4. Petri Nets
4.1 Introduction

Goals

- modeling and analysis of the behavior of arbitrary systems => models of system dynamics (processes)
- mathematical foundation (based on Linear Algebra and Graph Theory) which allows to formally prove aspects like
  - a given state is reachable (i.e. may be provoked by an event)
  - a given sequence of states is admissible
  - certain states may never be reached
  - deadlocks may occur or not ('liveness')
  - channels/storages do not overflow
  - starvation of (sub)processes
- clear distinction between parallelism and concurrency
- theoretical foundation (theory of non-sequential processes, Carl Adam Petri, Diss. 1962: "Kommunikation mit Automaten")

4. Petri Nets
4.1 Introduction

Basic Idea

- introduction of a formal language (terminal symbols, syntactic rules, abstract semantics)
- "conditioning" for modeling purposes by **net interpretations,** i.e. by relating domain specific notions to the syntactic elements of the language
- transparent and domain independent representation concepts

=> modeling systems for different purposes based on the same theoretical and formal fundaments

4. Petri Nets
4.1 Introduction

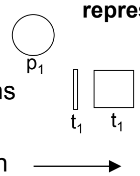
Typical application domains

- technical and industrial processes
- business process modeling
- hardware design and analysis
- communication protocols
- parallel programs
- distributed data bases
- requirements engineering
- simulation

4 Petri Nets
4.1 Introduction

Basic Definition

Graphical representation



- given
  - a set $P = \{p_0, p_1, ..., p_n\}$ of so-called places
  - a set $T = \{t_0, t_1, ..., t_n\}$ of so-called transitions
  - a relation $I \subseteq P \times T$ called input relation
  - a relation $O \subseteq P \times T$ called output relation
- the quadruple PN = (P,T, I, O) is called a **Petri Net** (**Transition Net**) if
  - $P \cup T \neq 0$, $P \cap T = 0$, $|P \cup T| < \infty$ and if
  - the un-directed graph $(P \cup T, I \cup O)$ is a connected one (i.e., there exists a path from each node to each other one)

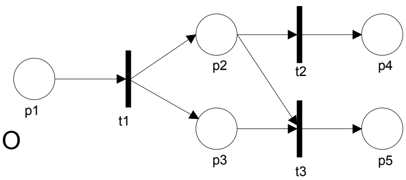
Note: A Petri Net is a bipartite directed graph

4 Petri nets
4.1 Introduction

Example

($P = \{p_1, p_2, p_3, p_4, p_5\}$,
$T = \{t_1, t_2, t_3\}$,
$I = \{(p_1, t_1), (p_2, t_2), (p_2, t_3), (p_3, t_3)\}$,
$O = \{(p_2, t_1), (p_3, t_1), (p_4, t_2), (p_5, t_3)\}$)

- input (upstream) $(p,t) \in I$
- output (downstream) $(p,t) \in O$

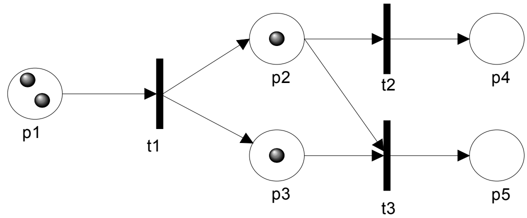


I,O may be multisets (extension that allows for more than 1 edge between a particular pair (p_i,t_j)); in that case I, O may be written as functions I,O: T -> ***P*** with I(t), O(t) is a multiset (collection) of places

4 Petri nets
4.2 Marked Petri nets

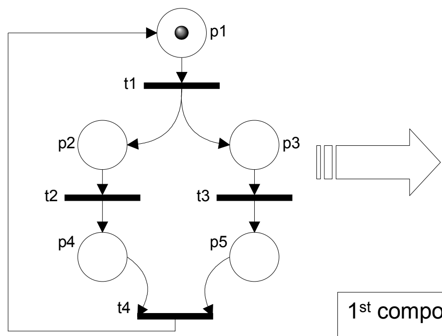
- Marks (tokens)
  - – M: P → $\mathbf{N}_0$: marking
  - – M(p) = n (place p contains n marks)
- formal definition

  A Marked PN is a **quintuple MPN = (PN; M)**, where PN=(P,T,I,O) is a Petri net and M: P → N_0 a marking
- representation of marks: dots



4 Petri nets
4.2 Marked Petri nets

- Initial Marking M_0
- marking at a given moment may be represented as a **column vector**



$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

1st component = marking of place p1,

2nd component = marking of place p2, ...

