Real Time Systems – SS2016

Prof. Dr. Karsten Weronek

Faculty 2

Computer Science and Engineering

Petri net

Issue: How to describe concurrent processes?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Concurrent process **does not mean** „at the same time".

Concurrent **means** „independent".

However, when two concurrent processes
need to use a resource at the same time
then they become dependent.

To analyse concurrent processes a model is needed to descibe the states and transitions of the system.

The model Petri-Net (similar to the finite state machine (endlicher Zustandsautomat) is able to describe and to analyse a system with concurrent processes

Petri net also P/T-net

A Petri net is a mathematical modeling language to describe descrete, mainly distributed systems.

Like industry standards such as UML, activity diagrams, Business Process Model and Notation (BPMN) and Event-driven Process Chains (EPCs), Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. Unlike these standards, Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis.

Modeling

A Model is

- a simplified picture of the real world
- a des of functionality
- Unabiguous and complete
- easy to understand
- Making an abstraction of the details

Purpose

- Basic for optimisation. Validation, prognosis and decision finding
- Verification and simulation of system specifications

Knots and Edges (Knoten und Kanten)

Graph Theory:

In mathematics and computer science,
graph theory is the study of graphs.

Graphs are mathematical structures used to model
pairwise relations between objects.

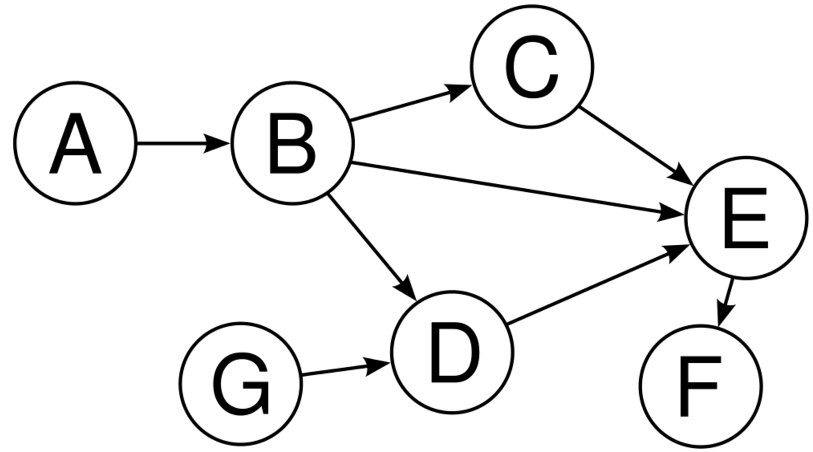
A graph in this context is made up of
vertices, nodes, or points (Knoten)
which are connected by
edges, arcs, or lines (Kanten).

A graph may be undirected, meaning that there is no distinction between the two nodes associated with each edge, or its edges may be directed from one node to another.

