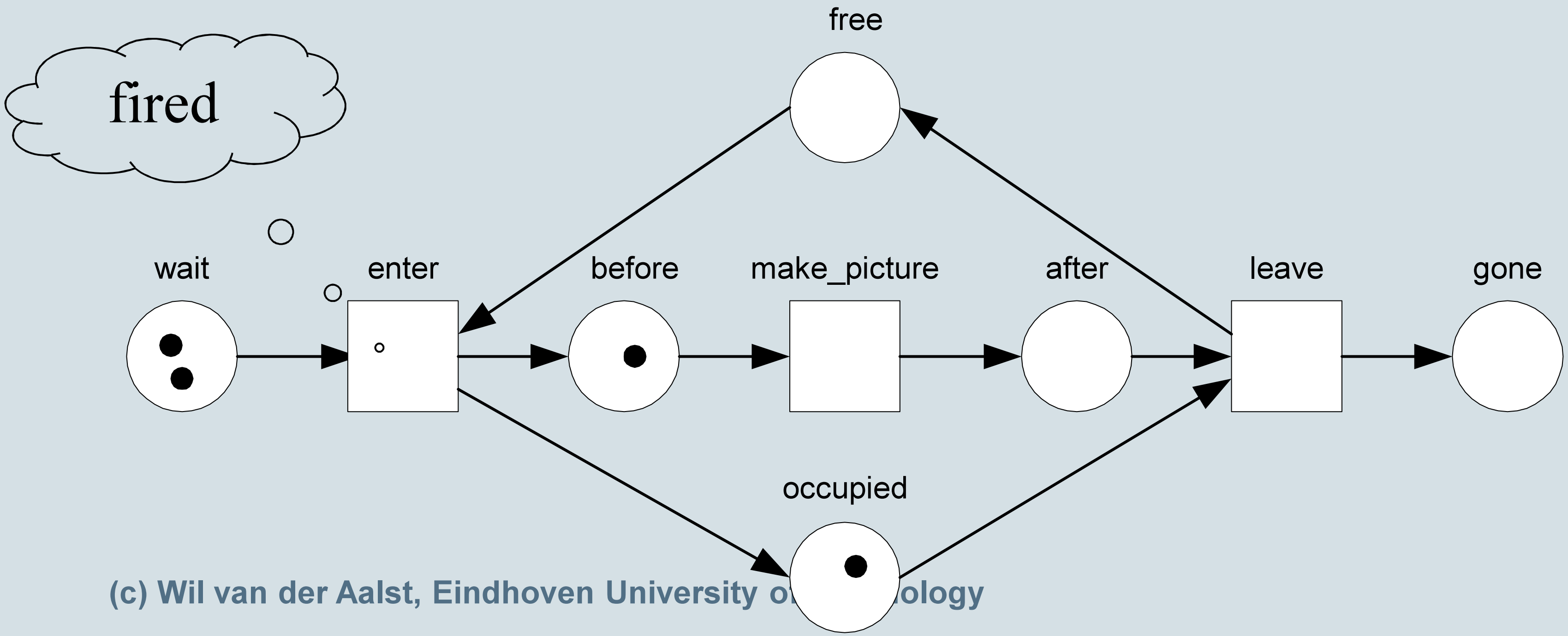
Marking and Enabled Transition

- The **state** of a net is a distribution of tokens over places (also referred to as **marking**).
- A transition is **enabled** if each of its input places contains at least one token.



