COMP201 Software Engineering I
Lecture 11 – Petri Nets

Lecturer: T Carroll

Email: Thomas.Carroll2@Liverpool.ac.uk

Office: G.14

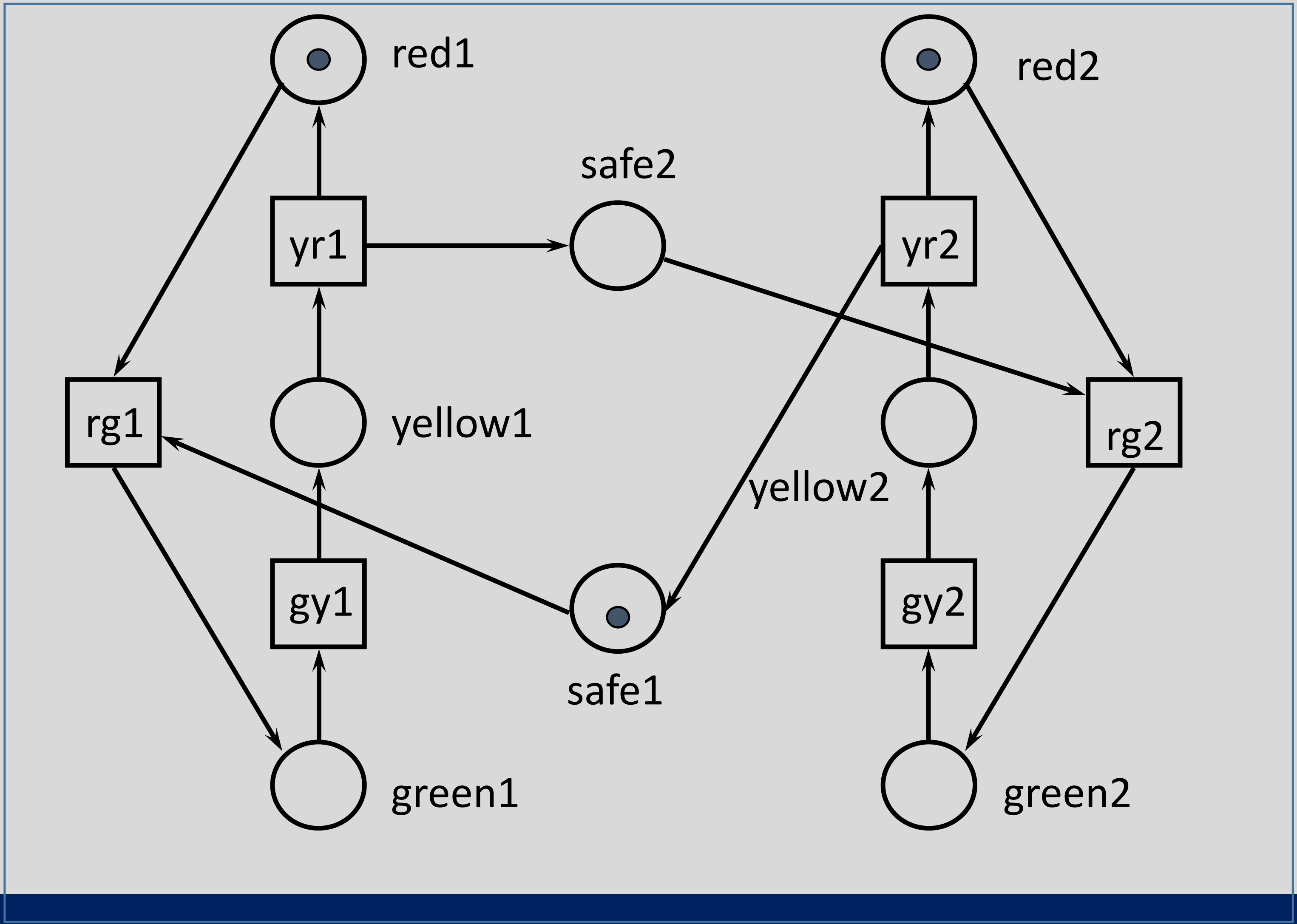
See Vital for all notes

Today

Overview

- Petri Nets
- Extensions to Petri Nets

Two Safe and Fair Traffic Lights

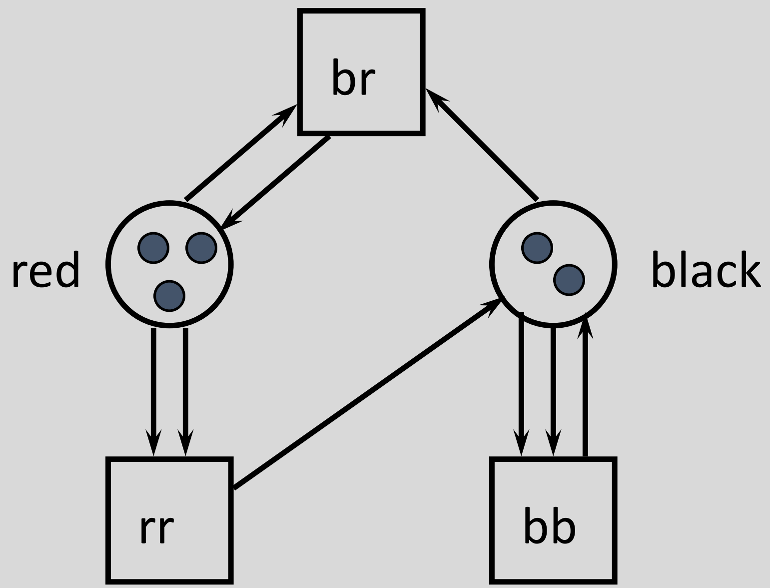


Exercise

- 1) Can you prove that the Petri net from the previous slide will never allow two red lights to be shown simultaneously?

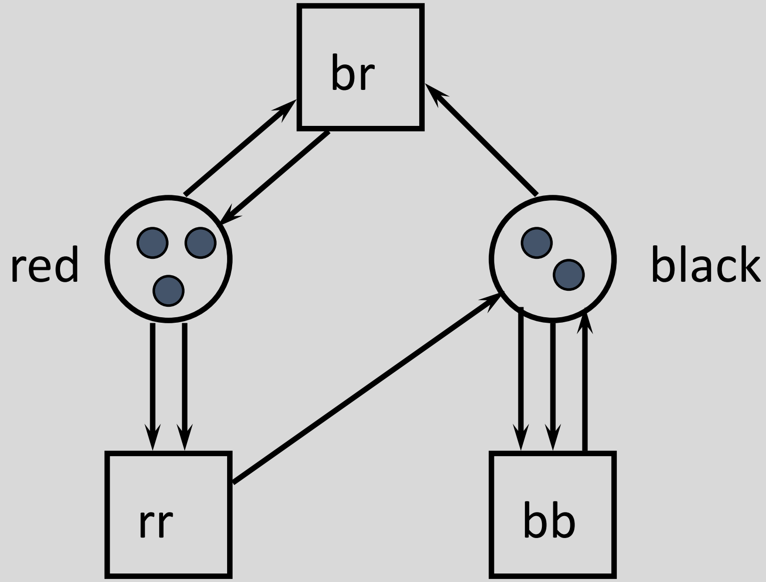
Arcs in Petri Nets

- Arc weights between objects specifies number of tokens tobe consumed/produced
- This can also be specified by number of arcs
- This can be used to model (dis)assembly processes.