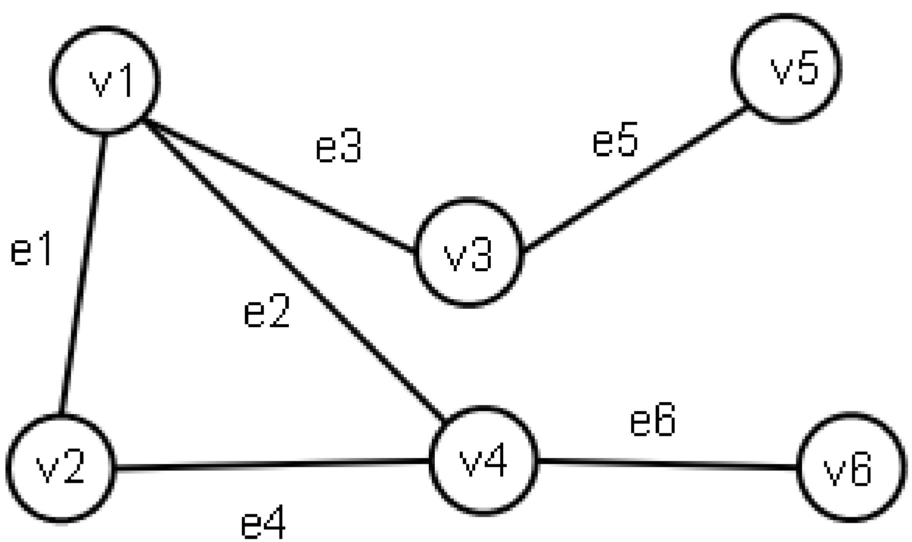
Examples of Graphs

A simple graph

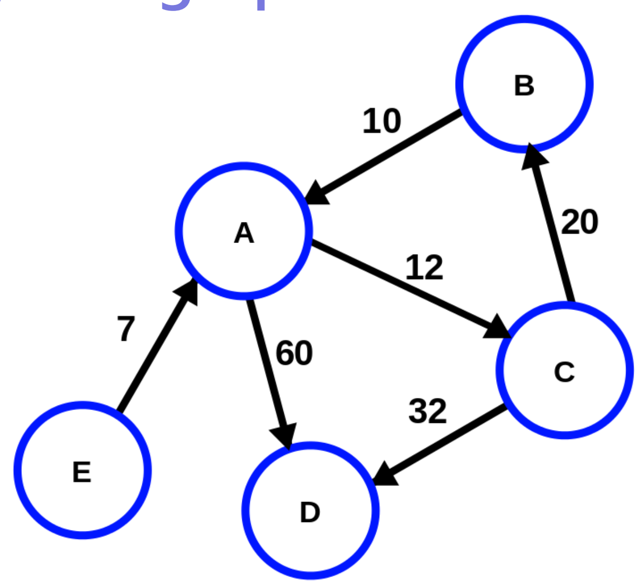
A directed graph



A named graph



A weighted graph



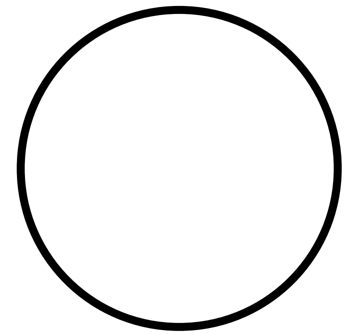
Nodes: Places and Transitions

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

In Petri nets there are two different nodes:

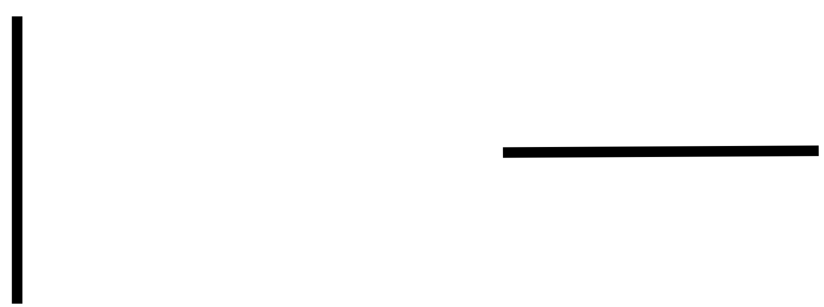
1. Places (Stellen)

   Places represent states (Zustände)



2. Transitions (Transitionen)

   Transitions represent actions or events (Ereignisse)

