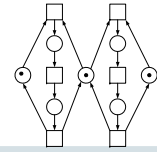


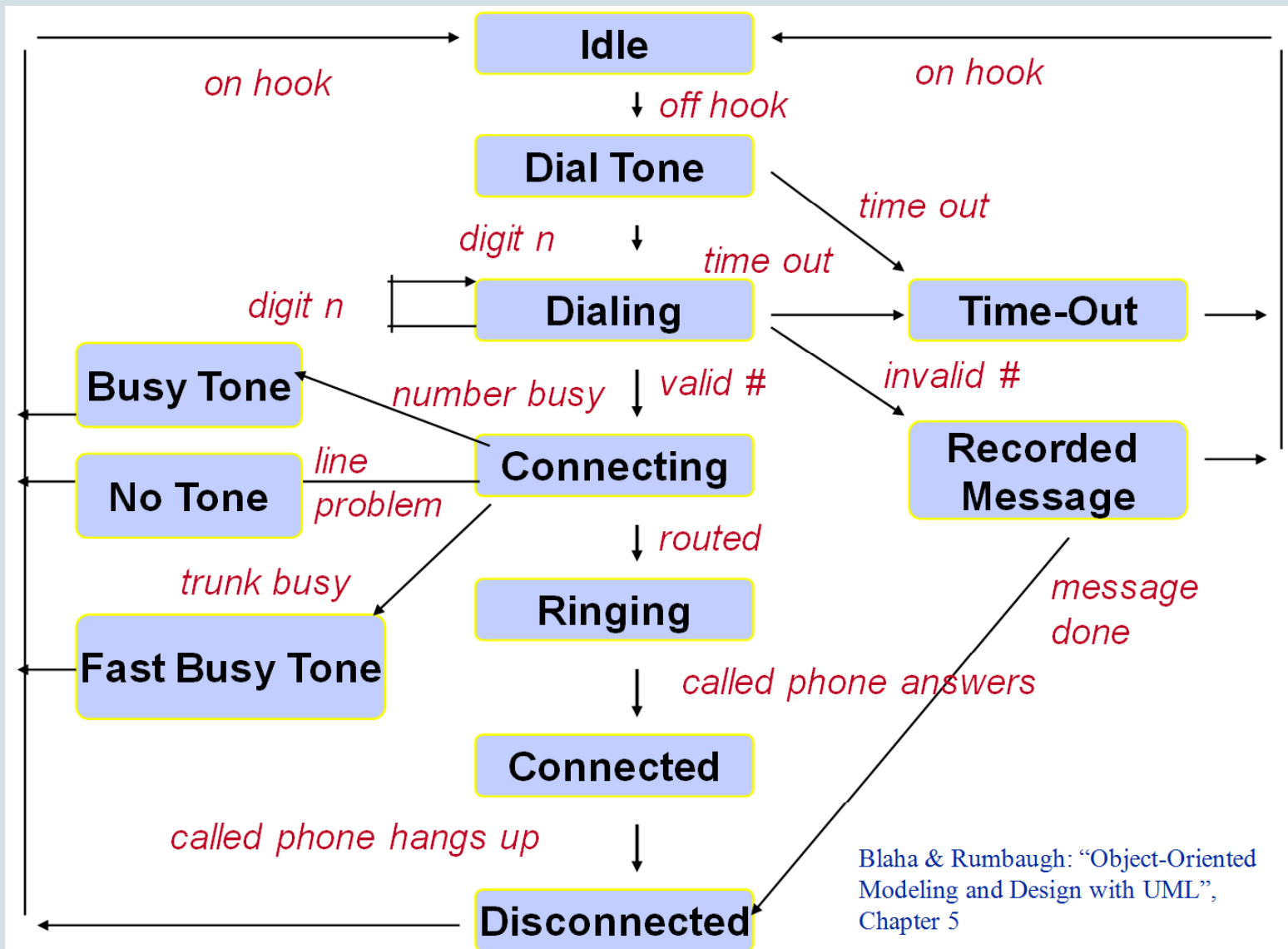
Concurrent Systems Modeling using Petri Nets

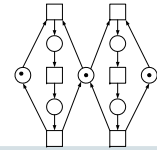
Marlon Dumas

*(Based on lecture material by Wil van der Aalst
Eindhoven University of Technology, The Netherlands
<http://www.workflowcourse.com>)*



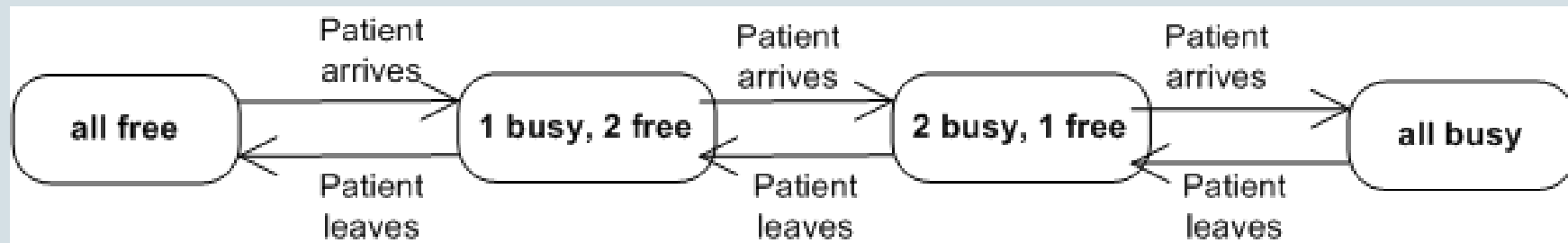
Behavior Modeling: State Machines



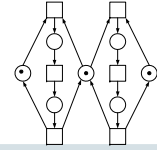


Limitations of state machines

- Three doctors in a medical centre

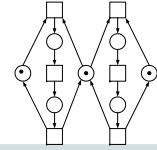


- What if there are 6 doctors?
- What if doctors can arrive and leave (so long as they are not busy)?
- State explosion...



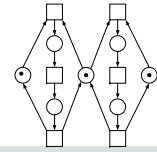
Concurrent systems modeling

- State machines are useful to model behaviour of sequential systems
- But many systems are concurrent by nature
- Statechart diagrams overcome some of the limitations (through compound states), but not all
- Petri nets are a family of techniques for modeling systems with concurrency, communication and synchronization