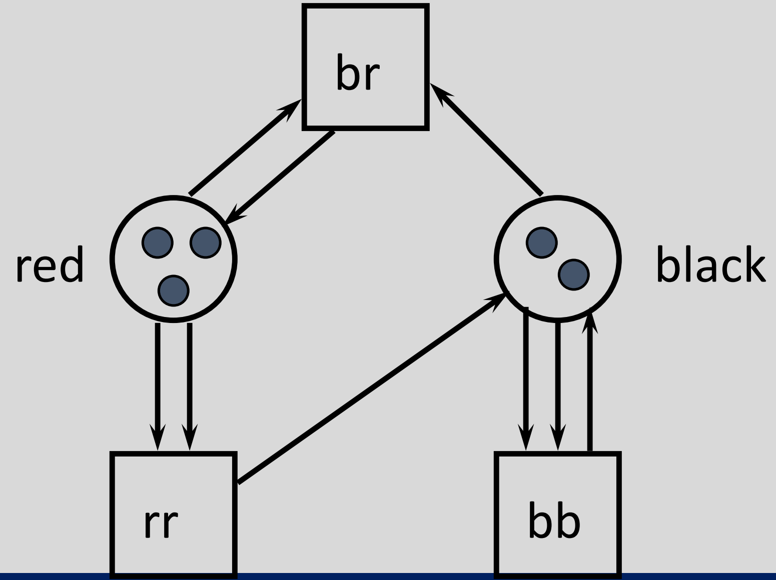
Some Definitions

- Current state (also called current marking) - The configuration of tokens over the places.
- Reachable state - A state reachable form the current state by firing a sequence of enabled transitions.
- Deadlock state - A state where no transition is enabled.

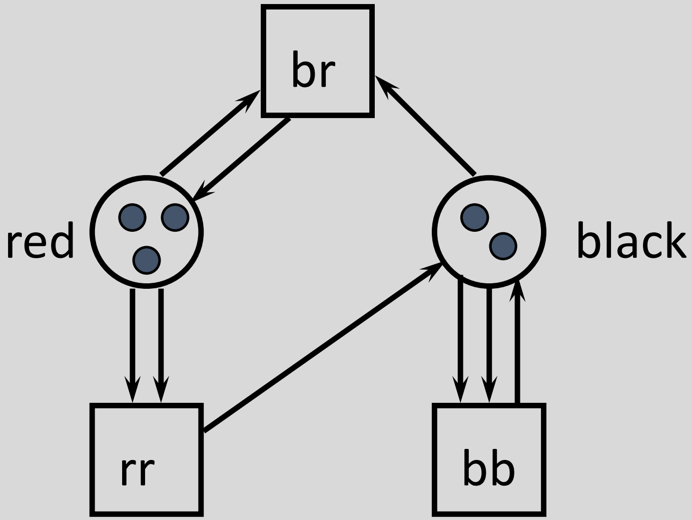


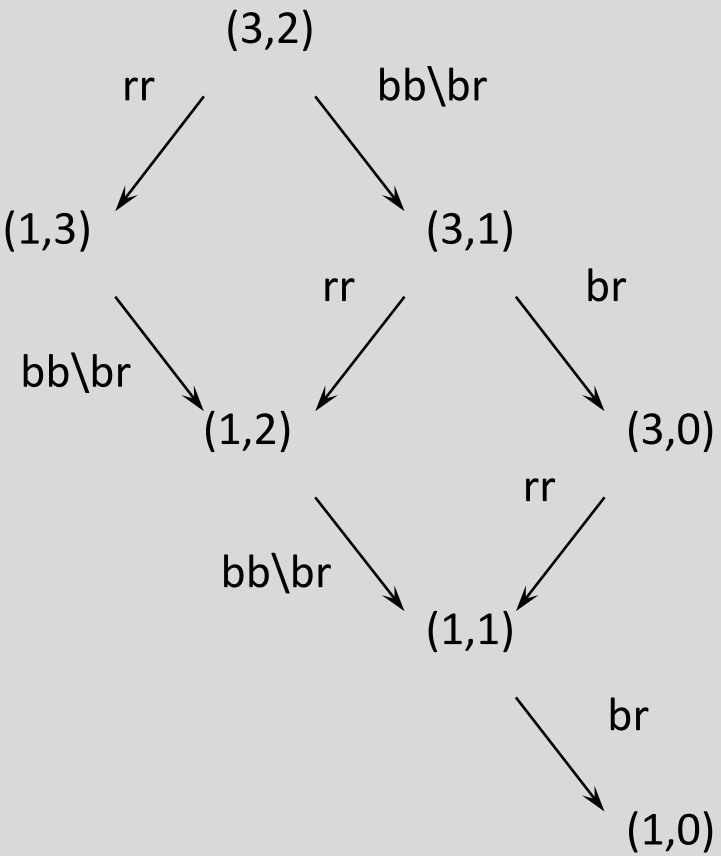
Notation

- If we write the places in some fixed order (red, black say), then we can use a tuple: (n,m) to denote the number of tokens in each corresponding place (n tokens in “red” and m tokens in “black”).
- The example below is thus in state (3,2).
- After firing transition “rr”, it will move to state (1,3) etc..



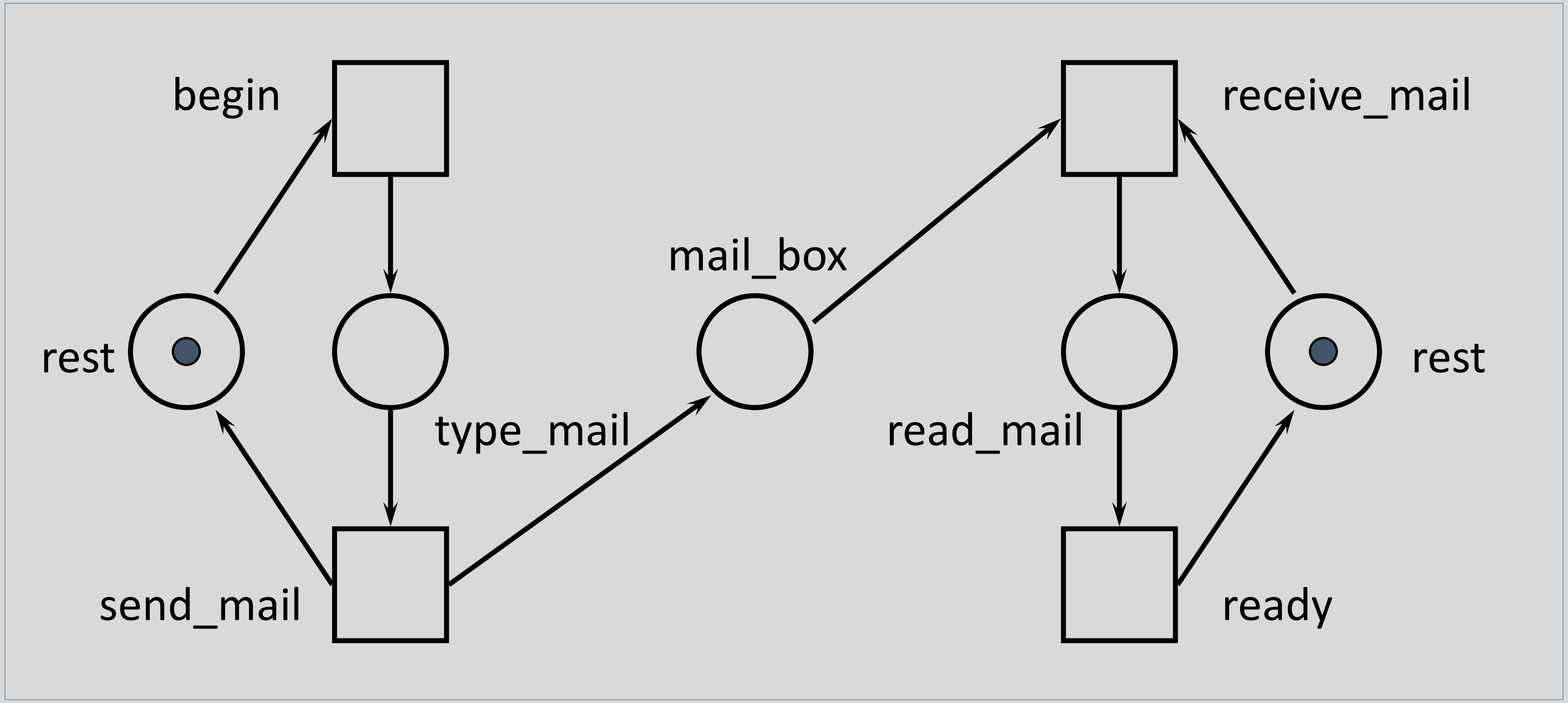
- 7 reachable states, 1 deadlock state.





