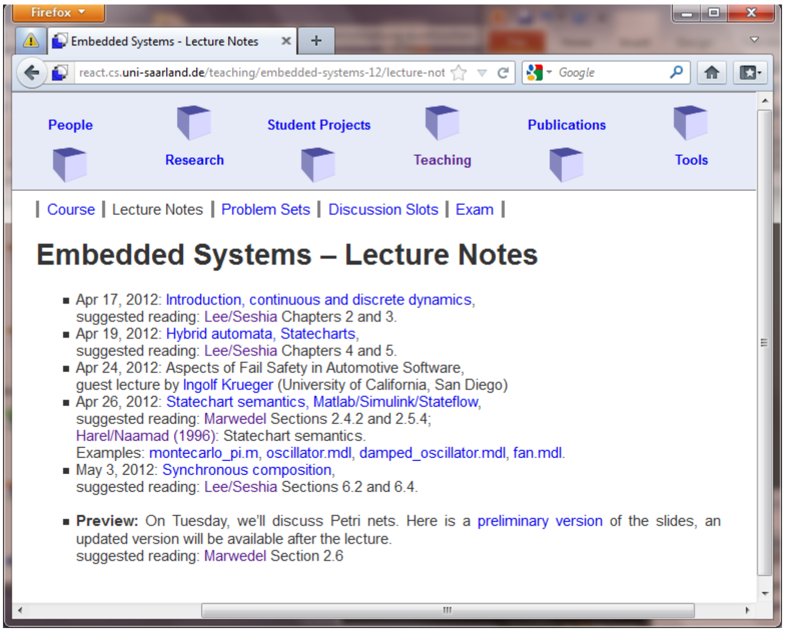
Embedded Systems 6

Please register!

- Please register in HISPOS for the exam
- In case of problems: studium@cs.uni-saarland.de
- If you cannot register (non-CS, Erasmus, …) please send email to finkbeiner@cs.uni-saarland.de



Petri nets

Introduced in 1962 by Carl Adam Petri

Application areas:

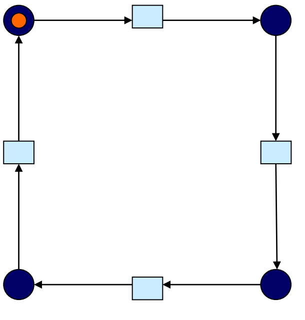
- modelling, analysis, verification of distributed systems
- automation engineering
- business processes
- modeling of resources
- modeling of synchronization

Focus on modeling causal dependencies;
no global synchronization assumed (message passing only).

Concurrency and parallelism

- Concurrency is central to embedded systems. A computer program is said to be **concurrent** if different parts of the program conceptually execute simultaneously.
- A program is said to be **parallel** if different parts of the program physically execute simultaneously on distinct hardware (multi-core, multi-processor or distributed systems)

Example 1: The four seasons



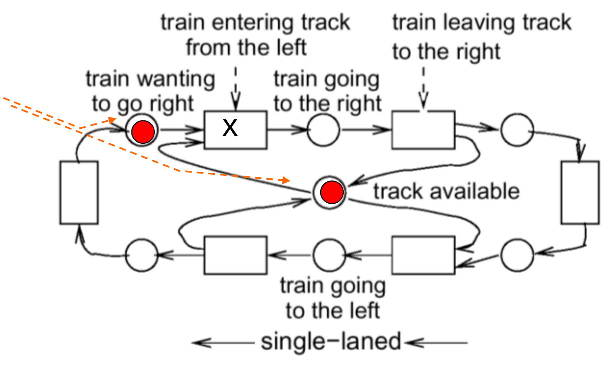
Key Elements

- **Conditions**
  Either met or not met. Conditions represent “local states”. Set of conditions describes the potential state space.
- **Events**
  May take place if certain conditions are met. Event represents a state transition.
- **Flow relation**
  Relates conditions and events, describes how an event changes the local and global state.
- **Tokens**
  Assignments of tokens to conditions specifies a global state.

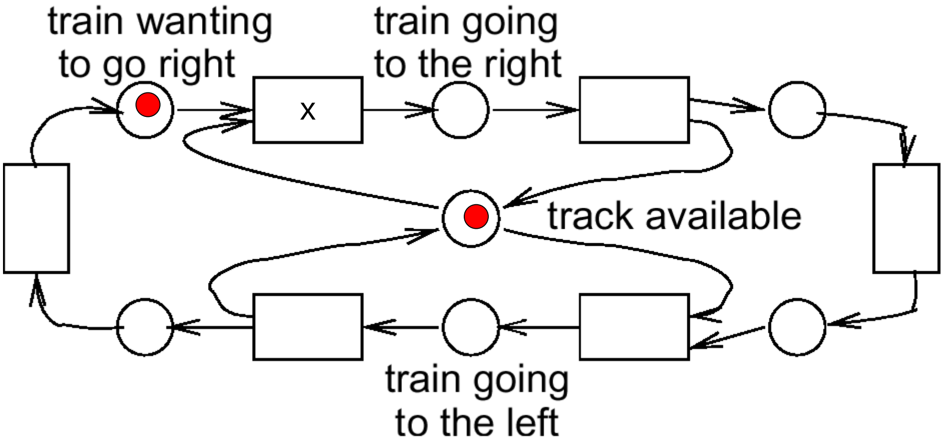
Example 2: Synchronization at single track rail segment