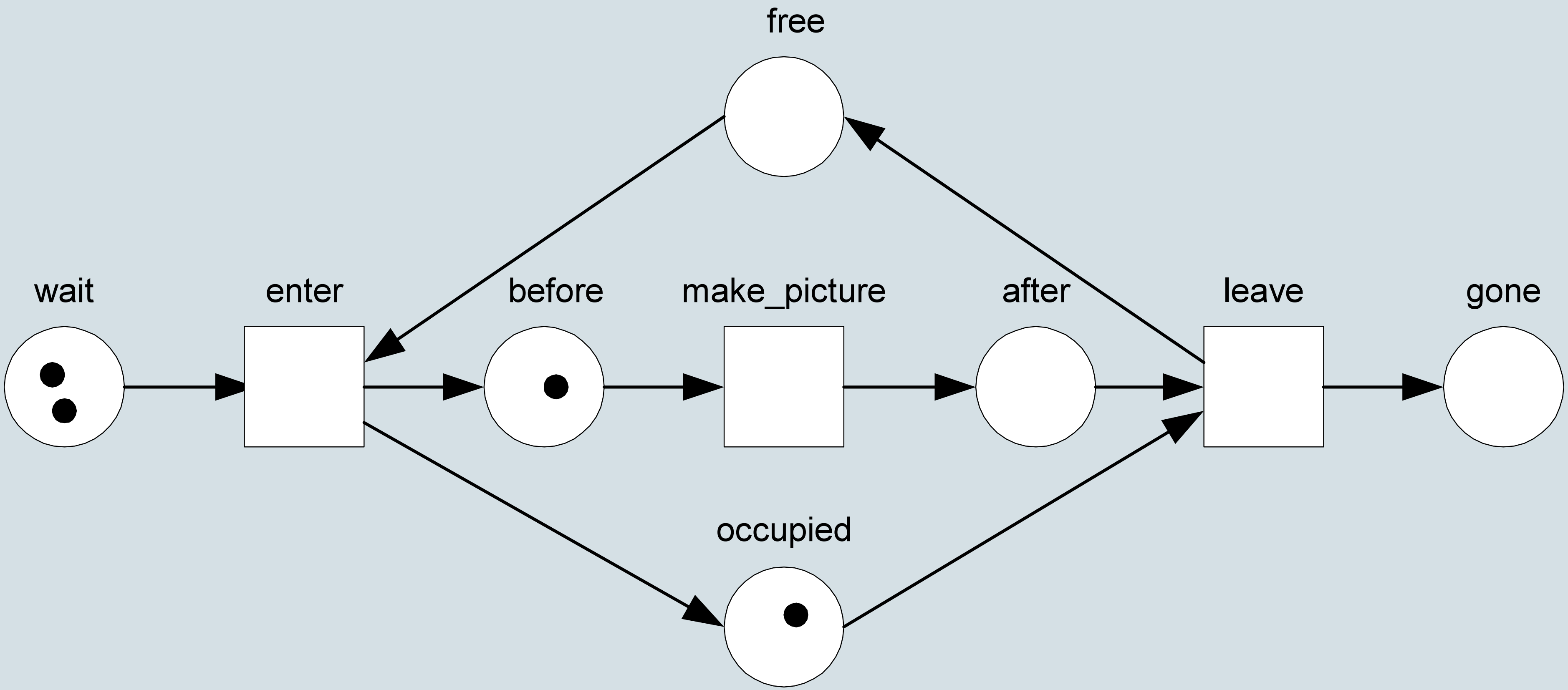
Firing

- An **enabled** transition can **fire** (i.e., it occurs).
- When it **fires** it **consumes** a token from each input place and **produces** a token for each output place.
- Which transitions are enabled now?



“Token Game”

- In the new state, *make_picture* is enabled. It will fire, etc.