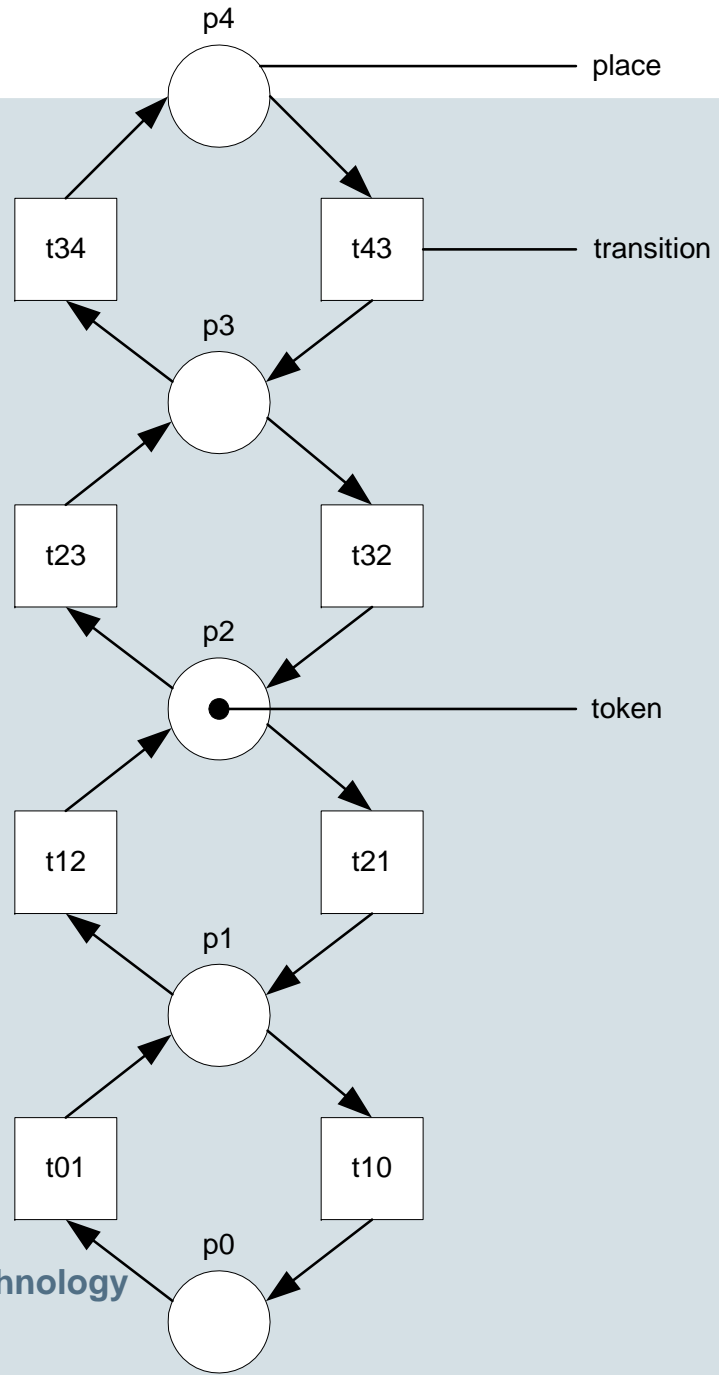
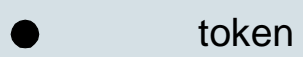
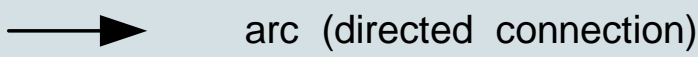
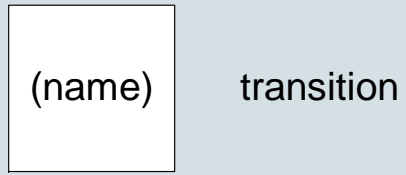
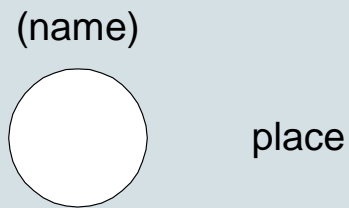