- **mutual exclusion: there is at most one train using the track rail**

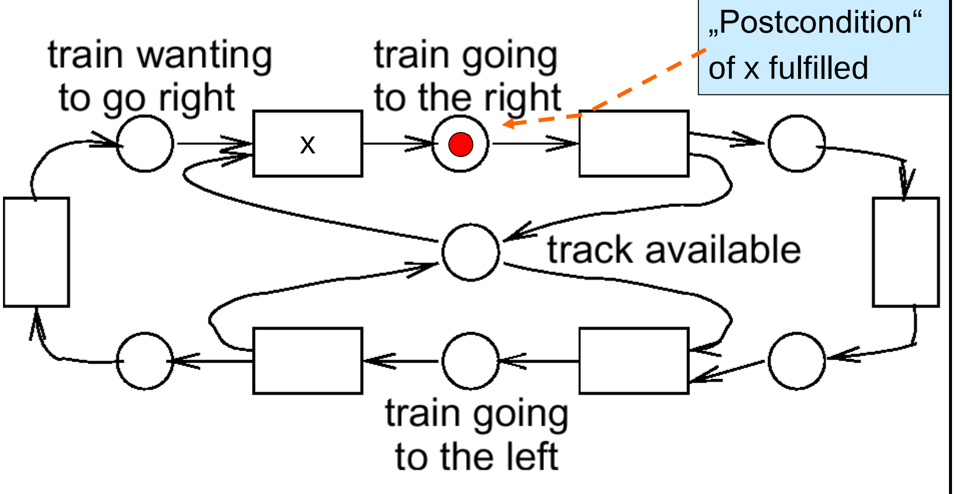
„Preconditions“ of x fulfilled



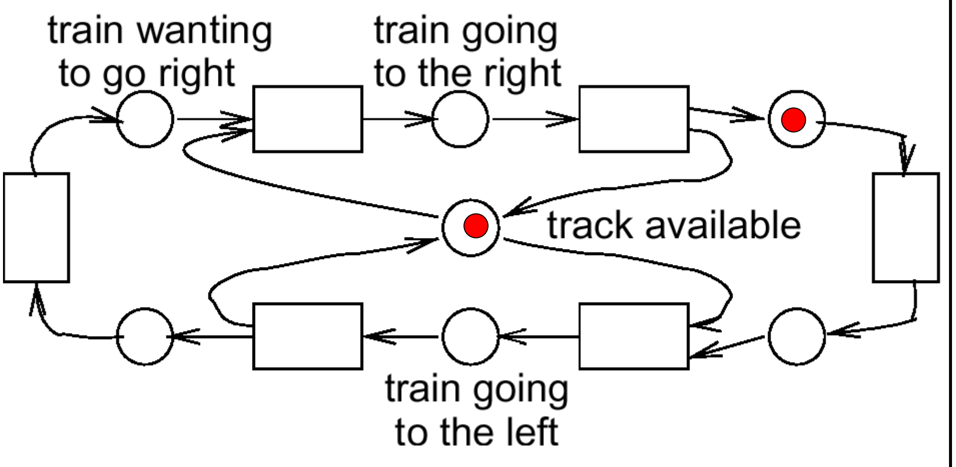
Playing the „token game“: dynamic behavior



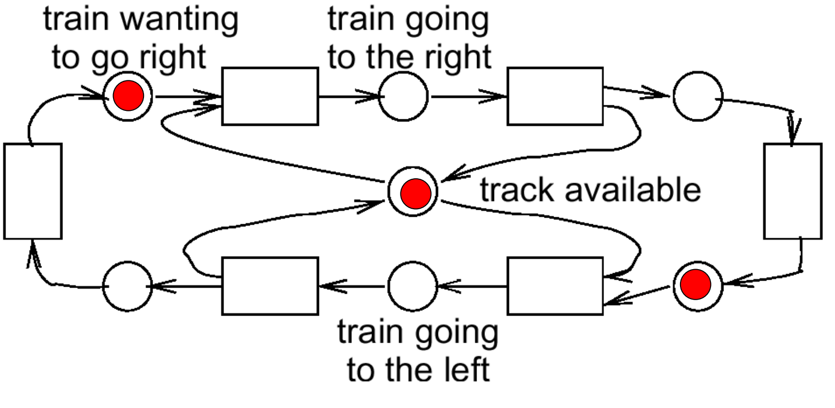
Playing the „token game“: dynamic behavior