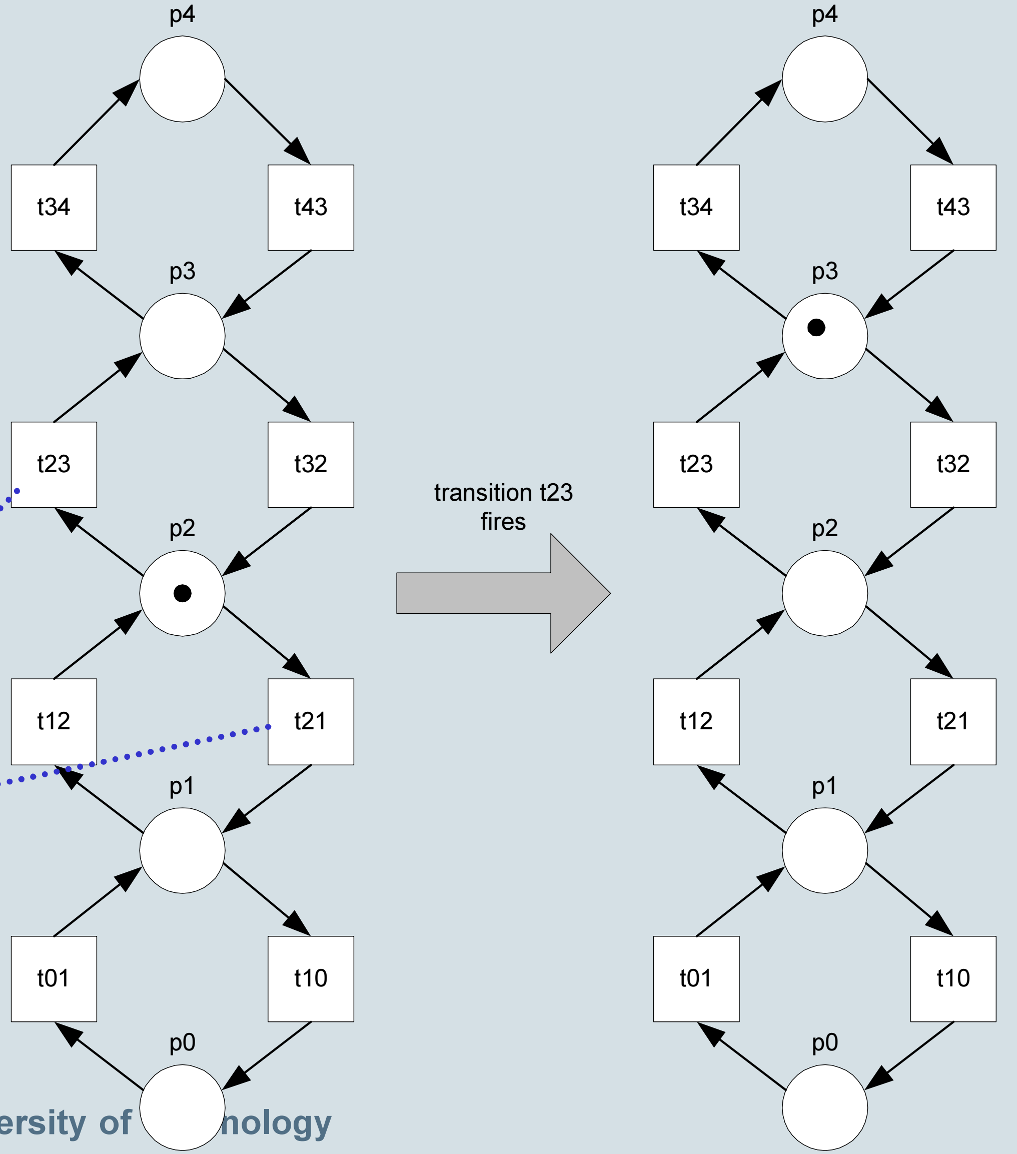


(c) Wil van der Aalst, Eindhoven University of Technology

Remarks

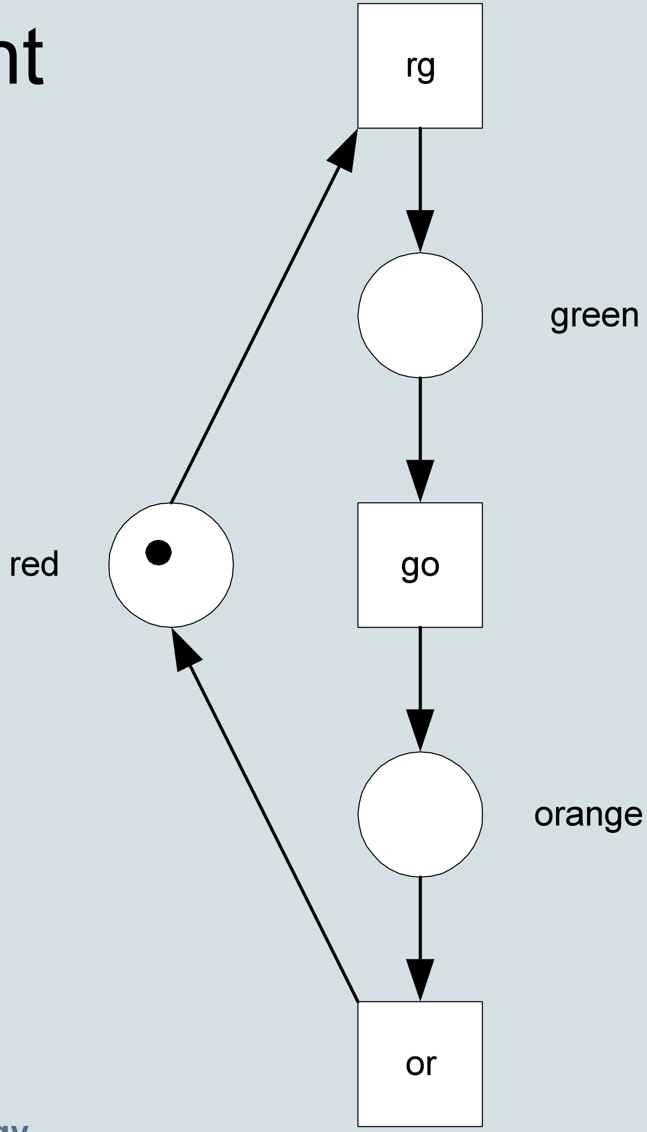
- Firing is **atomic**.
- Multiple transitions may be enabled, but only one fires at a time
- By default, choice is *non-deterministic*
- Any state machine can be trivially converted into a Petri net – How?