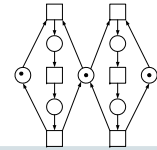
Classical Petri net

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

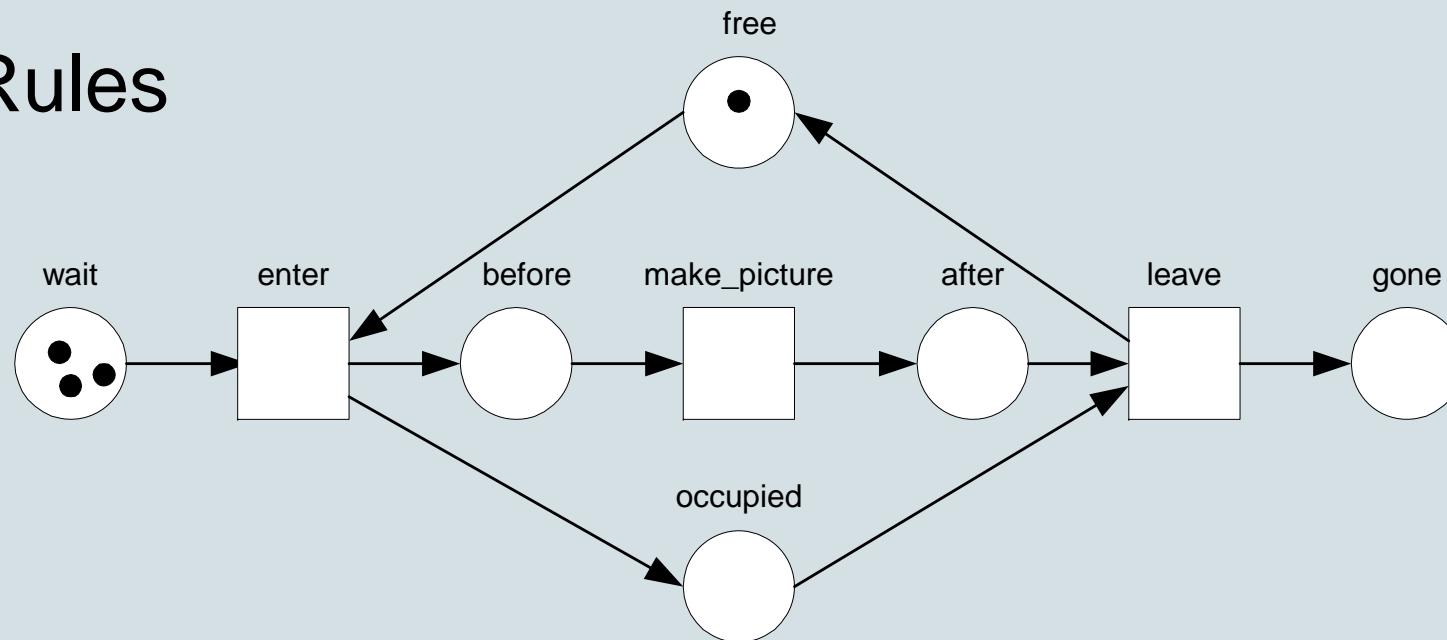


Elements

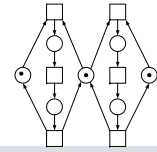




Rules

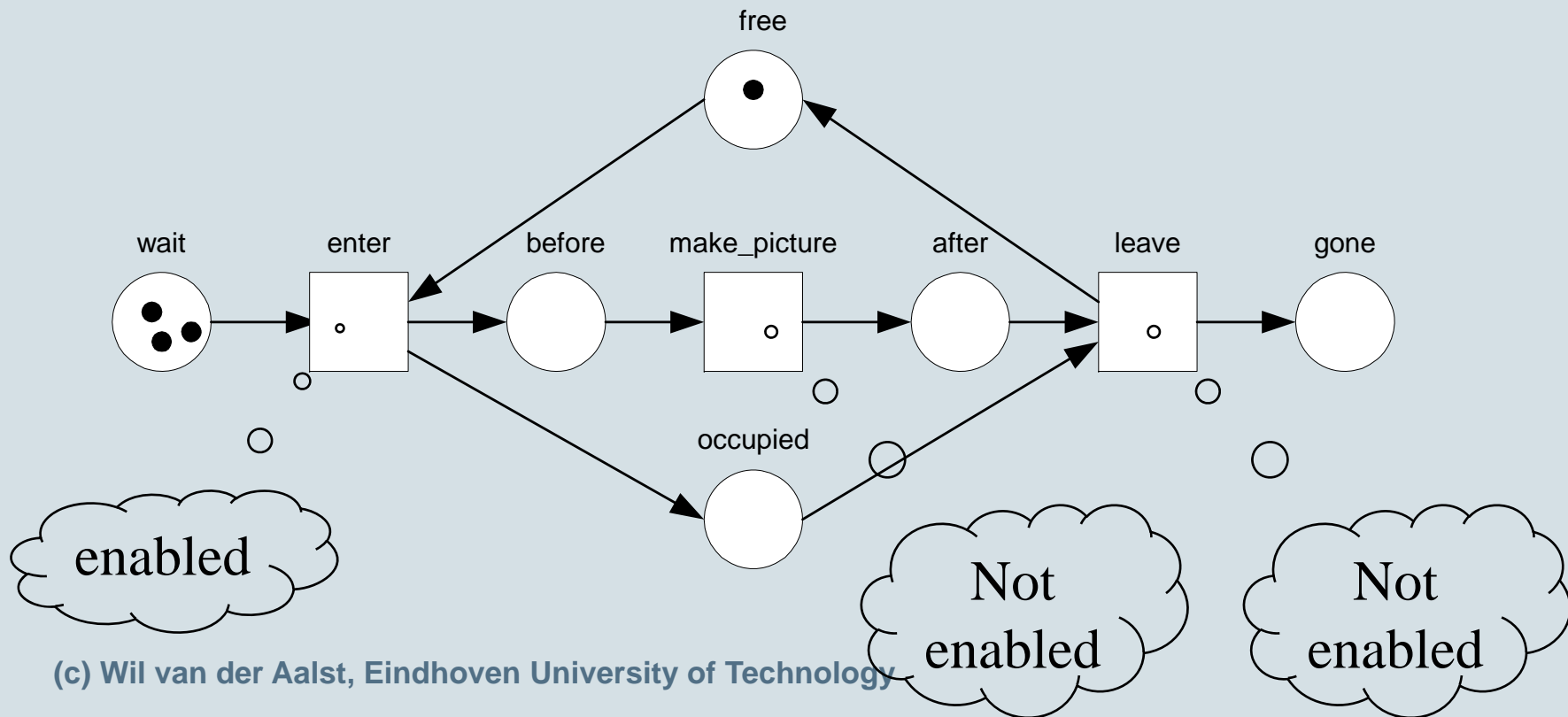


- Connections are directed.
- No connections between two places or two transitions.
- Places may hold zero or more tokens.
- First, we consider the case of at most one arc between two nodes.

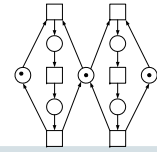


Enabled Transition

- A transition is **enabled** if each of its input places contains at least one token.

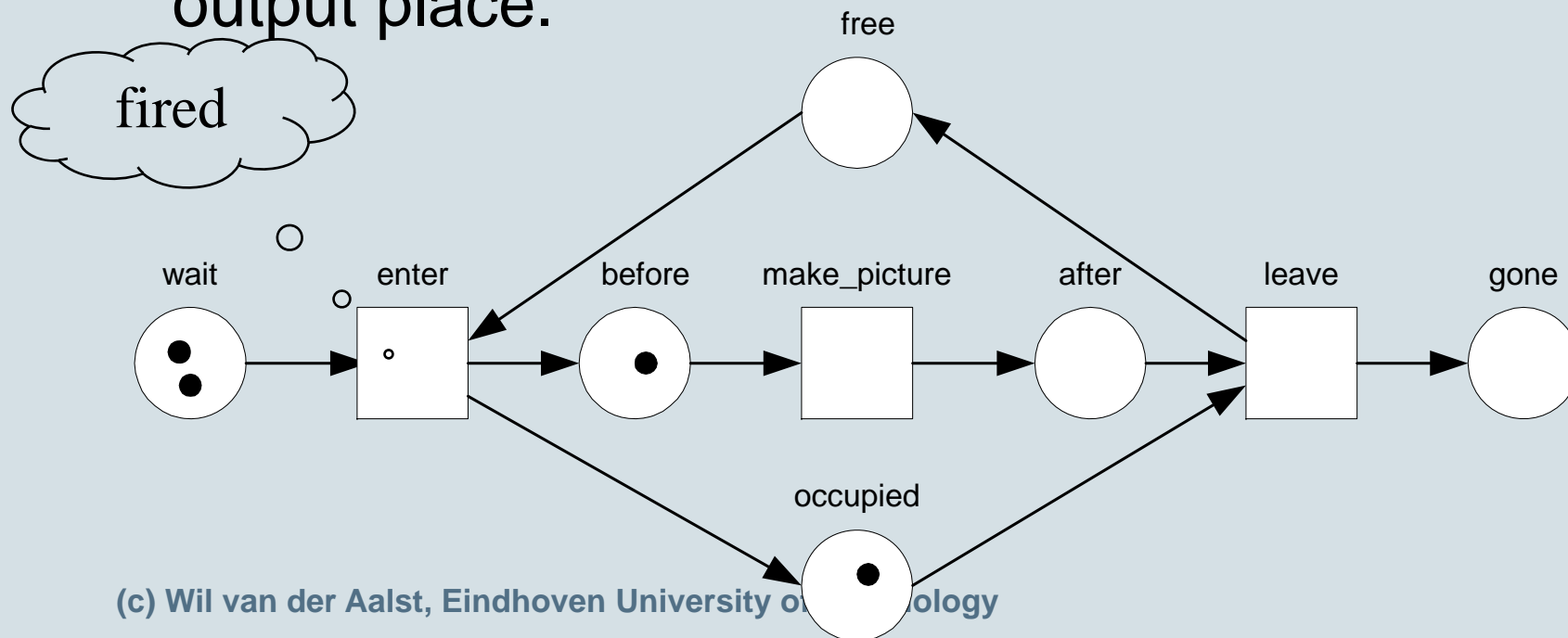


(c) Wil van der Aalst, Eindhoven University of Technology

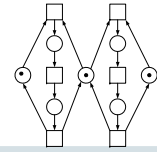


Firing

- 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.