Playing the „token game“: dynamic behavior



Conflict for resource „track“: two trains competing



Condition/event Petri nets

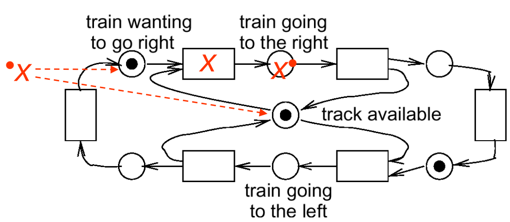
single token per place

Def.: $N=(C,E,F)$ is called a **Petri net**, iff the following holds

1. C and E are disjoint sets
2. $F \subseteq (C \times E) \cup (E \times C)$; is binary relation, („**flow relation**")

Def.: Let N be a net and let $x \in (C \cup E)$.
$^\bullet x := \{y \mid y\ F\ x\}$ is called the set of **preconditions.**
$x^\bullet := \{y \mid x\ F\ y\}$ is called the set of **postconditions.**

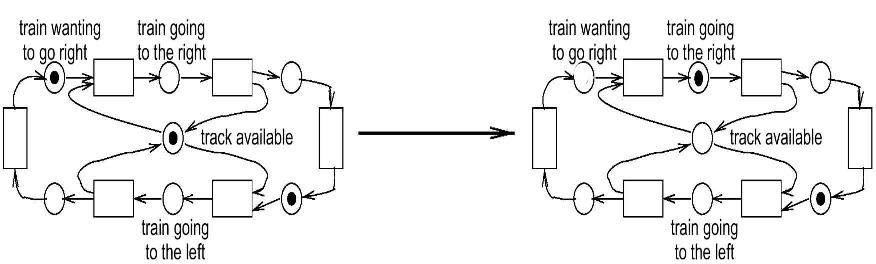
Example:



Boolean marking
and computing changes of markings

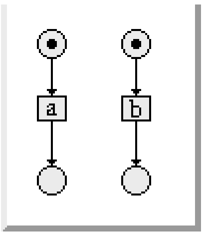
- A Boolean marking is a mapping M: C $\rightarrow$ { 0,1 }.
- „Firing" events x generate new markings on each of the conditions c according to the following rules:

  a transition at x can be fired, iff $^\bullet x$, *i.e.* all preconditions of x are marked and $x^\bullet$ is not marked, after firing $^\bullet x$ *is* unmarked and $x^\bullet$ is marked
- M $\rightarrow$ M', iff M' results from M by firing exactly one transition

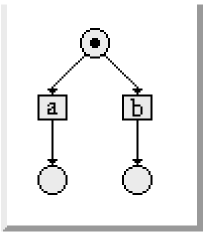


Expressiveness: basic examples

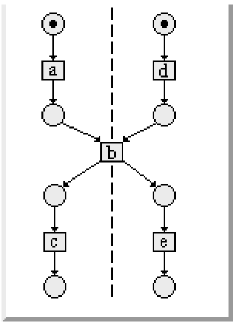
- concurrency of transitions



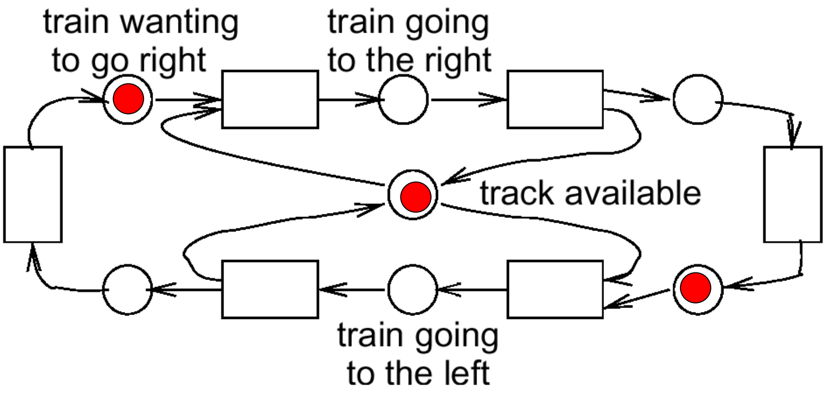
- alternative or conflict



- synchronization

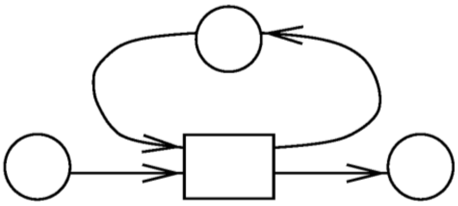


Competing Trains Example: Conflict for resource „track“



Basic structural properties: Loops and pure nets

Def.: Let $(c,e) \in C \times E$. (c,e) is called a **loop** iff $cFe \wedge eFc$.

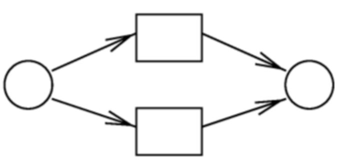


Def.: Net $N=(C,E,F)$ is called **pure**, if F does not contain any loops.