Exercise: Readers and Writers

- How many states are reachable?
- Are there any deadlock states?

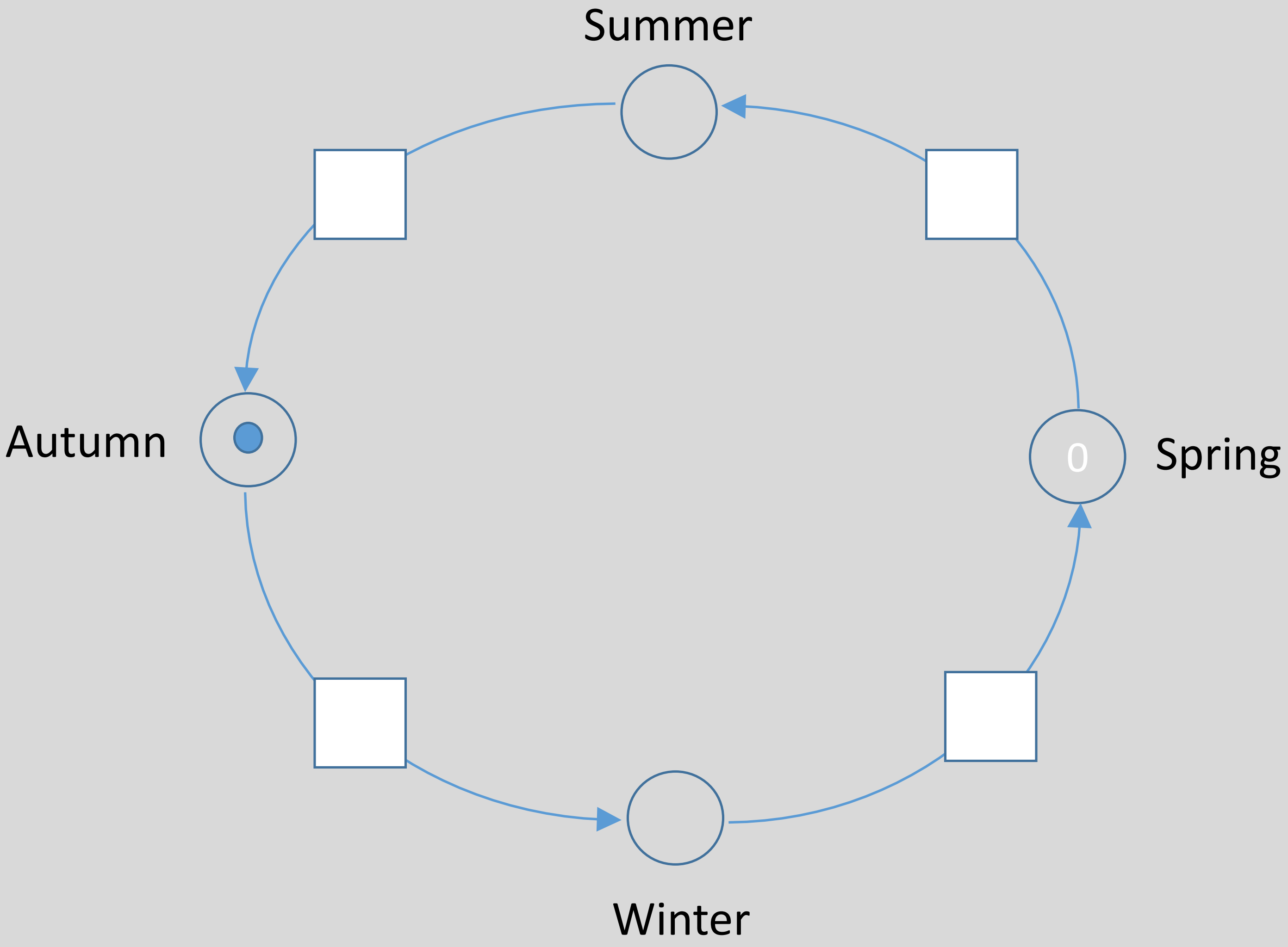


Exercise

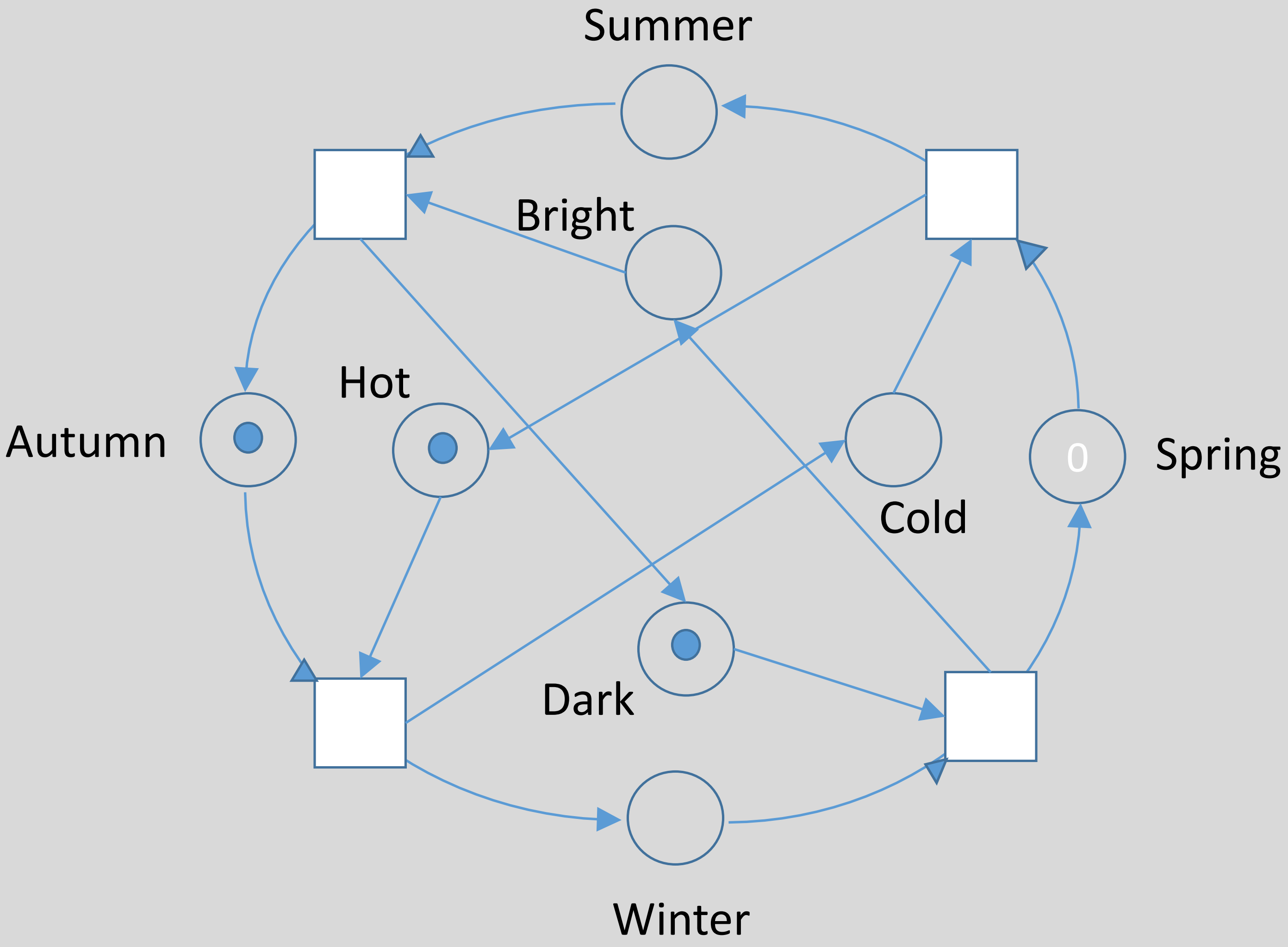
The Four Seasons

- Let us try to model the four seasons of the year together with their properties by a Petri net.
- We would like to denote the current season {spring, summer, autumn, winter}, the temperature {hot, cold} and the light level {bright, dark}.
- As a first step, let us model the seasons (with a token to represent that it is currently autumn).