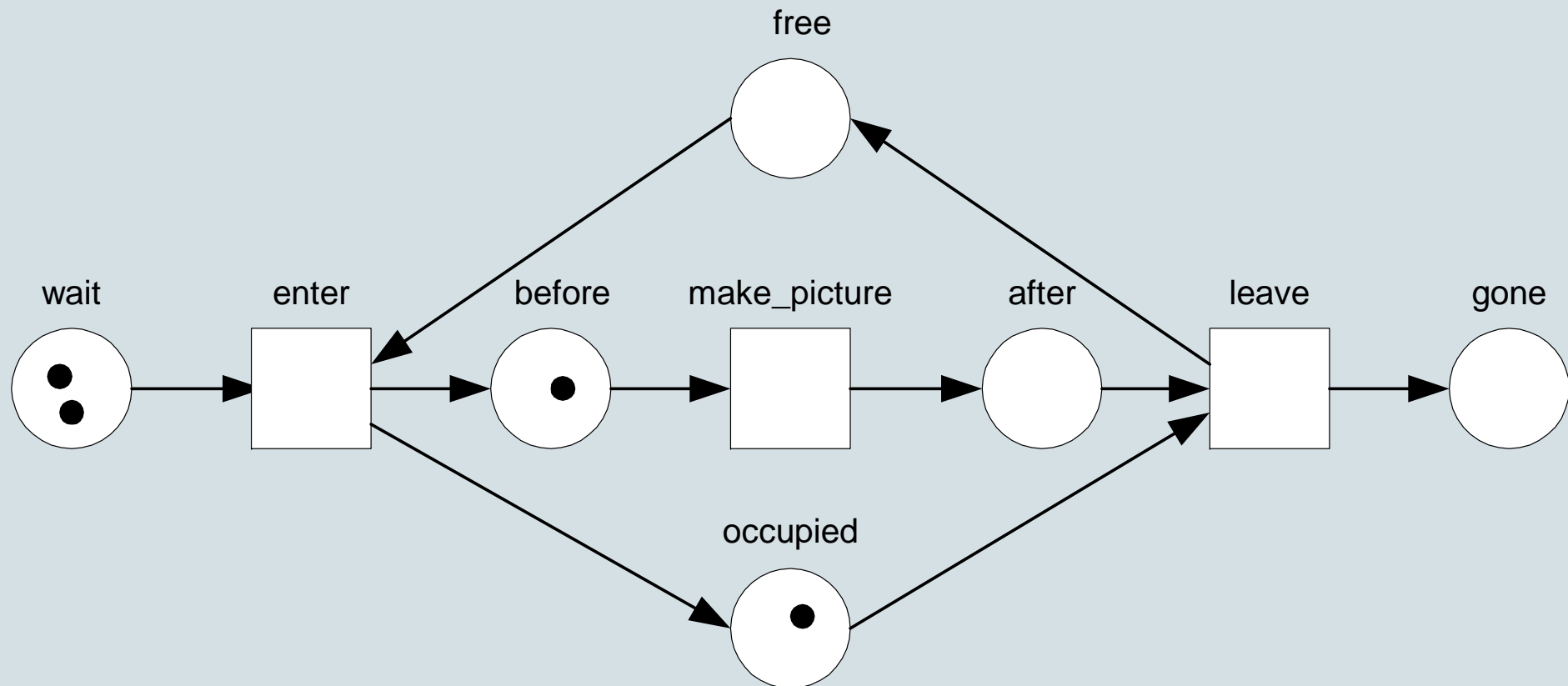


(c) Wil van der Aalst, Eindhoven University of Technology

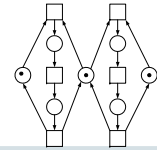


“Token Game”

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

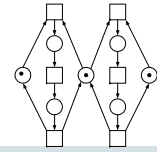


(c) Wil van der Aalst, Eindhoven University of Technology



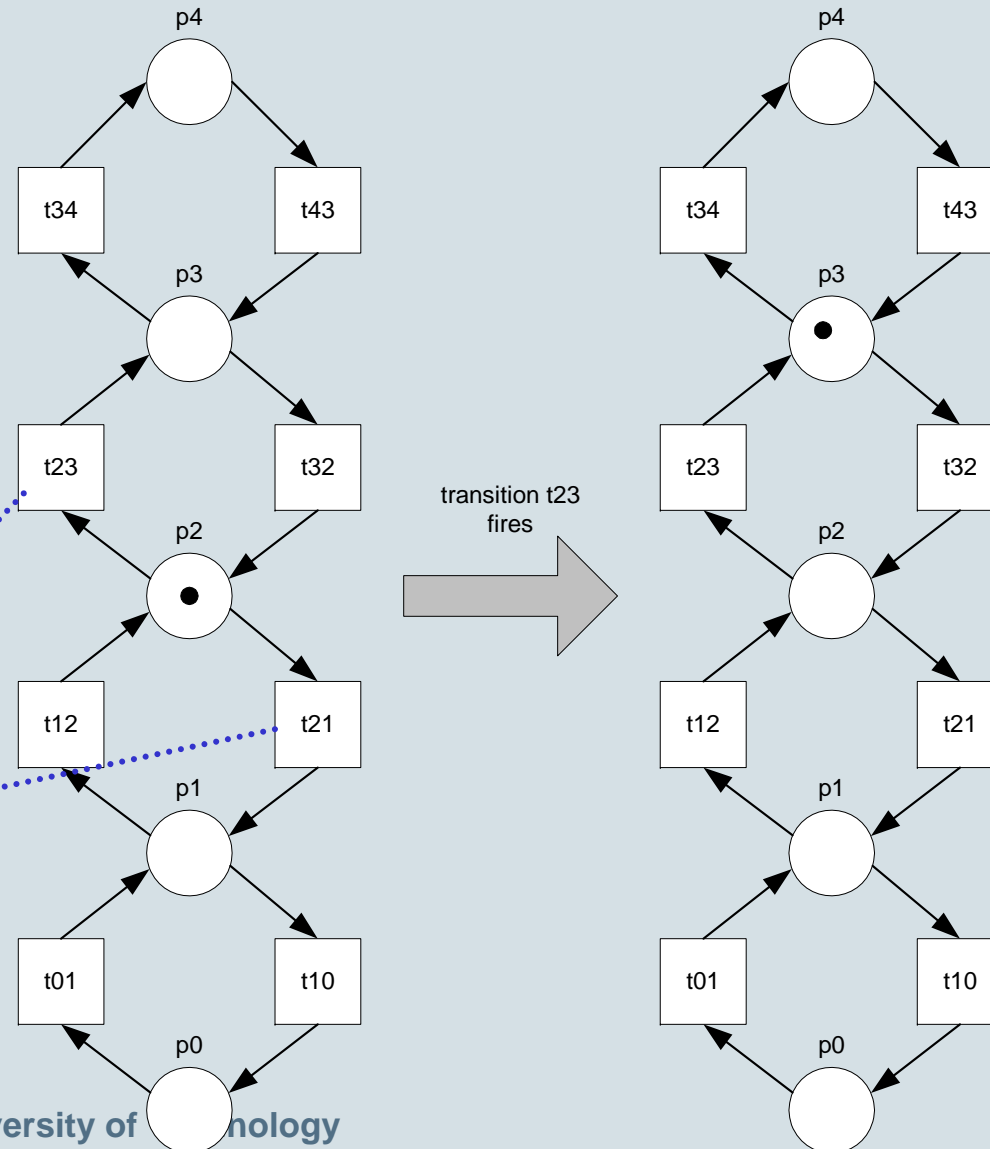
Remarks

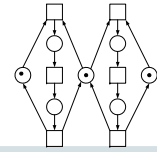
- Firing is **atomic**.
- Multiple transitions may be enabled, but only one fires at a time (called: interleaving semantics).
- The number of tokens may vary if there are transitions for which the number of input places is not equal to the number of output places.
- The **state** is represented by the distribution of tokens over places (also referred to as **marking**).
- Any state machine can be trivially converted into a Petri net – How?



