Discrete and continuous dynamic systems

Petri Nets
Dynamics and analysis

Katalin Hangos

University of Pannonia
Faculty of Information Technology
Department of Electrical Engineering and Information Systems

hangos.katalin@virt.uni-pannon.hu

Apr 2018

Lecture overview

1. Previous notions
   - Automata models
   - Petri nets
2. Generalized Petri net models
   - Hierachical Petri nets
   - Timed Petri nets
   - Coloured Petri nets
3. Reachability graph of Petri nets
   - Operation (dynamics) of Petri nets
   - Parallel and conflicting execution steps
   - Solution of Petri net models
   - The reachability graph
4. Analysis of discrete event system models

Discrete event systems

Characteristic properties:

- the *range space* of the signals (input, output, state) is **discrete**: $x(t) \in \mathbf{X} = \{x_0, x_1, ..., x_n\}$
- *event*: the occurrence of change in a discrete value
- *time is also* ***discrete***: $T = \{t_0, t_1, ..., t_n\} = \{0, 1, ..., n\}$

Only the **order of the events** is considered

- description of sequential and parallel events
- **application area**: scheduling, operational procedures, resource management

Automaton - abstract model: $\mathbf{A} = (Q, \Sigma, \delta; \Sigma_O, \varphi)$

- **Set of states**: Q
- **finite alphabet** of the input tape: $\Sigma = \{\#; a, b, ...\}$
- **State transition function**: $\delta : Q \times \Sigma \rightarrow Q$
- *Set of initial and final states*: $Q_I, \ Q_F \ \subseteq \ Q$
- **finite alphabet** of the output tape: $\Sigma_O = \{\#; \alpha, \beta, ...\}$
- **Output function**: $\varphi : Q \rightarrow \Sigma_O$

Graphical description: weighted directed graph

- **Vertices**: states (Q)
- **Edges**: state transitions (δ)
- **Edge weights**: input symbols (Σ)

Automata - discrete event systems

	Automaton model	Discrete event state space model
State space	Q	$\mathcal{X} \in \mathbb{Z}^n$
Input u	string from Σ	discrete time discrete valued signal
Output y	string from Σ_O	discrete time discrete valued signal
State equation	$q(k+1) = \delta(q(k), u(k))$	$x(k+1) = \Psi(x(k), u(k))$
Output equation	$y(k) = \varphi(x(k))$	$y(k) = h(x(k), u(k))$

Petri net - abstract description: $\mathbf{PN} = (P, T, I, O)$

Static description (structure)

- set of **places (conditions)**: P
- set of **transitions (events)**: T
- **Input (pre-condition) function**: $I : T \rightarrow P^{\infty}$
- **Output (consequence) function**: $O : T \rightarrow P^{\infty}$

Graphical description: bipartite directed graph

- **Vertices**: places (P) and transitions (T) (partitions)
- **Edges**: input and output functions (I, O)

Overview - Generalized Petri nets

Generalized Petri net models

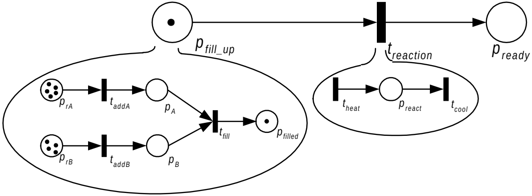
- **Hierarchical Petri nets**
- **Timed Petri nets**: using inscriptions
  - clock: built in (or special "source" place)
  - firing time to transitions
  - (waiting time for places)
- **Coloured Petri nets**: using inscriptions
  - tokens have discrete value ("colour")
  - colour set to places
  - discrete functions to the transitions and arcs

Hierarchical Petri nets

Super net - subnets:

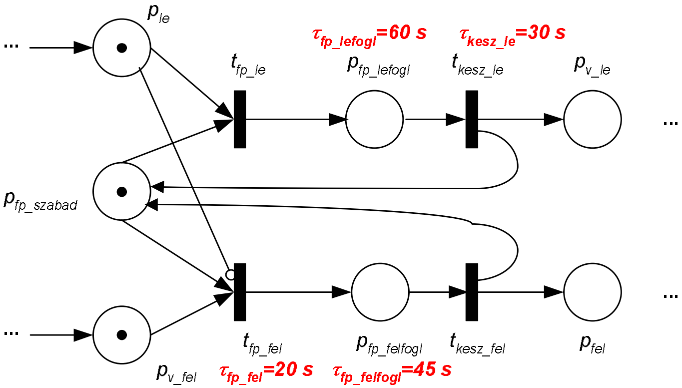
building in: to any place or transition

similar repetitive net-fragments



Petri net model of a runway – 3

Timed Peri net model



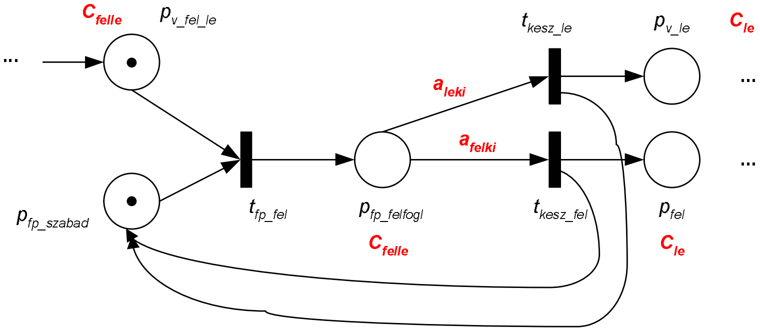
Petri net model of a runway – 4

Coloured Peri net model: "insriptions"

Edge fucntion: a_{felki} : **if** $val(p_{fp\_lefogl}) = "\uparrow"$ **then** "*true*"

$a_{fel} = val(p_{fp\_lefogl})$, $val(p_{fel}) = a_{fel}$

Colour set: $C_{felle} = \{ \uparrow , \downarrow \}$



Overview - Petri nets: operation and reachability graph

Dynamics of Petri nets

Marking function: marking points (**tokens**)

$$\mu : \mathbf{P} \rightarrow \mathcal{N} \quad , \quad \mu(p_i) = \mu_i \geq 0$$
$$\underline{\mu}^T = [\mu_1, \mu_2, \ldots, \mu_n] \quad , \quad n = |\mathbf{P}|$$

Transition **fires** (operates): when its pre-conditions are "true" (there is a **token** on its input places)

$\underline{\mu}^{(i)}[t_j > \underline{\mu}^{(i+1)}$

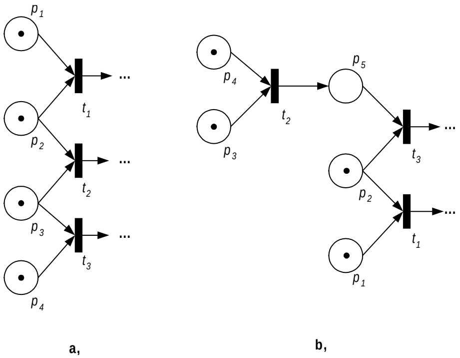
after firing the consequences become "true"

Firing (operation) sequence

$$\underline{\mu}^{(0)}[t_{j0} > \underline{\mu}^{(1)}[t_{j1} > ...[t_{jk} > \underline{\mu}^{(k+1)}$$

Parallel events

More than one enabled (fireable) transition:
concurrency (independent conditions), conflict, confusion



The solution problem

Abstract problem statement

Given:

- a *formal description* of a discrete event system model
- *initial state(s)*
- *external events*: system inputs

Compute:

- the sequence of *internal (state and output) events*

The solution is **algorithmic**! **The problem is NP-hard!**

Petri net models – reachability graph

Solution: marking (systems state) sequences
reachability graph (tree) (weighted directed graph)

- *vertices*: markings
- *edges*: if exists transition the firing of which connects them
- *edge weights*: the transition and the external events

Construction:

1. *start*: at the given initial state (marking)
2. *adding a new vertex*: by firing an enabled transition (with the effect of inputs!)