The Four Seasons



The Four Seasons



High-Level Petri Nets

In practice, classical Petri nets have some modelling problems:

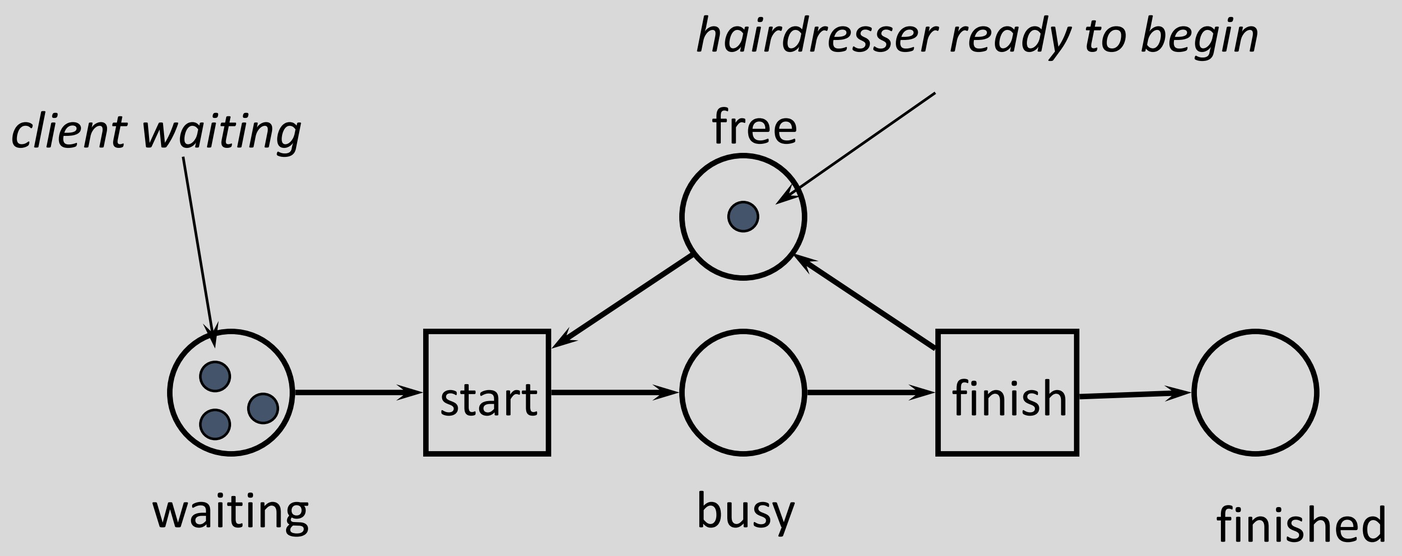
- The Petri net becomes too large and too complex.
- It takes too much time to model a given situation.
- It is not possible to handle time and data.

Therefore, we use high-level Petri nets, i.e. Petri nets extended with:

- colour
- time
- hierarchy

Example - High-Level Petri Nets

To explain the three extensions we use the following example of a hairdresser's salon:

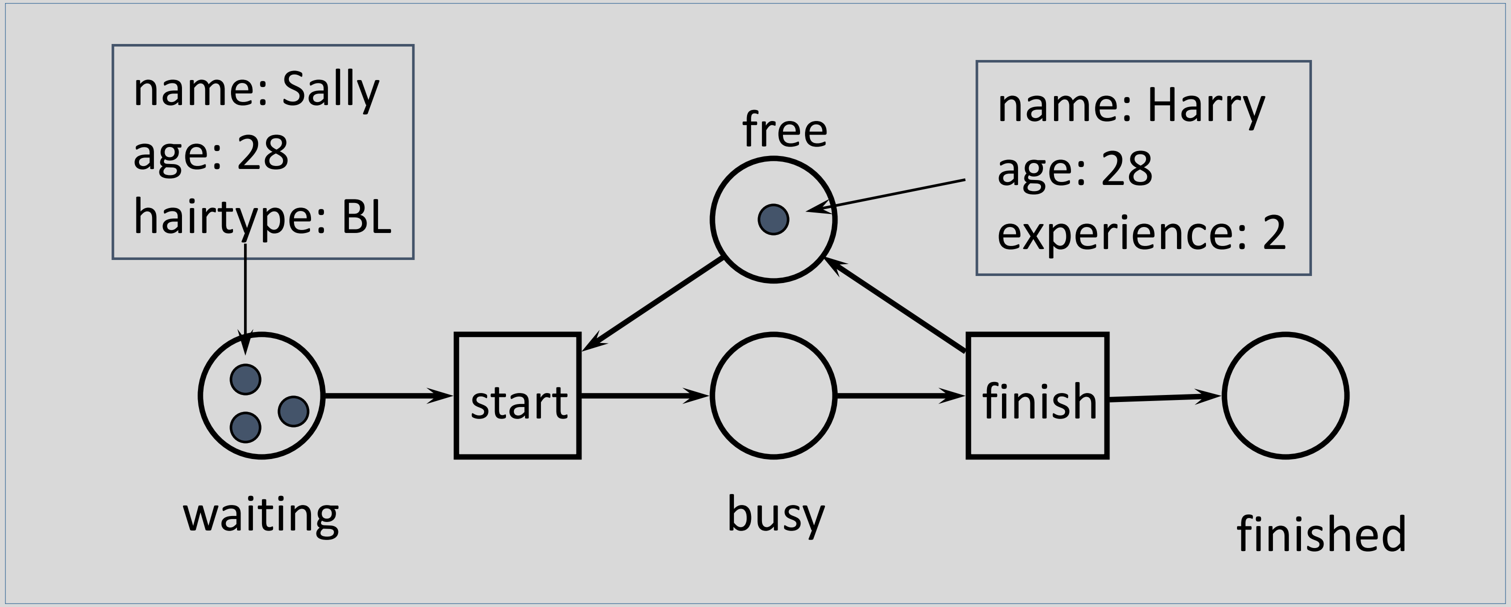


Note how easy it is to model the situation with multiple hairdressers..

The Extension with Colour

A token often represents an object having all kinds of attributes.

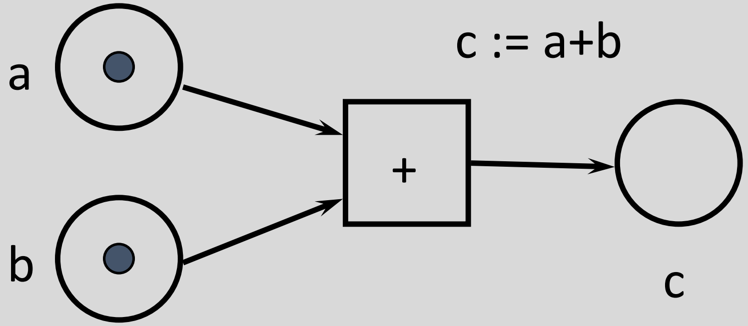
Therefore, each token has a **value** (colour) with refers to specific features of the object modelled by the token.