Non-determinism

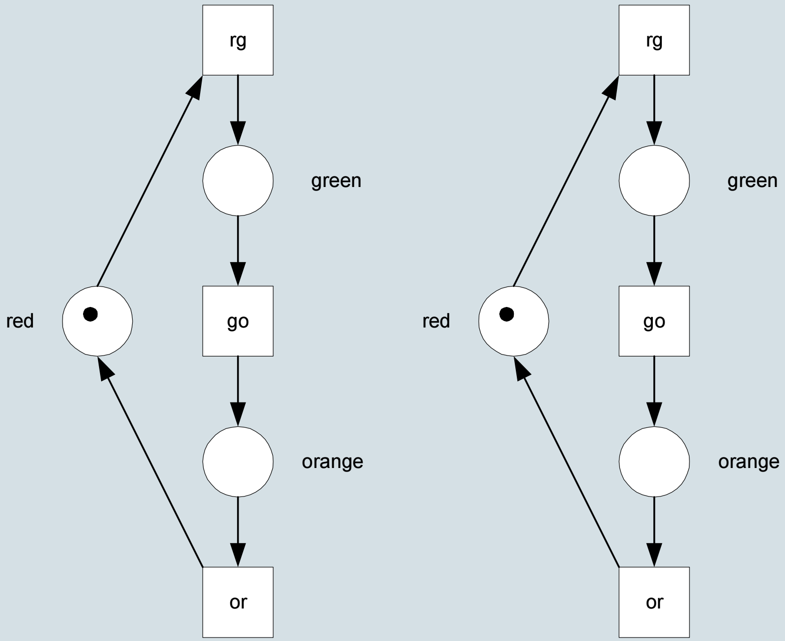


Two transitions are enabled but only one can fire

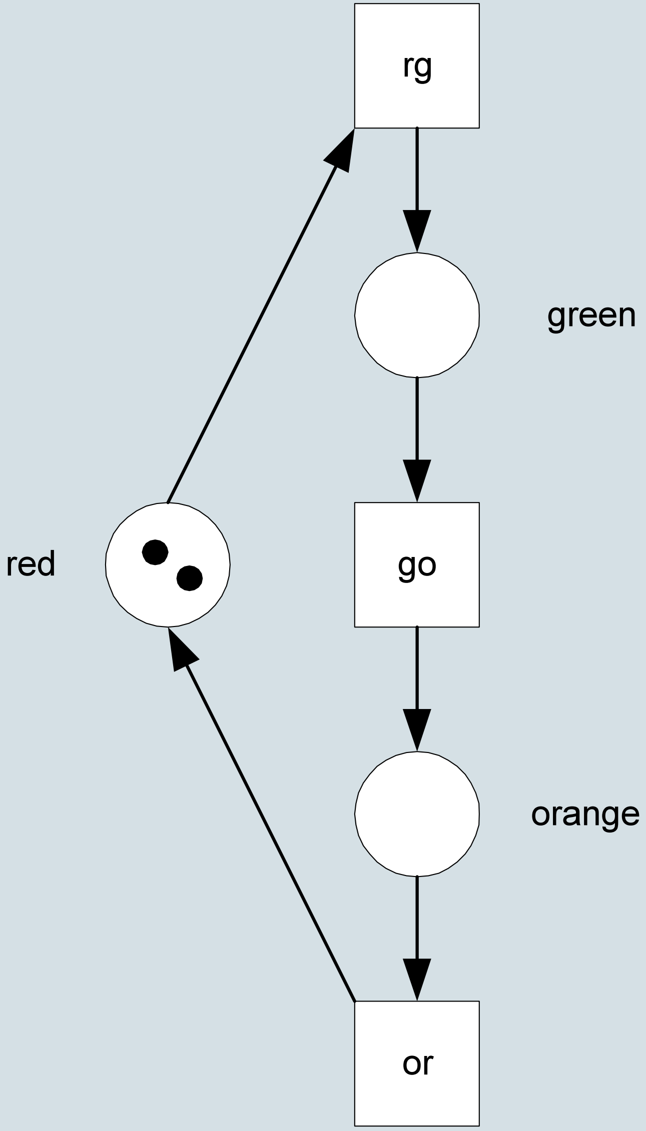