Structural properties: Simple nets

Def.: A net is called **simple**, iff
$[x,y \in (C \cup E) \wedge ({}^\bullet x = {}^\bullet y) \wedge (x^\bullet = y^\bullet)] \rightarrow x = y$

Example (not a simple net):

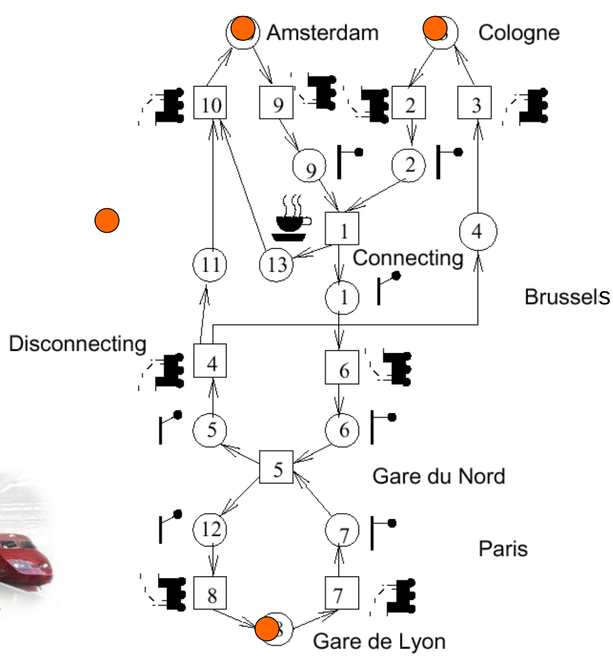


Properties of C/E

Def.:

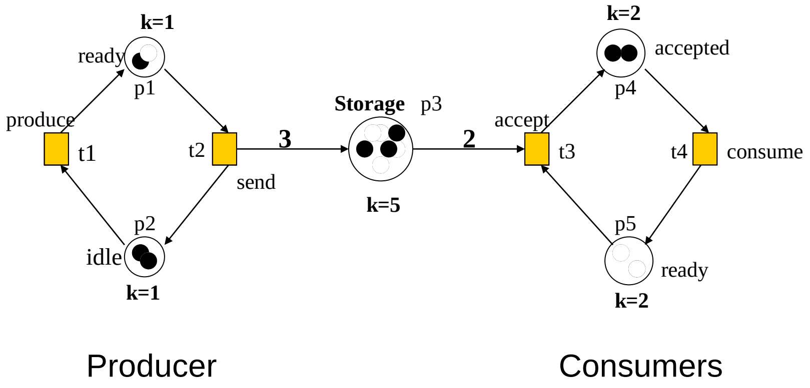
- Marking M' is **reachable** from marking M, iff there exists sequence of firing steps transforming M into M' (Not.: M [*> M')
- A C/E net is **cyclic**, iff any two different markings are reachable from each other.
- A C/E net fulfills **liveness,** iff for each marking M and for each event e there exists a reachable marking M' that activates e for firing

Thalys trains example



Place/transition nets

- More than one token per condition, capacities of places
- weights of edges



From conditions to resources

- c/e nets model the flow of information at a fundamental level (true/false)
- there are natural application areas for which the flow/transport of resources and the number of available resources is important (data flow, document-/workflow, production lines, communication networks, www, ..)
- place/transition nets are a generalization of c/e nets

From conditions to resources

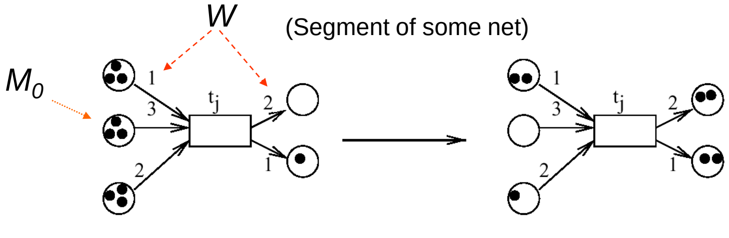
- place/transition nets are a generalization of c/e nets:
  - state elements represent places where resources (tokens) can be stored
  - transition elements represent local transitions or transport of resources
- a transition is enabled if and only if
  - sufficient resources are available on all its input places
  - sufficient capacities are available on all its output places
- a transition occurrence
  - consumes one token from each input place and
  - produces one token on each output place

Place/transition nets

multiple tokens per place

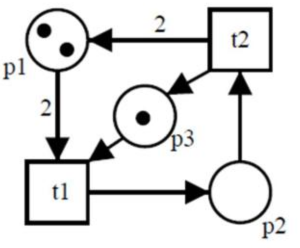
Def.: (P, T, F, K, W, M_0) is called a **place/transition net (P/T net)** iff

1. $N=(P,T,F)$ is a **net** with places P and transitions T
2. $K: P \rightarrow (\mathbb{N}_0 \cup \{\omega\}) \setminus \{0\}$ denotes the **capacity** of places (ω symbolizes infinite capacity)
3. $W: F \rightarrow (\mathbb{N}_0 \setminus \{0\})$ denotes the **weight of graph edges**
4. $M_0: P \rightarrow \mathbb{N}_0 \cup \{\omega\}$ represents the **initial marking** of places