4 Petri nets
4.2 Marked Petri nets

- A transition $t \in T$ of a marked Petri net MPN = (PN; M) is said to be **fireable (enabled; en(t,M))** if $M(p_i) \geq I_{p_i}(t)$ for each $p_i \in P$, [$I_{p_i}(t)$ being the number of occurrences of p_i in I(t)]

  $\Rightarrow$ a source transition (transition without an input place) is always fireable

- The **firing** of an enabled transition $t \in T$ within a marked Petri net MPN = (PN; M) results in a marking M‘ = δ(M,t) with

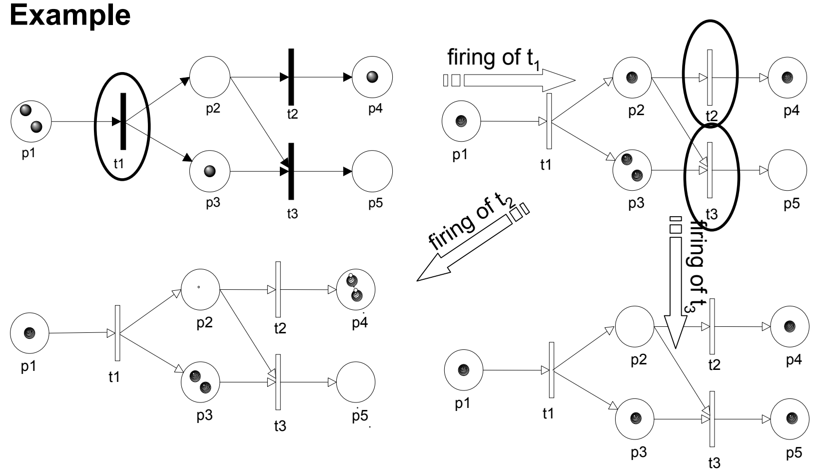
  $M'(p_i) = M(p_i) - I_{p_i}(t) + O_{p_i}(t)$ for each $p_i \in P$,

  (take $I_{p_i}(t)$ tokens from each of t’s input places and add $O_{p_i}(t)$ tokens to each of its output places)

  Remark:
  - the firing of a transition is indivisible (i.e. atomic)
  - the firing of a transition has no duration (time is not considered)
  - concurrency: a transition may fire before, after or in parallel with another fireable one (if they are not in "conflict“, see below)

4 Petri nets
4.2 Marked Petri nets

Example



4 Petri nets
4.2 Marked Petri nets

Transition function and reachability

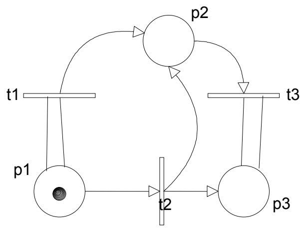
Given a Petri net PN = (P,T,I,O) with |P| = n.

- the mapping δ: $N^n \times T \rightarrow N^n$ defined by $\delta(M,t) = M'$ for each $M \in N^n$ is called the **transition function** of PN
- M_s is said to be **reachable** from M_0 in (PN; M_0) if there is a sequence $t_1, t_2, \ldots, t_k$ of transitions such that $M_s = \delta(\delta(\ldots \delta(M_0, t_1), t_2) \ldots), t_k)$.
  If k=1 then M_s is said to be **immediately reachable** from M_0 in (PN; M_0)
- The set R(PN; M_0) = $\{M_0\} \cup \{M \mid M$ reachable from M_0 in (PN; M_0)$\}$ is called the **reachability set** of (PN; M_0)

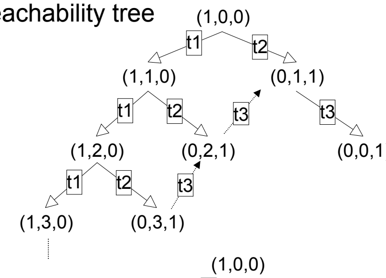
Note: R(PN; M_0) may be visualized by a **reachability tree**