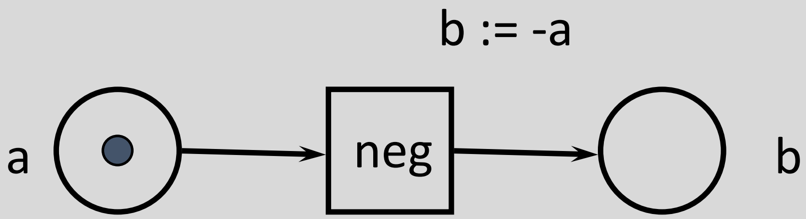


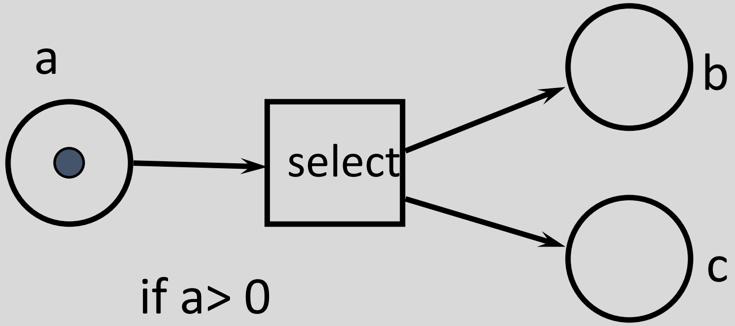
The Extension with Colour

- Each transition has an (in)formal specification which specifies:
  - the number of tokens to be produced,
  - the values of these tokens,
  - and (optionally) a precondition.
- The complexity is divided over the network and the values of tokens.
- This results in a compact, manageable and natural process description.

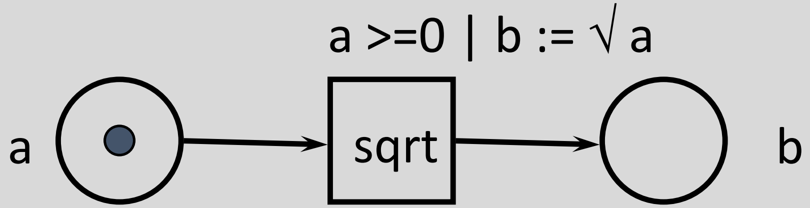
Examples







if a> 0
then b:= a
else c:=a
fi

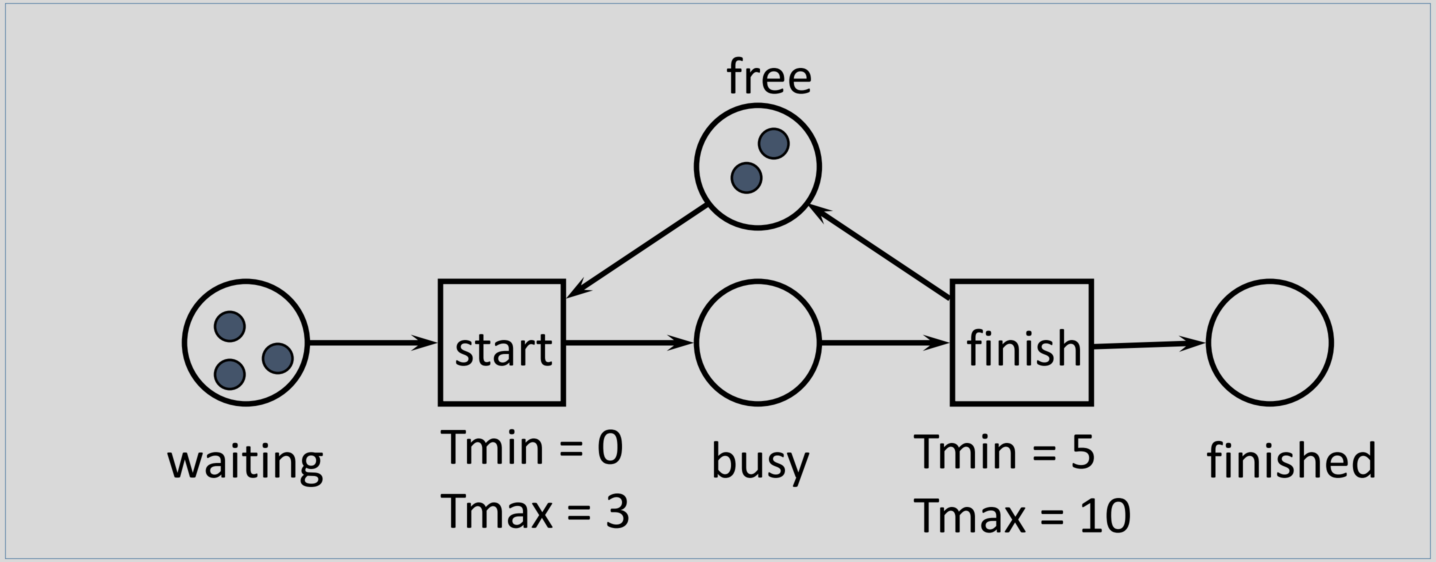


Exercise:
calculate √ |a+b| using these buiding blocks

The Extension with Time

To analyse performance, we must model durations, delays, etc.

A timed Petri net associates a pair t_{min} and t_{max} with each transition (there are other possible definitions for timed Petri net, but we shall only consider this one).

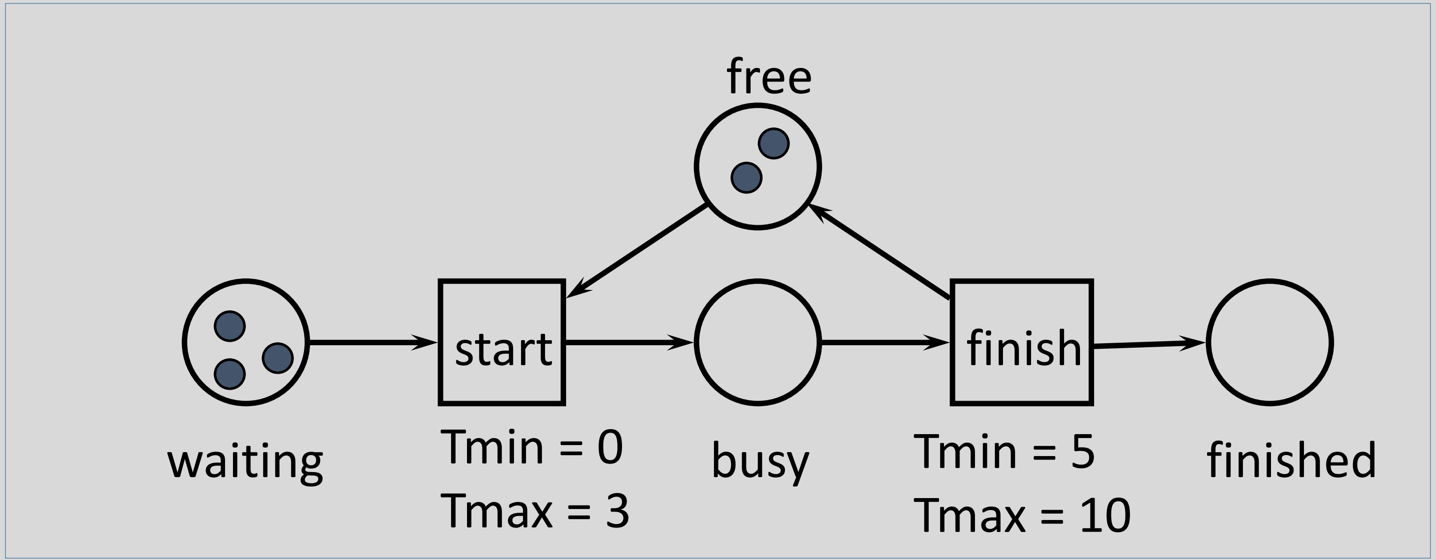


The Extension with Time

The values t_{min} and t_{max}, tell us the minimum and maximum time that a transition will take to fire once enabled.

