Modelling, analyzing and control of DES and DEDS

(Compendium of the recent research)

František Čapkovič

Presentation in University of Aizu

October 11, 2006

Author's address

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences

Dúbravská cesta 9, 845 07 Bratislava, Slovak Republic

e-mail: Frantisek.Capkovic@savba.sk

http://www.ui.sav.sk/home/capkovic/capkhome.htm

Contents

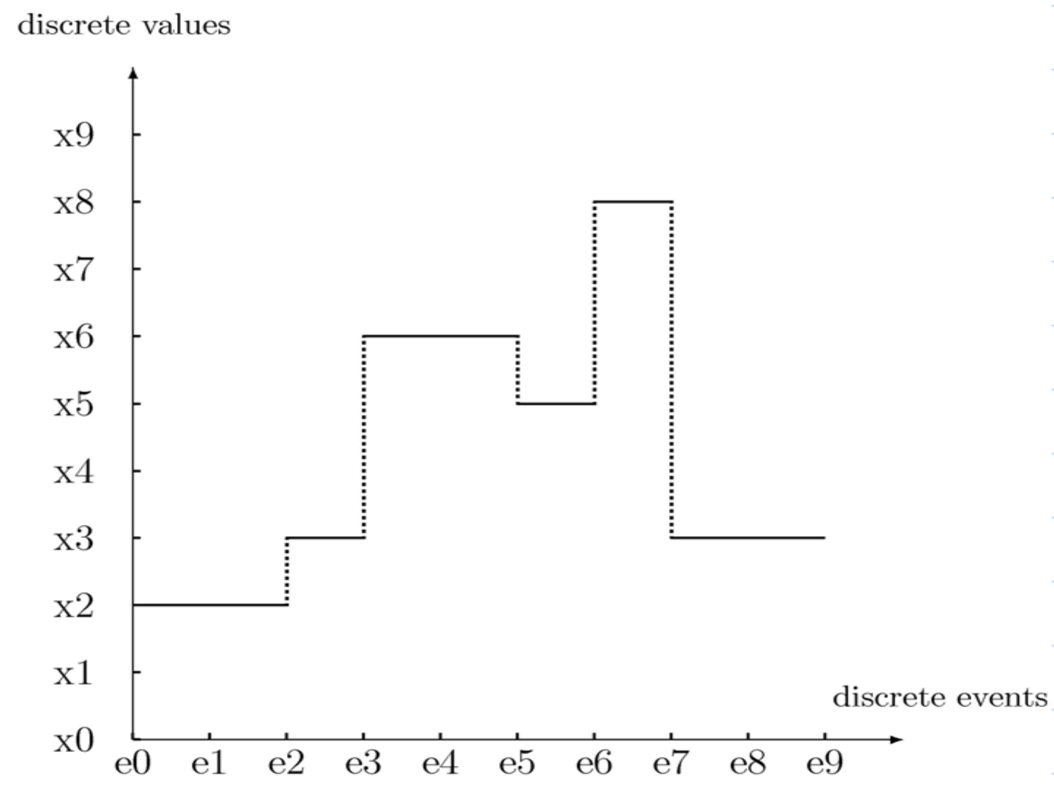
- Introduction
  - Basic definition of DEDS
  - Petri nets in DEDS modelling
  - Directed graphs in DEDS modelling
- DEDS control synthesis
  - Definition of the control synthesis
  - Basic principle of the proposed control synthesis method

- Examples
  - Client-server cooperation
  - Two agents cooperation
- Adaptivity
  - Simple example
- Petri nets in problem solving
  - Problem solving and causality
  - Petri nets & reachability graphs in problem solving

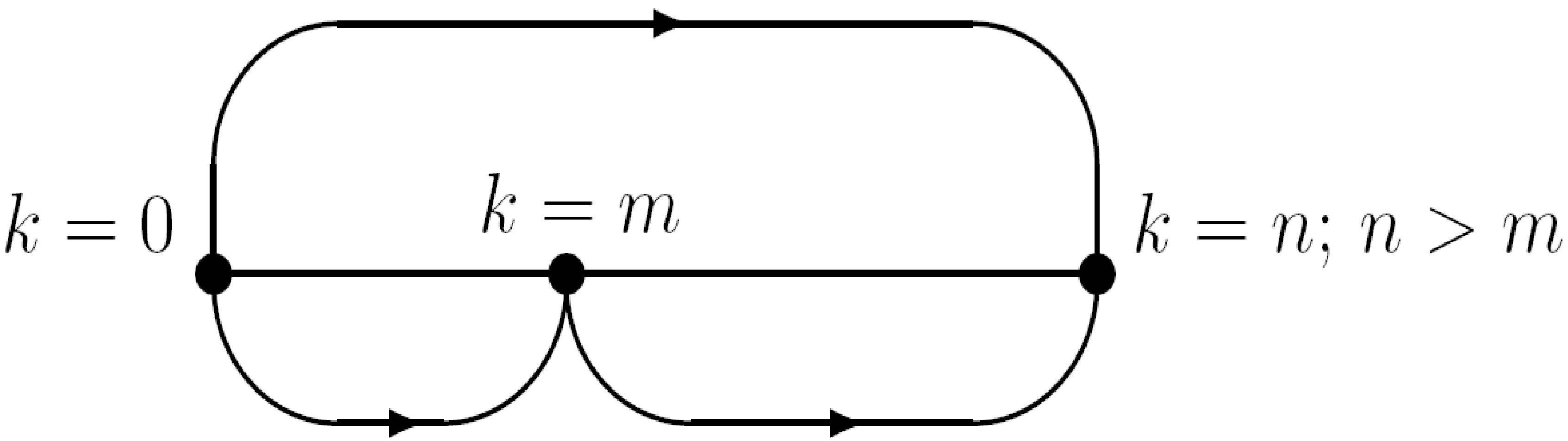
- Problem solving & DES
  - Solving the DES control synthesis problem
  - Agent-based approach to DES control synthesis
  - Example – the maze problem solving
- Approaches to solving the problem
  - Mutual intersection of autonomous solutions
  - Solving the global problem in the whole
  - Utilizing P-invariants of Petri net based model

- Modular approach to agents modelling
- Engineering applications
  - Assembly & disassembly process
  - Flexible manufacturing system
- Conclusions

The graphical expression of a DEDS dynamics development



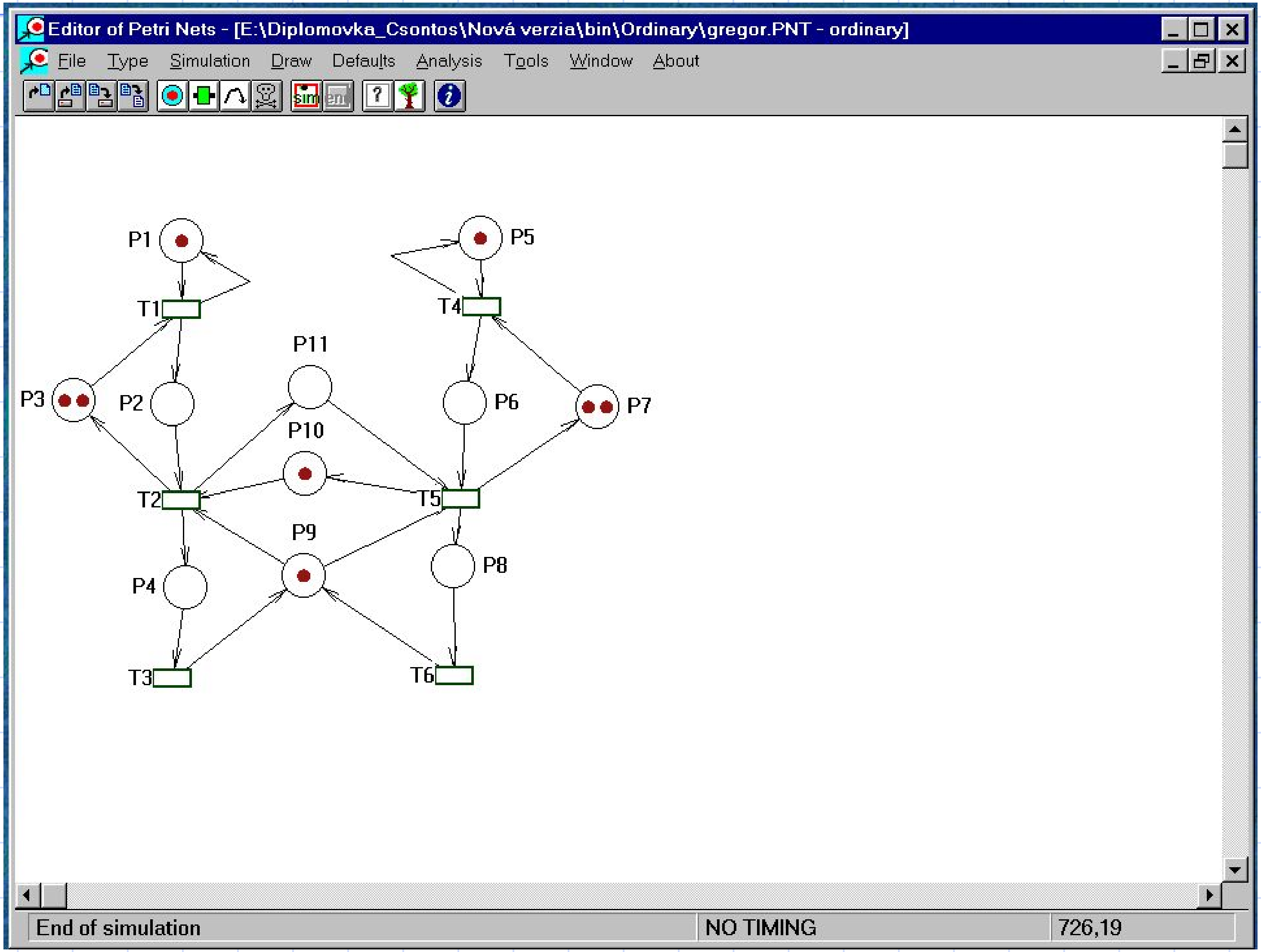
Causality



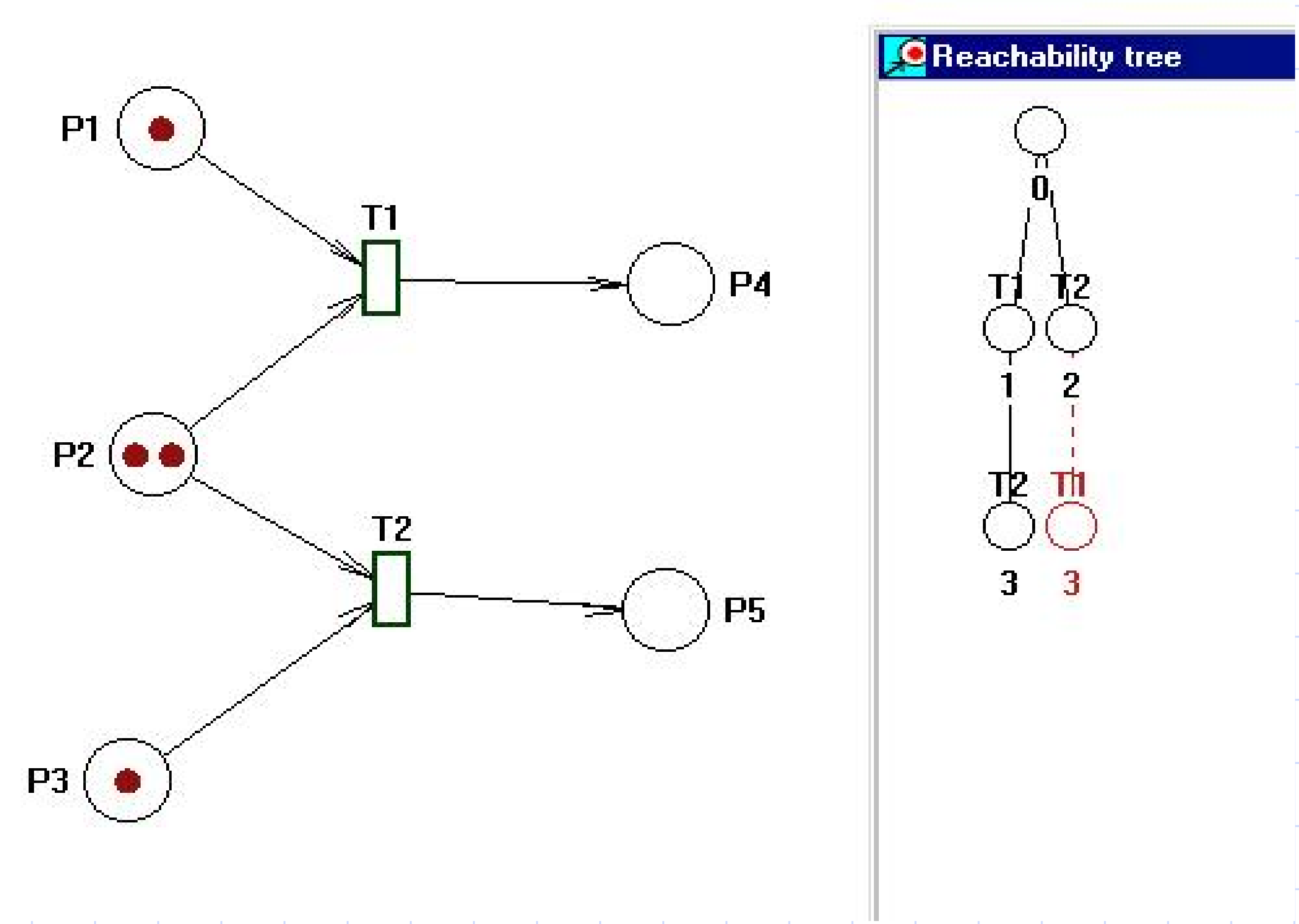
Why Petri nets in DEDS modelling ????

- PN are able to express parallelism and conflict situations
- PN can be expressed in analytical terms (in the form of the linear discrete system) as well as in the graphical form
- PN properties can be tested by means of the reachability tree and invariants
- PN allow to use analytical approach to the DEDS control synthesis
- PN make possible to quantify (model) problems that are given e.g. only verbally

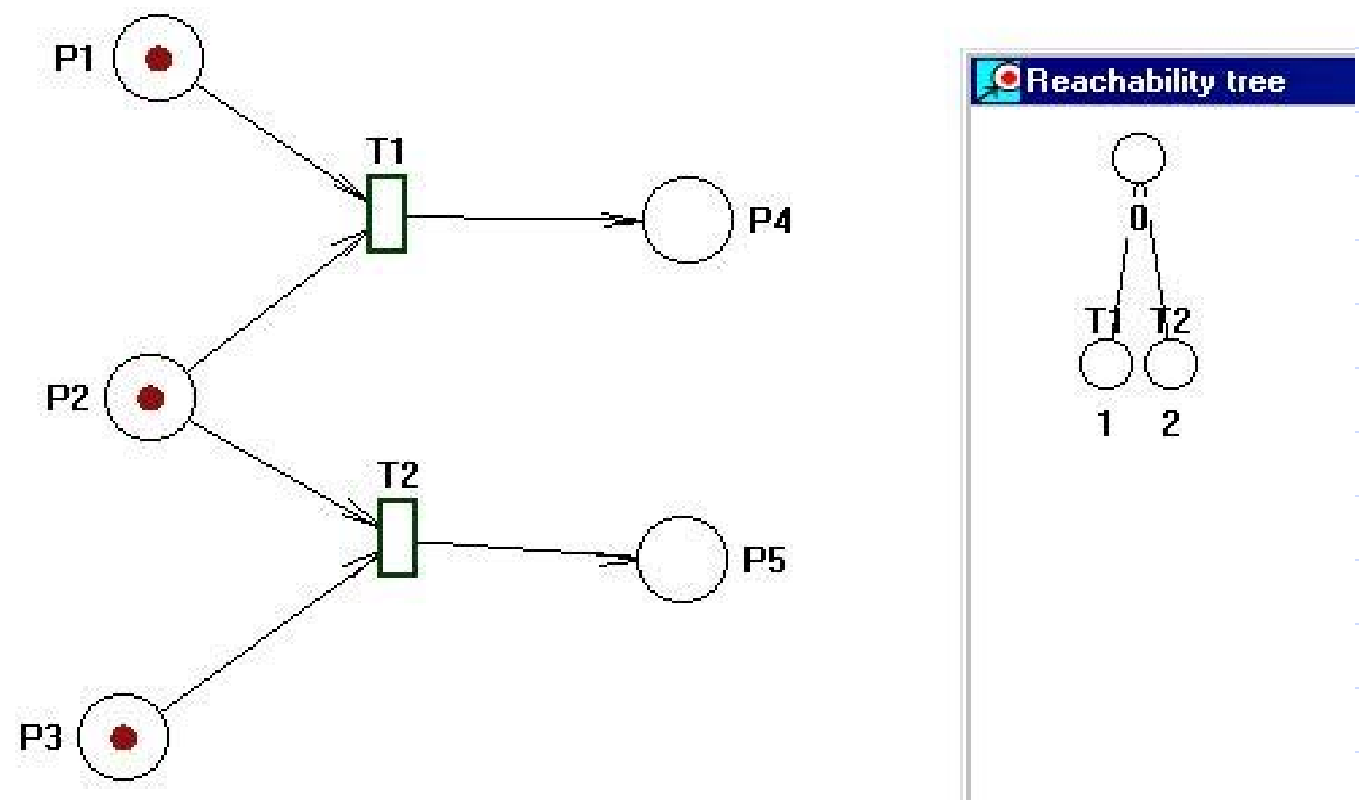
An example of a Petri net



Parallelism



Conflict situation



Formal expression of the Petri net structure

$$\langle P, T, F, G \rangle; \quad P \cap T = \emptyset; \quad F \cap G = \emptyset$$

$P = \{p_1, ..., p_n\}$ Set of the PN places

$T = \{t_1, ..., t_m\}$ Set of PN transitions

$F \subseteq P \times T$ Set of PN arcs from places to transitions

$G \subseteq T \times P$ Set of PN arcs from transitions to places

Formal expression of the Petri net dynamics

$$\langle X, U, \delta, \mathbf{x}_0 \rangle ; \quad X \cap U = \emptyset$$

$$X = \{\mathbf{x}_0, \ \mathbf{x}_1 \ldots, \ \mathbf{x}_N\}$$ Set of state vectors

$$U = \{\mathbf{u}_0, \ \mathbf{u}_1 \ \ldots, \ \mathbf{u}_N\}$$ Set of control vectors

$$\delta : X \times U \longrightarrow X$$ Transition function

$\mathbf{x}_0$ is an initial state

Mathematical model of the Petri net

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k \quad , \quad k = 0, N$$

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F}$$

$$\mathbf{F}.\mathbf{u}_k \leq \mathbf{x}_k$$

$$\mathbf{x}_k = (\sigma_{p_1}^k, \ldots, \sigma_{p_n}^k)^T$$

$$\sigma_{p_i}^k \in \{0, c_{p_i}\}, \; i = 1, n$$

$$\mathbf{u}_k = (\gamma_{t_1}^k, \ldots, \gamma_{t_m}^k)^T$$

$$\gamma_{t_j}^k \in \{0, 1\}, \; j = 1, m$$

A more general PN-based mathematical model of DEDS (with the multiplicity of arcs)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k\,, \quad k = 0, \ldots, N \qquad (1)$$

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} \qquad (2)$$

$$\mathbf{F}.\mathbf{u}_k \leq \mathbf{x}_k \qquad (3)$$

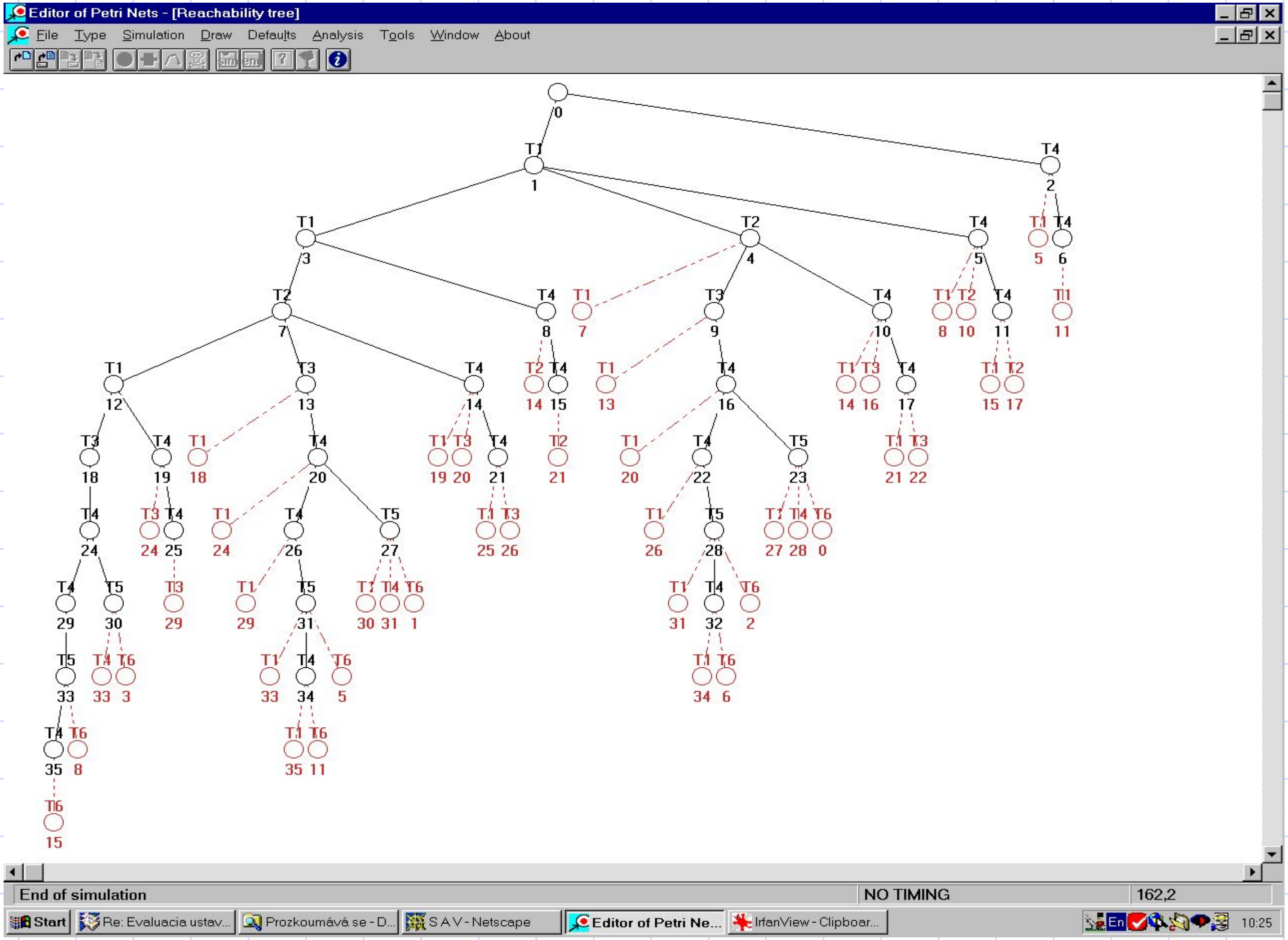
$$\mathbf{x}_k = (\sigma_{p_1}^k, \ldots, \sigma_{p_n}^k)^T \quad \sigma_{p_i}^k \in \{0, c_{p_i}\}$$

$$\mathbf{u}_k = (\gamma_{t_1}^k, \ldots, \gamma_{t_m}^k)^T, \quad \gamma_{t_j}^k \in \{0, 1\}$$