4 Petri nets
4.2 Marked Petri nets

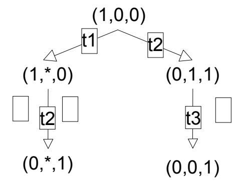
Example



Reachability tree



finite representation



4 Petri nets
4.3 Important net properties

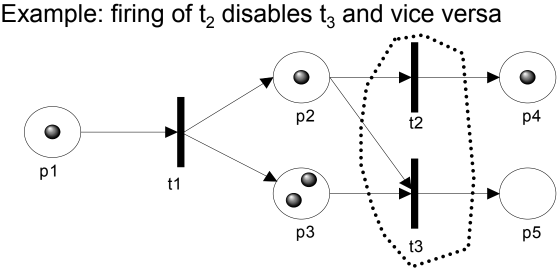
Conflict

2 fireable transitions t_i and t_j are said to be "in conflict", if there is a place p in $I(t_i) \cap I(t_j)$ such that $M(p) < I_p(t_i) + I_p(t_j)$

$\Rightarrow$ structural conflict; firing of t_i disables t_j and vice versa

$\Rightarrow$ t_i and t_j are causally dependent $\Rightarrow$ no concurrency

Example: firing of t_2 disables t_3 and vice versa



4 Petri nets
4.3 Important net properties

- **Coverability**

  A marking M‘ in a marked Petri net (PN; M_0) is said to be coverable if there exists a marking M“ in R(PN; M_0) such that M" ≥ M‘
  (M“ covers M‘)

- **Boundedness**
  - A place $p \in P$ is called k-bounded (k-safe) within a MPN = (PN; M_0), if $M'(p) \leq k \in \mathbf{N}$ for each $M' \in R(PN; M_0)$
  - A Petri net is said to be k-bounded (k-safe) if all its places $p \in P$ are k-bounded

  Note:
  The reachability set R(PN; M_0) of a k-bounded Petri net is finite

4 Petri nets
4.3 Important net properties

- **Conservation**
  - A marked Petri net MPN = (PN; M_0) is said to be **strictly conservative**, if for each $M \in RP(PN; M_0)$: $\Sigma\, M(p_i) = \Sigma\, M_0(p_i)$
    i.e., the total number of tokens remains constant

    This implies for each $t_i \in T$: $|I(t_j)| = |O(t_j)|$

  - A marked Petri net MPN = (PN; M_0) is said to be **conservative**, if there is a weighting vector $w = (w_1, w_2, ..., w_n) \neq$ null_vector such that $w_i \in \{0, 1\}$ and $w * M = w * M_0$ for all $M \in R(PN ; M_0)$ (vector product)

4 Petri nets
4.3 Important net properties

- **Liveness of transitions**

  Given a marked Petri net MPN = (PN; M_0); we distinguish the following levels of lifeness:

  – **Level 0**: a transition t is not enabled in any M‘$\in$ R(PN; M_0); in this case the transition is called **dead**
  – **Level 1**: a transition t is potentially fireable, i.e. there exists a M‘$\in$ R(PN; M_0) such that en(t,M_0)
  – **Level 2**: a transition t is at least n times fireable where n $\in$ **N**
  – **Level 3**: a transition t can be fired infinitely
  – **Level 4**: for each M' $\in$ R(PN; M_0) there is a M‘‘$\in$ R(PN; M‘) such that en(t, M‘‘)

4 Petri nets
4.3 Important net properties

- **Liveness of Petri nets**

  Given a marked Petri net MPN = (PN; M_0):
  – MPN is **live at level i** if all its transitions are live at level i
  – MPN is **live** if all its transitions are live at level 4
  – MPN is **dead** if all its transitions are dead

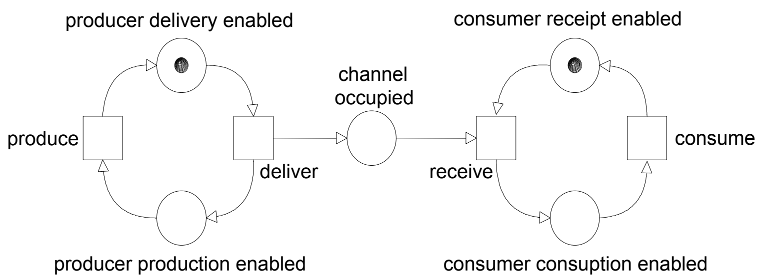
- **Deadlock**

  – A **deadlock** (sink state) is a marking M such that no transition is enabled (i.e. the marking can no longer evolve); this implies R(PN; M) = {M}
  – A marked Petri net MPN = (PN; M_0) is said to be deadlock-free (pseudolive) if no reachable marking M' $\in$ R(PN; M_0) is a deadlock