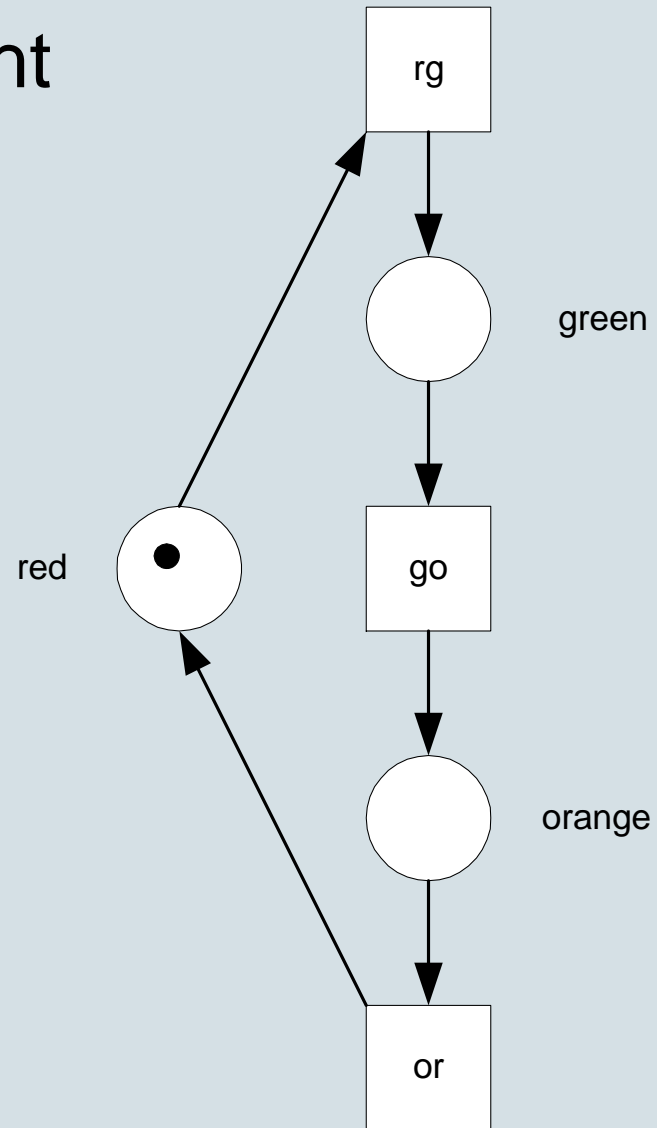
Non-determinism

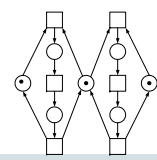
Two transitions are enabled but only one can fire