May be NP-hard (in conflict situation or non-finite operation)

The state space of Petri net models

State vector: marking in *internal* places

in- and out-degree is at least 1

$$x(k) \quad \sim \quad \underline{\mu}_x^{(k)}$$

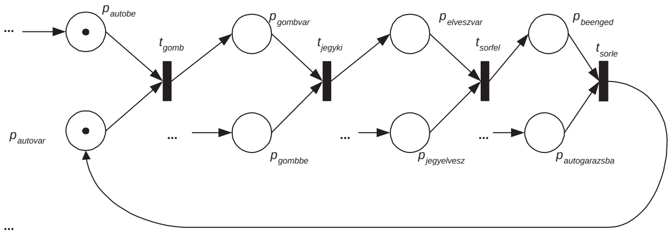
Inputs: marking in *input* places

in-degree is zero

$$u(k) \quad \sim \quad \underline{\mu}_u^{(k)}$$

Example: garage gate

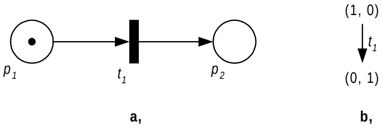
Petri net model



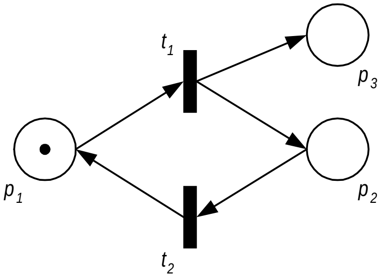
$$\underline{\mu}_x^T = [\mu_{autovar}, \mu_{gombvar}, \mu_{elveszvar}, \mu_{beenged}]$$
$$\underline{\mu}_u^T = [\mu_{autobe}, \mu_{gombbe}, \mu_{jegyelevesz}, \mu_{autogarazsba}]$$

Reachability graphs

Finite case

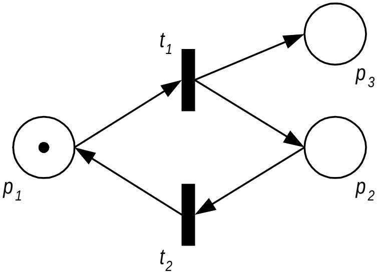


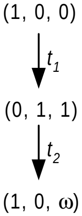
Non-finite case



Non-finite reachability graph

Reduction: using the ω symbol





Analysis of Petri net models

Dynamic properties

- *behavioural* (initial state dependent)
- *structural* (only depends on the structure graph)

Behavioural properties

- *reachabiliy* (coverability, controllability)
- *deadlocks*, liveness
- *boundedness*, safeness
- (token) conservation

Structural properties

- *state and transition invariant*: cyclic behaviour

Reachability of Petri net models

The notion of **reachability**: whether there exists

- to a given *[initial state* ($\underline{\mu}^{(I)}$)*, final state* ($\underline{\mu}^{(F)}$)*]* pair
- *a firing sequence*, such that

$$\underline{\mu}^{(I)}[t_{j0} > \underline{\mu}^{(1)}[t_{j1} > ...[t_{jk} > \underline{\mu}^{(F)}$$

The notion of **coverability**:

$$\underline{\mu}'' \geq \underline{\mu}' \quad \Leftrightarrow \quad \forall i : \ \mu_i'' \geq \mu_i'$$

The same as the usual controllability

Boundedness of Petri nets

Related properties to **boundedness**

- *finiteness (boundedness)*: Is the number of tokens finite for every initial state?
- *Safeness*: the bound is 1 for each place

Can be defined (examined) for the **whole net** or only for **a given set of places**

Conservative Petri net: the number of tokens is constant (resource-conservation)

Liveness of Petri nets

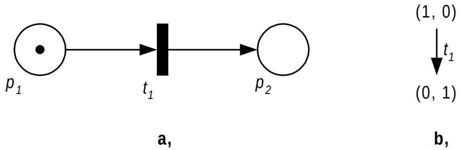
The notion of **liveness**: from a given initial state

- for a *transition*: is there a firing sequence when the transition is active?
- for a *set of transition*, for the whole net

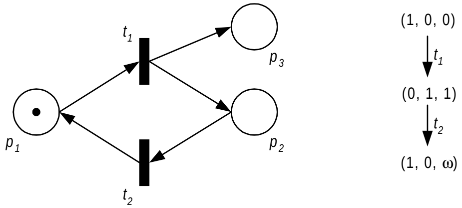
Deadlock: a non-final state from where there is no enabled (fireable) transition