4 Petri nets
4.4 Net interpretations

- **Condition-event nets**
  - place: condition
  - transition: event

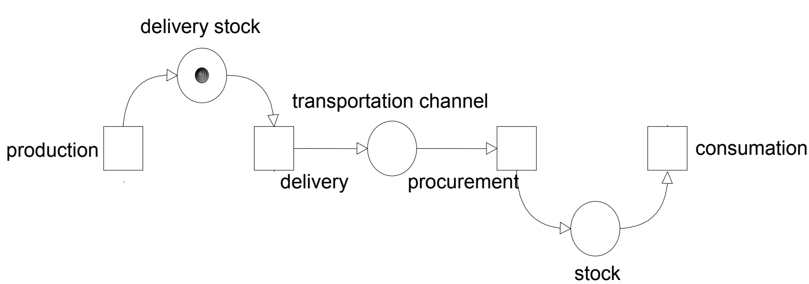
  ⇒ I(t): preconditions of t, O(t): postconditions of t



4 Petri nets
4.4 Net interpretations

- **Channel-instance nets**
  - place: channel/storage
  - transition: instance (activity)

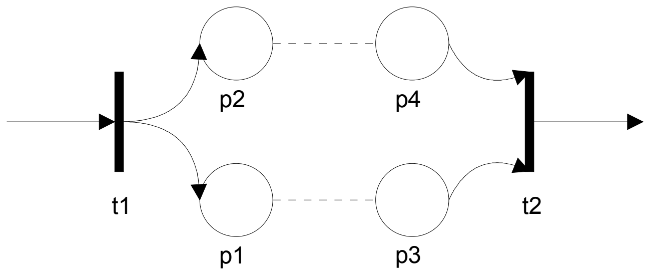
  ⇒ I(t): input channels t, O(t): output channels of t



4 Petri nets
4.5 Some patterns for system analysis

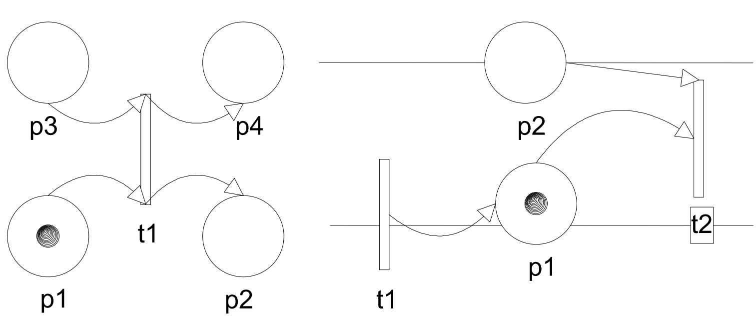
- **Concurrency:** transitions are causally independent
- **Parallelism**: transitions are temporarily independent

Note: concurrent transitions may fire before, after or concurrently with each other



4 Petri nets
4.5 Some patterns for system analysis

- **Synchronization**

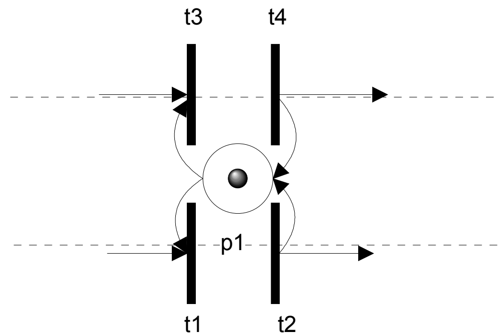


Rendez-vous type *Semaphore type*

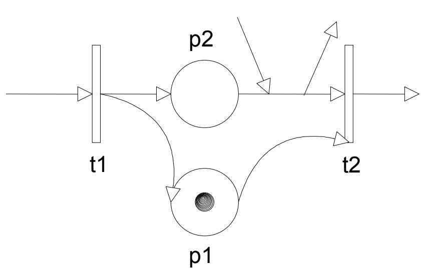
4 Petri nets
4.5 Some patterns for system analysis