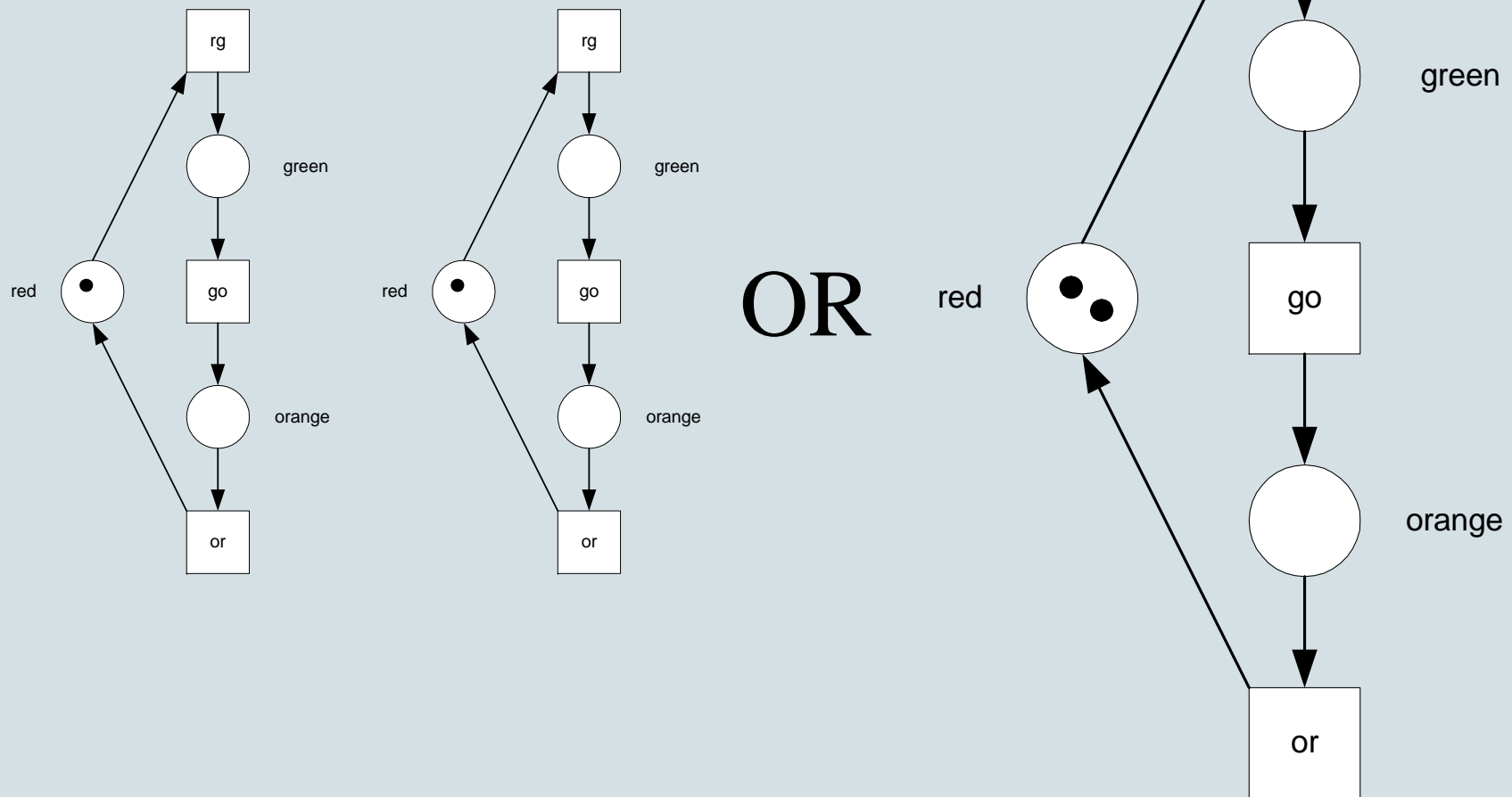


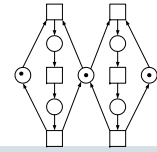
Example: Single traffic light



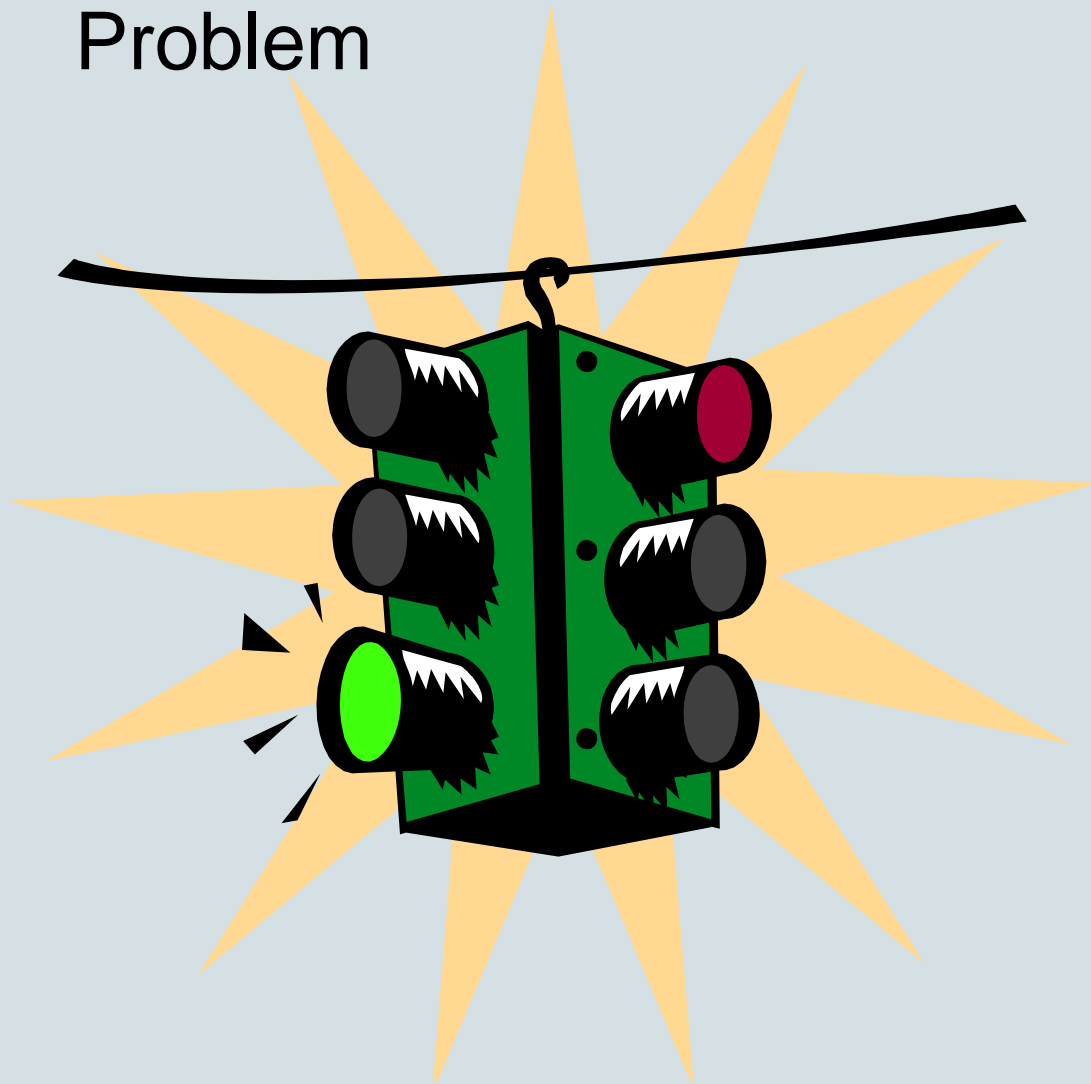


Two traffic lights

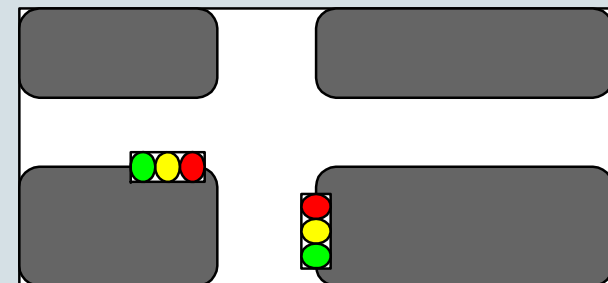


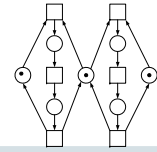


Problem

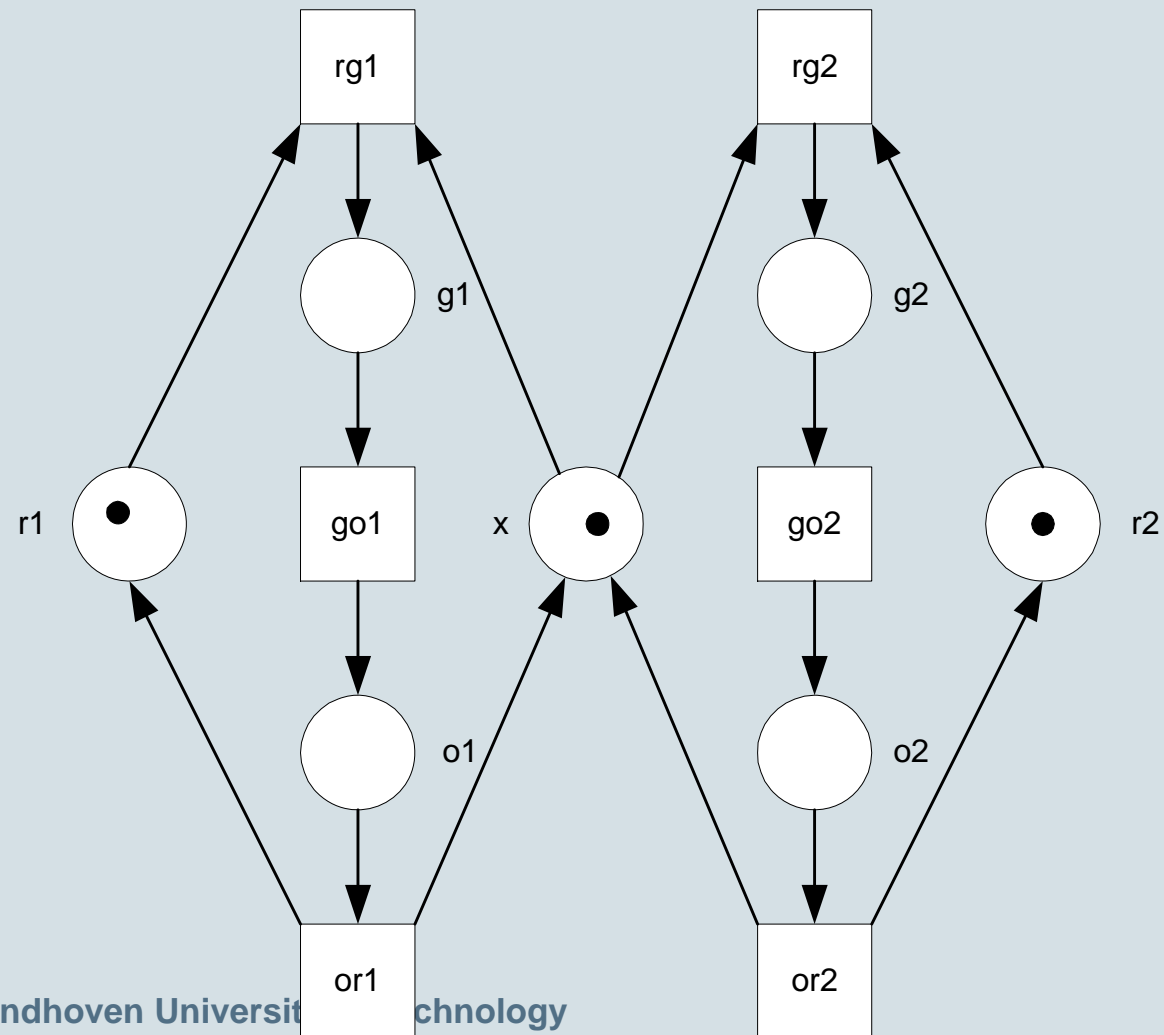


(c) Wil van der Aalst, Eindhoven University of Technology

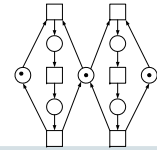




Solution



How to make them alternate?