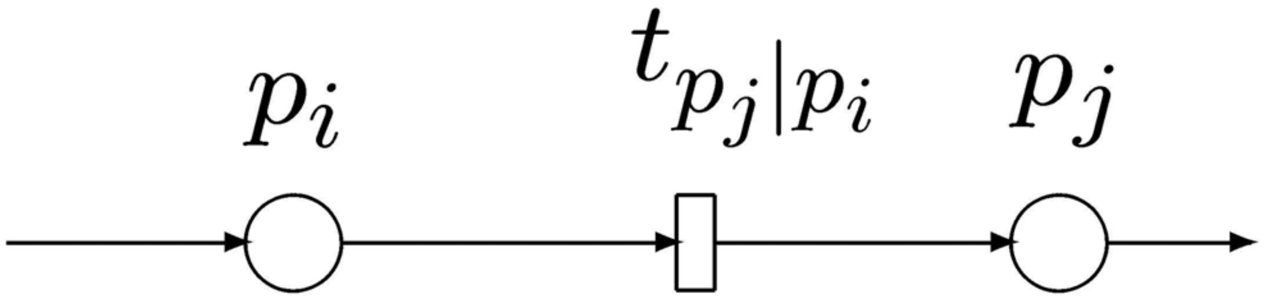
$$\mathbf{F} = \{f_{ij}\};\ i = 1, \ldots, n,\ j = 1, \ldots, m;\ f_{ij} \in \{0, M_{f_{ij}}\}$$

$$\mathbf{G} = \{g_{ij}\};\ i = 1, \ldots, m,\ j = 1, \ldots, n;\ g_{ij} \in \{0, M_{g_{ij}}\}$$

Reachability tree of the above introduced Petri net



Directed graphs (DG) in DEDS modelling



$$\gamma^{(k)}_{t_{\pi_i|\pi_j}} \in \{0,\ 1\}$$

$$\mathbf{X}(k+1) = \mathbf{\Delta}_k.\mathbf{X}(k) \quad , \quad k = 0, N$$

$$\mathbf{X}(k) = (\sigma_{\pi_1}^{(k)}(\gamma), \ldots, \sigma_{\pi_{n_{RT}}}^{(k)}(\gamma))^T, \; k = 0, N$$

$$\mathbf{\Delta}_k = \{\delta_{ij}^{(k)}\}_{n_{RT} \times n_{RT}}$$

$$\delta_{ij}^{(k)} = \gamma_{t_{\pi_i|\pi_j}}^{(k)}, \; i = 1, n_{RT}, \; j = 1, n_{RT}$$

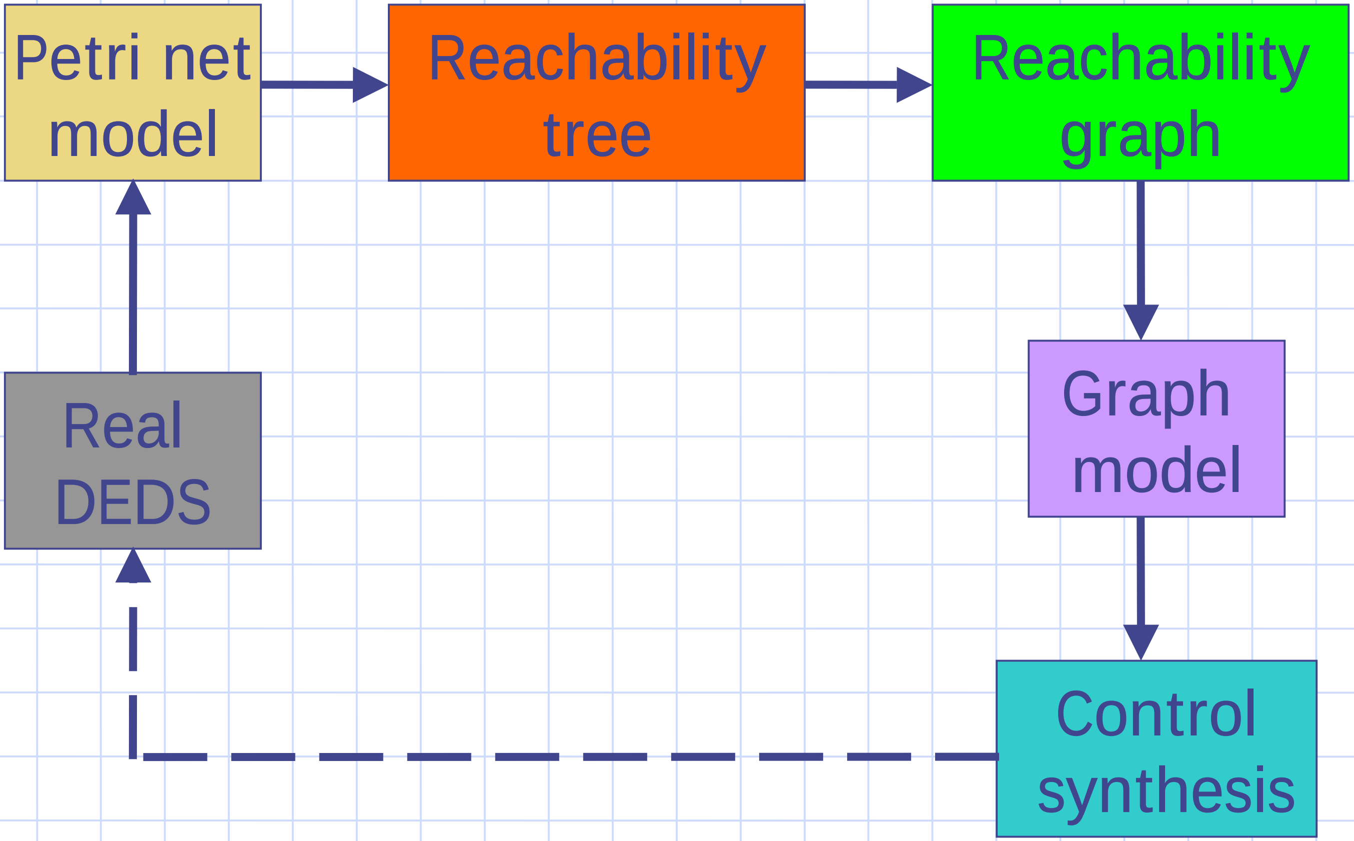
State machines

Petri nets where each transition has only one input and only one output position are named state machines. They can be modelled by directed graphs (DG) without any problem.

Petri nets with general structure

In case of the general structure, when any transition is allowed to have more input positions and more output ones, the PN reachability graph has to be used.

Transforming the PN model to the DG model



DEDS control synthesis

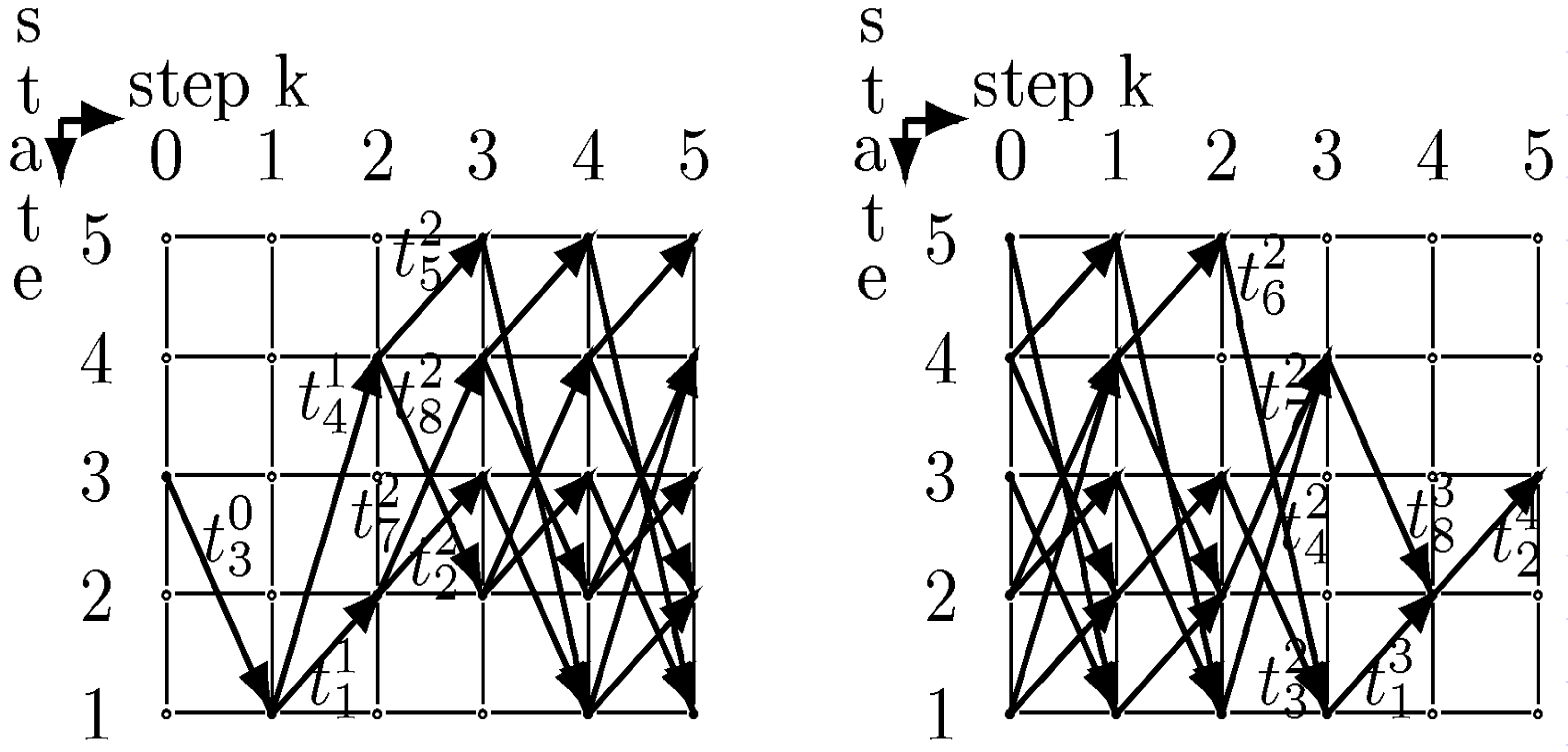
Definition of the control synthesis

Control synthesis = finding the most suitable sequence of discrete events (control interferences) which is able to ensure the transition (transformation) of the system from a given initial state into a prescribed terminal state at simultaneous fulfilling control task specifications that are imposed on the control task.

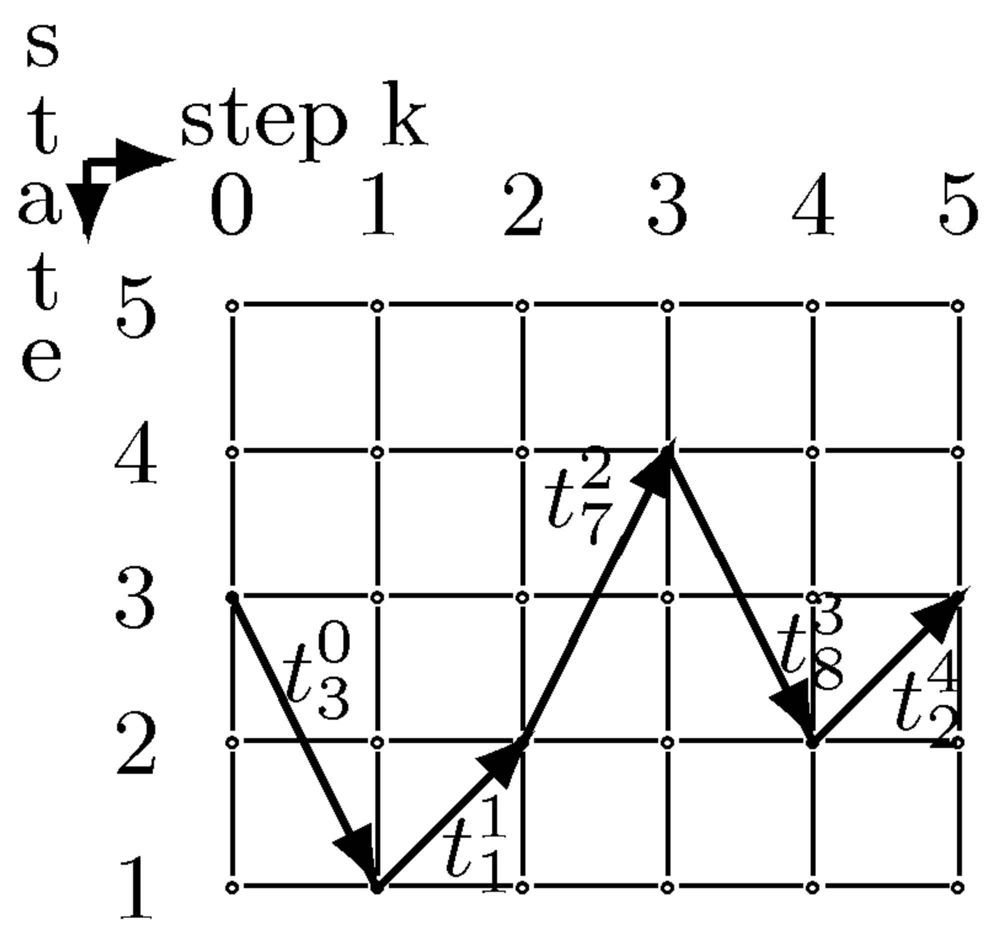
Control task specifications = criteria, constraints, etc. Usually, they are not given in analytical terms. Even, often they are given only verbally.

Basic principle of the proposed control synthesis method

Straight-lined reachability tree and the backtracking one



Intersection of the trees yields the state trajectory(-ies)



Procedure in analytical terms

- The staight-lined reachability tree (SLRT)

$$
\begin{aligned}
\{\mathbf{X}_1\} &= \mathbf{\Delta}.\mathbf{X}_0 \\
\{\mathbf{X}_2\} &= \mathbf{\Delta}.\{\mathbf{X}_1\} = \mathbf{\Delta}.(\mathbf{\Delta}.\mathbf{X}_0) = \mathbf{\Delta}^2.\mathbf{X}_0 \\
& \quad \dots \qquad \dots \qquad \dots \\
\{\mathbf{X}_N\} &= \mathbf{\Delta}.\{\mathbf{X}_{N-1}\} = \mathbf{\Delta}^N.\mathbf{X}_0
\end{aligned}
$$

- The backtracking (backward) reachability tree (BTRT)

$$
\begin{aligned}
\{\mathbf{X}_{N-1}\} &= \mathbf{\Delta}^{T}.\mathbf{X}_{N} \\
\{\mathbf{X}_{N-2}\} &= \mathbf{\Delta}^{T}.\{\mathbf{X}_{N-1}\} = (\mathbf{\Delta}^{T})^{2}.\mathbf{X}_{N} \\
& \cdots \qquad \cdots \qquad \cdots \\
\{\mathbf{X}_{0}\} &= \mathbf{\Delta}^{T}.\{\mathbf{X}_{1}\} = (\mathbf{\Delta}^{T})^{N}.\mathbf{X}_{N}
\end{aligned}
$$

The intersection of the SLRT and BTRT

$$
\begin{aligned}
\mathbf{M}_1 &= (\mathbf{X}_0, {}^1\{\mathbf{X}_1\}, \ldots, {}^1\{\mathbf{X}_{N-1}\}, {}^1\{\mathbf{X}_N\}) \\
\mathbf{M}_2 &= ({}^2\{\mathbf{X}_0\}, {}^2\{\mathbf{X}_1\}, \ldots, {}^2\{\mathbf{X}_{N-1}\}, \mathbf{X}_N) \\
\mathbf{M} &= \mathbf{M}_1 \cap \mathbf{M}_2 \\
\mathbf{M} &= (\mathbf{X}_0, \{\mathbf{X}_1\}, \ldots, \{\mathbf{X}_{N-1}\}, \mathbf{X}_N)
\end{aligned}
$$

Using the principle of causality

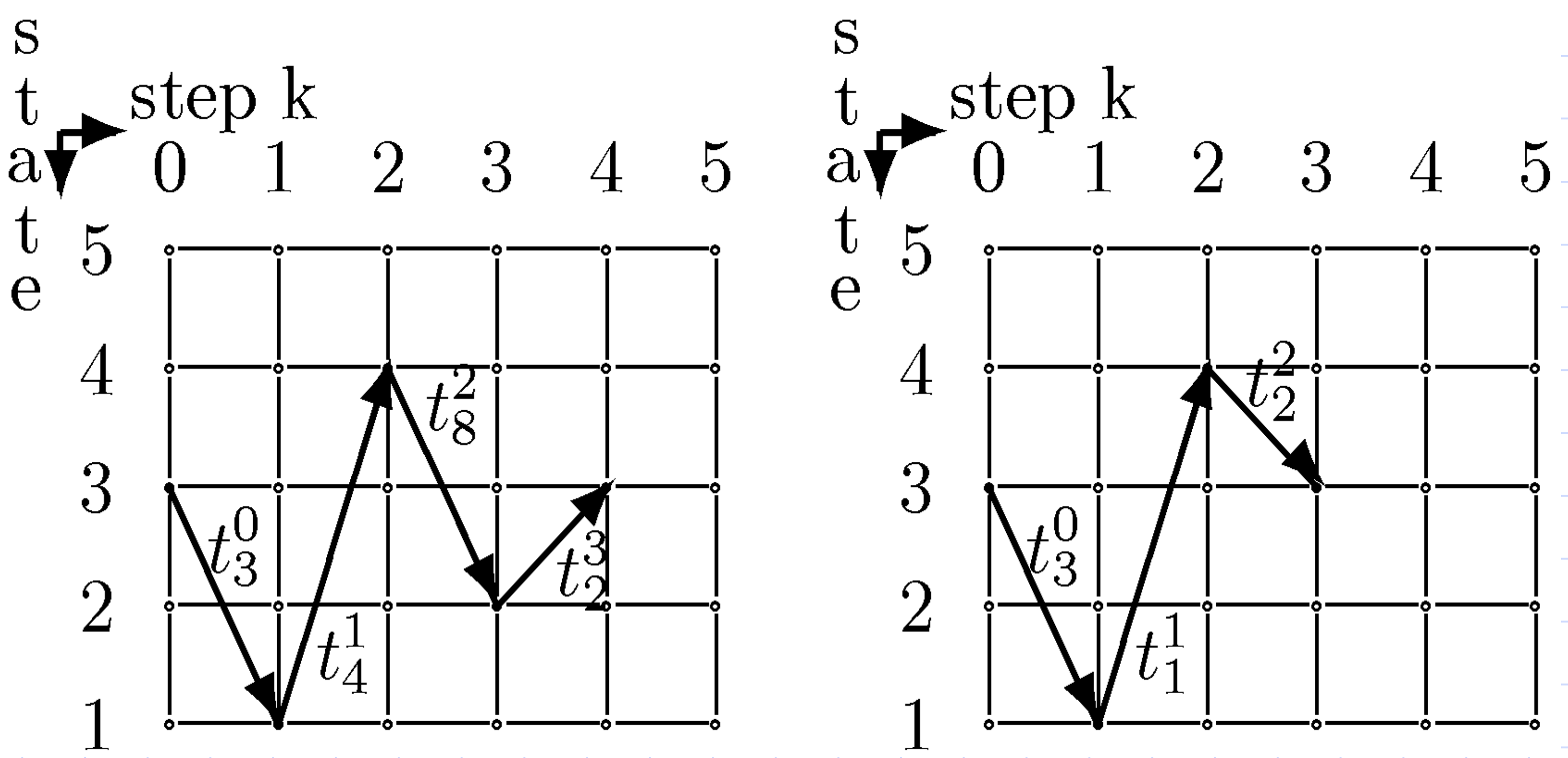
- Due to the principle of causality any shorter feasible solution is involved in the longer feasible one. Hence, when

$\mathbf{M}_2$ is shifted to the left before the intersection.

$$^{-1}\mathbf{M} = (\mathbf{x}_0, \{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_{N-2}\}, \mathbf{x}_{N-1}) \qquad (18)$$

where $\mathbf{x}_{N-1} = \mathbf{x}_t$.
Shifting (finding the $(n \times (N-k+1))$ matrices $^{-k}\mathbf{M}, k = 1, 2, \ldots$) can continue until the intersections exists, i.e. until $\mathbf{x}_0 \in {}^2\{\mathbf{x}_k\}$ and $\mathbf{x}_t \in {}^1\{\mathbf{x}_{N-k}\}$.