Example: Single traffic light



Two traffic lights

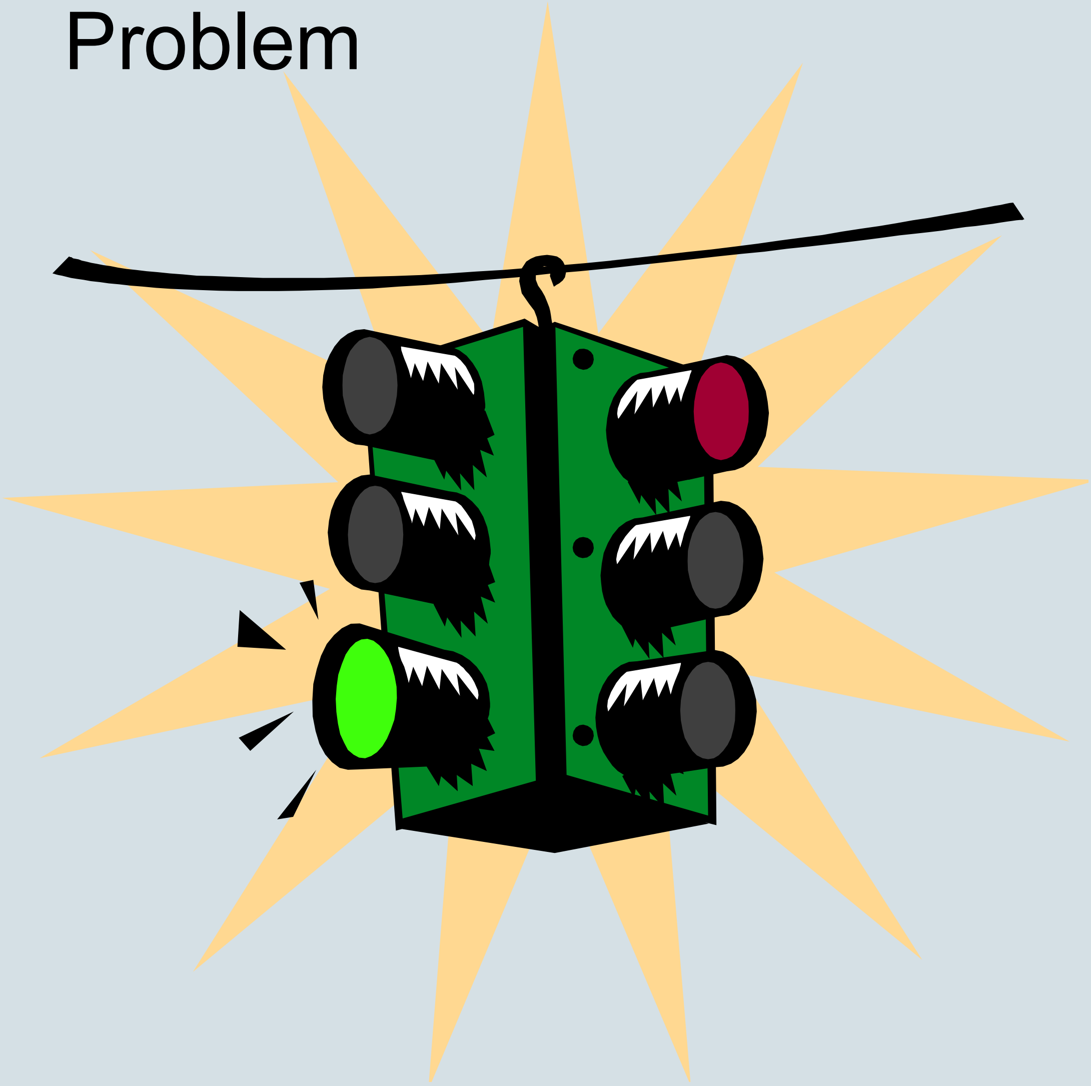