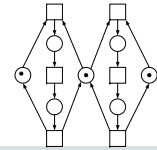
Playing the “Token Game” on the Internet

- FLASH animations:

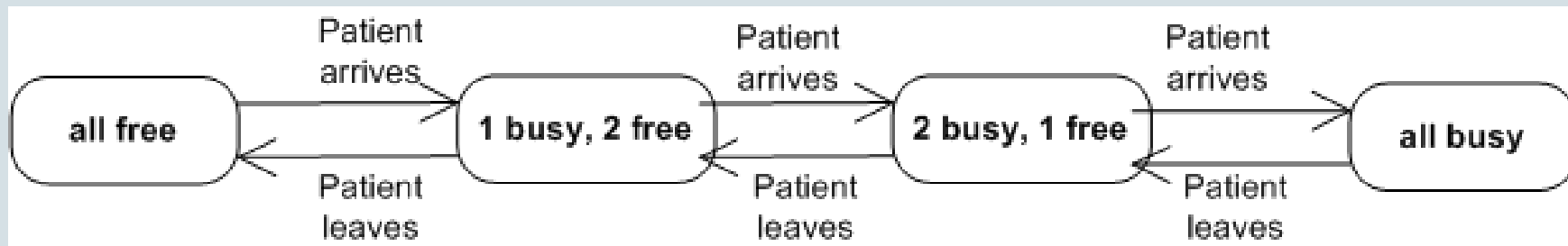
<http://is.tm.tue.nl/staff/wvdaalst/workflowcourse/lecture2.htm>

- Applet to build your own Petri nets and execute them:

http://is.tm.tue.nl/staff/wvdaalst/workflowcourse/pn_applet/pn_applet.htm



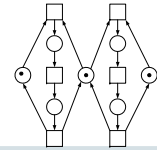
Exercise: Doctor's scenario in Petri nets



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)

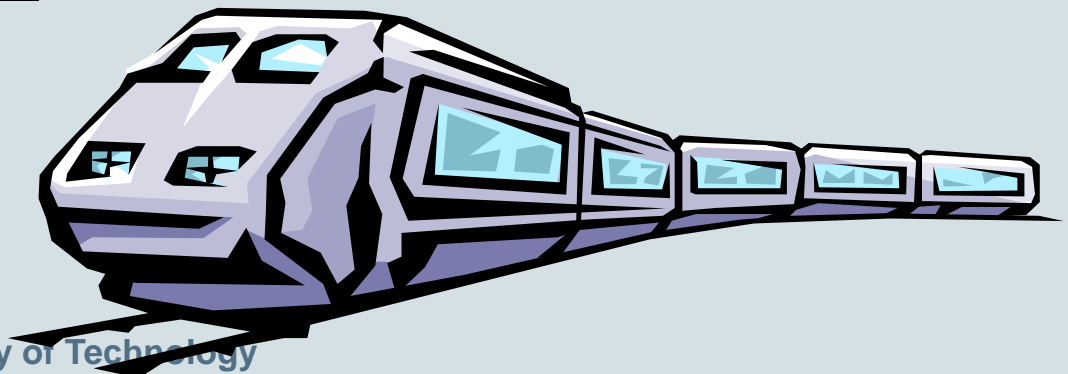
Case 3: When patients arrive, they are classified into simple and complex cases. Simple cases require only a doctor, complex cases require a doctor and a nurse. (Assume doctors and nurses do not arrive nor leave)

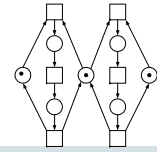


Exercise: Train system

- Consider a circular railroad system with 4 (one-way) tracks (1,2,3,4) and 2 trains (A,B). No two trains should be at the same track at the same time and we do not care about the identities of the two trains.

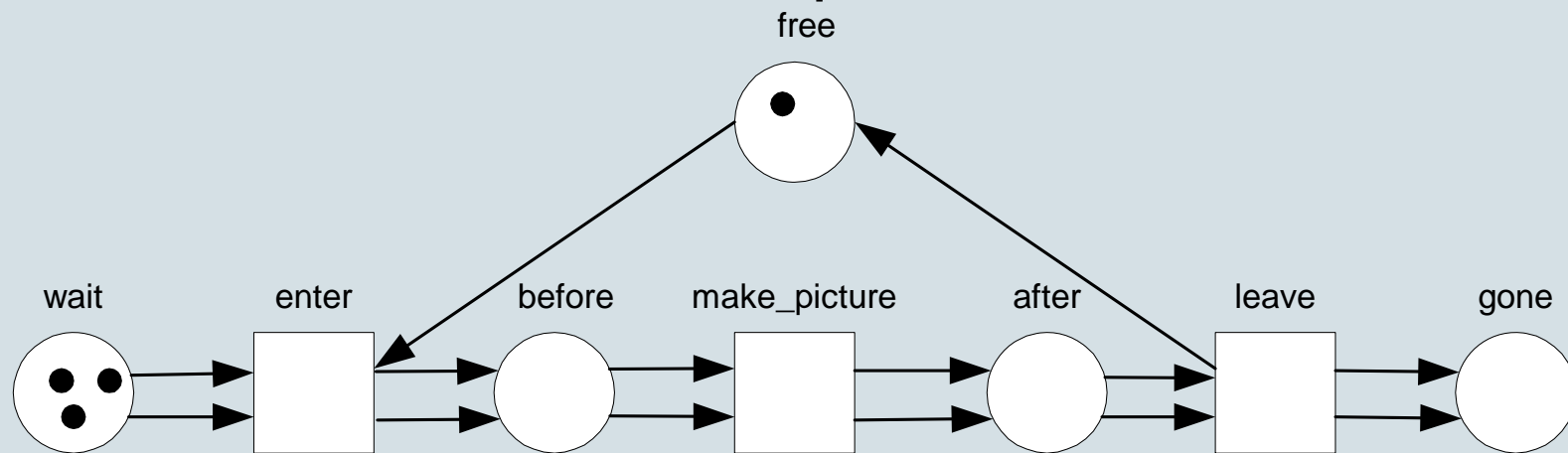
http://is.tm.tue.nl/staff/wvdaalst/workflowcourse/exercises/exercise2_3.htm



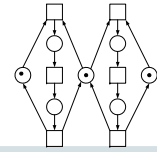


Multiple arcs connecting two nodes

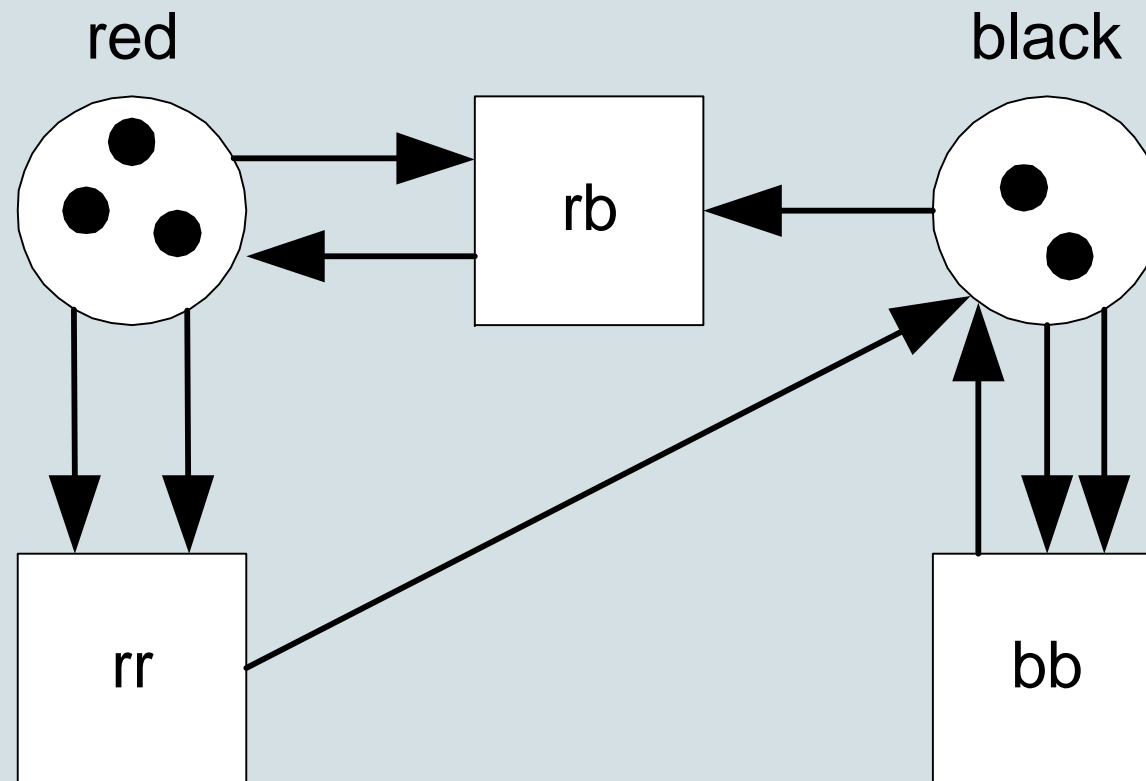
- The number of arcs between an input place and a transition determines the number of tokens required to be enabled.
- The number of arcs determines the number of tokens to be consumed/produced.