Shorter trajectories obtained by shifting



Example 1 – Client-server connection

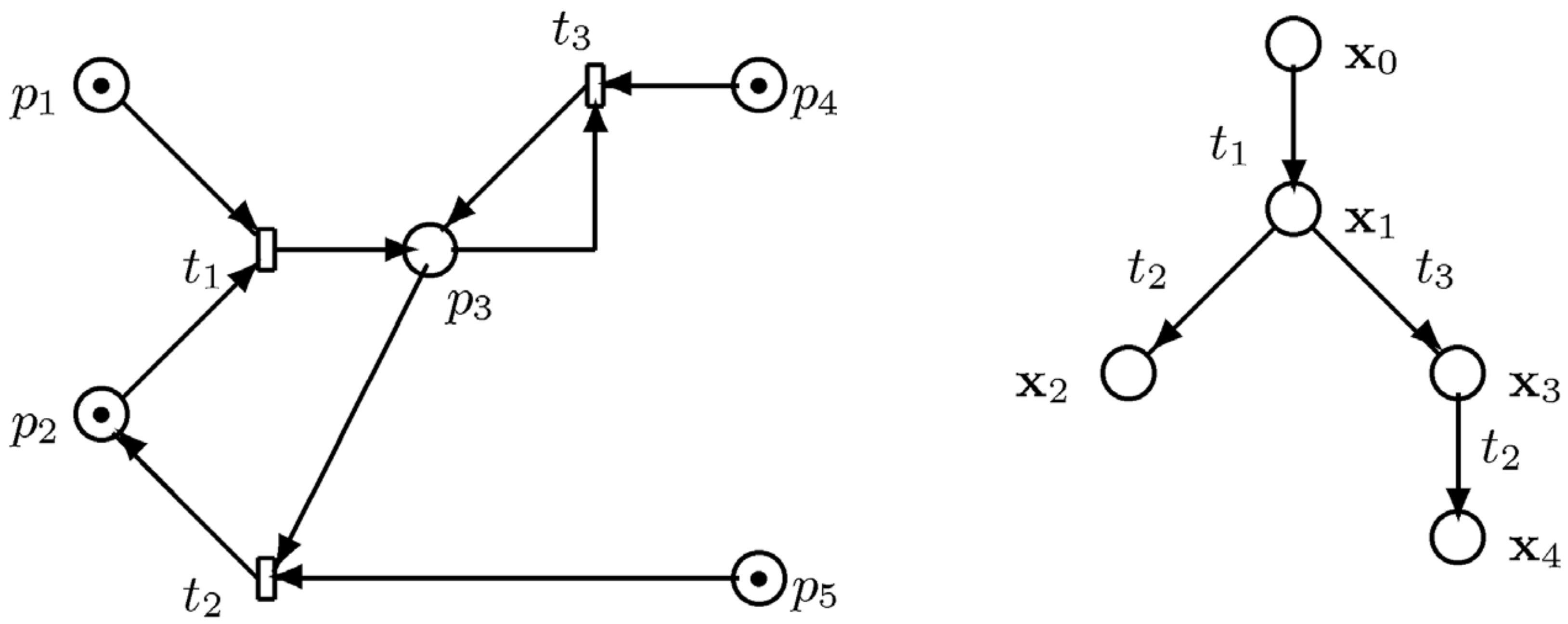
The places of the PN-based model

p1 - the client (C) requests for the connection
p2 - the server (S) is listening
p3 - the connection of C with S
p4 - the data sent by C to S
p5 - the disconnection of C by the C himself

The transitions of the PN-based model

t1, t2, t3 – discrete events realizing the system dynamics

PN-based model and corresponding reachability tree



Parameters of the PN-based model

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_0 = (1,\ 1,\ 0,\ 1,\ 1)^T$$

MATLAB procedure for enumerating the RT

```
Xreach=x0
Art=[0]
[n,m]=size(F);
B=Gt-F
i=0
while i < size(Xreach,2)
        i=i+1;
        for k=1:m
            x(k)=all(Xreach(:,i) >= F(:,k));
        end
        findx=find(x)
        for k=1:size(findx,2)
```

```
            bb = Xreach(:,i)+B(:,findx(k));
            matrix=[];
            for j=1:size(Xreach,2)
                matrix=[matrix,bb];
            end;
            v=all(matrix == Xreach);
            j=find(v);
            if any(v)
                Art(i,j)= findx(k);
            else
                Xreach=[Xreach,bb];
                Art(size(Art,1)+1,size(Art,
                Art(i,size(Art,2))=findx(k)
            end;
        end;
Xreach;
Art;
end
```

Enumerated RT

Quasi-functional adjacency matrix of RT

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \boldsymbol{\Delta}_k^T = \begin{pmatrix} 0 & t_1 & 0 & 0 & 0 \\ 0 & 0 & t_2 & t_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Space of reachable states

$$\mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Example 2 – Two agents cooperation

The agent A needs to do the activity (i.e. to solve a problem) P. However, A is not able to do P. Consequently, A requests the agent B to do P for him.

The places of the PN-based model:

p1 – A wants to do P
p2 - A waits for an answer from B
p3 - A waits for a help from B
p4 - the failure of the cooperation
p5 - the satisfying cooperation
p6 - A requests B to do P
p7 - B refuses to do P

p8 - B accepts the request of A to do P

p9 - B is not able to do P

p10 - doing P by B

p11 - B receives the request of A

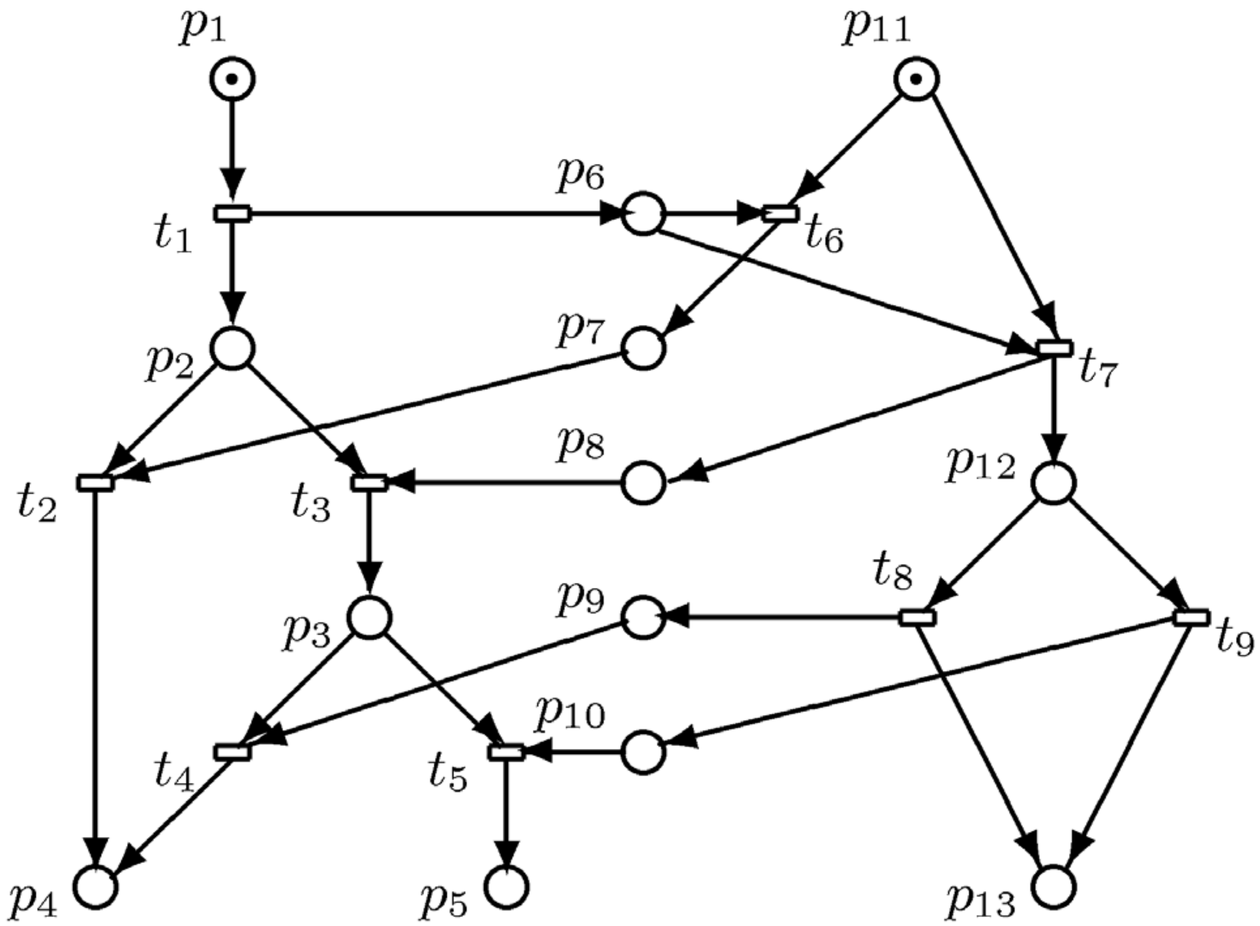
p12 - B is willing to do P for A

p13 - the end of the work of B

The transitions of the PN-based model:

t1 – t9 represent discrete events realizing the system dynamics

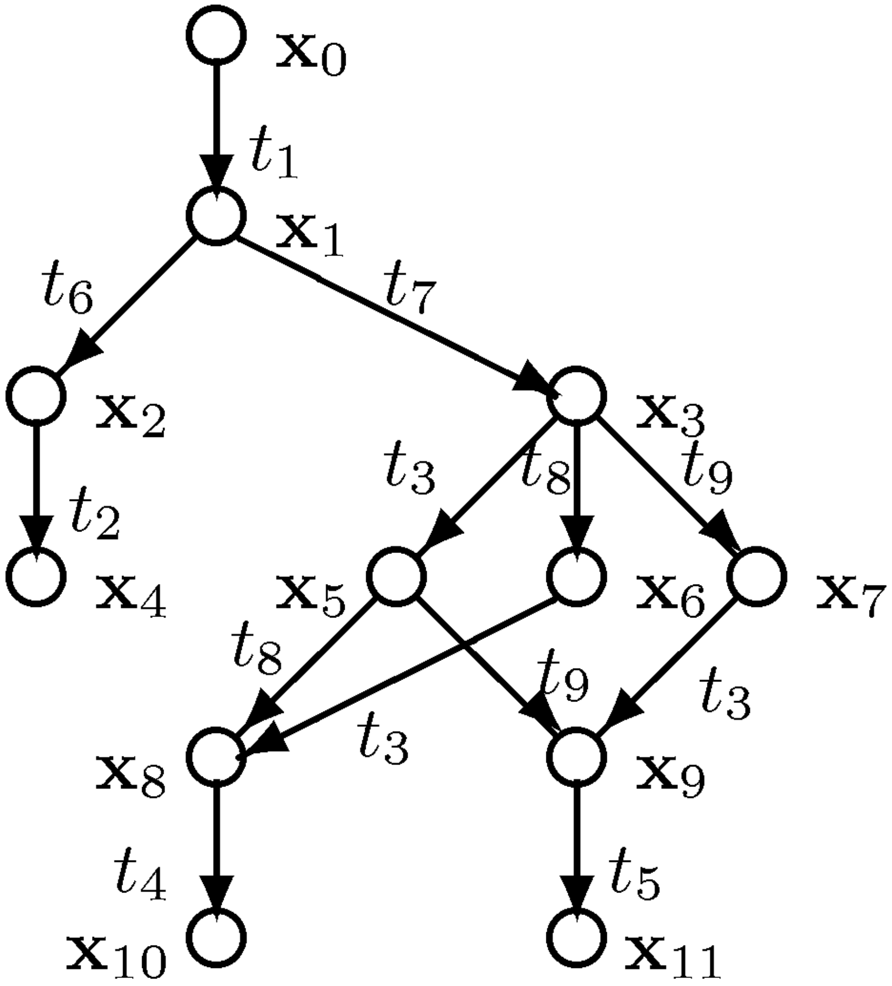
PN-based model



Enumerated RT

$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 8 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The reachability graph



The space of reachable states

$$\mathbf{X}_{reach} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

Control synthesis

The initial state

$$\mathbf{x}_0 = (1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0)^T$$

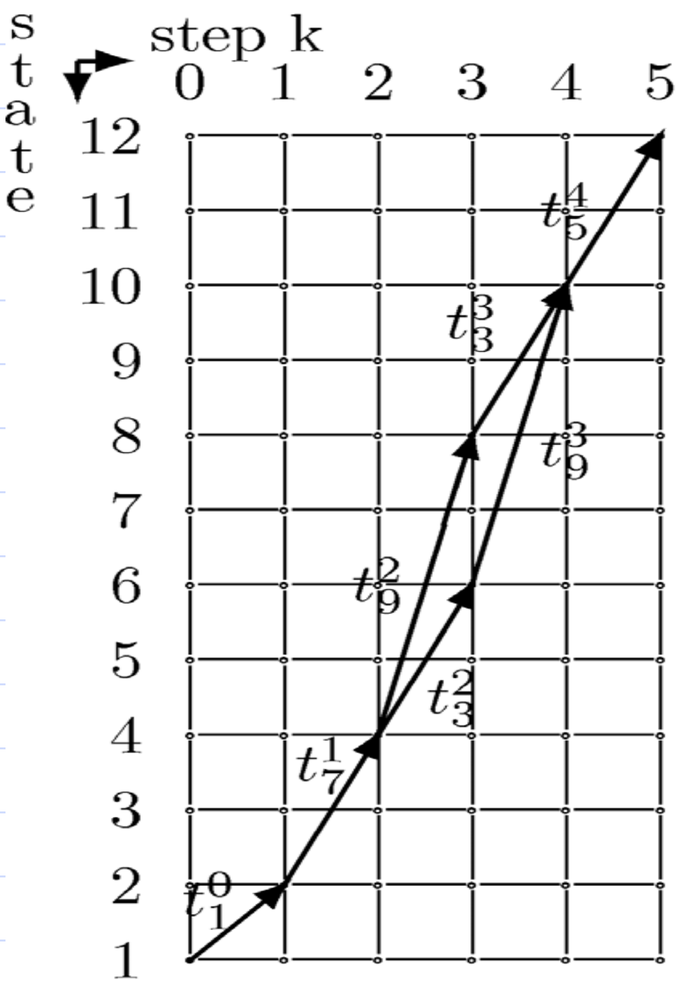
The terminal state – the successful cooperation

$$\mathbf{x}_N = (0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1)^T$$

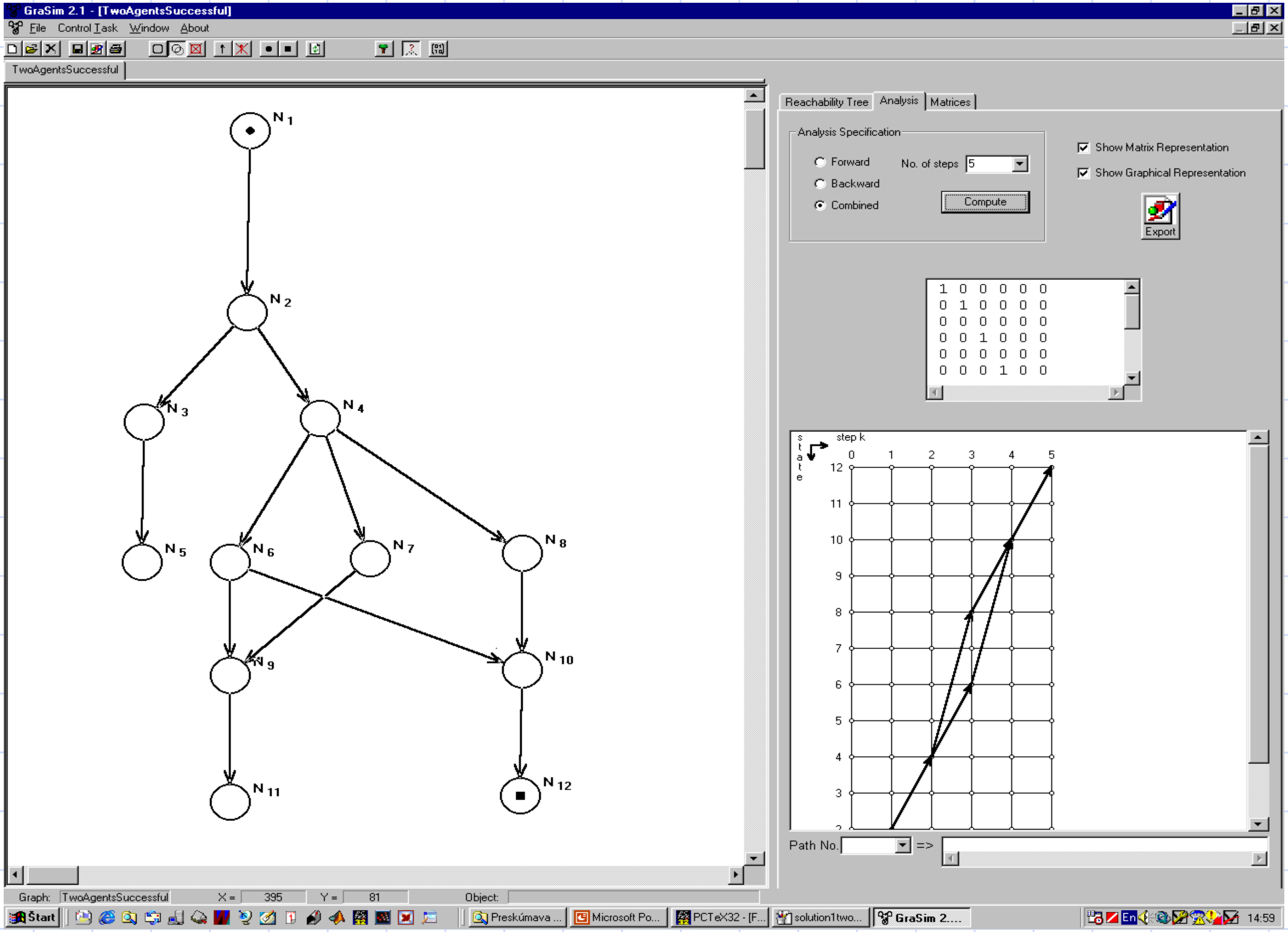
The intersection of the SLRT and BTRT

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

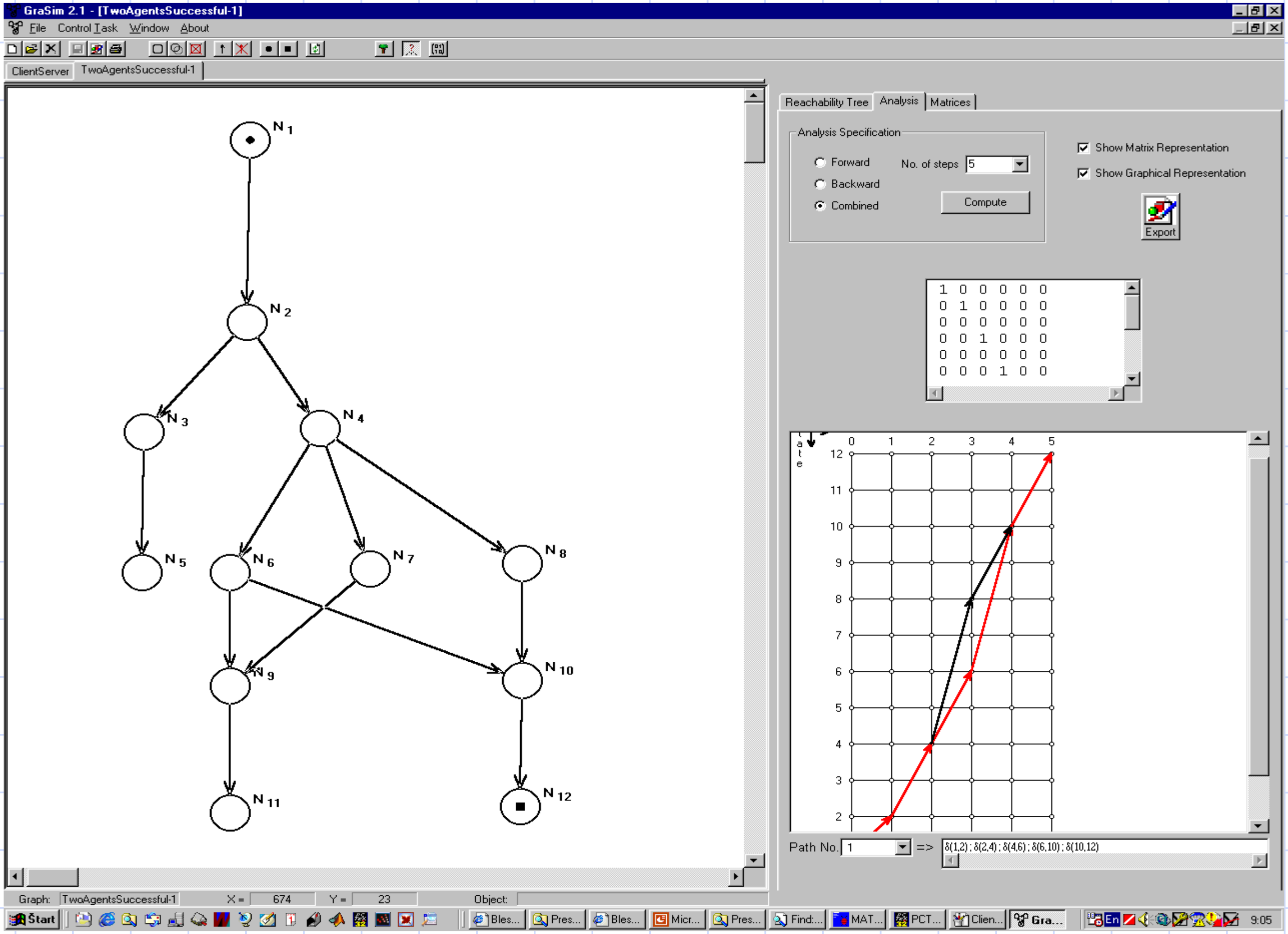
The state trajectories – the successful cooperation