This allows us to model performance properties of the system

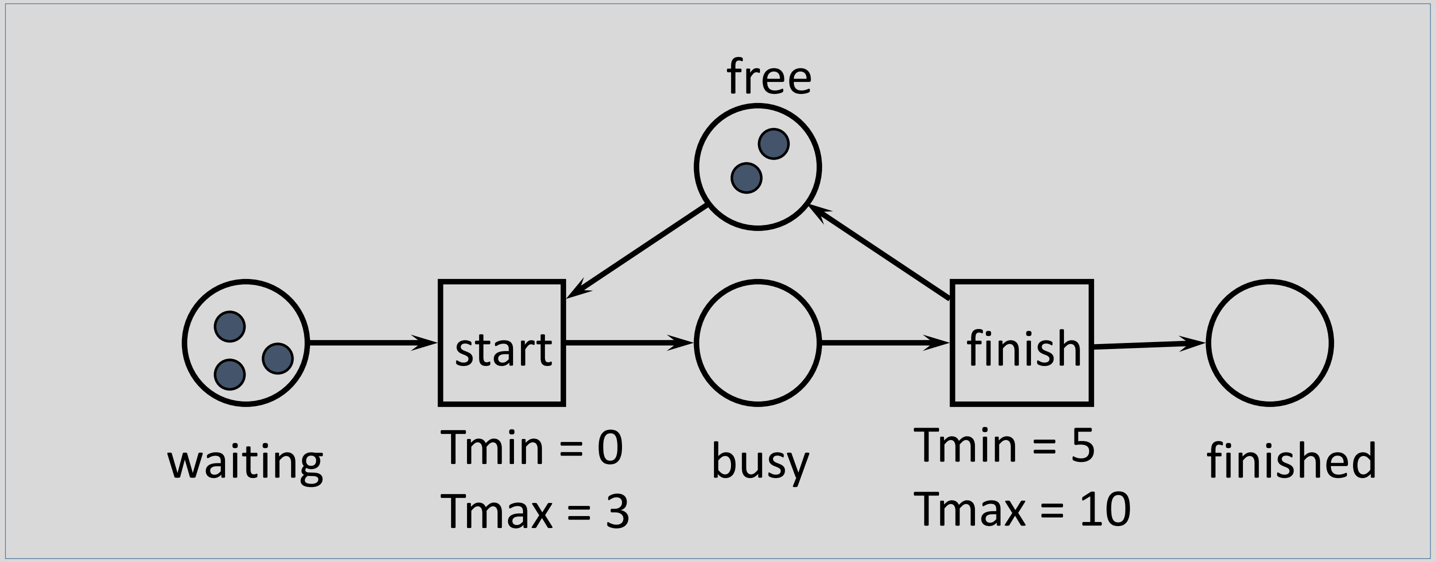
The analysis of such systems may be more difficult.



The Extension with Time

Question: What is the minimum/maximum time for all three people to have their hair cut in this system?

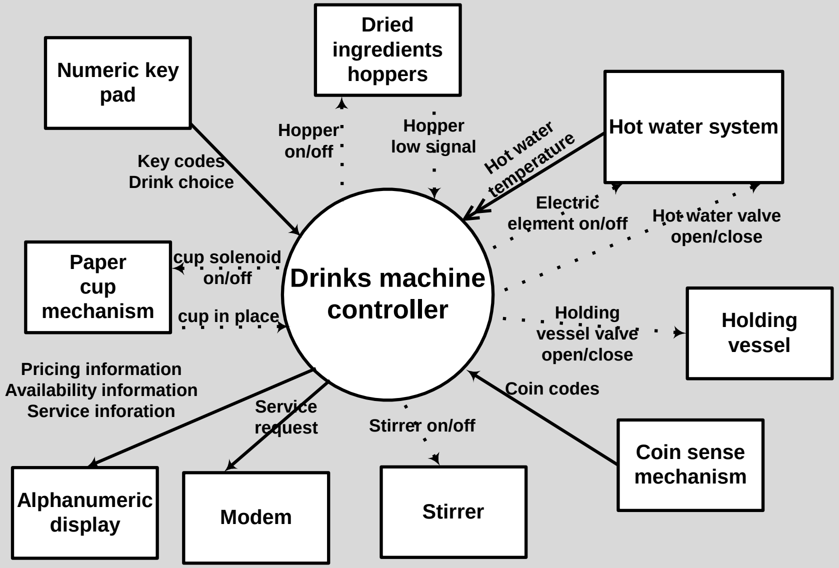
(Harder) Question: What about with n clients and m hairdressers? Is there a general formula for the required time?