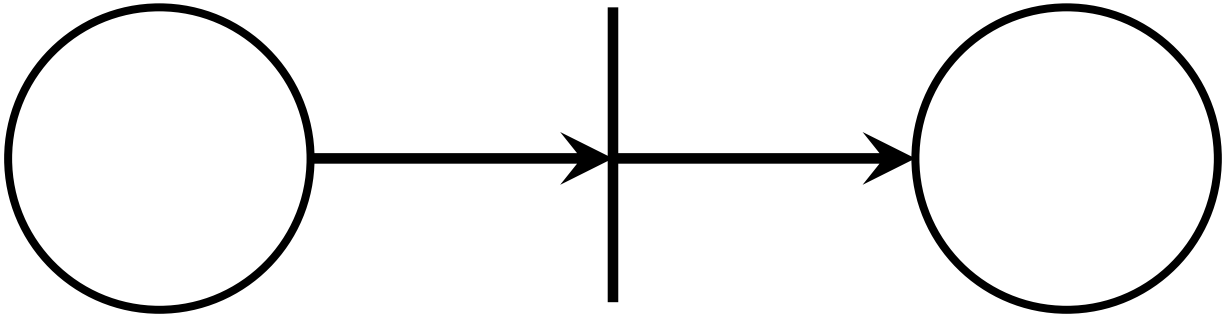


Edges

Edges (Kanten) in Petri are called arcs (Kanten, Bögen)

Arcs (flow relations) connect places and transitions where

- An arc is directed either from place to transition or from transition to place
- An arc never connects a transition with a transition
  an arc never connects a place with a place



Simple net

A elementary **net (!!!)** is a triple of sets: PN=(**P, T, F**)

where

- P is a not empty set of **p**laces
- T is a not empty set of **t**ransitions
- F is a not empty set of **f**low relations (arcs)

Marks

In contrast to Finite State Machines a Petri net is able to model a dynamic behaviour. Therefore so called marks (Marken) are defined.

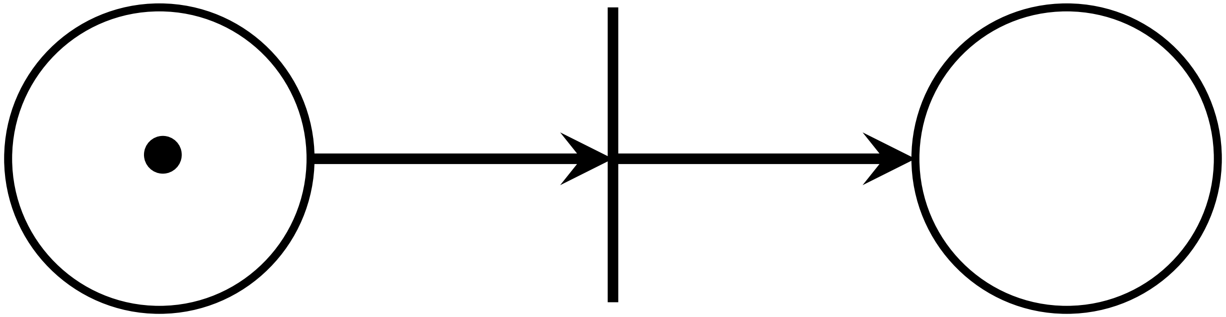
Marks can move from place to place via a transition along the related arcs, but only when the predefined switch condition is fulfilled.

Marks are also called tokens

A switch of tokens are an atomic action (not interruptible)

A switch takes no time (instantaneous)