- **Mutual exclusion**

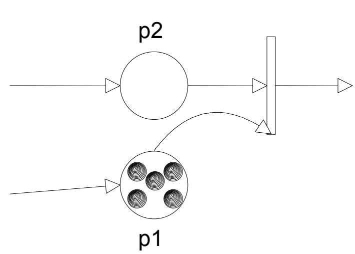
  e.g. program parts which use the same resource (e.g. common memory)



4 Petri nets
4.5 Some patterns for system analysis



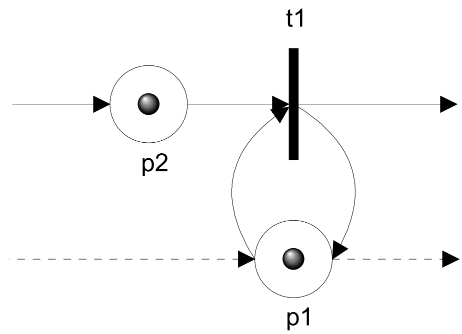
Memorizing of a firing



Memorizing of a number

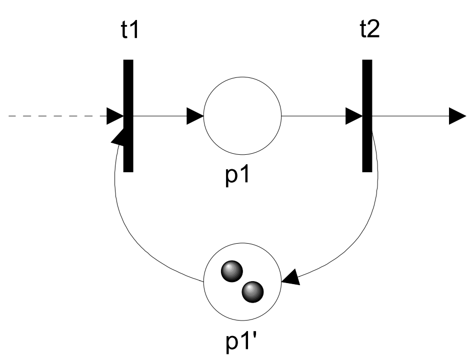
4 Petri nets
4.5 Some patterns for system analysis

- Firing of t_1 is conditioned by the marking of p_1 without modifying this marking: limited capacity



4 Petri nets
4.5 Some patterns for system analysis

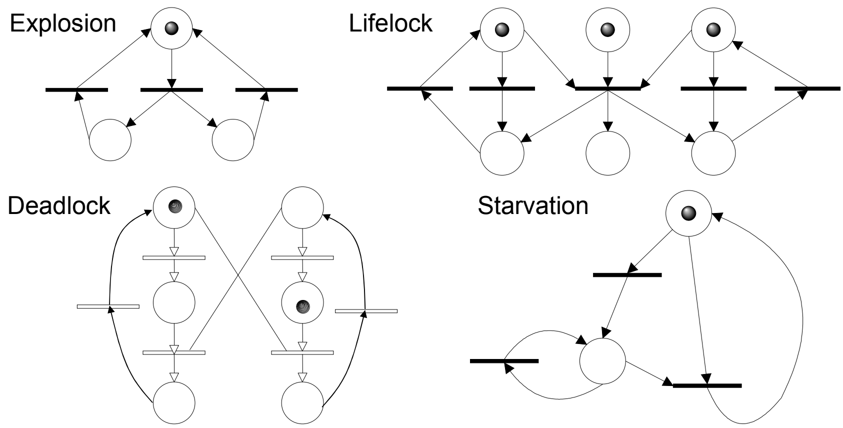
- **limited capacity:** Firing of t_1 is only possible if place p_1 contains fewer than two tokens



Heinrich C. Mayr: Fundamentals of Modeling
© Inst. für Wirtschaftsinformatik und Anwendungssysteme 2002

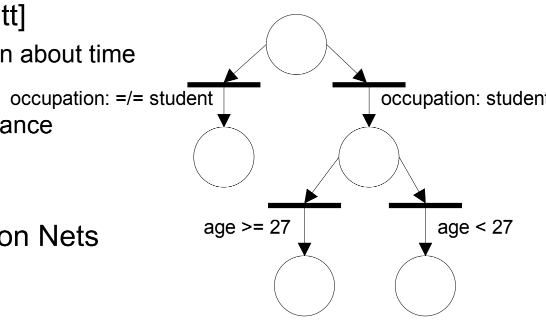
4 Petri nets
4.5 Some patterns for system analysis

Prototypical Constellations



4 Petri nets
4.6 Extended Petri nets

- **basic idea:**
  valuable information transported by tokens within a Petri net
- E-Nets [Noe & Nutt]
  - contain information about time
  - models for computer performance evaluation
- Predicate Transition Nets



4 Petri nets
4.6 Extended Petri nets

- XML and Petri nets:
  - e.g. PNML – the Petri Net Markup Language (http://www.informatik.hu-berlin.de/top/pnml)

- Obiect oriented Petri nets