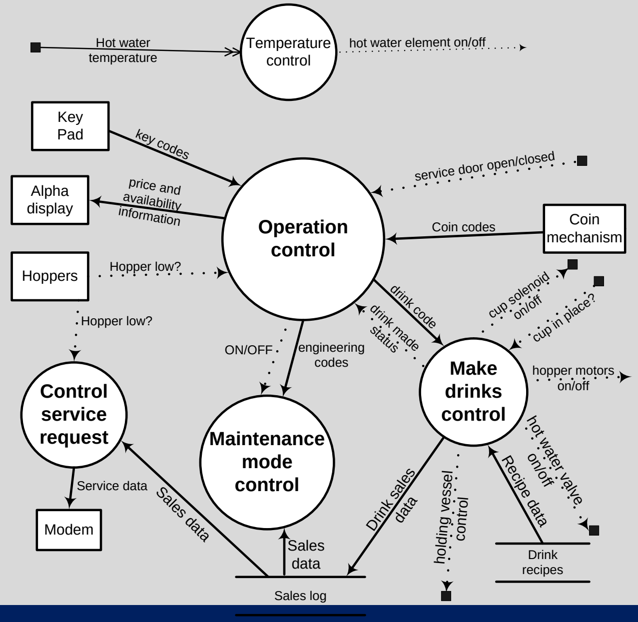


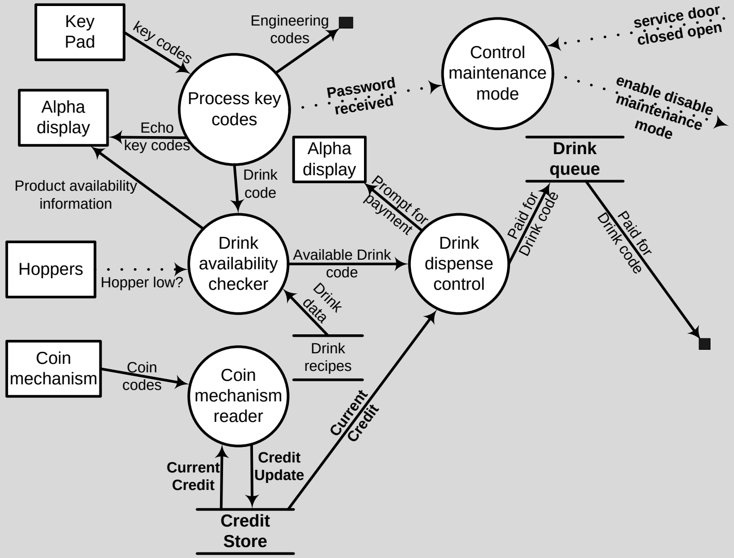
Exercise

The Extension with Hierarchy

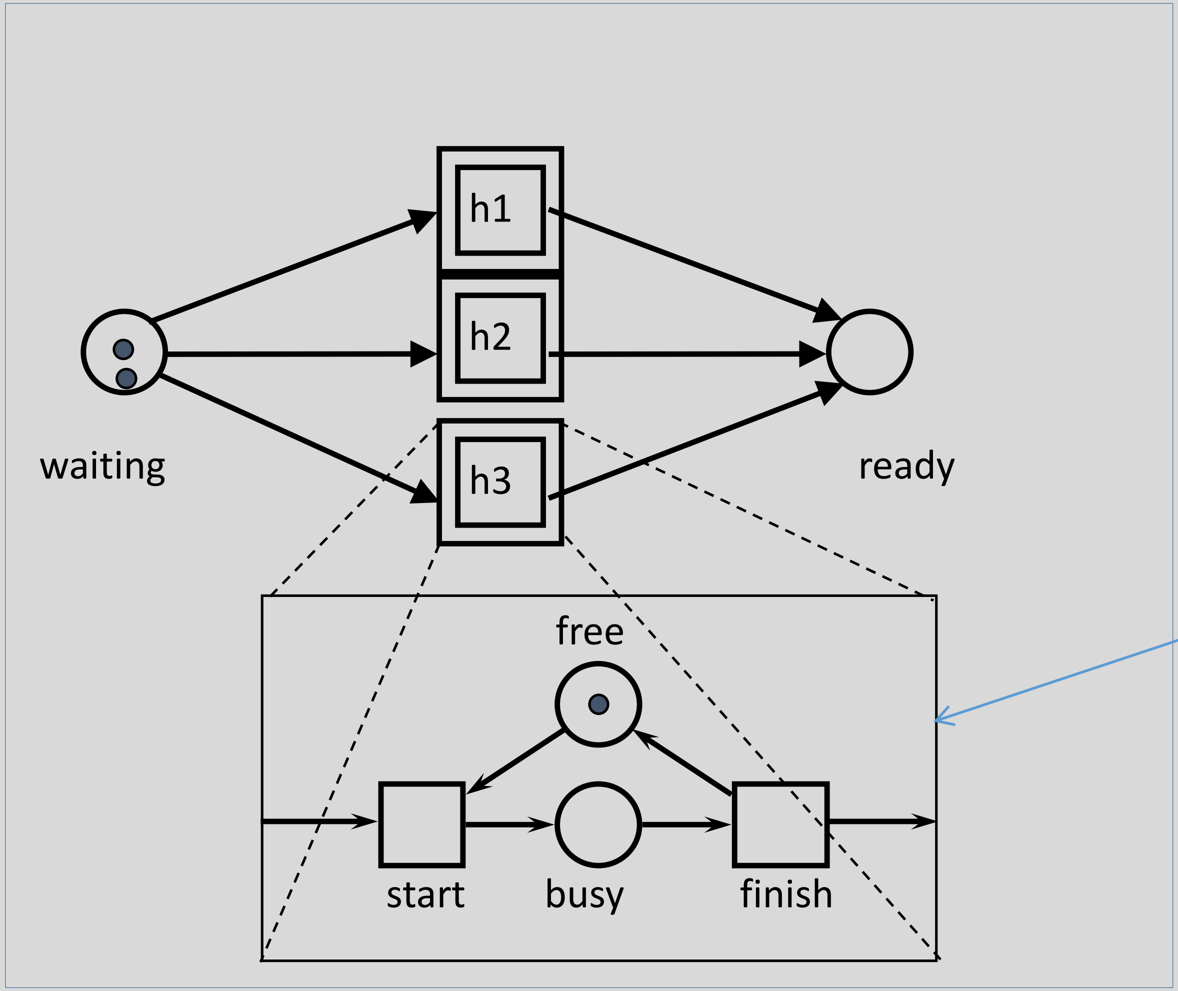
- A hierarchy is a mechanism to structure complex Petri nets comparable to Data Flow Diagrams.
- A **subnet** is a net composed out of places, transitions and other subnets.
- This allows us to model a system at different levels of abstraction and can reduce the complexity of the model.





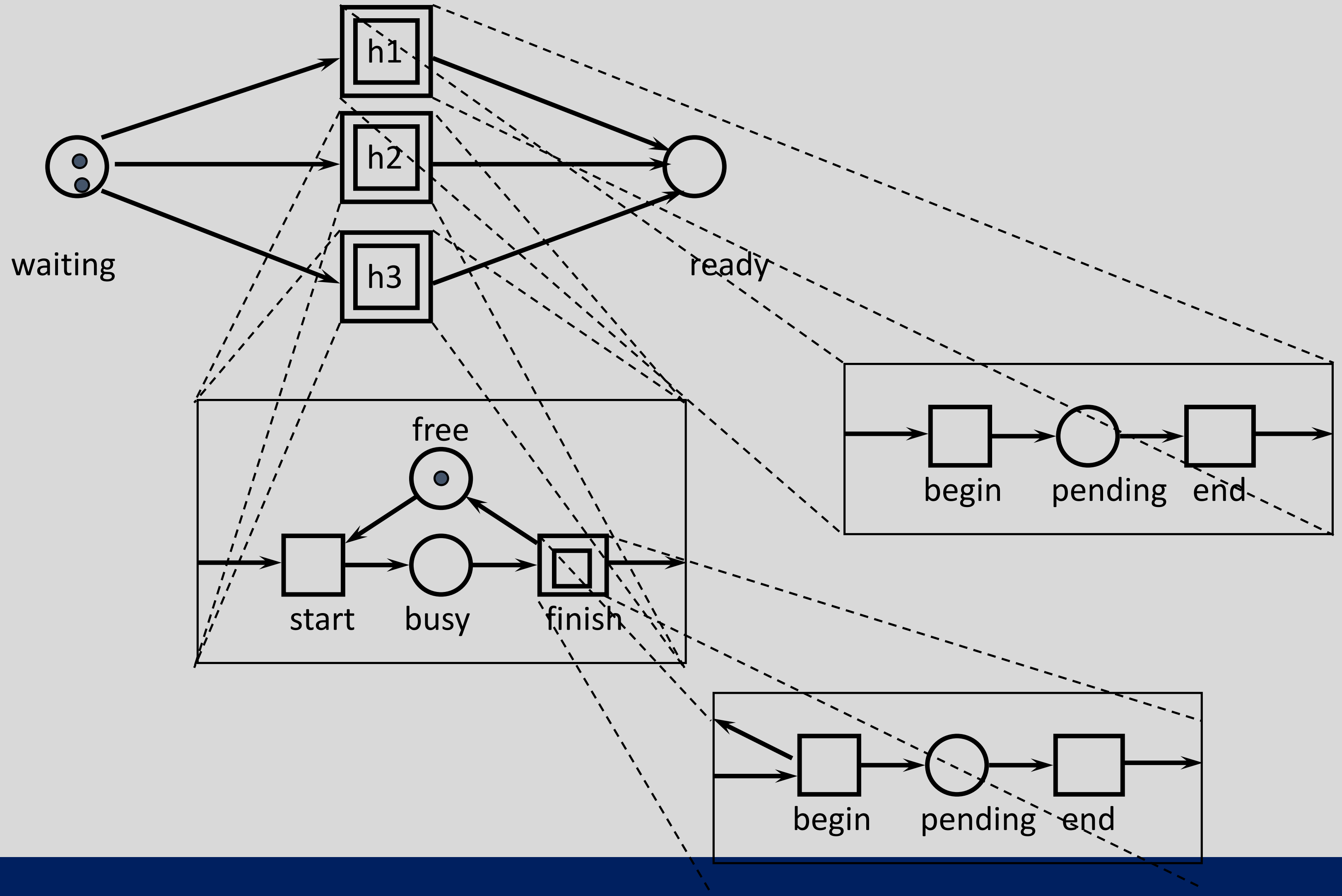


The Extension with Hierarchy



Here we expand subnet h3..