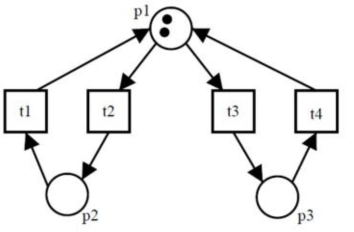
default:
K = ω
W = 1

Example

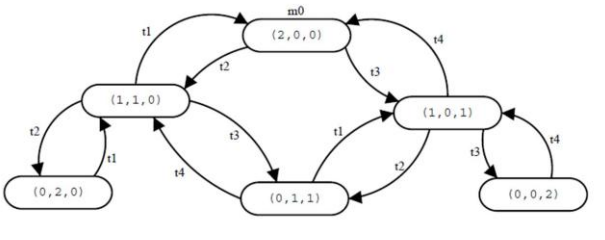


- P = {p1, p2, p3}
- T = {t1, t2}
- F = {(p1, t1), (p2, t2), (p3, t1), (t1, p2), (t2, p1), (t2, p3)}
- W = {(p1, t1) $\rightarrow$ 2, (p2, t2) $\rightarrow$ 1, (p3, t1) $\rightarrow$ 1, (t1, p2) $\rightarrow$ 1, (t2, p1) $\rightarrow$ 2, (t2, p3) $\rightarrow$ 1}
- m0 = (2, 0, 1)

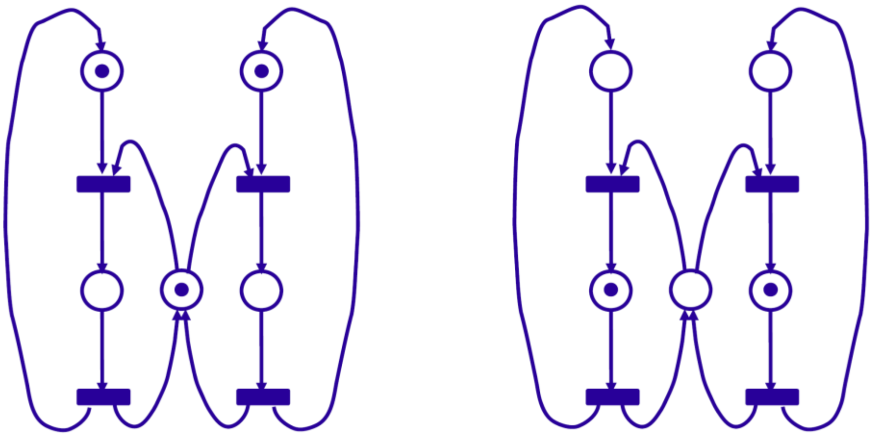
Reachability



Reachability graph:



Reachability



Marking
M

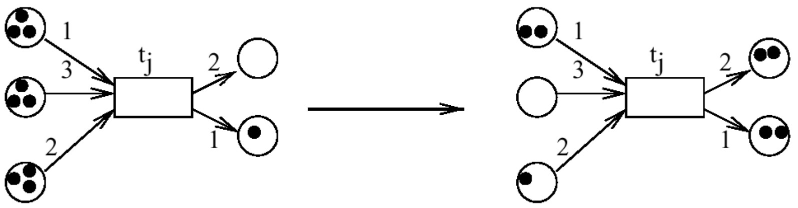
Is there a sequence of
transition firings such
that M $\longrightarrow$ M'?

Marking
M'

Computing changes of markings

- „Firing“ transitions *t* generate new markings on each of the places *p* according to the following rules:

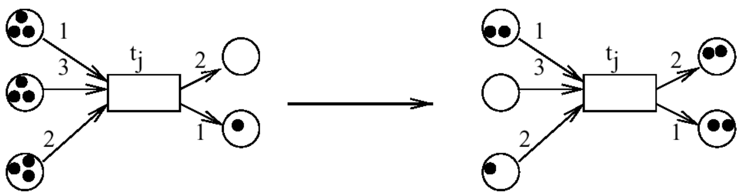
$$M'(p) = \begin{cases} M(p) - W(p,t), & \text{if } p \in {}^\bullet t \setminus t^\bullet \\ M(p) + W(t,p), & \text{if } p \in t^\bullet \setminus {}^\bullet t \\ M(p) - W(p,t) + W(t,p), & \text{if } p \in {}^\bullet t \cap t^\bullet \\ M(p) & \text{otherwise} \end{cases}$$



Activated transitions

- Transition *t* is „activated“ iff

$$(\forall p \in {}^{\bullet}t : M(p) \geq W(p,t)) \wedge (\forall p \in t^{\bullet} : M(p) + W(t,p) \leq K(p))$$

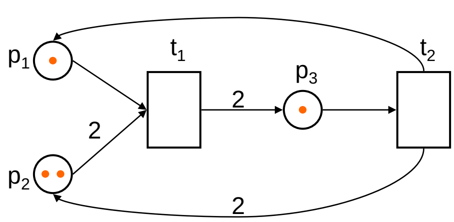


Activated transitions can „take place“ or „fire“, but don‘t have to.
The order in which activated transitions fire is not fixed (it is non-deterministic).