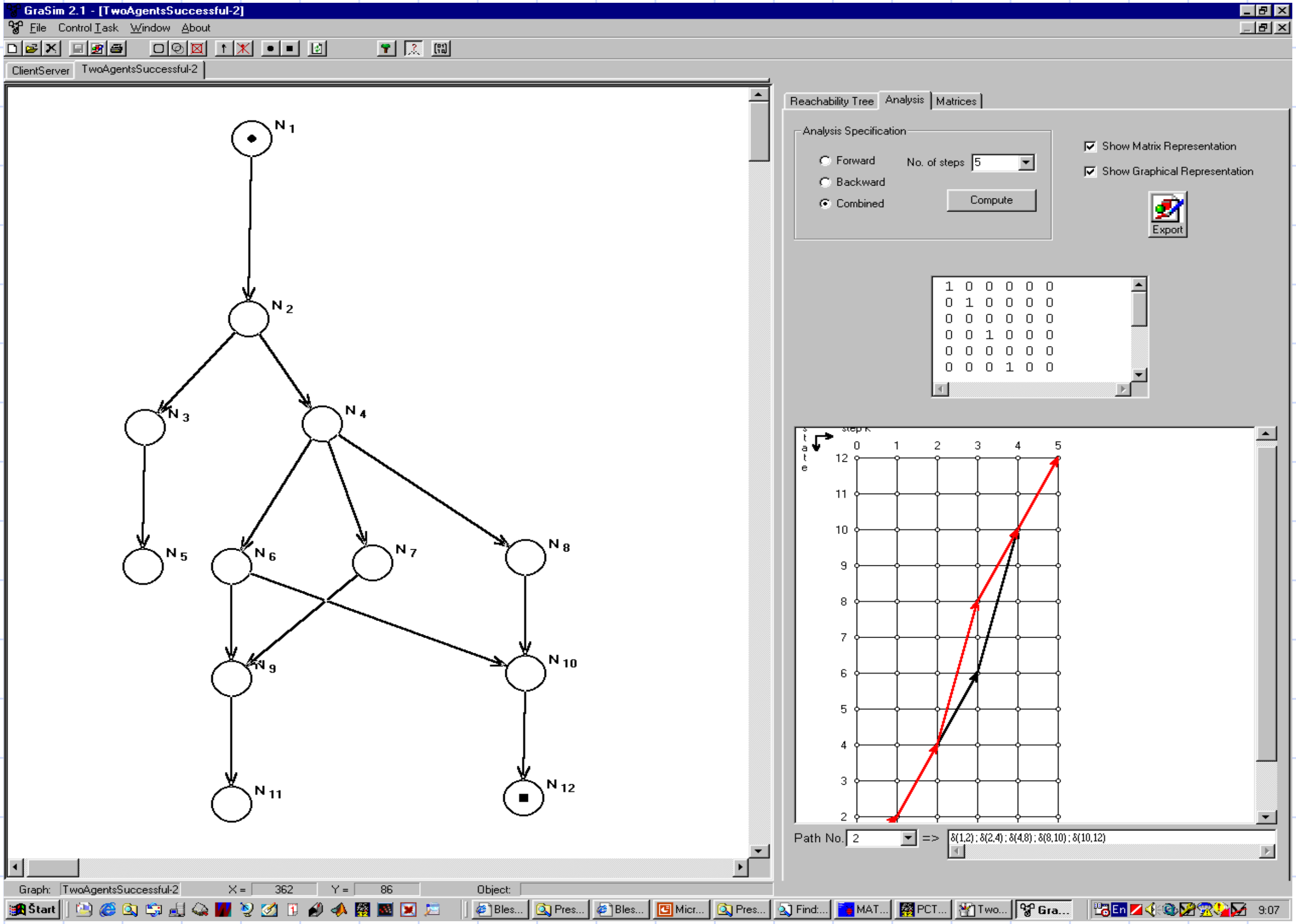
Graphic tool - GraSim



Succesfull cooperation 1

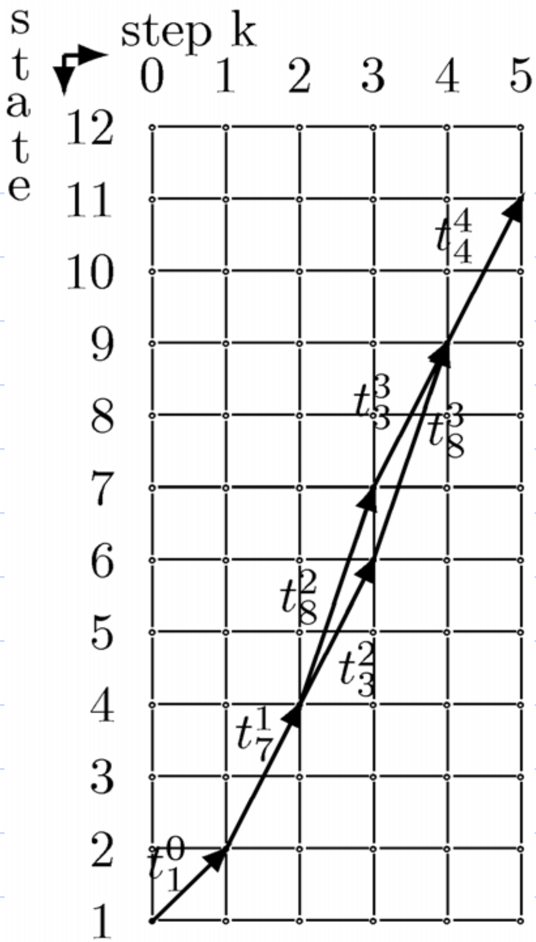


Succesfull cooperation 2



The failured cooperation – when B is not able to do P

$$\mathbf{x}_N = (0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1)^T$$

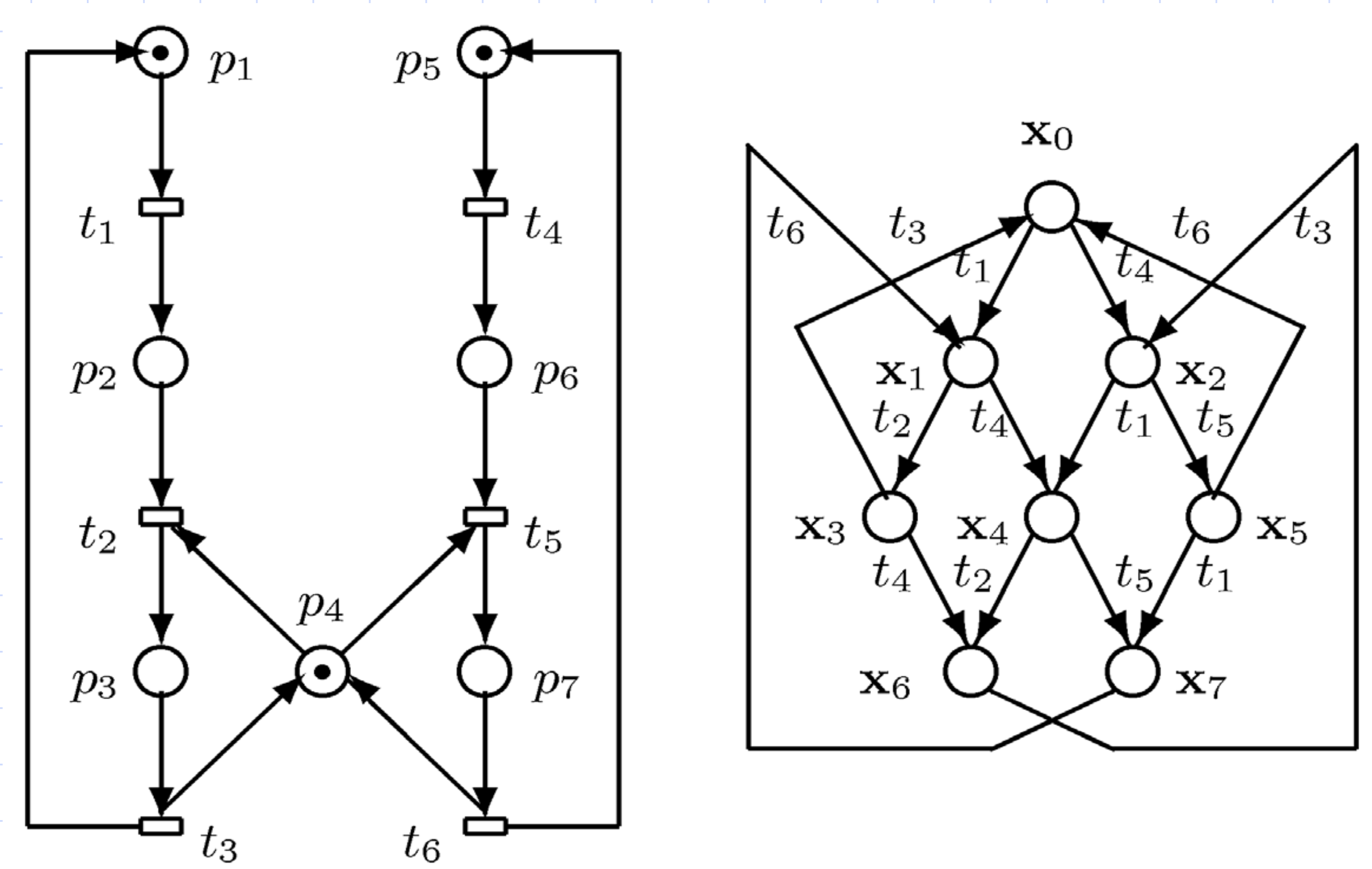


Adaptivity (flexibility)

There are two kinds of the adaptivity in the DEDS control synthesis

- Choosing the most suitable trajectory from the feasible ones in order to adapt the system behaviour to external demands (conditions)
- Changing the structure of the system model in order to express more kinds of the system behaviour.
  - Choosing the most suitable behaviour from the feasible ones. It is illustrated in the next example.

Example 3 – Two processes



{p1, p2, p3} – 1st process ***P1***
{p5, p6, p7} – 2nd process ***P2***

p4 – the structural element that is able to influence

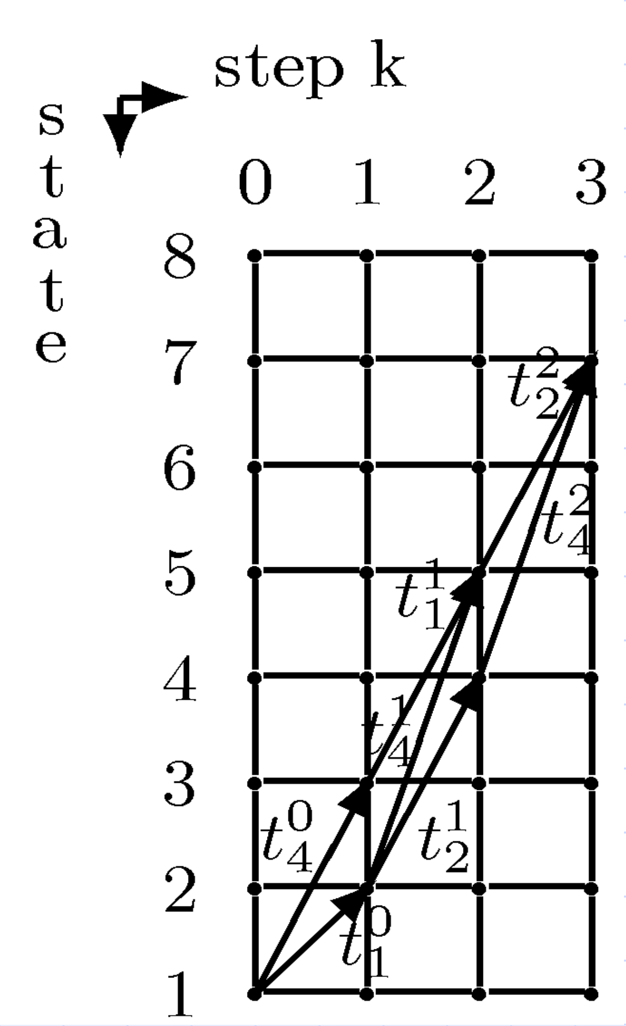
- the mutual exclusion of ***P1*** and ***P2***
- the sequencing of ***P1*** and ***P2***
- the re-running of ***P1*** and ***P2***

$$\mathbf{x}_0 = (1,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0)^T \qquad \mathbf{x}_6 = (0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0)^T$$

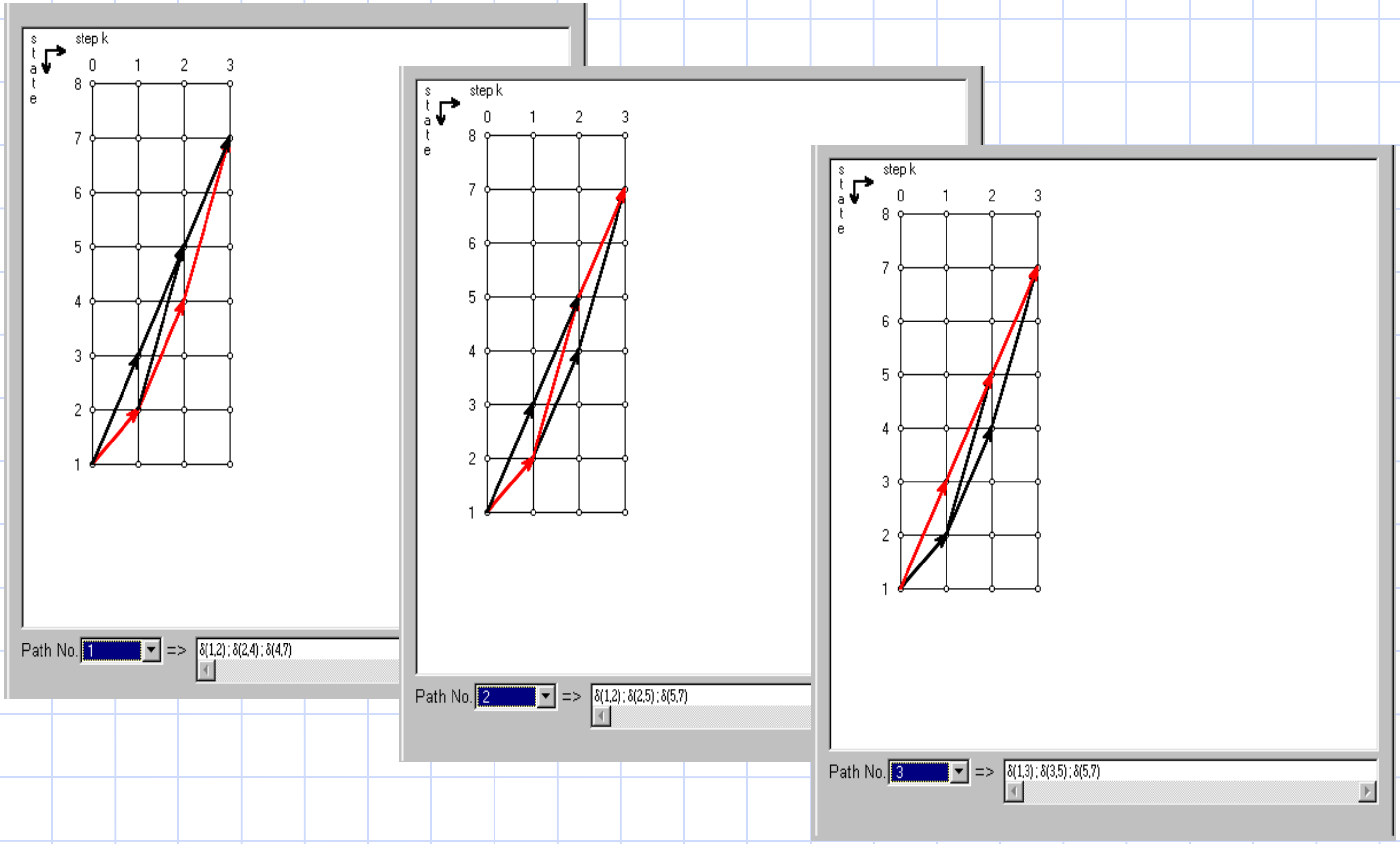
$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Exclusion of the process *P2*



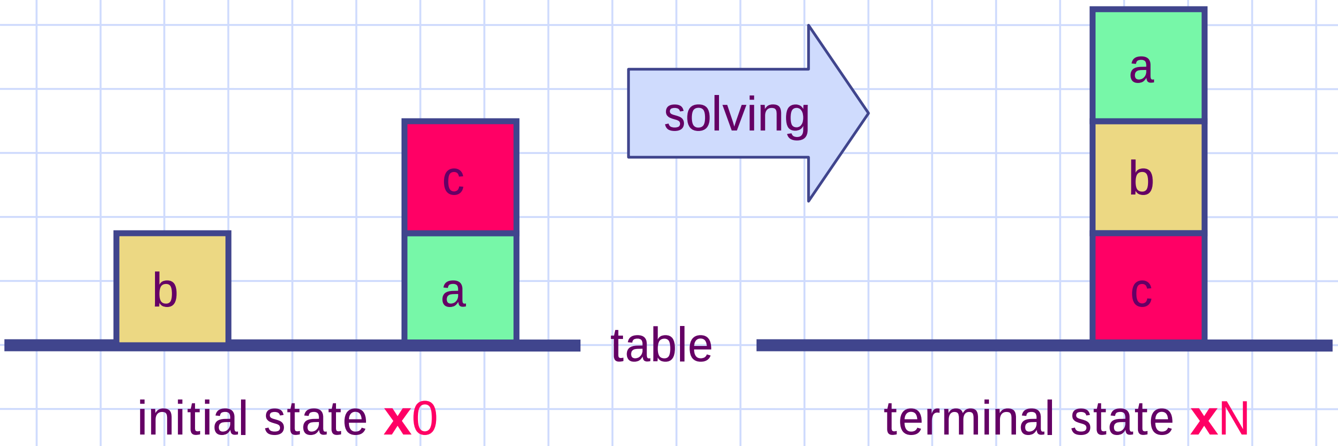
Three possibilities of the ***P2*** exclusion



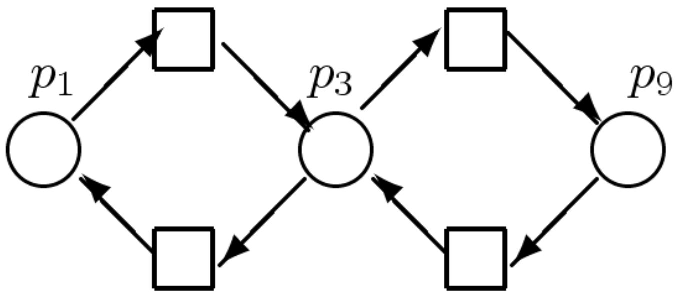
Petri nets in problem solving

Causality and problem solving

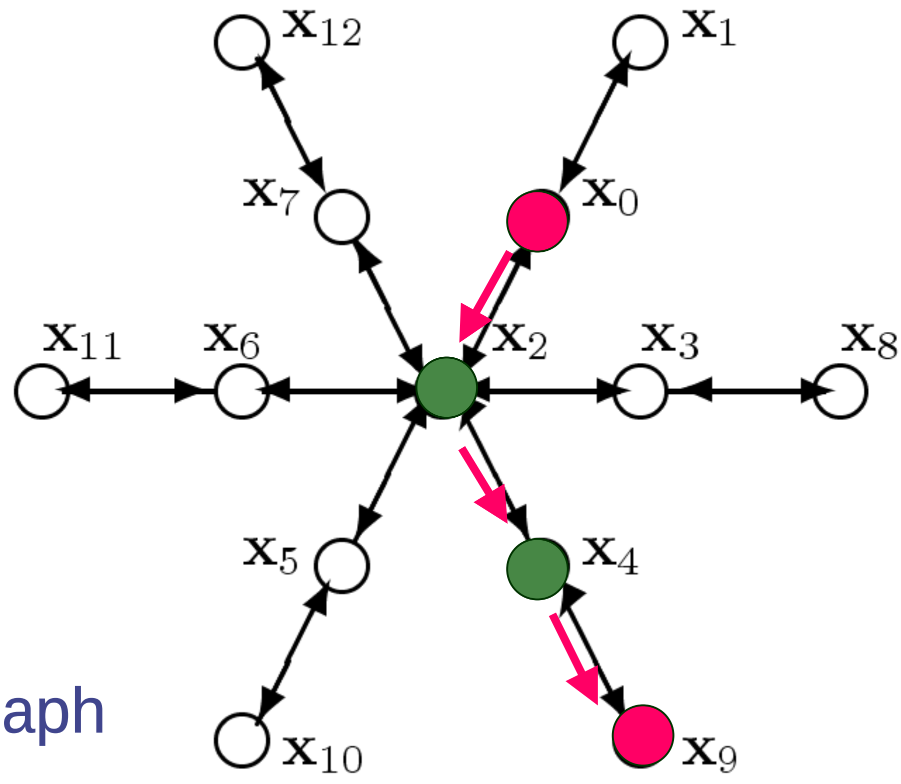
STRIPS (STanford Research Institute Problem Solver)
[Fikes and Nilsson]
is associated with the so called *block world* paradigm.



Expressing causality by Petri nets and reachability graphs



A fragment of PN



Full reachability graph

Interpretation of the Petri net places *wrt.* the problem

p1 – a, b, c are separated on the table (T); {a,b,c}
p2 – a, b are separated on T; c is on b; {a,c/b}
p3 – a, c are separated on T; b is on c; {a,b/c}
p4 – a, b are separated on T; c is on a; {c/a,b}
p5 – b, c are separated on T; a is on c; {b,a/c}
p6 – b, c are separated on T; a is on b; {a/b,c}
p7 – a, c are separated on T; b is on a; {b/a,c}
p8 – b is on T ; c is on b ; a is on c; {a/c/b}
p9 – c is on T ; b is on c ; a is on b; {a/b/c}
p10 - c is on T ; a is on c ; b is on a; {b/a/c}
p11 - a is on T ; c is on a ; b is on c; {b/c/a}
p12 - b is on T ; a is on b ; c is on a; {c/a/b}
p13 - a is on T ; b is on a ; c is on b; {c/b/a}

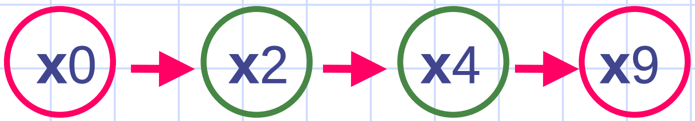
States as the nodes of the reachability tree

Solution of the problem

x0 = (0,0,0,1,0,0,0,0,0,0,0,0,0)' - the initial state

x9 = (0,0,0,0,0,0,0,0,1,0,0,0,0)' - the terminal state

The solution is



where

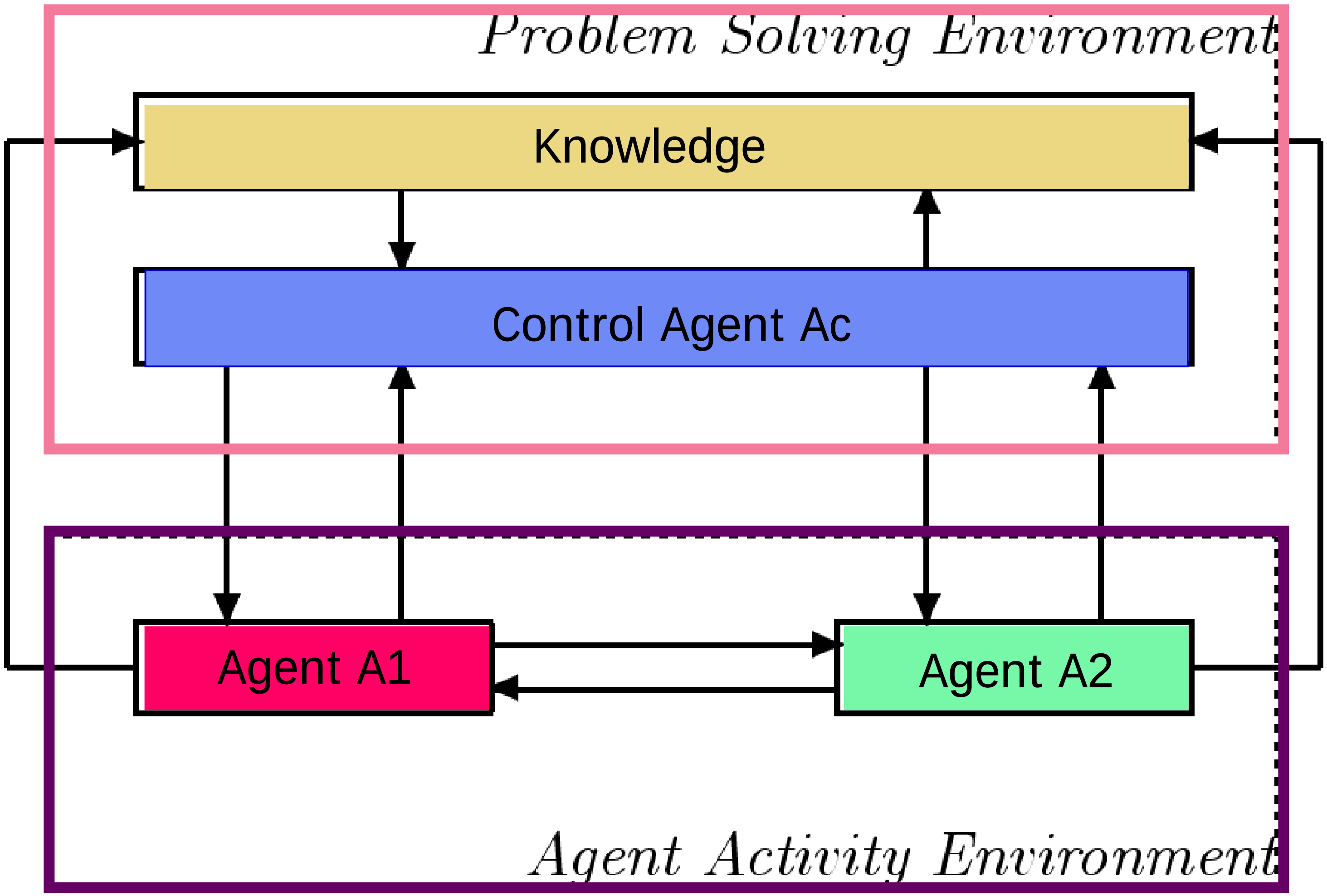
x2 = (1,0,0,0,0,0,0,0,0,0,0,0,0)'

x4 = (0,0,1,0,0,0,0,0,0,0,0,0,0)'