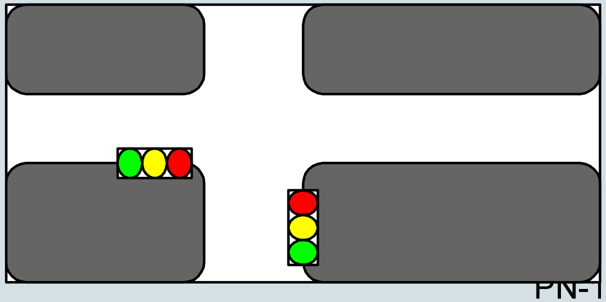
OR



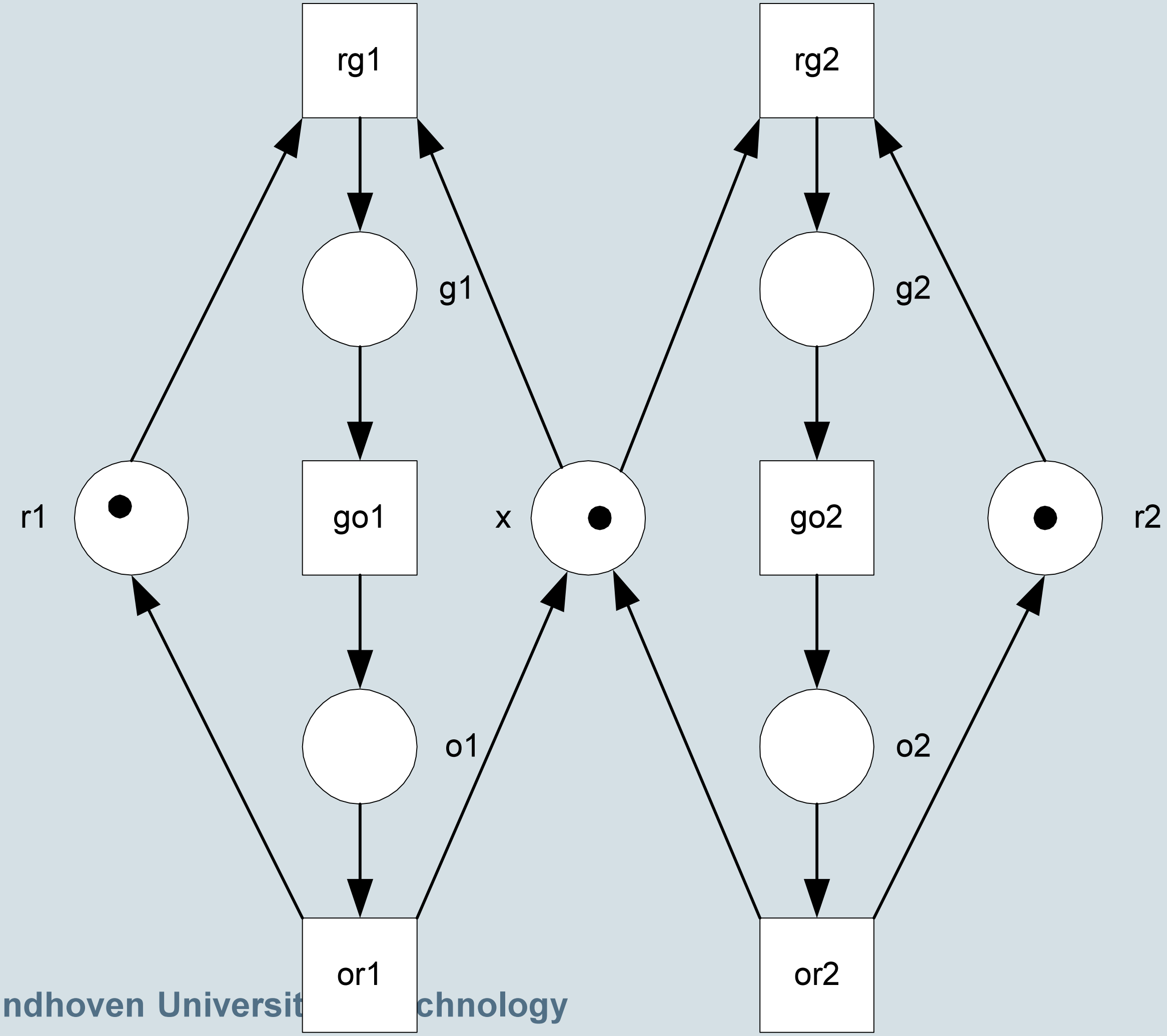
(c) Wil van der Aalst, Eindhoven University of Technology