Marks

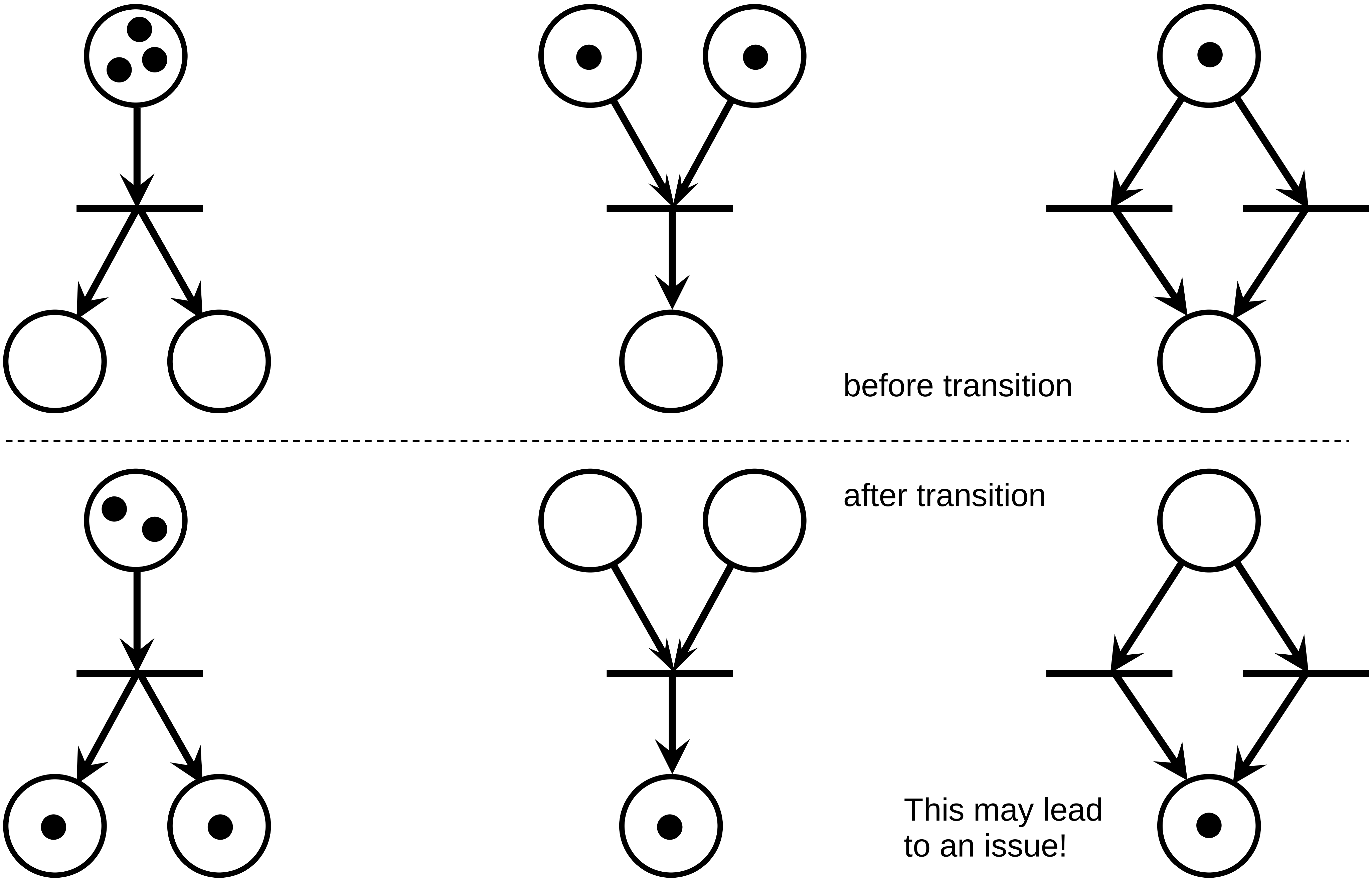


Marks

The number of marks are **not** constant. There are not balls they just represent a state. That means
out of transition marks flow as many as arcs are available.

When two arcs join at a transition the incoming marks will be joined.