Solving the DES control synthesis problems

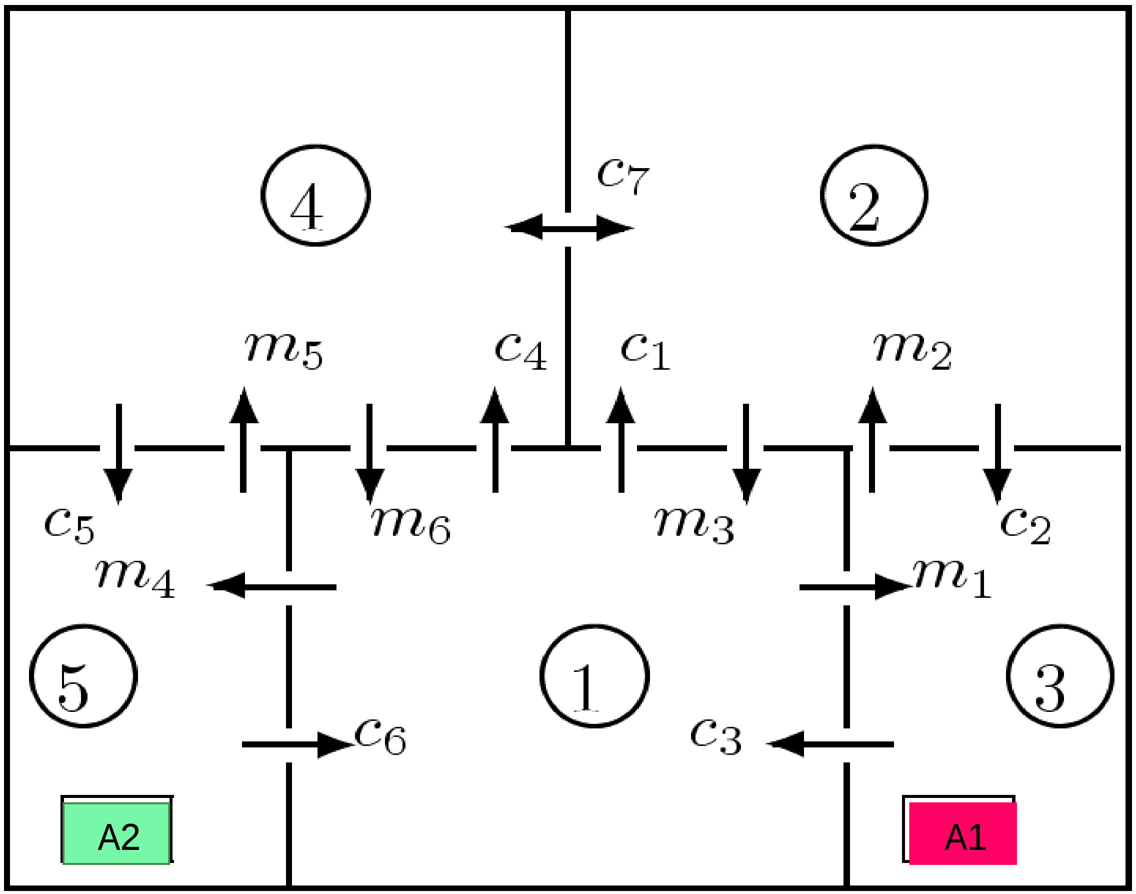
- Agent-based approach
- Engineering applications of AI methods
  - Block world paradigm
  - Hanoi tower paradigm
  - Control of the flexible manufacturing system

Agent-based approach to DES control synthesis



Example 4 – Agent-based control synthesis

Problem to be solved – the maze problem



Problem formulation

Maze consists of 5 rooms denoted as 1, 2, 3, 4, 5 connected by doors c1, c2, ..., c7, c8 for A1 and doors m1, m2, ..., m6 for A2.

Initially, A1 is in the room 3, A2 is in the room 5. Doors can be open (closed) by the control agent Ac. Only door c7 is permanently open (uncontrollable).

Agent Ac observes only discrete events from sensors built-in the doors.

The control synthesis problem is the following:

Control synthesis problem to be solved

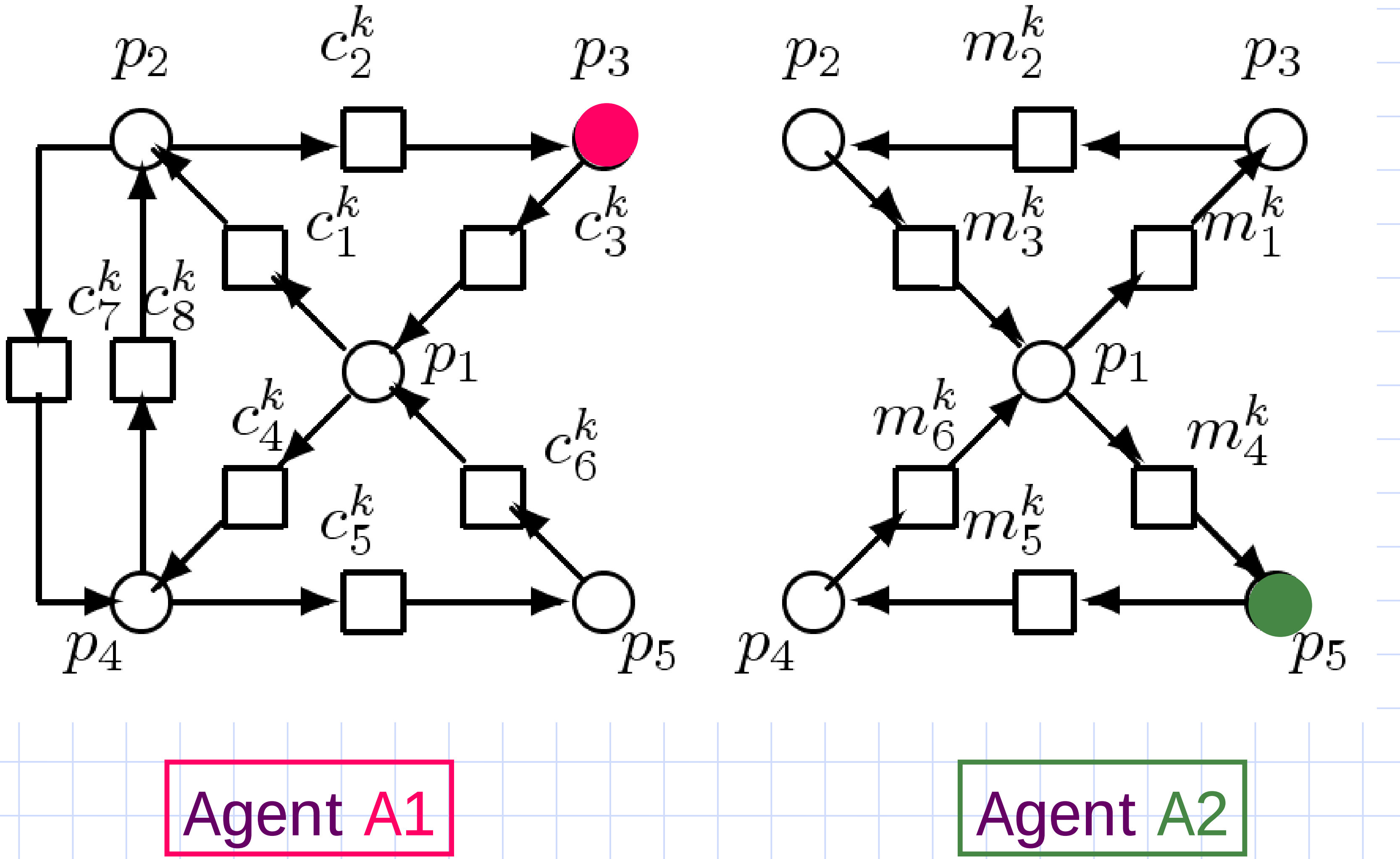
To find the feedback controller (a policy of the agent Ac) that will fulfil the conditions:

1. A1, A2 never occupy the same room simul-taneously

2. It is always possible for both of them to return to their initial position

3. The agent Ac should enable both of them to behave as freely as possible (with respect to (1.), (2.))

Petri net-based model of the agents A1, A2



Parameters of the mathematical models

Agent A1

Parameters of the PN-based model

$$\mathbf{F}_c = \begin{pmatrix} 1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0 \end{pmatrix};\ \mathbf{G}_c = \begin{pmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0 \end{pmatrix}$$

Adjacency matrix of RG and the space of reachable states

$$\mathbf{A}_{k_c} = \begin{pmatrix} 0 & c_3 & 0 & 0 & 0 \\ 0 & 0 & c_1 & c_4 & 0 \\ c_2 & 0 & 0 & c_7 & 0 \\ 0 & 0 & c_8 & 0 & c_5 \\ 0 & c_6 & 0 & 0 & 0 \end{pmatrix}; \ \mathbf{X}_{r_c} = \begin{pmatrix} 01000 \\ 00100 \\ 10000 \\ 00010 \\ 00001 \end{pmatrix}$$

Agent A2

Parameters of the PN-based model

$$\mathbf{F}_m = \begin{pmatrix} 1\,0\,0\,1\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0 \end{pmatrix} ; \; \mathbf{G}_m = \begin{pmatrix} 0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,1\,0 \\ 1\,0\,0\,0\,0 \end{pmatrix}$$

Adjacency matrix of RG and the space of reachable states

$$\mathbf{A}_{k_m} = \begin{pmatrix} 0 & m_5 & 0 & 0 & 0 \\ 0 & 0 & m_6 & 0 & 0 \\ m_4 & 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & 0 & m_2 \\ 0 & 0 & m_3 & 0 & 0 \end{pmatrix}; \quad \mathbf{X}_{r_m} = \begin{pmatrix} 00100 \\ 00001 \\ 00010 \\ 01000 \\ 10000 \end{pmatrix}$$

Approaches to solving the problem

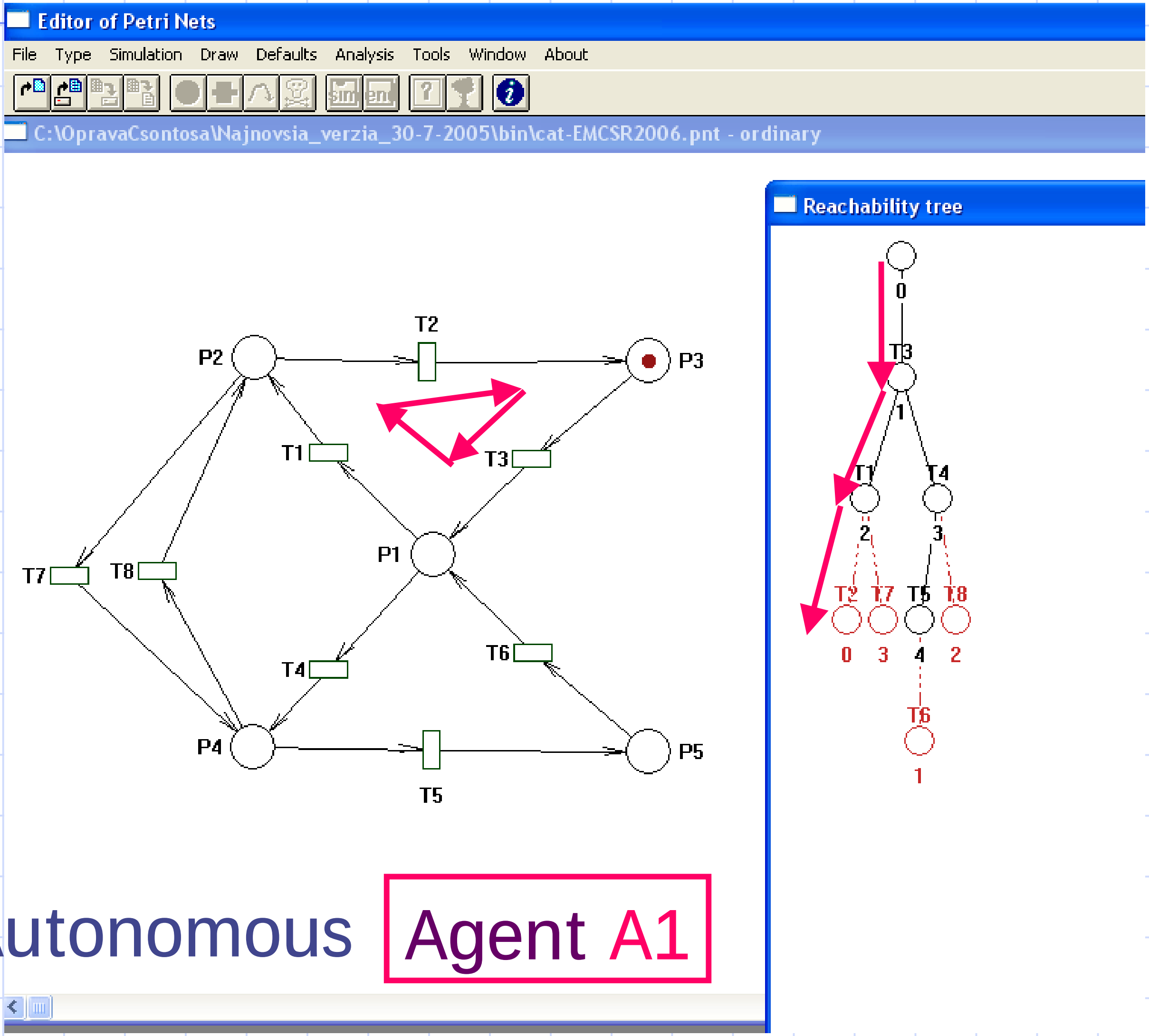
I. Mutual intersection of autonomous solutions

II. Solving the global problem in the whole

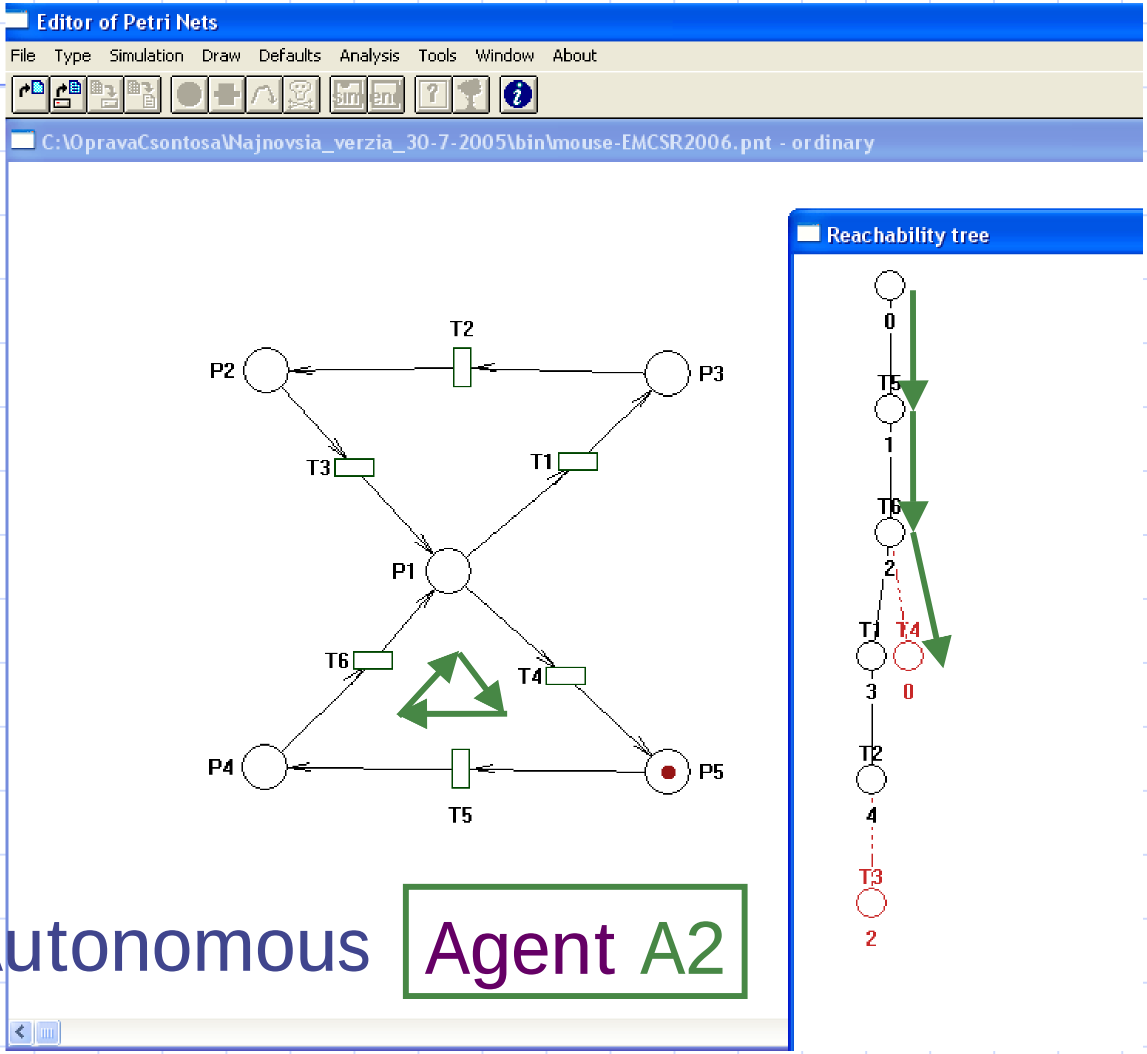
III. Utilizing the invariants of the Petri nets model