Boundedness

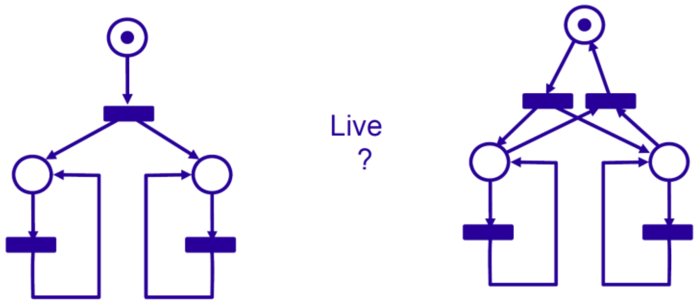
- A place is called **k-bounded** or **k-safe** if it contains in all reachable markings at most k tokens
- A net is **bounded** if each place is bounded

Application: places represent buffers and registers
$\rightarrow$ avoid buffer overflow



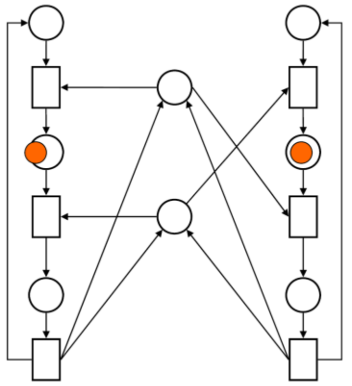
Liveness

- A transition is **live** if in every reachable marking there exists a firing sequence such that the transition becomes enabled
- A net is **live** if all its transitions are live



Deadlock

- A **dead marking** (**deadlock**) is a marking where no transition can fire
- A net is **deadlock-free** if no dead marking is reachable

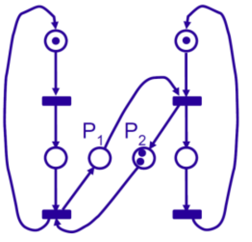


Computation of Invariants

We are interested in subsets consisting of places whose number of tokens remain invariant under transitions,

e.g. the number of trains commuting between Amsterdam and Paris (Cologne and Paris) remains constant

Important for correctness proofs, e.g. the proof of liveness



$P_1 + P_2 = 2$

Shorthand for changes of markings

Firing transition:

$$M'(p) = \begin{cases} M(p) - W(p,t), & \text{if } p \in {}^{\bullet}t \setminus t^{\bullet} \\ M(p) + W(t,p), & \text{if } p \in t^{\bullet} \setminus {}^{\bullet}t \\ M(p) - W(p,t) + W(t,p), & \text{if } p \in {}^{\bullet}t \cap t^{\bullet} \\ M(p) & \text{otherwise} \end{cases}$$

Let

$$\underline{t}(p) = \begin{cases} -W(p,t) \text{ if } p \in {}^{\bullet}t \setminus t^{\bullet} \\ +W(t,p) \text{ if } p \in t^{\bullet} \setminus {}^{\bullet}t \\ -W(p,t) + W(t,p) \text{ if } p \in t^{\bullet} \cap {}^{\bullet}t \\ 0 \end{cases}$$

$\Rightarrow$ $\forall p \in P$: $M'(p) = M(p) + \underline{t}(p)$

$\Rightarrow$ $M' = M + \underline{t}$ +: vector add

Matrix $\underline{N}$ describing all changes of markings

$$\underline{t}(p)=\begin{cases}-W(p,t) \text{ if } p\in {}^\bullet t\setminus t^\bullet\\ +W(t,p) \text{ if } p\in t^\bullet \setminus {}^\bullet t\\ -W(p,t)+W(t,p) \text{ if } p\in t^\bullet \cap {}^\bullet t\\ 0\end{cases}$$

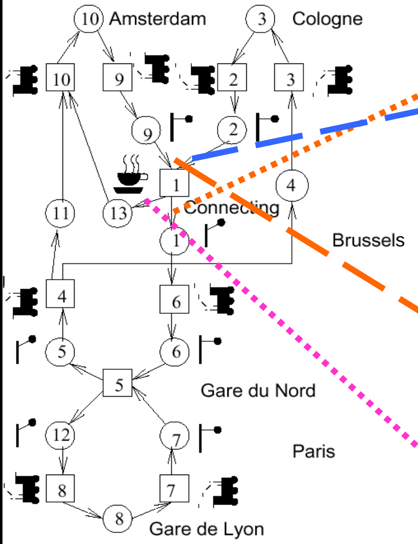
Def.: Matrix $\underline{N}$ of net N is a mapping

$\underline{N}: P \times T \rightarrow Z$ (integers)

such that $\forall\, t \in T$: $\underline{N}(p,t)=\underline{t}(p)$

Component in column t and row p indicates the change of the marking of place p if transition t takes place.