Problem





Solution

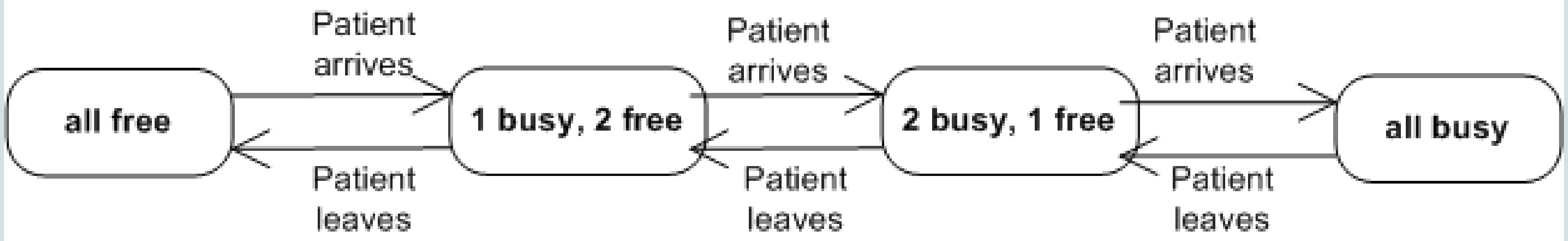


How to make them alternate?

Playing the "Token Game"

- FLASH animations:
  - http://wwwis.win.tue.nl/~wvdaalst/workflowcourse/
- Woped: A more sophisticated Petri net drawing and animation tool:
  http://www.woped.org/

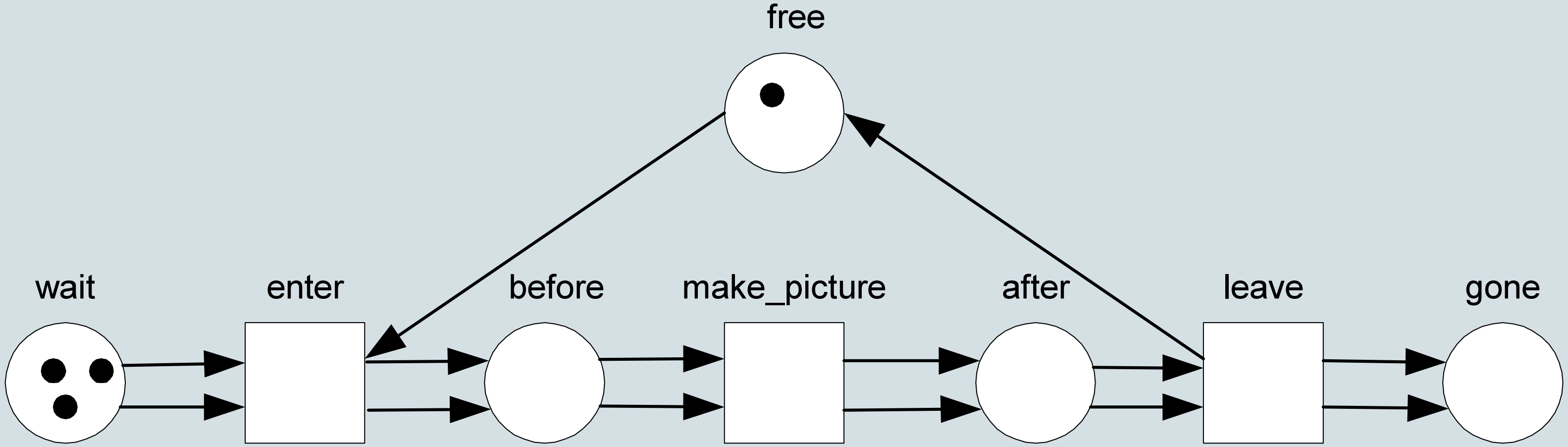
Exercise: Doctor’s scenario in Petri nets