Illustrative example

I. Mutual intersection of autonomous solutions

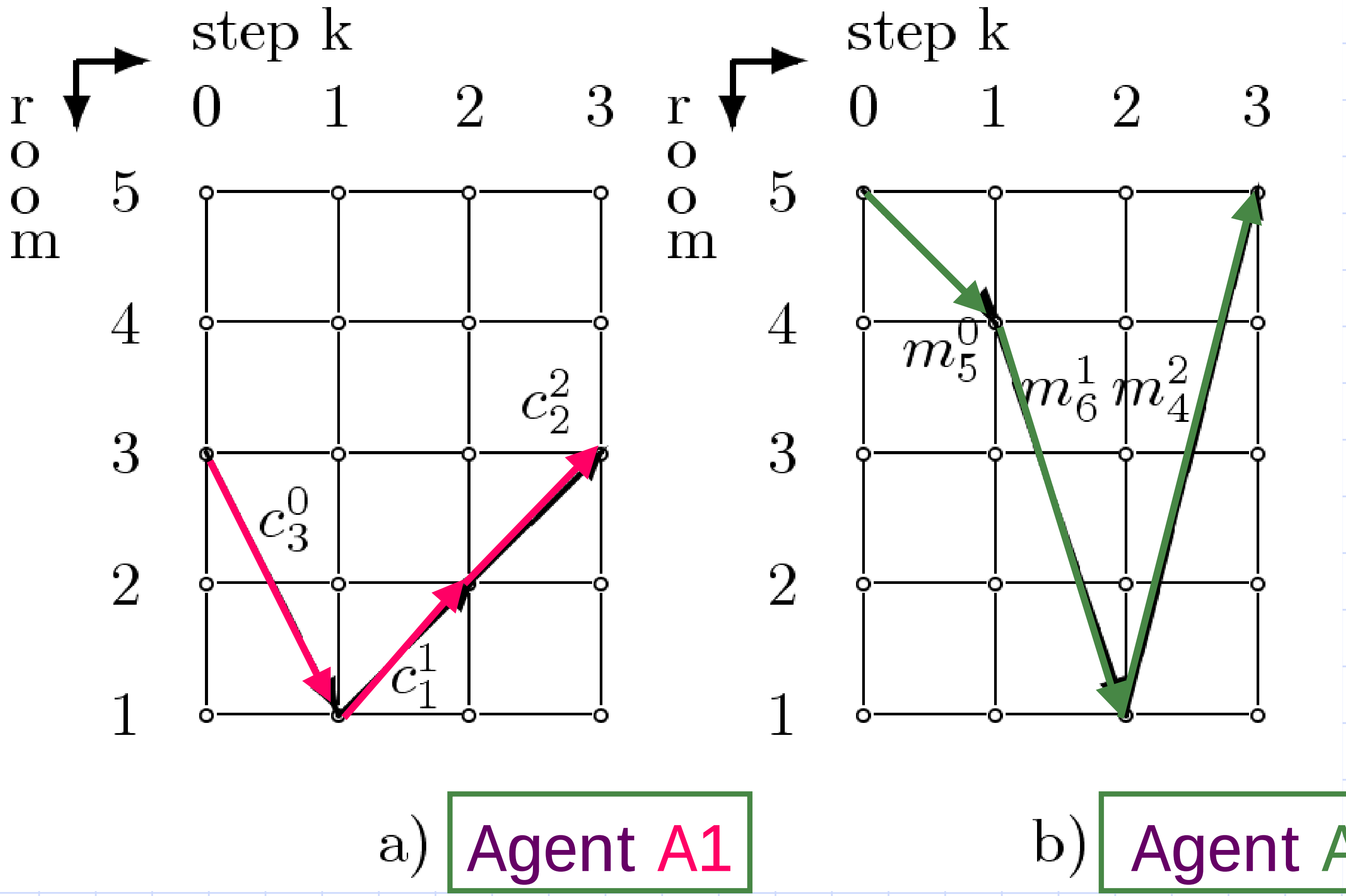


Autonomous Agent A1



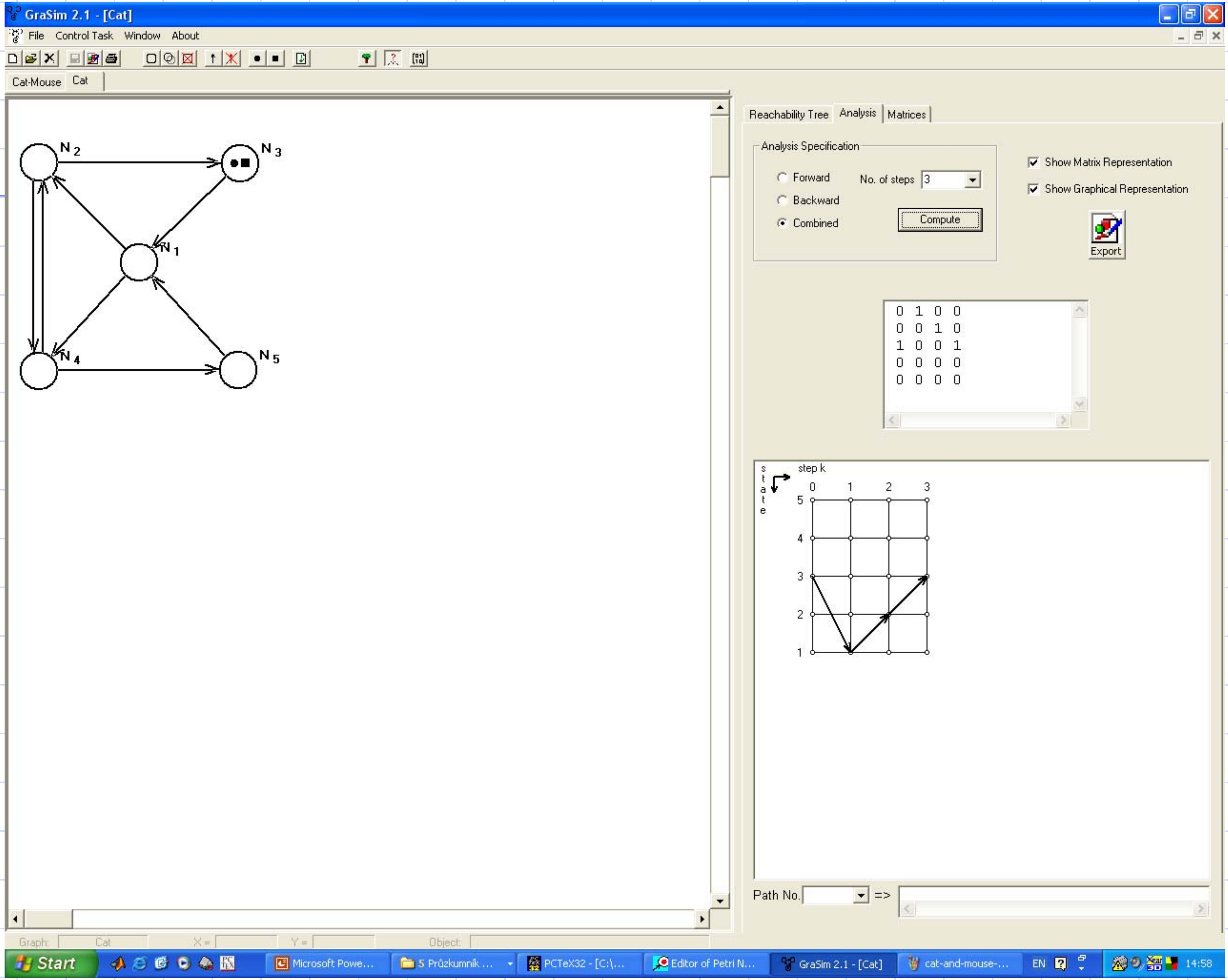
Autonomous Agent A2

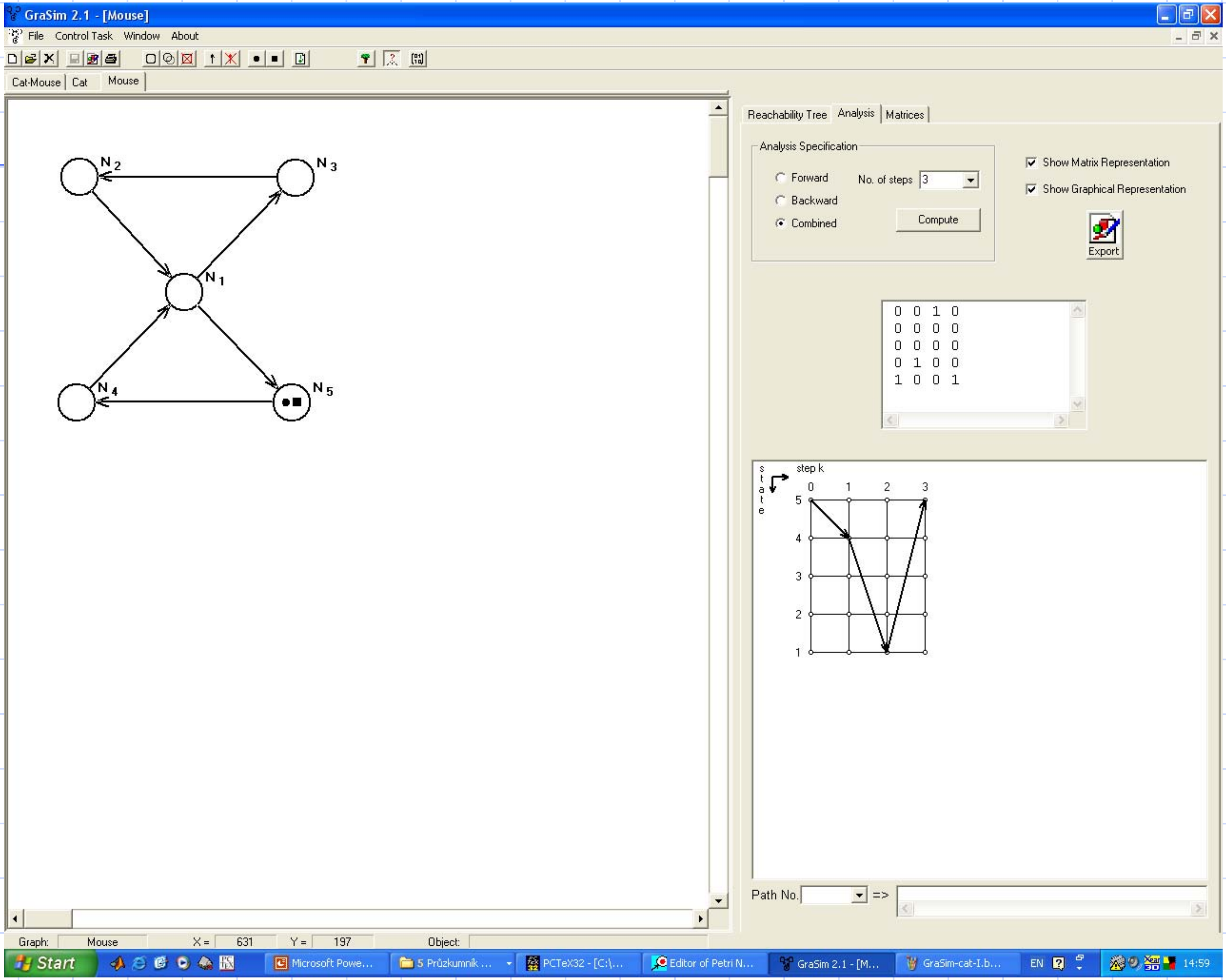
The final solution



Agent A1

Agent A2


GraSim 2.1 - [Cat]
File Control Task Window About
Cat-Mouse Cat
N 2
N 3
N 1
N 4
N 5
Reachability Tree Analysis Matrices
Analysis Specification
Forward
Backward
Combined
No. of steps 3
Compute
Show Matrix Representation
Show Graphical Representation
Export
0 1 0 0
0 0 1 0
1 0 0 1
0 0 0 0
0 0 0 0
state
step k
0 1 2 3
5
4
3
2
1
Path No. =>
Graph: Cat X = Y = Object:
Start Microsoft Powe... 5 Průzkumník ... PCTeX32 - [C:\... Editor of Petri N... GraSim 2.1 - [Cat] cat-and-mouse-... EN 14:58


GraSim 2.1 - [Mouse]
File Control Task Window About
Cat-Mouse Cat Mouse
N 1
N 2
N 3
N 4
N 5
Reachability Tree Analysis Matrices
Analysis Specification
Forward
Backward
Combined
No. of steps 3
Compute
Show Matrix Representation
Show Graphical Representation
Export
0 0 1 0
0 0 0 0
0 0 0 0
0 1 0 0
1 0 0 1
state
step k
0 1 2 3
5 4 3 2 1
Path No. =>
Graph: Mouse X = 631 Y = 197 Object:
Start
Microsoft Powe...
5 Průzkumník ...
PCTeX32 - [C:\...
Editor of Petri N...
GraSim 2.1 - [M...
GraSim-cat-I.b...
EN
14:59

II. Solving the global problem in the whole

The PN-based mathematical model

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_c & \emptyset \\ \emptyset & \mathbf{F}_m \end{pmatrix}, \quad \mathbf{G}^T = \begin{pmatrix} \mathbf{G}_c^T & \emptyset \\ \emptyset & \mathbf{G}_m^T \end{pmatrix}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_m \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_m \end{bmatrix}$$

The global solution

Agent A1 Agent A2

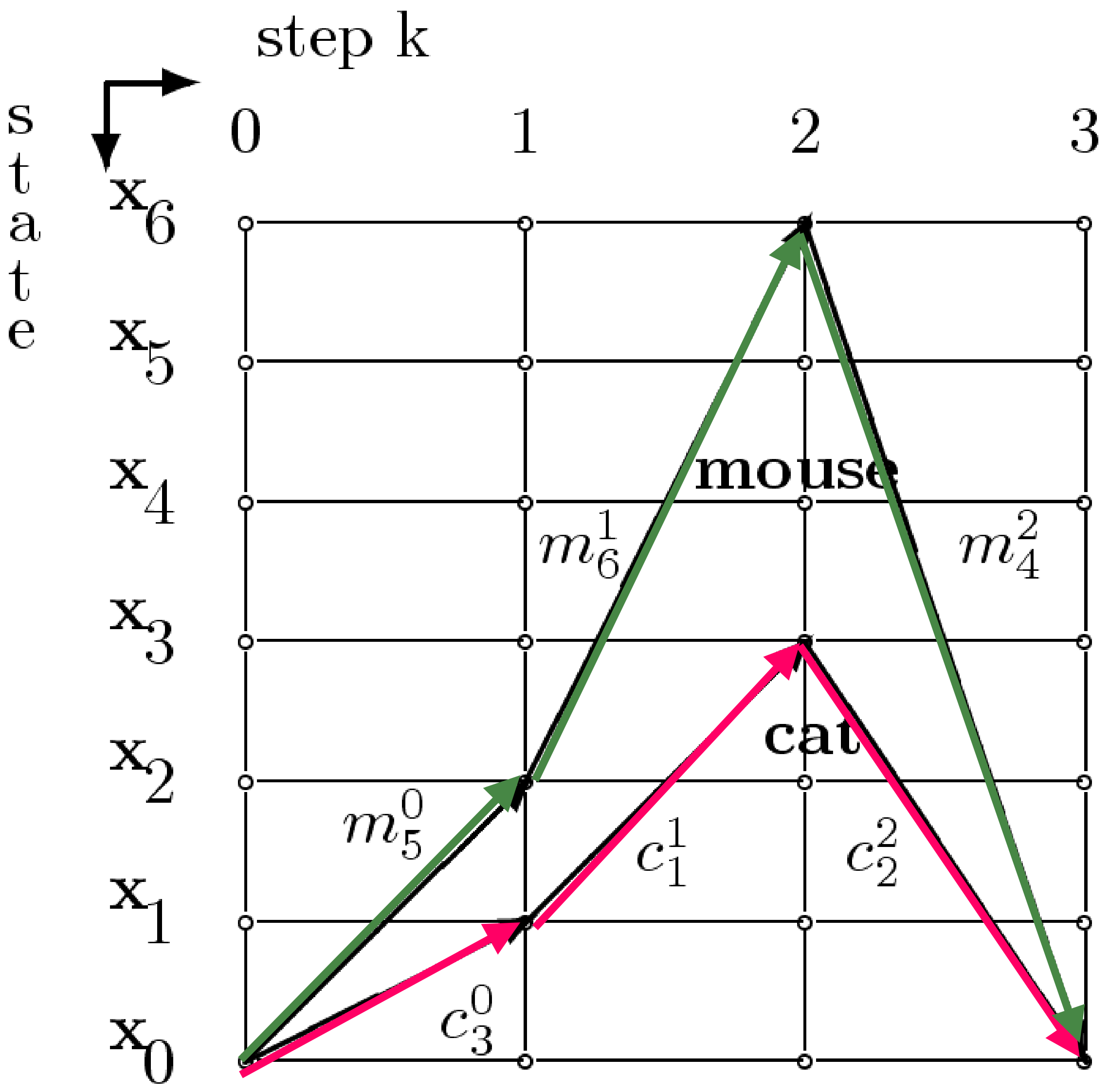
$$\mathbf{x}_0 = (\mathbf{x}_{0_c}^T, \mathbf{x}_{0_m}^T)^T = (\overbrace{0\,0\,1\,0\,0}\,|\,\overbrace{0\,0\,0\,0\,1})^T$$

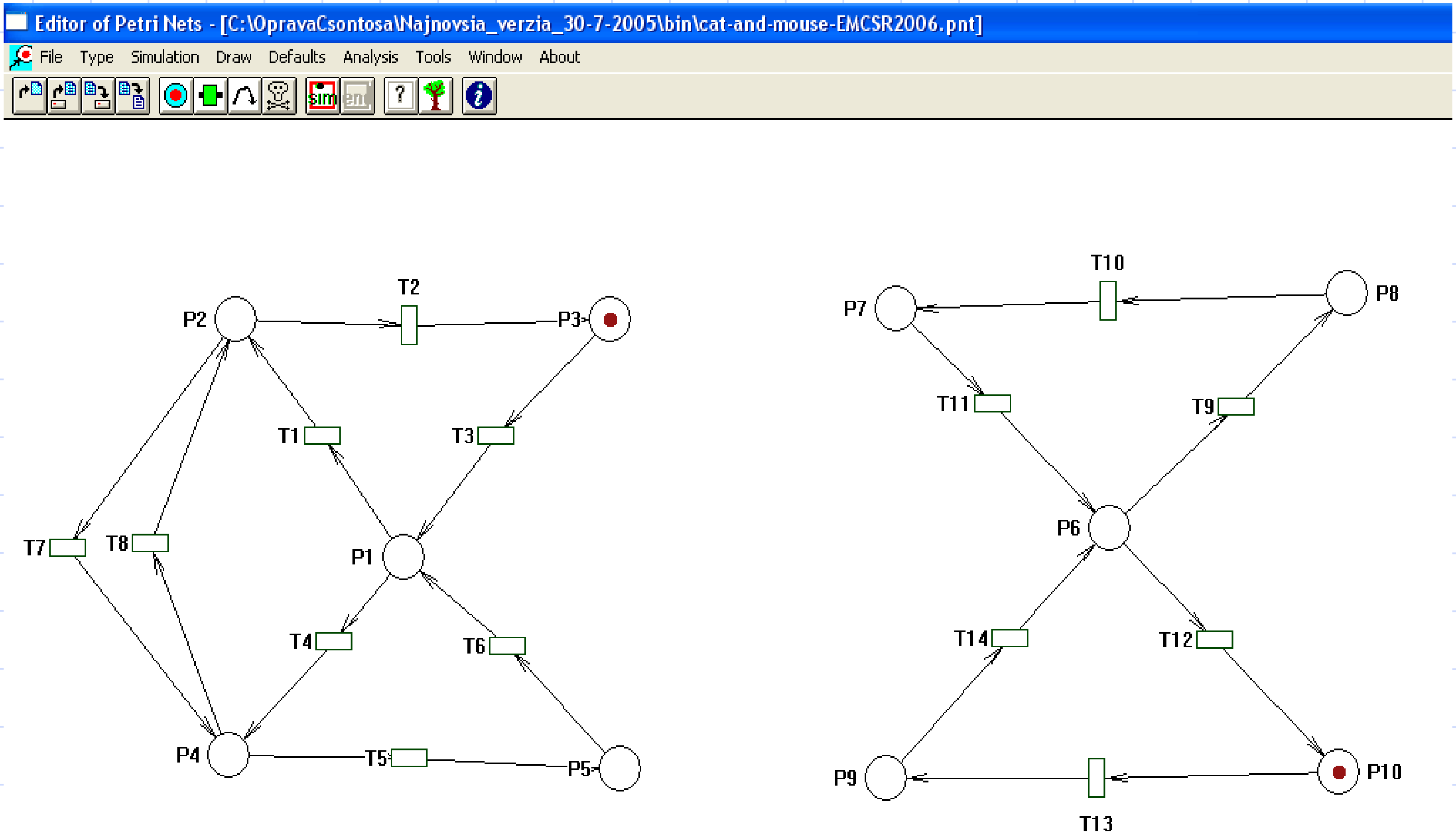
$$\mathbf{x}_1 = (\mathbf{x}_{1_c}^T, \mathbf{x}_{1_m}^T)^T = (1\,0\,0\,0\,0\,|\,0\,0\,0\,0\,1)^T$$

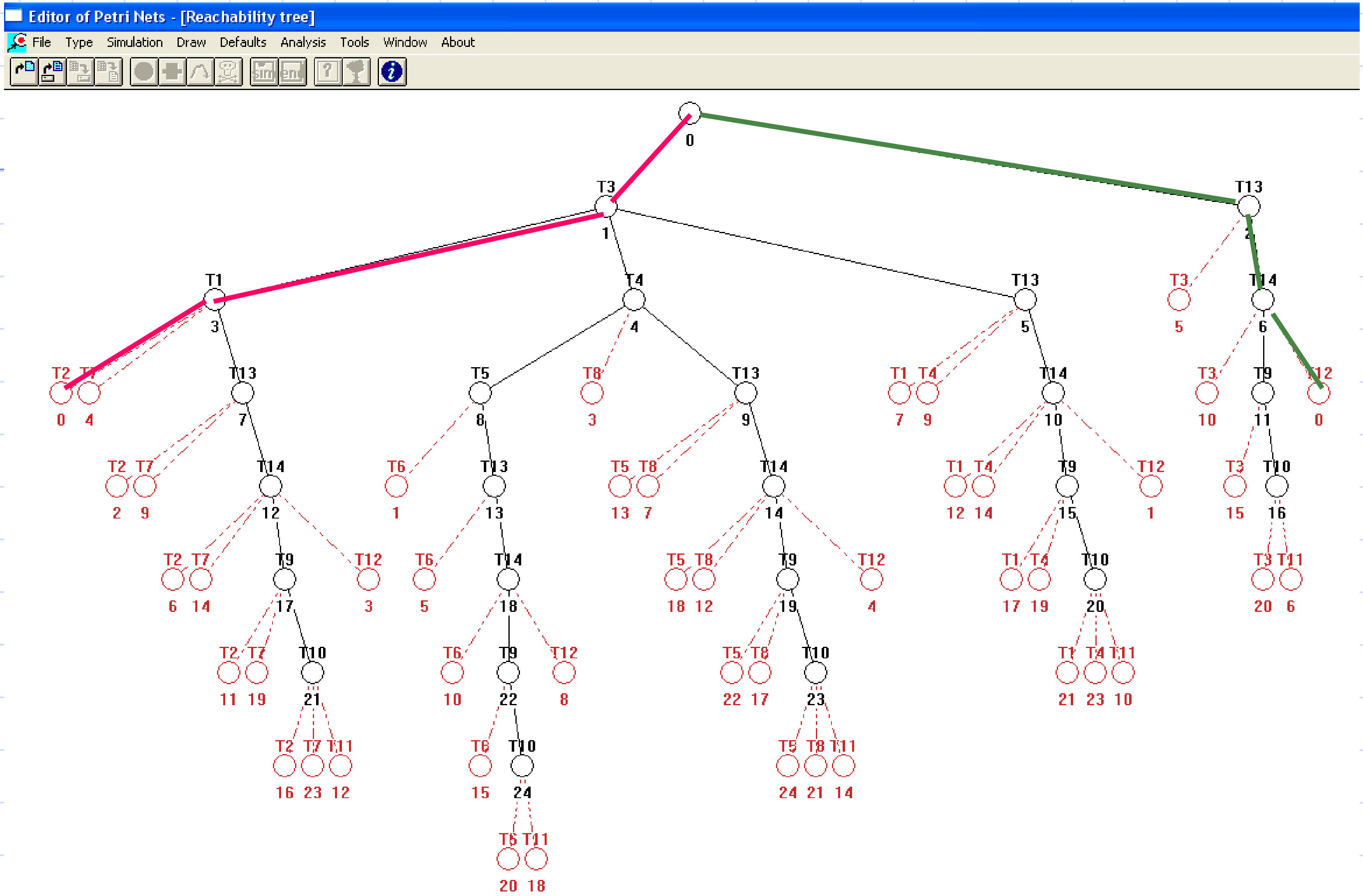
$$\mathbf{x}_2 = (\mathbf{x}_{2_c}^T, \mathbf{x}_{2_m}^T)^T = (0\,0\,1\,0\,0\,|\,0\,0\,0\,1\,0)^T$$

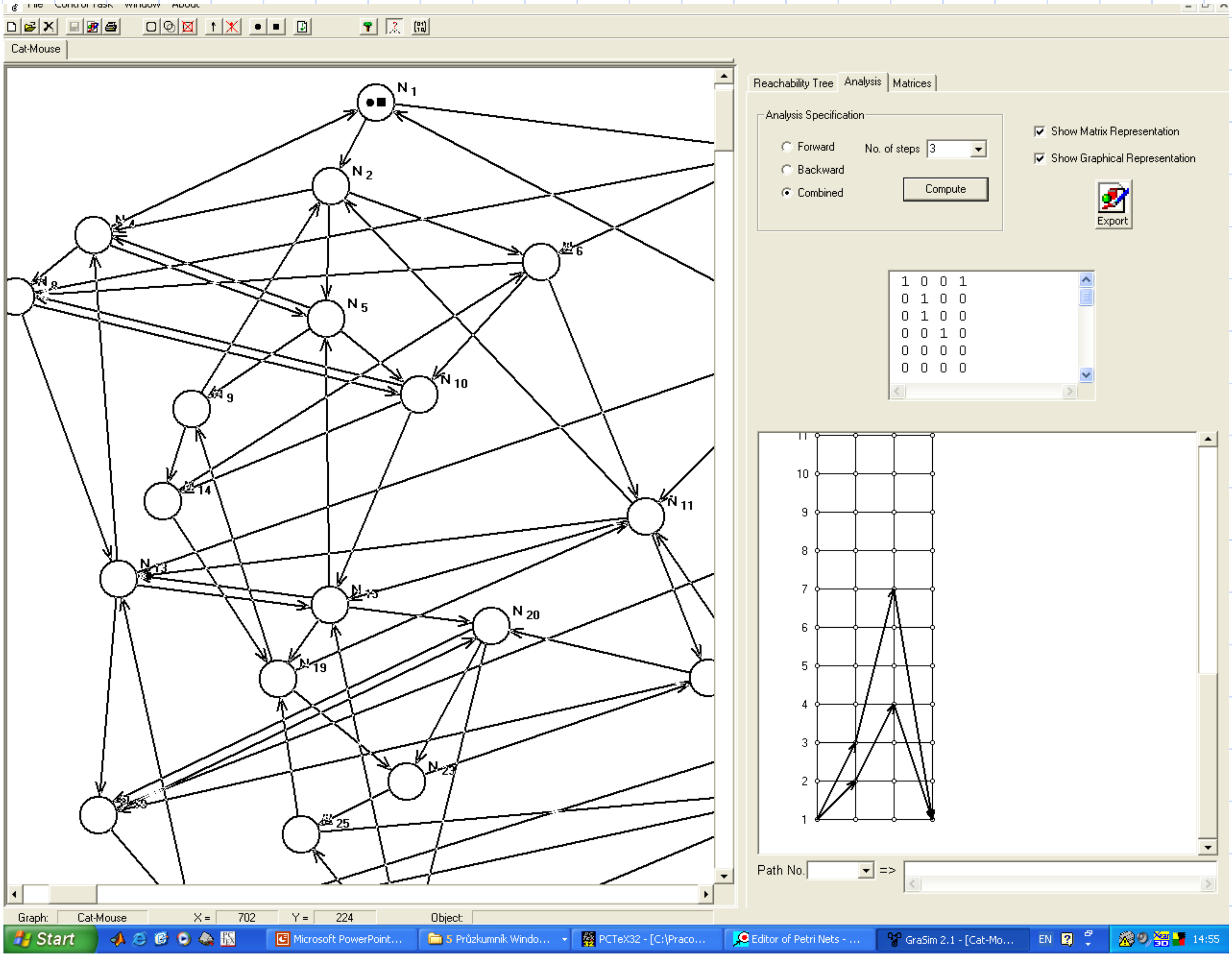
$$\mathbf{x}_3 = (\mathbf{x}_{3_c}^T, \mathbf{x}_{3_m}^T)^T = (0\,1\,0\,0\,0\,|\,0\,0\,0\,0\,1)^T$$

$$\mathbf{x}_6 = (\mathbf{x}_{6_c}^T, \mathbf{x}_{6_m}^T)^T = (0\,0\,1\,0\,0\,|\,1\,0\,0\,0\,0)^T$$


step k
state
0
1
2
3
x6
x5
x4
x3
x2
x1
x0
mouse
cat
m_6^1
m_4^2
m_5^0
c_1^1
c_2^2
c_3^0