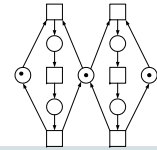
(c) Wil van der Aalst, Eindhoven University of Technology



Example: Ball game

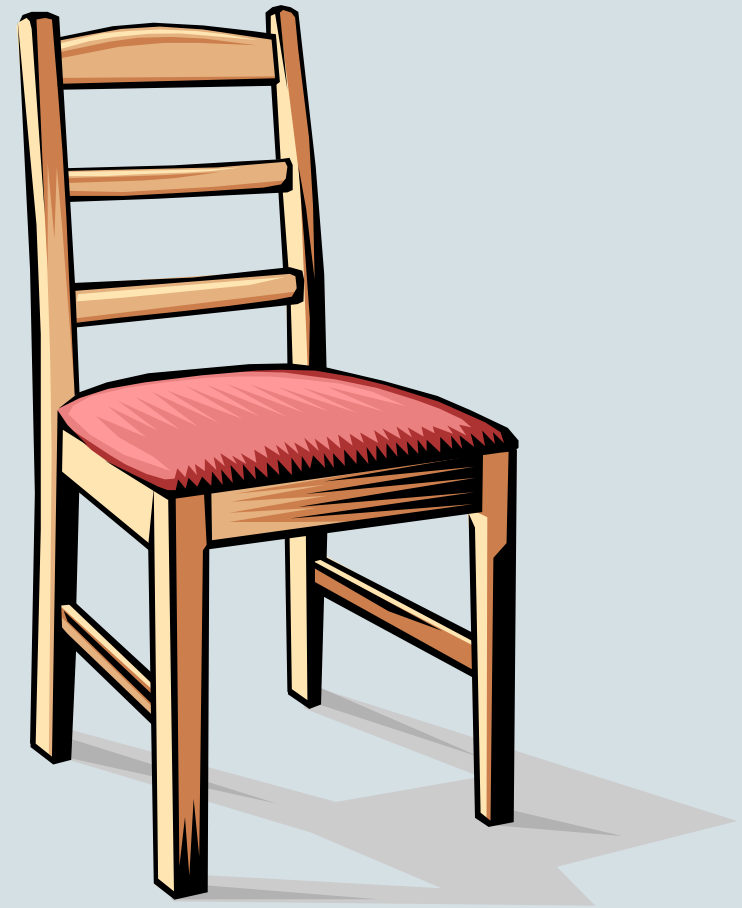


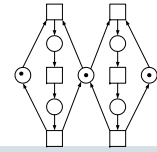
Which transition(s) is/are enabled?



Exercise: Manufacturing a chair

- Model the manufacturing of a chair from its components: 2 front legs, 2 back legs, 3 cross bars, 1 seat frame, and 1 seat cushion as a Petri net.
- Select some sensible assembly order.

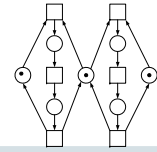




Exercise: Burning alcohol.

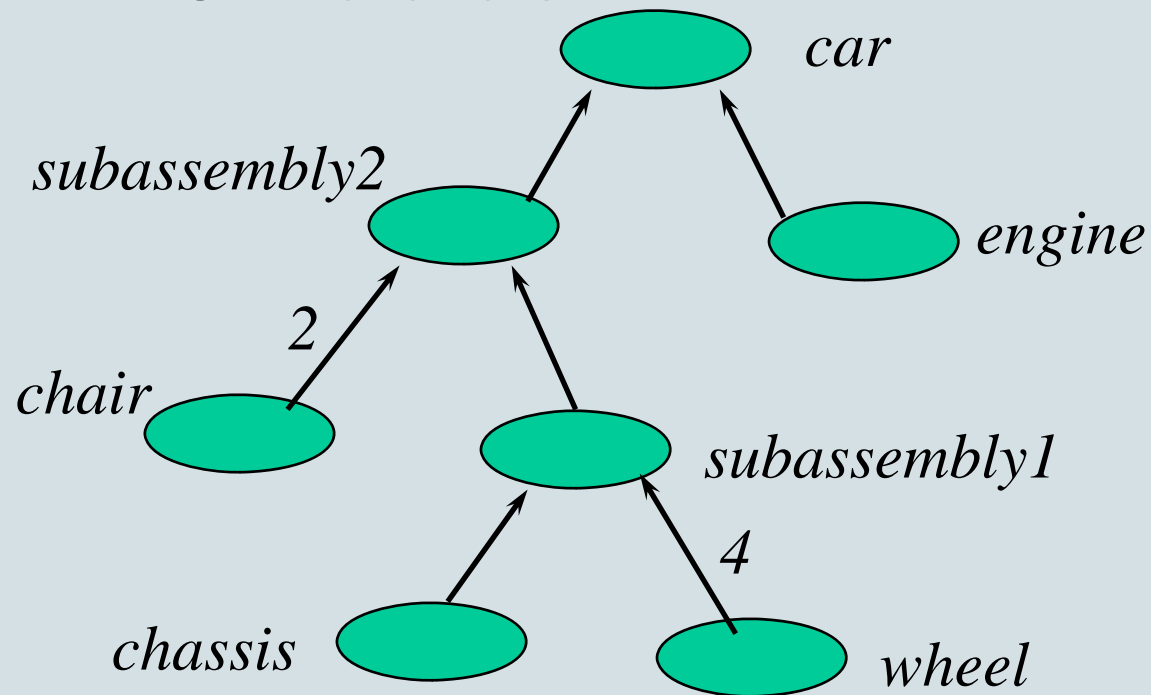
- Model $\text{C}_2\text{H}_5\text{OH} + 3 * \text{O}_2 \Rightarrow 2 * \text{CO}_2 + 3 * \text{H}_2\text{O}$
- Assume that there are two steps: first each molecule is disassembled into its atoms and then these atoms are assembled into other molecules.

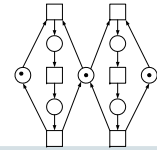




Exercise: Manufacturing a car

- Model the production process shown in the Bill-Of-Materials.





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.