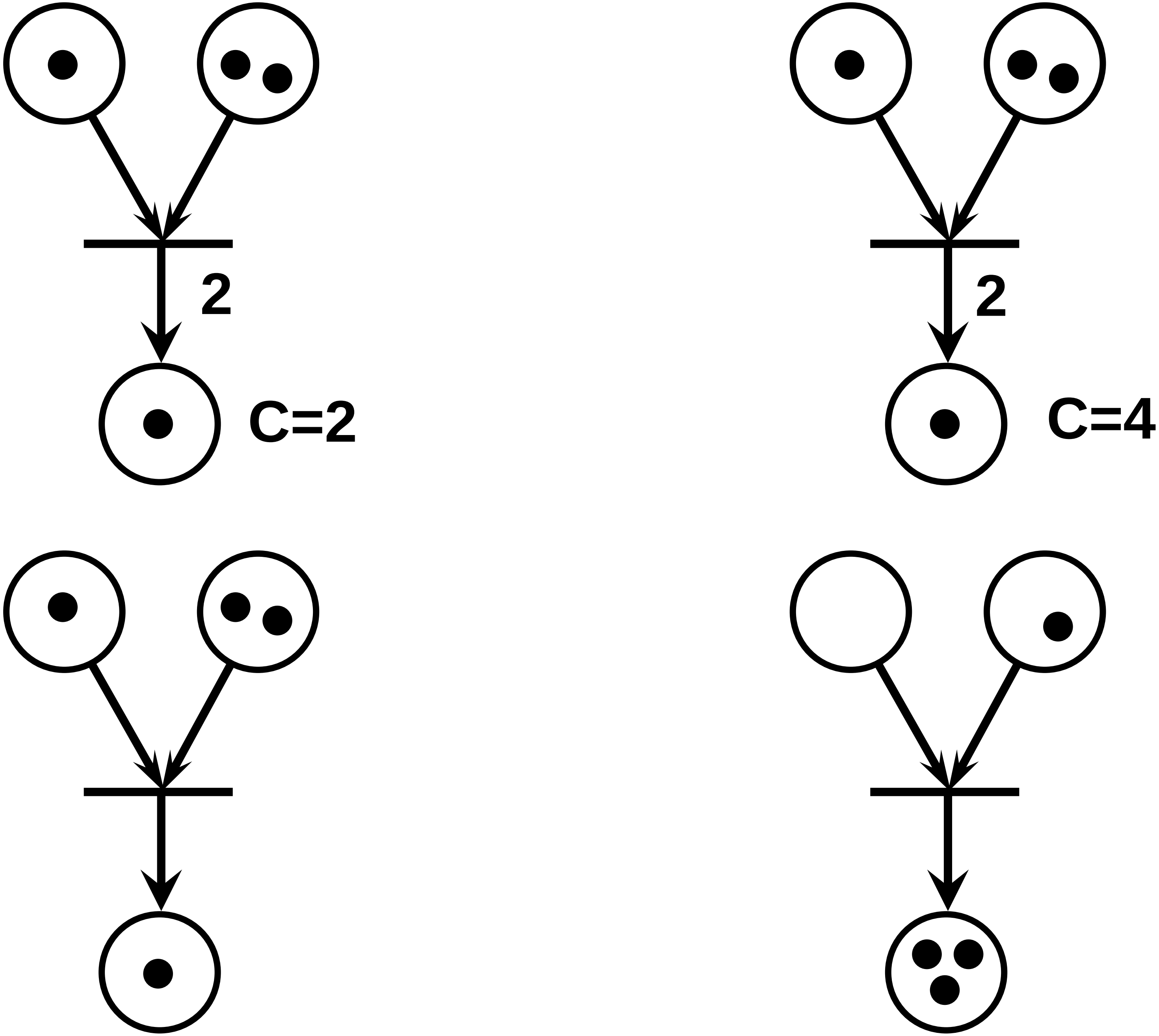
Marks


before transition
after transition
This may lead
to an issue!

Weighted transitions

Default = 1

Capacity



Petri net (one formal definition)

Petri net

A Petri net is a 6-tuple G(P,T,F,K,W,M0) with:

- P : is a finite set of places
- T : is a finite set of transitions
- F : is a set of flow relations
- C : P → N capacity of the places
- W : F → N weight of the arcs
- M0 : initial marking

A transition t in a petri net may fire(switch), if

1. all input places have enough tokens and
2. there are enough free places at the destination places.

Analysability of Petri nets

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

A Petri net has a good analysability, e.g.