Exercise: Remove Hierarchy



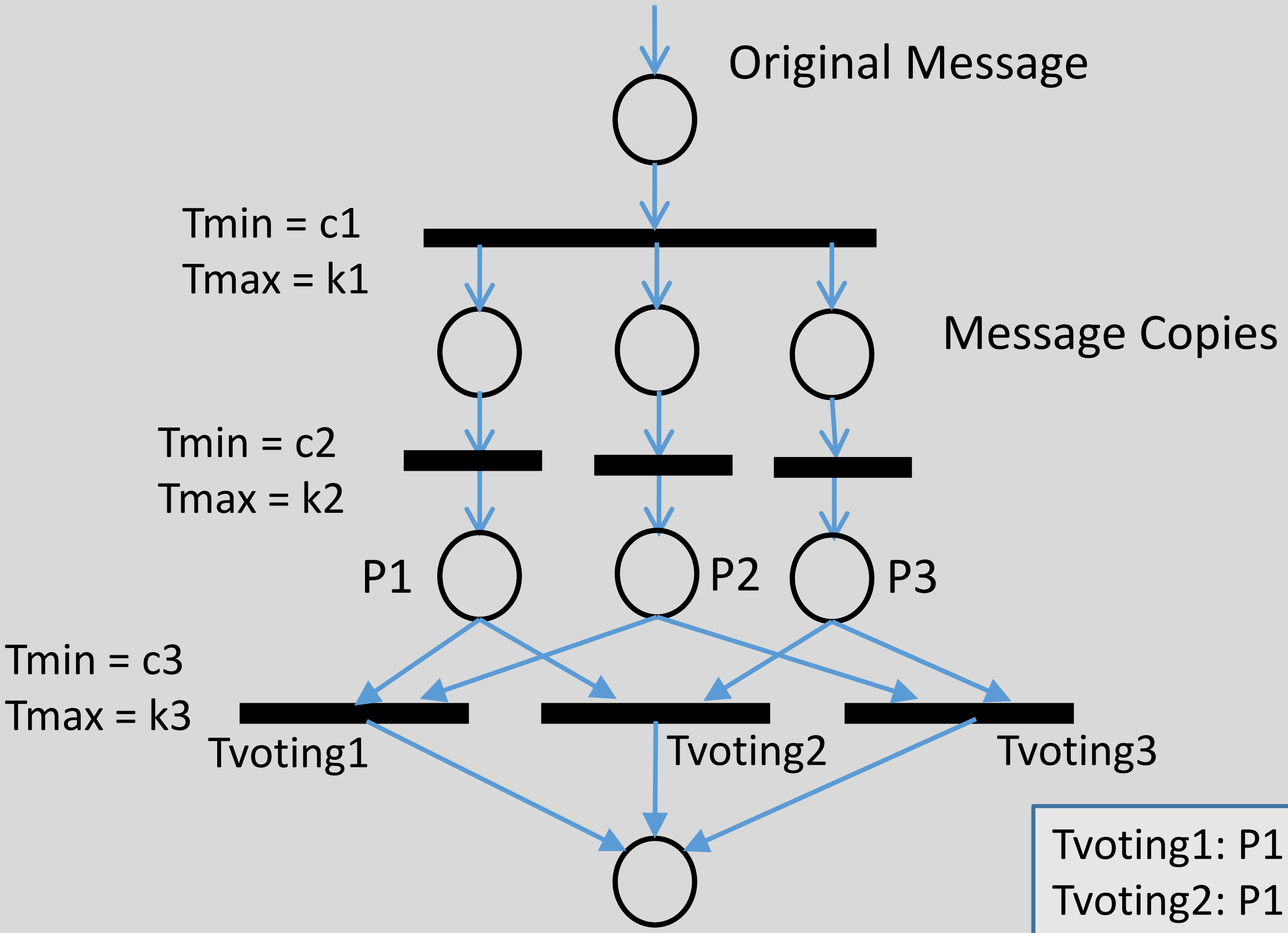
Excercise

Another Example

- Recall the following example of an informal specification from a critical system [1] :
  - The message must be triplicated. The three copies must be forwarded through three different physical channels. The receiver accepts the message on the basis of a two-out-of-three voting policy.
- Questions: Can you identify any ambiguities in this specification?
- How could we model this system with a Petri net?

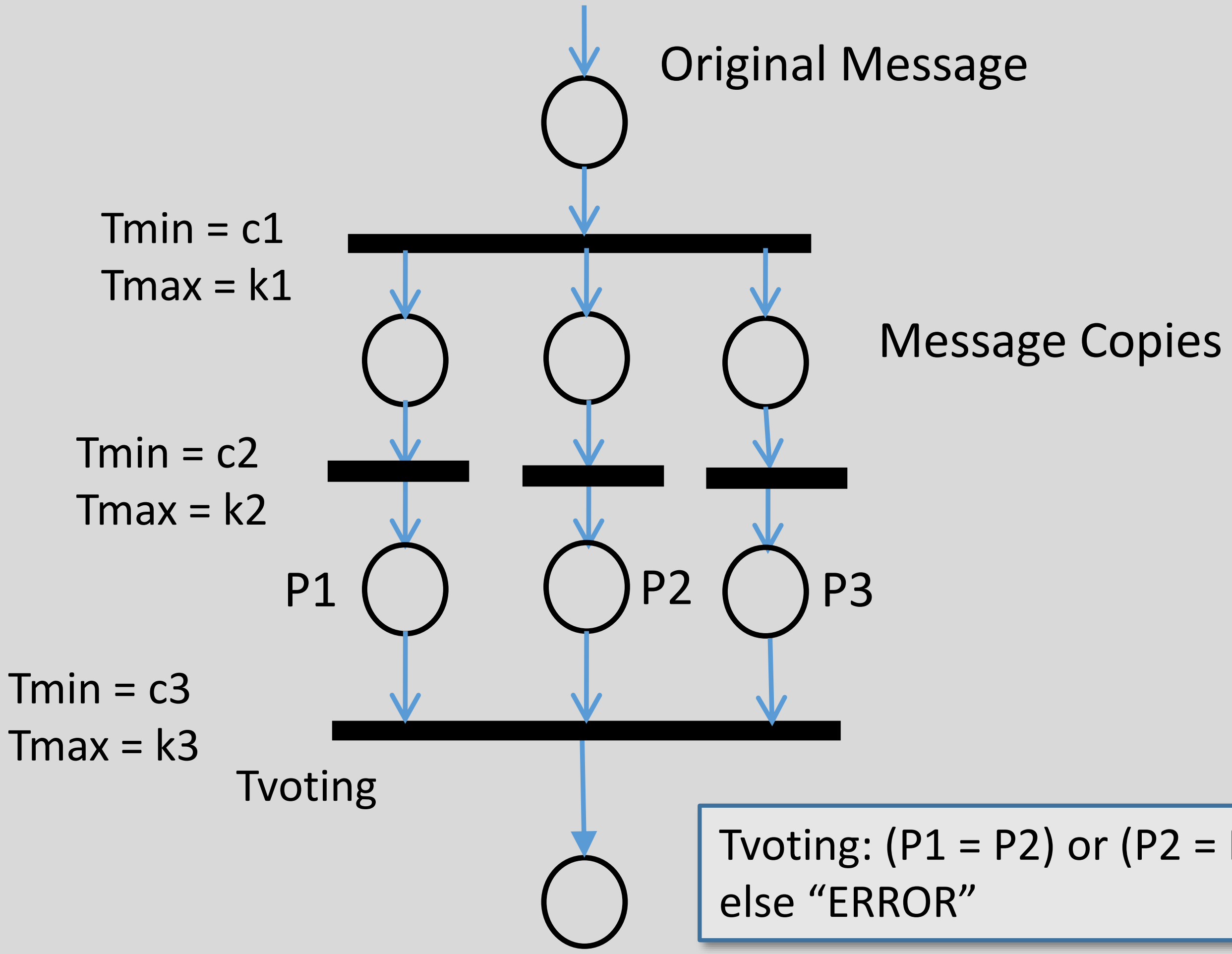
[1] - C. Ghezzi, M. Jazayeri, D. Mandrioli, “Fundamentals of Software Engineering”, Prentice Hall, Second Edition, page 196 - 198

Message Triplication



Tvoting1: P1 = P2
Tvoting2: P1 = P3
Tvoting3: P2 = P3

Message Triplication (2)



Tvoting: (P1 = P2) or (P2 = P3) or (P1 = P3)
else “ERROR”

A Final Note on Petri Nets

- We can see from the previous example that the ambiguity (or impreciseness) in the informal specification for the message triplication protocol is clearly highlighted by the more formal Petri net model.
- We can also perform some analysis on the model itself, for example to see if certain “bad” states ever occur or if deadlock/livelock is possible in the model.
- Finally we can represent timing constraints (to encode even more constraints on the system) and use hierarchical models to show different levels of abstration.

A Final Note on Petri Nets

- Imagine modelling the elevator system of a skyscraper which contains three elevators and twenty floors.
- What would be some of the advantages of using a Petri net model for this?
  - We can ensure if someone at a floor pushes the lift button (up or down), the elevator will eventually come.
  - We can attempt to model the timing constraints of the system (Timed Petri net).
  - We can also use hierarchies to simplify the system.
  - Finally we could try to optimize the model in some way if its performance is not optimal.
  - Etc..

Lecture Key Points

- Petri nets have Arcs, Places and Transitions.
- The state or marking of a net is an assignment of tokens to places.
- Petri nets are **non-deterministic** and thus may be used to model discrete distributed systems.
- They have a well defined semantics and many variations and extensions of Petri nets exist.