Example: $\underline{N}$ =



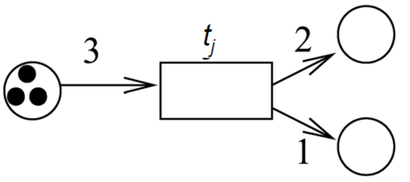
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
p_1	1					−1				
p_2	−1	1								
p_3		−1	1							
p_4			−1	1						
p_5				−1	1					
p_6					−1	1				
p_7					−1		1			
p_8							−1			
p_9	−1							1	1	
p_{10}									−1	1
p_{11}				1						−1
p_{12}					1			−1		
p_{13}	1									−1

Place invariants

For any transition $t_j \in T$ we are looking for sets $R \subseteq P$ of places for which the accumulated marking is constant:

$$\sum_{p \in R} \underline{t}_j(p) = 0$$

Example:



Characteristic Vector

$$\sum_{p \in R} \underline{t}_j(p) = 0$$

Let: $$\underline{c}_R(p) = \begin{cases} 1 \text{ if } & p \in R \\ 0 \text{ if } & p \notin R \end{cases}$$

$$\Rightarrow \quad \sum_{p \in R} \underline{t}_j(p) = \underline{t}_j \cdot \underline{c}_R = \sum_{p \in P} \underline{t}_j(p)\, \underline{c}_R(p) = 0$$

Scalar product

Condition for place invariants

$$\sum_{p\in R} \underline{t}_j(p) = \underline{t}_j \cdot \underline{c}_R = \sum_{p\in P} \underline{t}_j(p)\, \underline{c}_R(p) = 0$$

Accumulated marking constant for **all** transitions if

$$\begin{array}{ccc} \underline{t}_1 \cdot \underline{c}_R & = & 0 \\ \ldots & \ldots & \ldots \\ \underline{t}_n \cdot \underline{c}_R & = & 0 \end{array}$$

Equivalent to $\underline{\boldsymbol{N}}^T\, \underline{\boldsymbol{c}}_R = \boldsymbol{0}$ where $\underline{N}^T$ is the transposed of $\underline{N}$

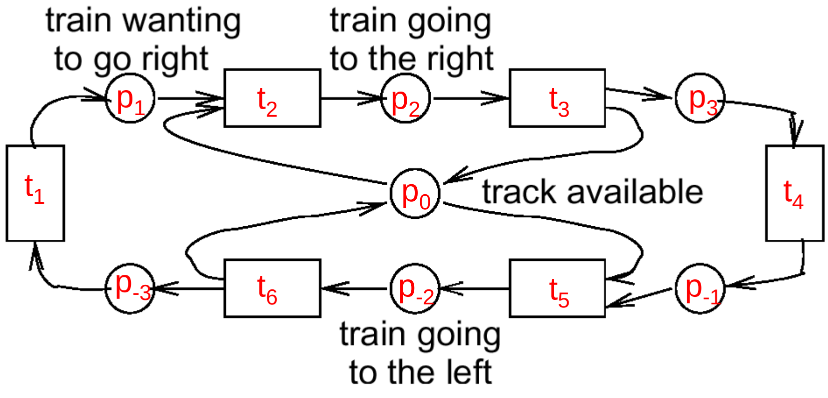
System of linear equations

$$\begin{pmatrix} \underline{t}_1(p_1) \ldots \underline{t}_1(p_n) \\ \underline{t}_2(p_1) \ldots \underline{t}_2(p_n) \\ \ldots \\ \underline{t}_m(p_1) \ldots \underline{t}_m(p_n) \end{pmatrix} \begin{pmatrix} \underline{c}_R(p_1) \\ \underline{c}_R(p_2) \\ \ldots \\ \underline{c}_R(p_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

System of linear equations.

Solution vectors must consist of zeros and ones.

Competing trains example



Application to Thalys example

$\underline{N}^T\ \underline{c}_R = 0$, with $\underline{N}^T$=

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
t_1	1	-1							-1				1
t_2		1	-1										
t_3			1	-1									
t_4				1	-1						1		
t_5					1	-1	-1					1	
t_6	-1					1							
t_7							1	-1					
t_8								1				-1	
t_9									1	-1			
t_{10}										1	-1		-1

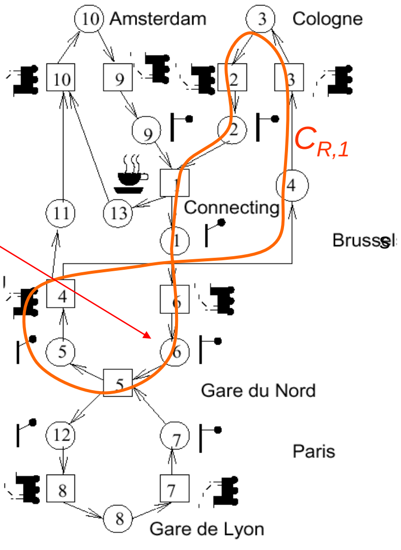
$$c_{R,1} = (1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

Interpretation of the 1st invariant

$$c_{R,1} = (1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

Characteristic vector describes places for Cologne train.

We proved that: the number of trains along the path remains constant.