Simple Petri net examples

Deadlock: the marking $(0, 1)$

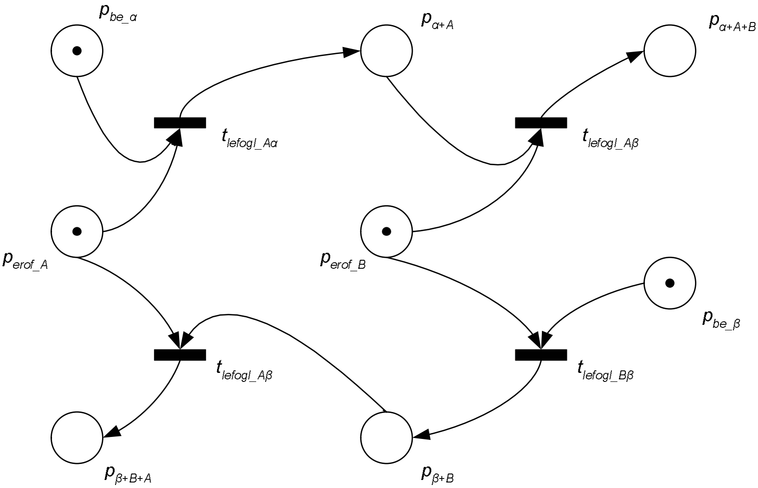


Non-bounded place: p_3

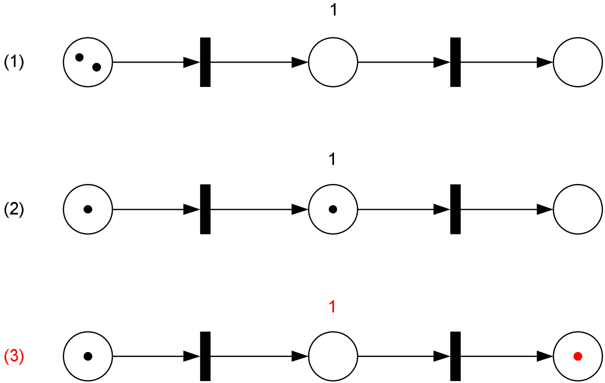


Resource allocation deadlock

Conflict situations



A safe net example



The capacity of the places changes the enabling of the transitions

Dynamic analysis methods of Petri net models – 1

Analysis of **behavioural properties**

- by constructing the *reachability graph*
- and *searching* on the vertices of the graph
- may be *NP-hard*

Problems:

- cyclic behaviour
- non-bounded places

Dynamic analysis methods of Petri net models – 2

Structural properties

- by constructing the *occurrence matrix* of the Petri net graph

$$H \in \mathbb{R}^{|P| \times |T|}$$

- and solving *linear set of equations*
- *polynomial time*, restricted importance

The elements of the occurrence matrix (for nets without loops)

$$h_{ij} = w(p_i, t_j) = \begin{cases} < 0 & \text{if} \quad p_i \quad \text{precondition} \\ > 0 & \text{if} \quad p_i \quad \text{consequence} \end{cases}$$

Place and transition invariants

Place invariant: set of conservation places $P_{INV} \subseteq P$
by solving the equation

$$z^T H = \underline{0}^T \quad , \quad z \in \mathbb{R}^{|P|}$$

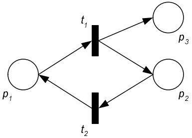
for its non-trivial solutions (z is the indicator vector)

Transition invariant: a set of transitions $T_{INV} \subseteq T$ that brings the system back to the initial state
by solving the equation

$$Hv = \underline{0} \quad , \quad v \in \mathbb{R}^{|T|}$$

for its non-trivial solutions (v is the indicator vector)

Place and transition invariants – Example



Place invariant:

$$\left[\, z_1 \;\; z_2 \;\; z_3 \,\right] \cdot \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix} = \left[\, 0 \;\; 0 \,\right] \quad \Rightarrow \quad z_1 = z_2$$

Transition invariant: without p_3 !!

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad v_1 = v_2$$