Case 1: Patients arrive and leave, number of doctors fixed
Case 2: Patients arrive and leave, doctors arrive and leave (but only leave when they are free)
Case 3: When patients arrive, they are classified into simple and complex cases. Simple cases require only a doctor, complex cases require a doctor and a nurse. (Assume doctors and nurses do not arrive nor leave)

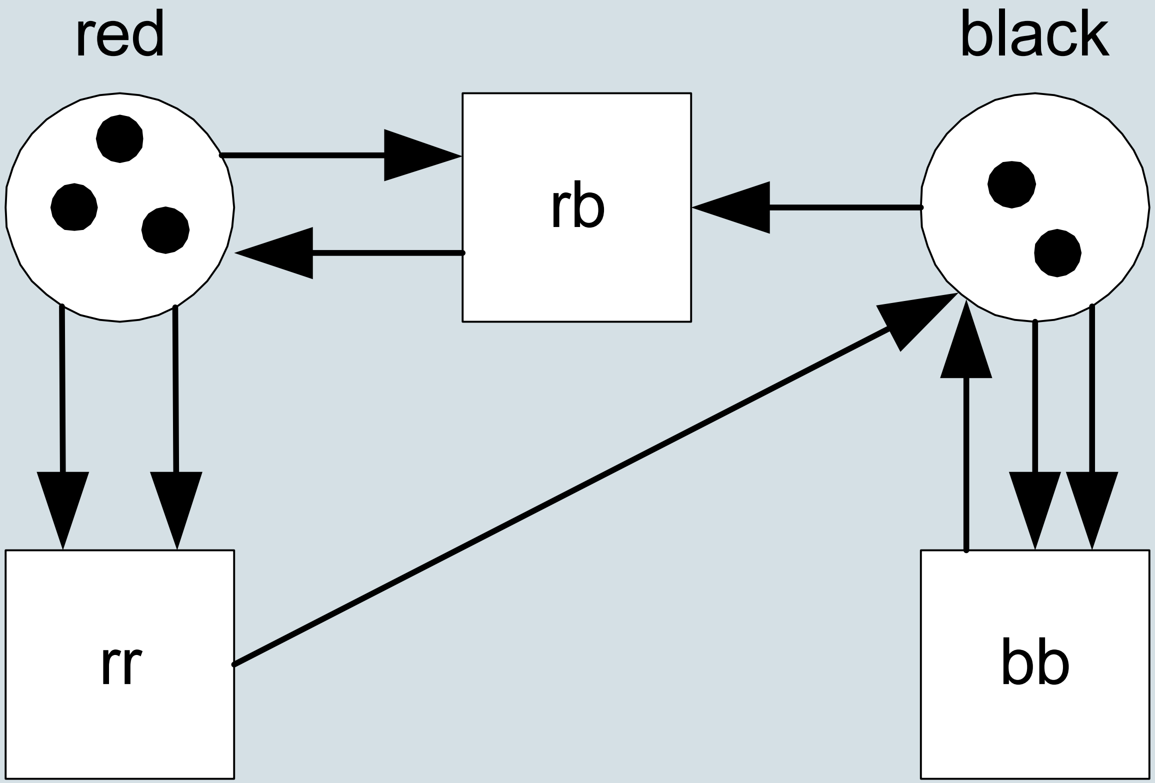
Multiple arcs connecting two nodes

- The number of arcs between an input place and a transition determines the number of tokens required to be enabled.
- The number of arcs determines the number of tokens to be consumed/produced.



(c) Wil van der Aalst, Eindhoven University of Technology

Example: Ball game



Which transition(s) is/are enabled?

(c) Wil van der Aalst, Eindhoven University of Technology

You should be able to ...

- Explain what is a Petri net and what are the basic elements of (plain) Petri nets
- Play a token game on a Petri net.
- Model simple concurrent systems using Petri nets.