Editor of Petri Nets - [C:\OpravaCsontosa\Najnovsia_verzia_30-7-2005\bin\cat-and-mouse-EMCSR2006.pnt]
File Type Simulation Draw Defaults Analysis Tools Window About
T2
P2
P3
T1
T3
T7
T8
P1
T4
T6
P4
T5
P5
T10
P7
P8
T11
T9
P6
T14
T12
P9
P10
T13


Editor of Petri Nets - [Reachability tree]
File Type Simulation Draw Defaults Analysis Tools Window About


Cat-Mouse
Reachability Tree
Analysis
Matrices
Analysis Specification
Forward
Backward
Combined
No. of steps
3
Compute
Show Matrix Representation
Show Graphical Representation
Export
1 0 0 1
0 1 0 0
0 1 0 0
0 0 1 0
0 0 0 0
0 0 0 0
Path No.
=>
Graph:
Cat-Mouse
X =
702
Y =
224
Object:
Start
Microsoft PowerPoint...
5 Průzkumník Windo...
PCTeX32 - [C:\Praco...
Editor of Petri Nets - ...
GraSim 2.1 - [Cat-Mo...
EN
14:55

III. Utilizing the invariants of the Petri nets model

Additional PN places – the so called slacks – are introduced

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix}; \quad \mathbf{F}_a = \begin{pmatrix} \mathbf{F} \\ \mathbf{F}_s \end{pmatrix}; \quad \mathbf{G}_a^T = \begin{pmatrix} \mathbf{G}^T \\ \mathbf{G}_s^T \end{pmatrix}$$

i.e. in our example

$$\sigma_{p_i} + \sigma_{p_{i+5}} \leq 1 \qquad (4)$$

$$\sigma_{p_i} + \sigma_{p_{i+5}} + \sigma_{s_i} = 1; \quad i = 1, 2, \ ..., \ 5 \qquad (5)$$

Principle of the invariants method

$$\mathbf{X}^T.\mathbf{B} = \mathbf{0} \quad \text{... in the general Petri net}$$

$$[\mathbf{L}, \mathbf{I}_s].\begin{pmatrix} \mathbf{B} \\ \mathbf{B}_s \end{pmatrix} = \mathbf{0} \quad \text{... in case of a Petri net with slacks}$$

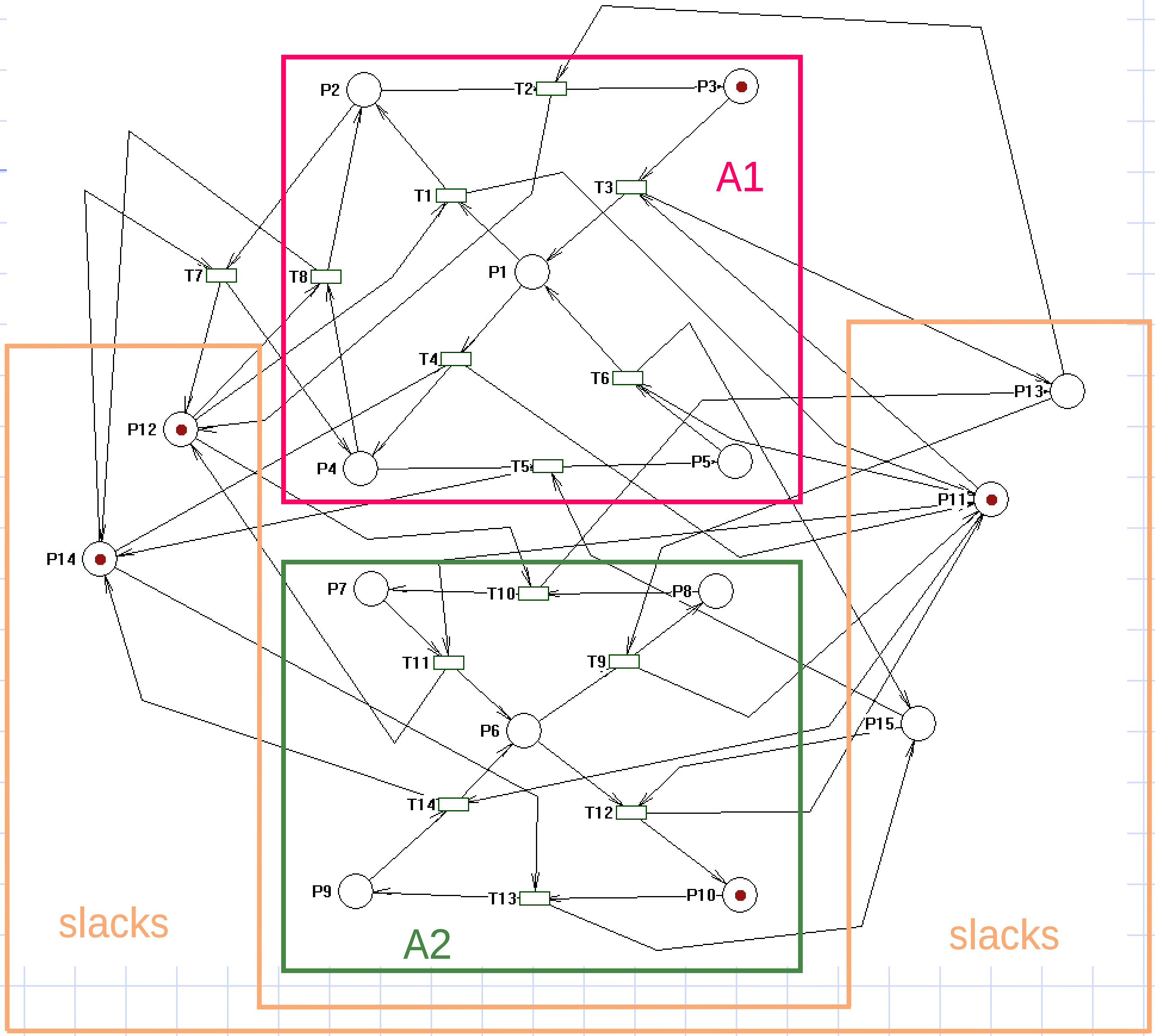
$$\mathbf{L}.\mathbf{B} + \mathbf{B}_s = \mathbf{0}$$

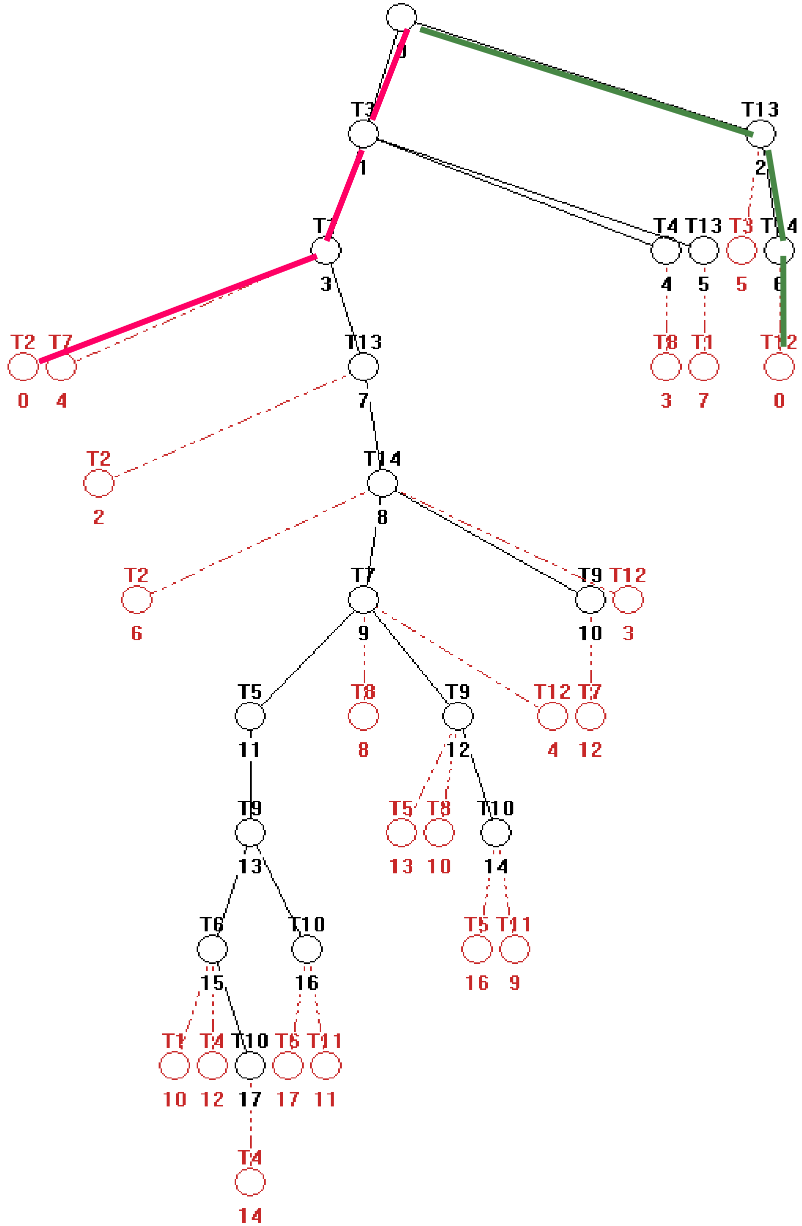
$$\mathbf{B}_s = -\mathbf{L}.\mathbf{B}$$

$$\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s$$

$$\mathbf{L} = [\mathbf{I}_5 \,|\, \mathbf{I}_5] = \begin{pmatrix} 10000|10000 \\ 01000|01000 \\ 00100|00100 \\ 00010|00010 \\ 00001|00001 \end{pmatrix}$$

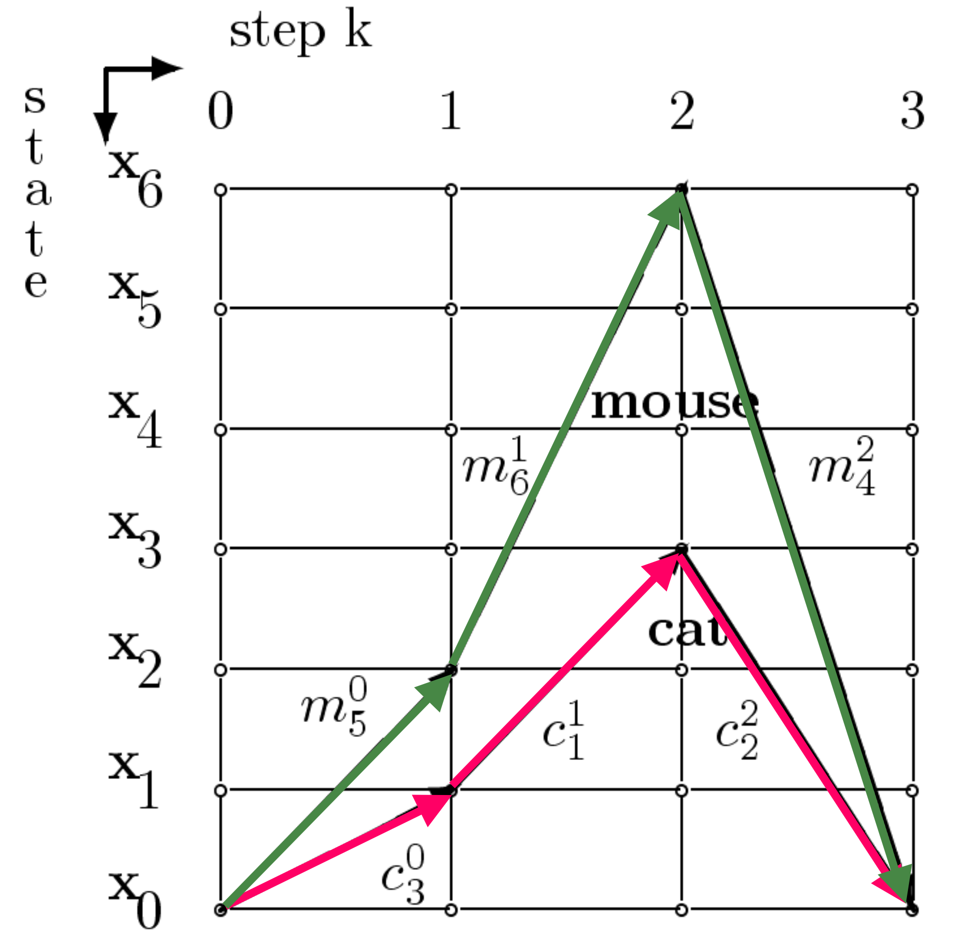
$$\mathbf{F}_s^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} ; \ \mathbf{G}_s = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

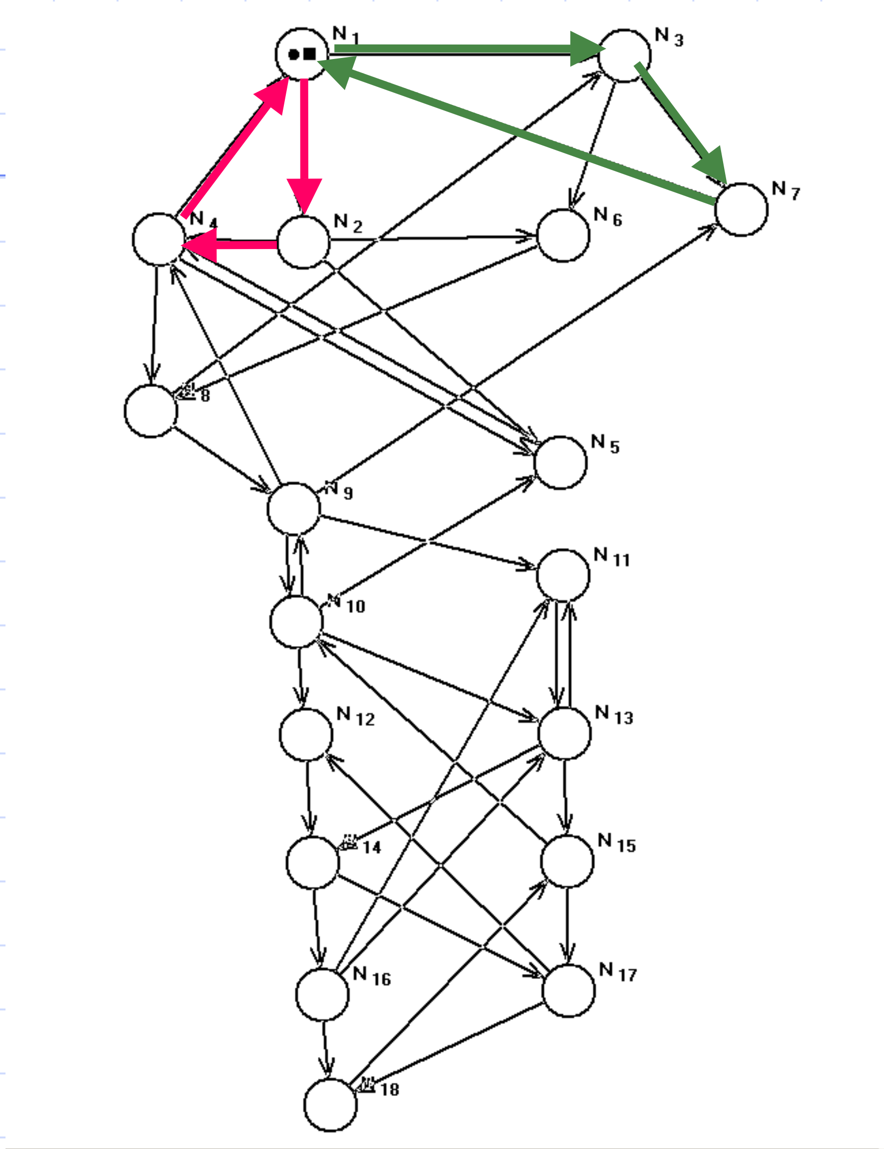

A1
A2
slacks
slacks
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14

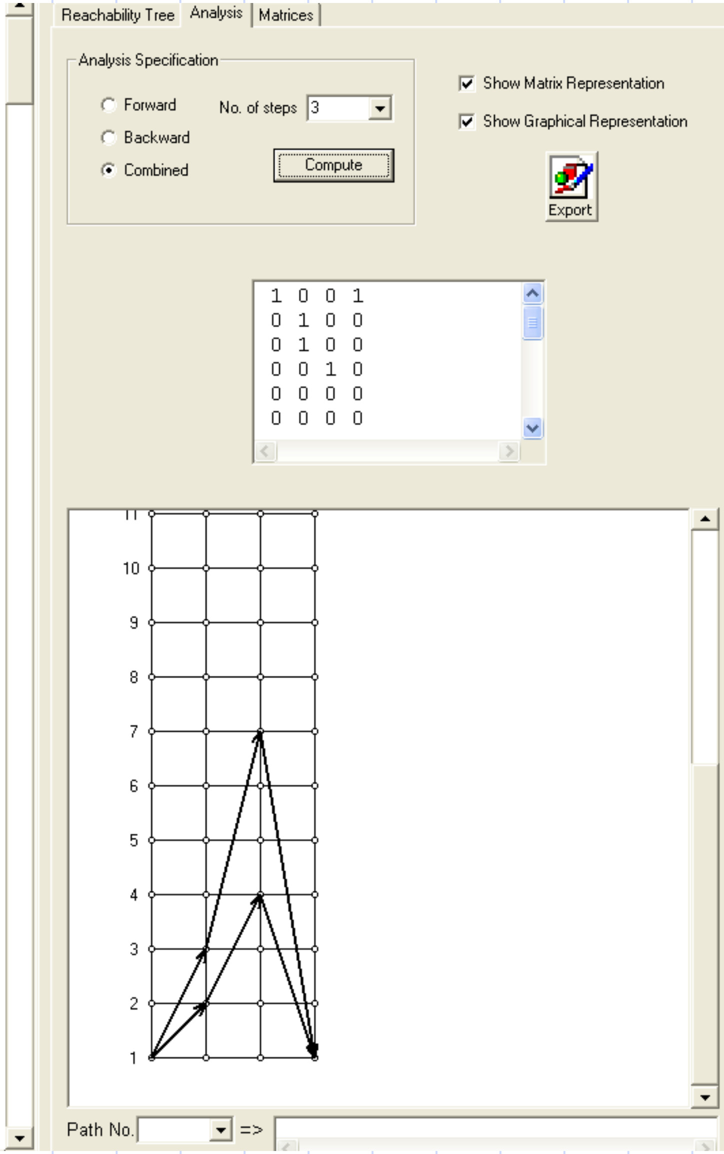

0
T3 T13
1 2
T1 T4 T13 T3 T14
3 4 5 5 6
T2 T7 T13 T8 T1 T12
0 4 7 3 7 0
T2 T14
2 8
T2 T7 T9 T12
6 9 10 3
T5 T8 T9 T12 T7
11 8 12 4 12
T9 T5 T8 T10
13 13 10 14
T6 T10 T5 T11
15 16 16 9
T1 T4 T10 T6 T11
10 12 17 17 11
T4
14

The final solution

- It is the same as in case II, i.e.:




N 1
N 3
N 7
N 4
N 2
N 6
8
N 5
N 9
N 11
N 10
N 12
N 13
14
N 15
N 16
N 17
18


Reachability Tree
Analysis
Matrices
Analysis Specification
Forward
Backward
Combined
No. of steps
3
Compute
Show Matrix Representation
Show Graphical Representation
Export
1 0 0 1
0 1 0 0
0 1 0 0
0 0 1 0
0 0 0 0
0 0 0 0
10
9
8
7
6
5
4
3
2
1
Path No.
=>

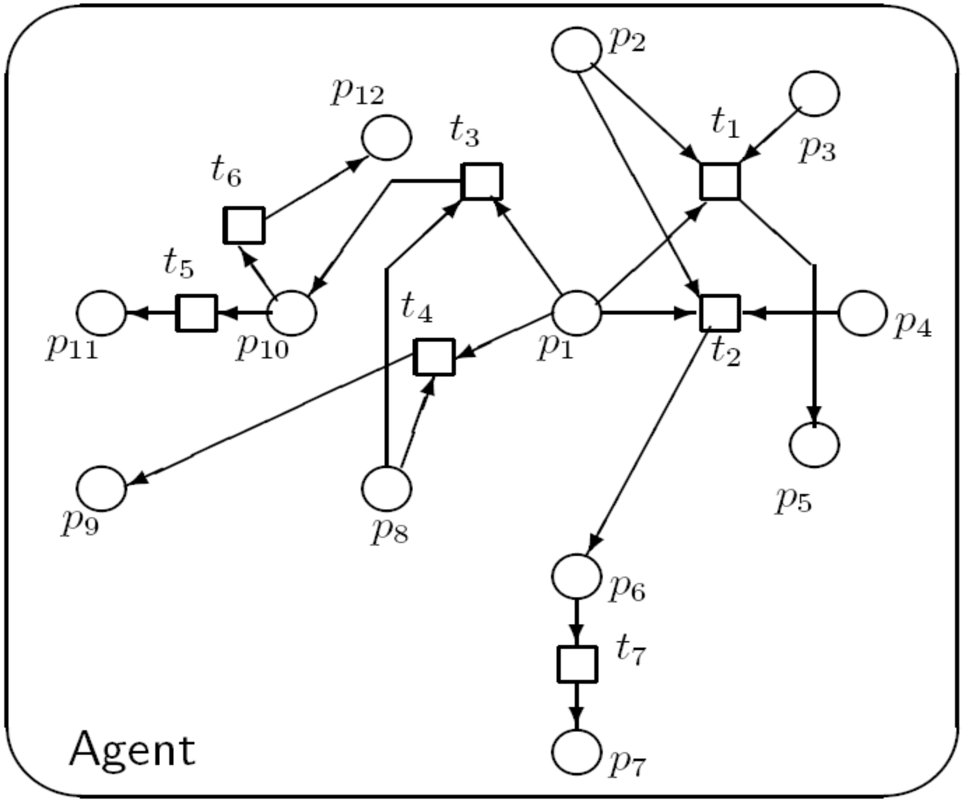
Comparison of the approaches

(I) The approach based on the intersection of the autonomous solutions is simple but multiplicative – i.e. the autonomous solution has to be found for any agent. In addition to this the process of the intersection has to be performed