Application to Thalys example

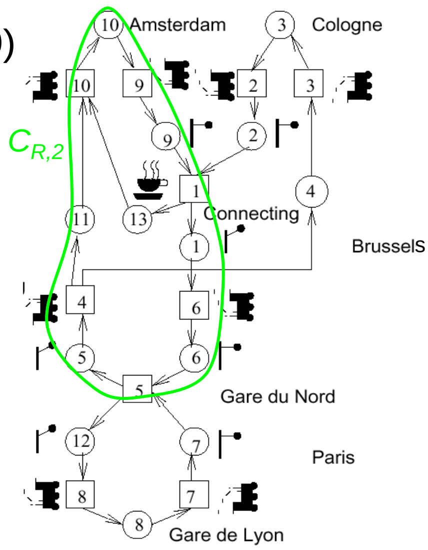
$\underline{\boldsymbol{N}}^T\ \underline{\boldsymbol{c}}_R = \boldsymbol{0}$, with $\underline{\boldsymbol{N}}^T$=

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
t_1	1	-1							-1				1
t_2		1	-1										
t_3			1	-1									
t_4				1	-1						1		
t_5					1	-1	-1					1	
t_6	-1					1							
t_7							1	-1					
t_8								1				-1	
t_9									1	-1			
t_{10}										1	-1		-1

$$c_{R,2} = (1,0,0,0,1,1,0,0,1,1,1,0,0)$$

Interpretation of the 2nd invariant

$$c_{R,2} = (1,0,0,0,1,1,0,0,1,1,1,0,0)$$



We proved that:
None of the Amsterdam trains gets lost.

Application to Thalys example

$\underline{N}^T \underline{c}_R = 0$, with $\underline{N}^T$=

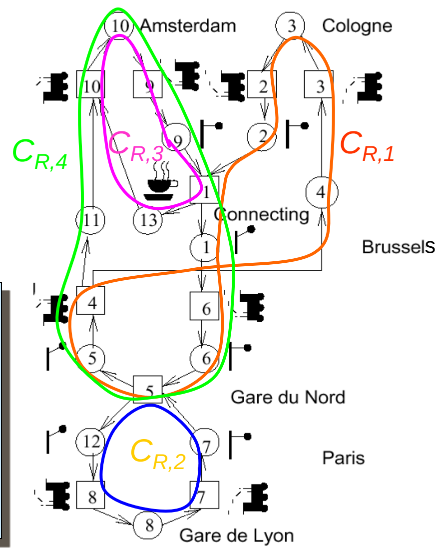
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
t_1	1	-1							-1				1
t_2		1	-1										
t_3			1	-1									
t_4				1	-1						1		
t_5					1	-1	-1					1	
t_6	-1					1							
t_7							1	-1					
t_8								1				-1	
t_9									1	-1			
t_{10}										1	-1		-1

$$c_{R,2} = \left(0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\right)$$

Solution vectors for Thalys example

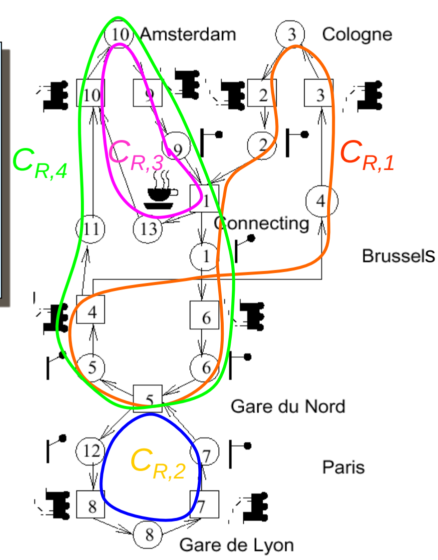
$$c_{R,1} = (1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0)$$
$$c_{R,2} = (0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0)$$
$$c_{R,3} = (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1)$$
$$c_{R,4} = (1\,0\,0\,0\,1\,1\,0\,0\,1\,1\,1\,0\,0)$$

We proved that:
- the number of trains serving Amsterdam, Cologne and Paris remains constant.
- the number of train drivers remains constant.



Solution vectors for Thalys example

It follows:
- each place invariant must have at least one label at the beginning, otherwise “dead”
- at least three labels are necessary in the example



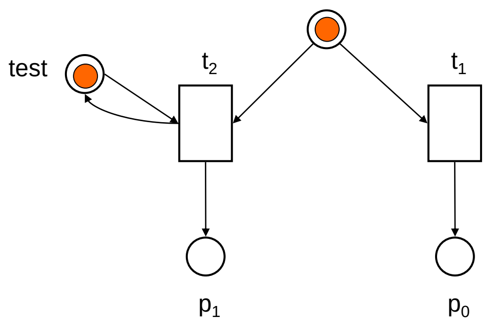
Invariants & boundedness

- A net is **covered** by place invariants iff every place is contained in some invariant.

Theorem 1:
a) If R is a place invariant and $p \in R$, then p is bounded.
b) If a net is covered by place invariants then it is bounded.

Extensions: Petri nets with priorities

- $t_1 \langle t_2$: t_2 has higher priority than t_1.



- Petri nets with priorities are Turing-complete.

Extensions: Predicate/transition nets

- Goal: compact representation of complex systems.
- Key changes:
  - Tokens are becoming individuals;
  - Transitions enabled if functions at incoming edges true;
  - Individuals generated by firing transitions defined through functions
- Changes can be explained by folding and unfolding C/E nets,
  ☞ semantics can be defined by C/E nets.

Predicate/transition model of the dining philosophers problem

- Let x be one of the philosophers,
- let l(x) be the left fork of x,
- let r(x) be the right fork of x.

Token: individuals.

Semantics can be defined by replacing net by equivalent condition/event net.

Model can be extended to arbitrary numbers.