- Boundedness (number of tokens is limited)
- Liveness (free of deadlocks)

Petri net: Boundedness

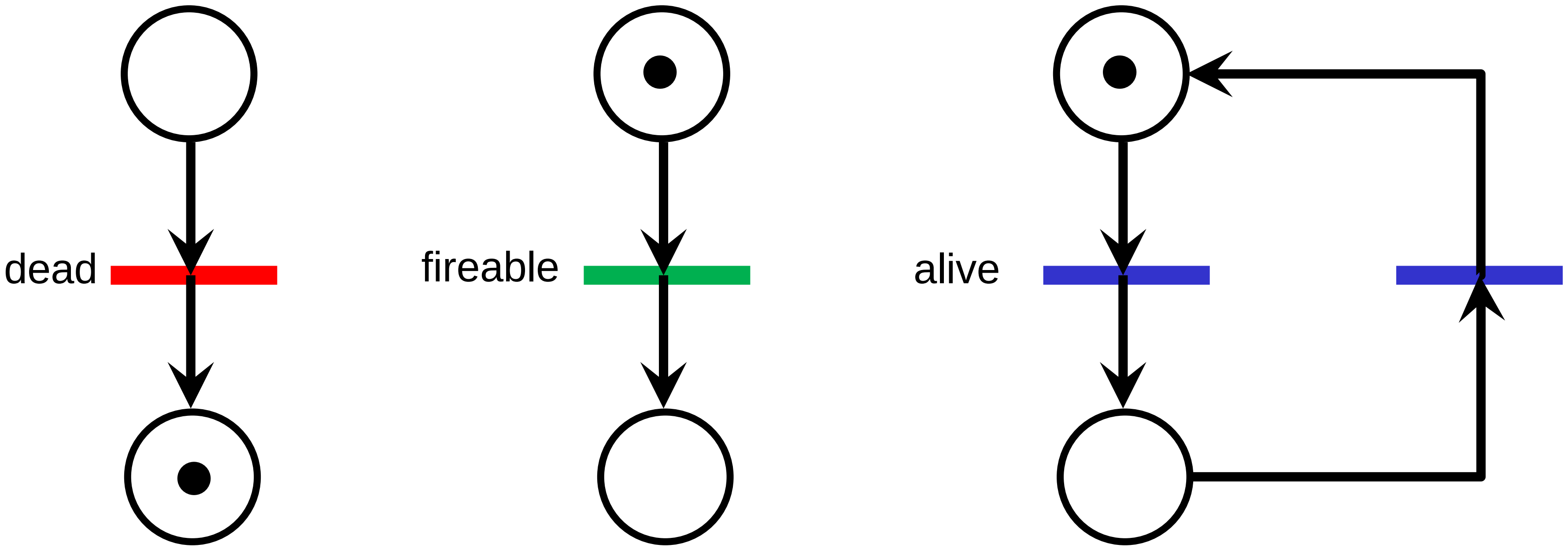
Boundedness:

- A place is bounded, if it has only one token either at the initial marking m_0 or at all reachable markings.
- A net is bounded, if all places are bounded.
- A place is called k-bounded , if
  it has only k tokens
  either at the initial marking m_0 or
  at all reachable markings.

The number of tokens is limited:

- No feedback or
- Feedback and
  the number of distributions is lower or
  equal to the number of mergers

Liveness



Petri net

Liveness:

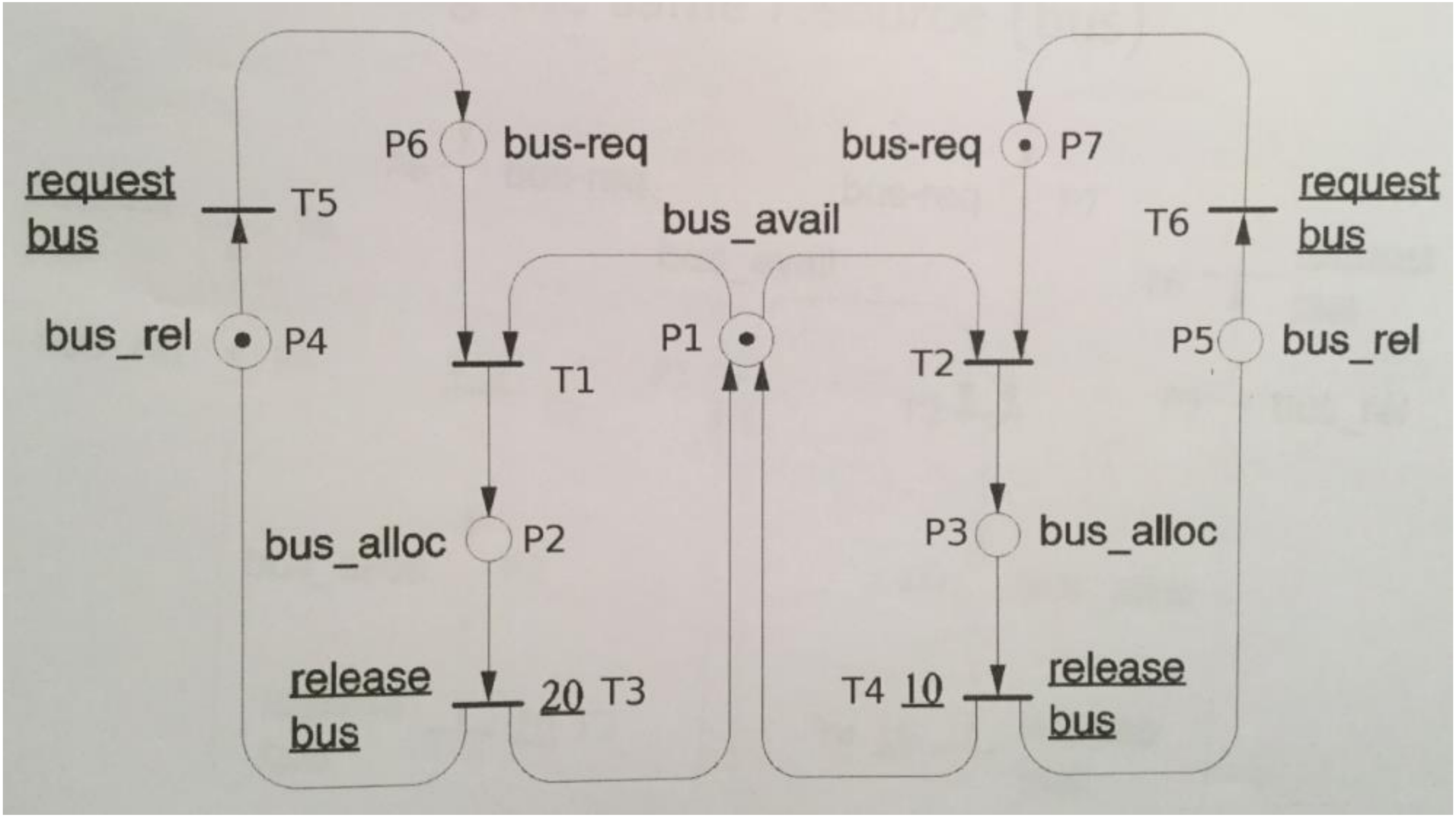
  - Liveness (free of deadlocks)

- ## Reachability
  - Using an analysis of reachability, it is tested, if all node could be reached from each node.
  - Reachability table
  - Reachability graph

Liveness: Reachability graph

1. **All active nodes are put into one state**
2. **One arc for each fireable transition**
3. **Again, after firing the transition, put all active nodes at the end of the arcs in a state**
    - Reoccurring states are reused
4. **Continue with 2, until all transitions have been executed**

Example: two task using one bus



Petri nets

Please be aware that Petri nets a mathematical construct. There are a lot of variation and a lot of use cases. There are complete series of lectures at some universities and there are internationals groups working on this topic only.

However, to model RTS it is essential
to know and understand the fundamentals and
to apply the graphical representation of Petri nets.