(II) The global approach solves the problem simultaneously, together with the interconnections among agents. It saves the computational time, but the problem dimensionality is greater

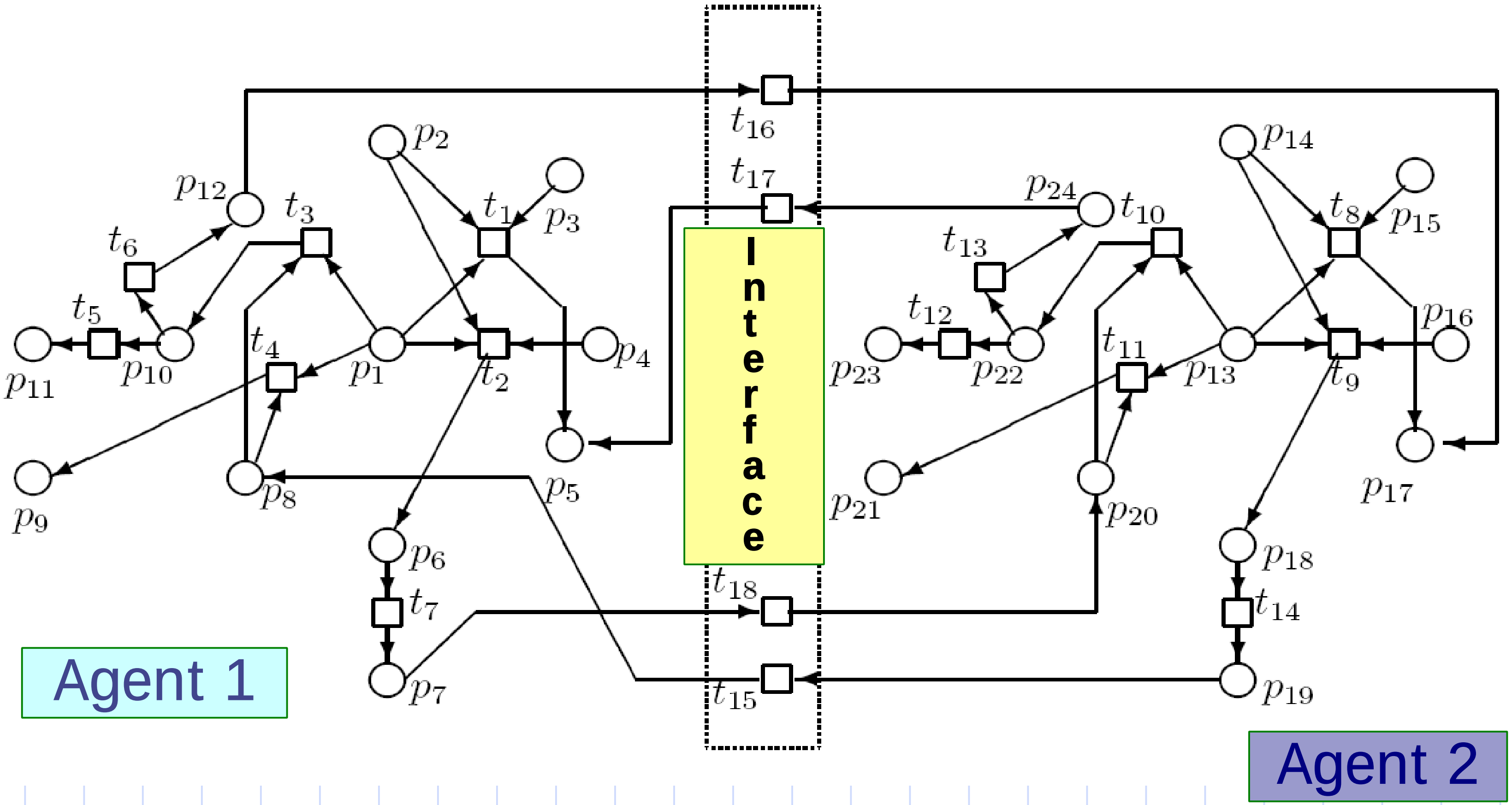
(III) The approach based on invariants decreases (in comparison with (II)) the RG, in spite of the fact that it increases the number of the PN places (it adds the slacks)

Modular approach to agents modelling



Interpretation of PN places

p1 – A is free
p2 – A has to solve a problem P
p3 – A is able to solve P
p4 – A is not able to solve P
p5 – P is solved
p6 – A contacts another agent
p7 – A asks another agent for help
p8 – A is asked by another agent for help to solve its problem
p9 – A refuses the help
p10 – A accepts the request for help
p11 – A is not able to help
p12 – A is able to help


Interface
Agent 1
Agent 2
t16
t17
t18
t15
p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12
t1 t2 t3 t4 t5 t6 t7
p13 p14 p15 p16 p17 p18 p19 p20 p21 p22 p23 p24
t8 t9 t10 t11 t12 t13 t14

Incidence matrix F of the PN model of MAS

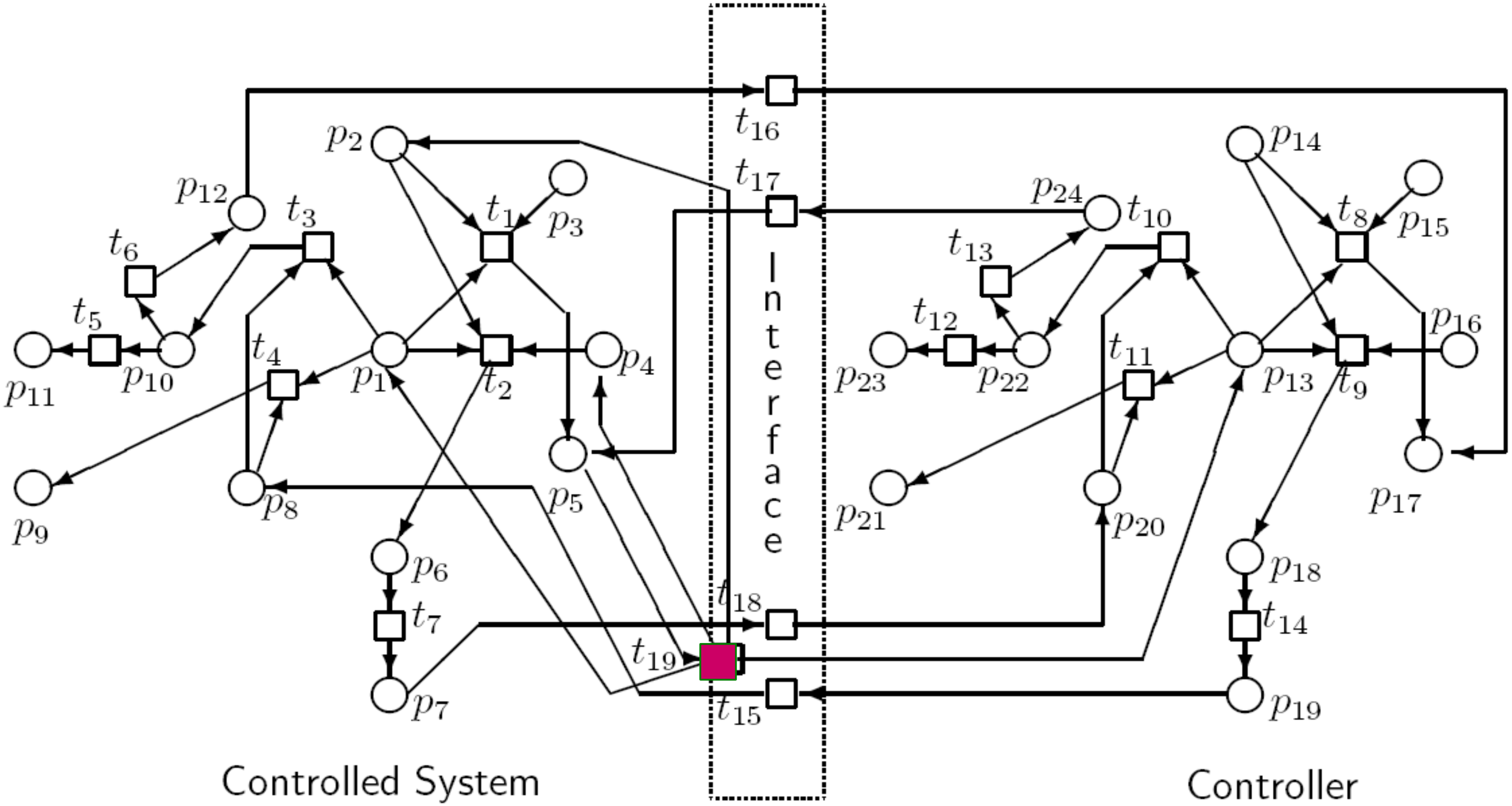
$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & | \; \mathbf{F}_{c_1} \\ \mathbf{0} & \mathbf{F}_2 & \dots & \mathbf{0} & \mathbf{0} & | \; \mathbf{F}_{c_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & | \; \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F}_{N_A-1} & \mathbf{0} & | \; \mathbf{F}_{c_{N_A-1}} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{F}_{N_A} & | \; \mathbf{F}_{c_{N_A}} \end{pmatrix}$$

Incidence matrix G of the PN model of MAS

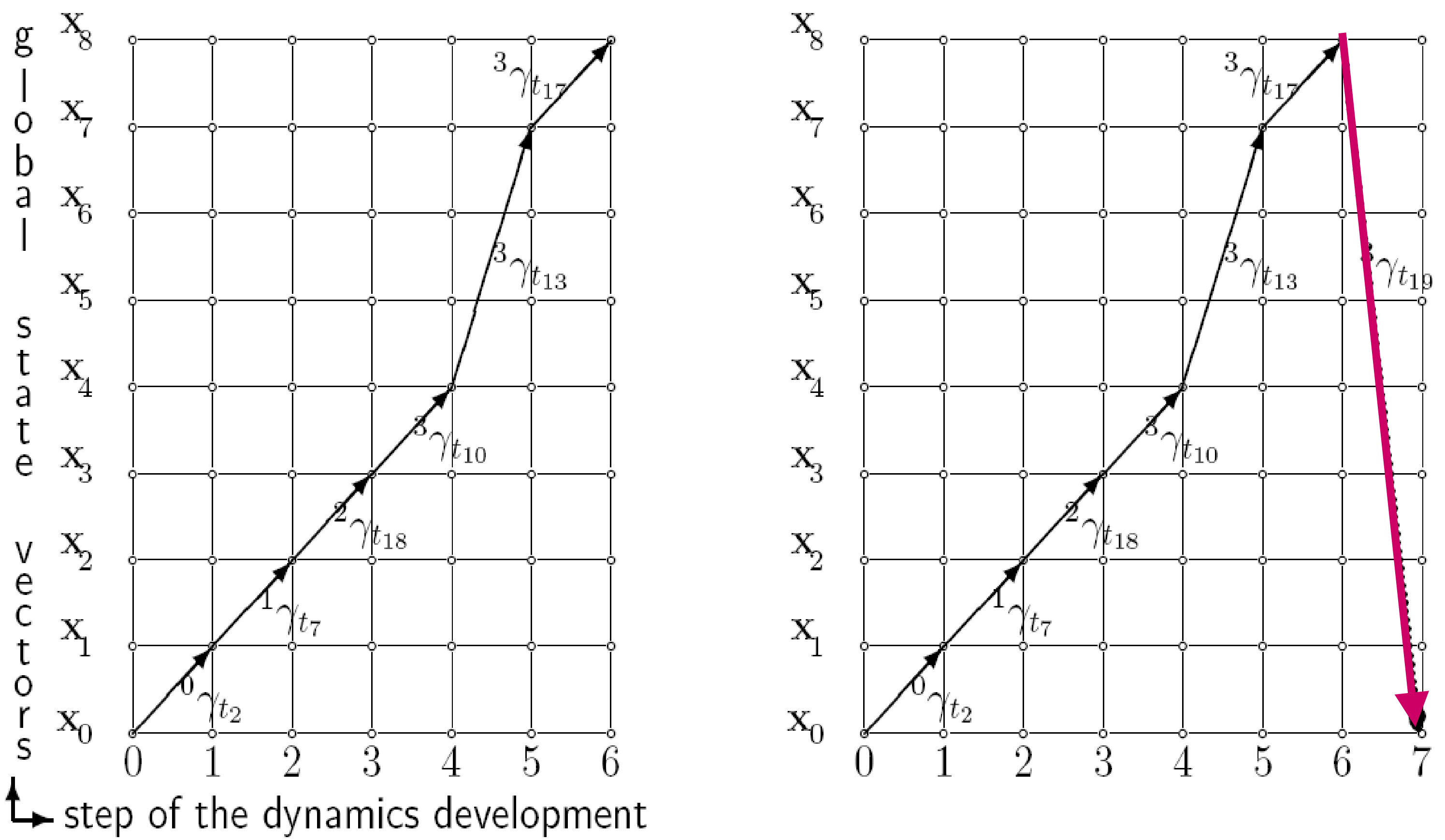
$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & 0 & \dots & 0 & 0 \\ 0 & \mathbf{G}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{G}_{N_A-1} & 0 \\ 0 & 0 & \dots & 0 & \mathbf{G}_{N_A} \\ ---- & ---- & ---- & -------- & ---- \\ \mathbf{G}_{c_1} & \mathbf{G}_{c_2} & \dots & \mathbf{G}_{c_{N_A-1}} & \mathbf{G}_{c_{N_A}} \end{pmatrix}$$

The feedback control

A1 is the **controlled system** , **A2** is the **controller**



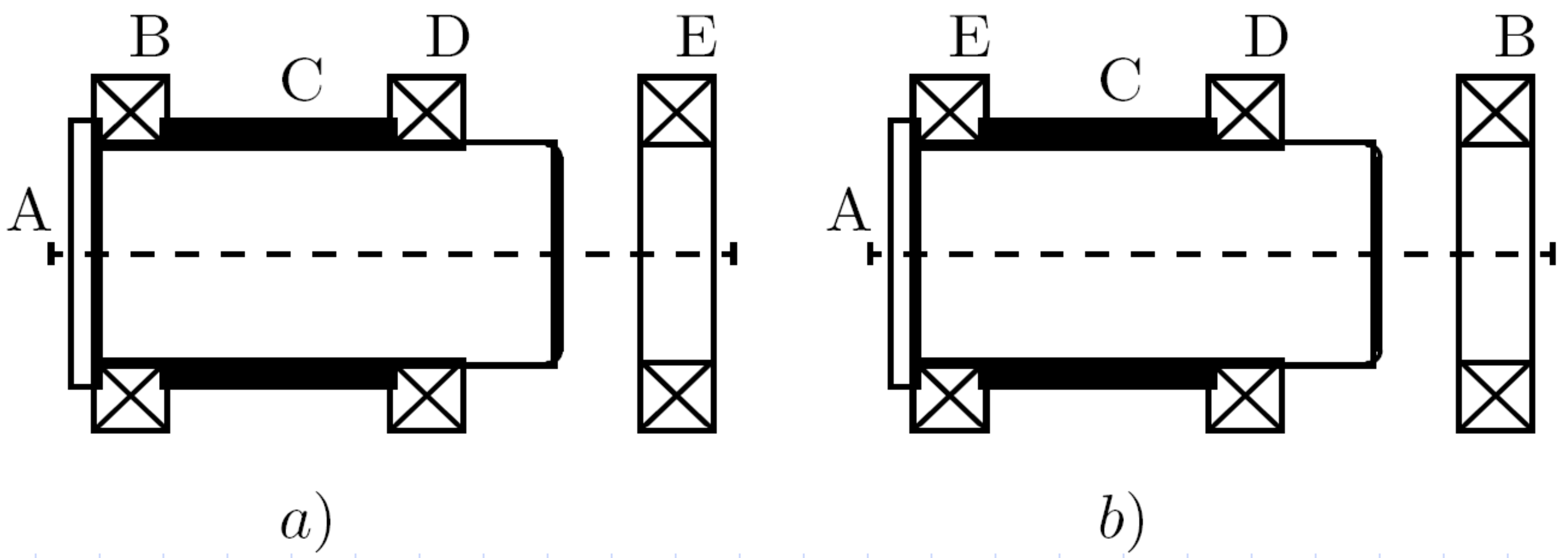
The feedback



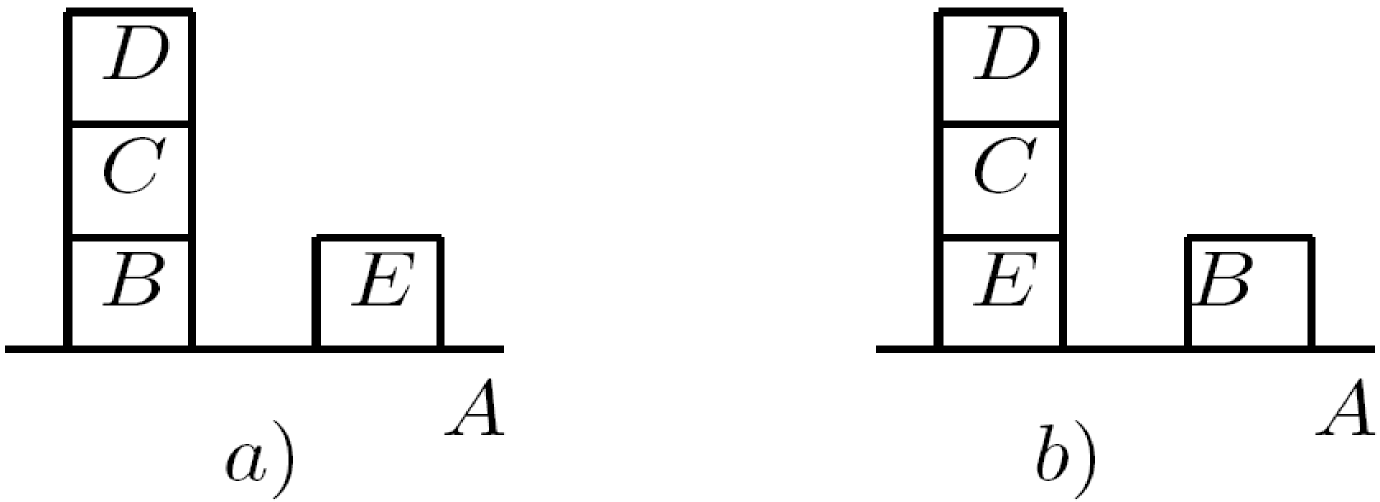
Engineering applications of AI

Assembly & disassembly problem solving:

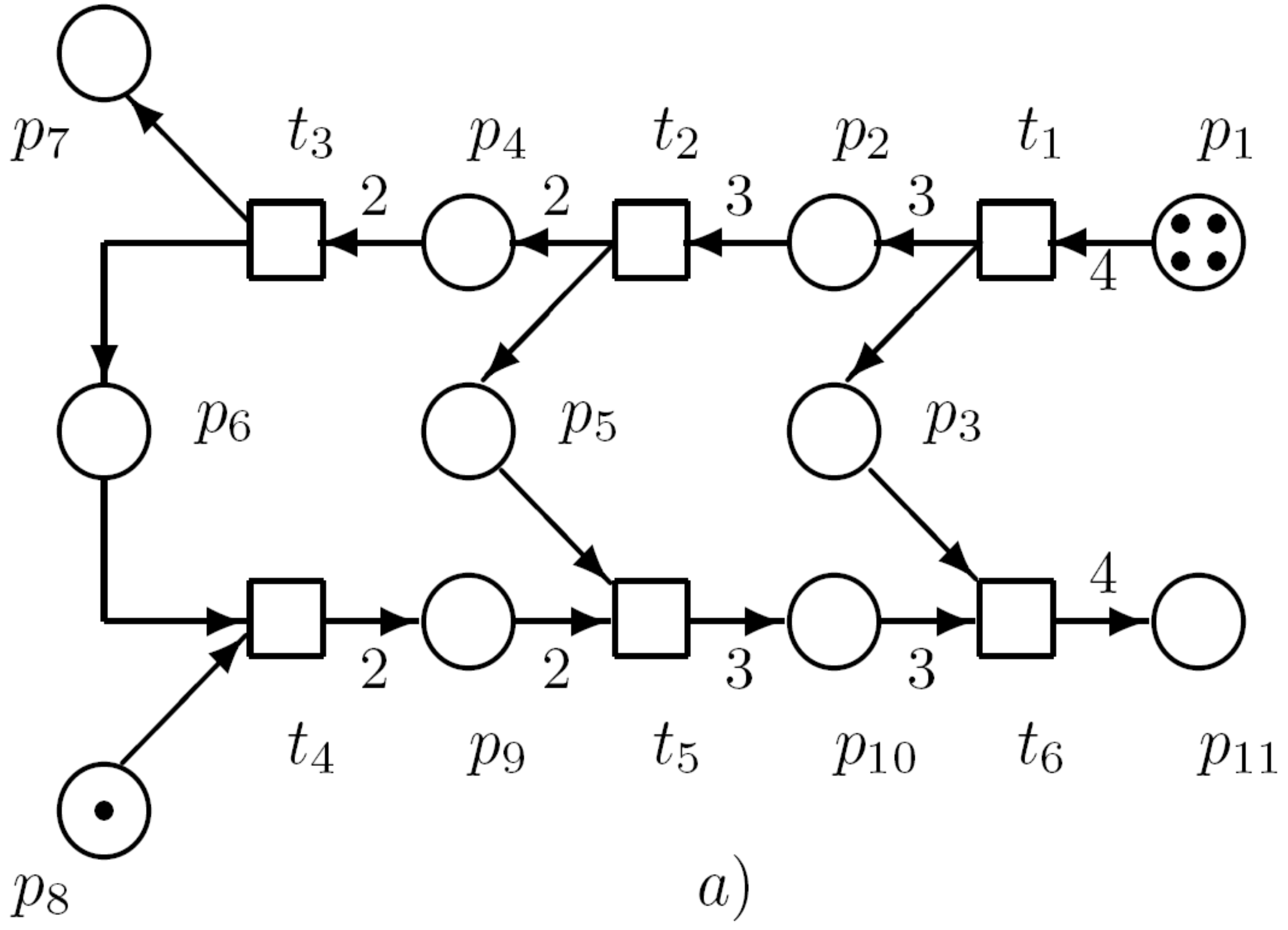
Exchange of the bad bearing B



Utilizing the block world paradigm



Petri net-based model



The interpretation of PN places

p1 – the initial configuration

p2 – D is disassembled

p3 – D is put aside

p4 – C is disassembled

p5 – C is put aside

p6 – A is free of parts

p7 – B is disassembled and put aside

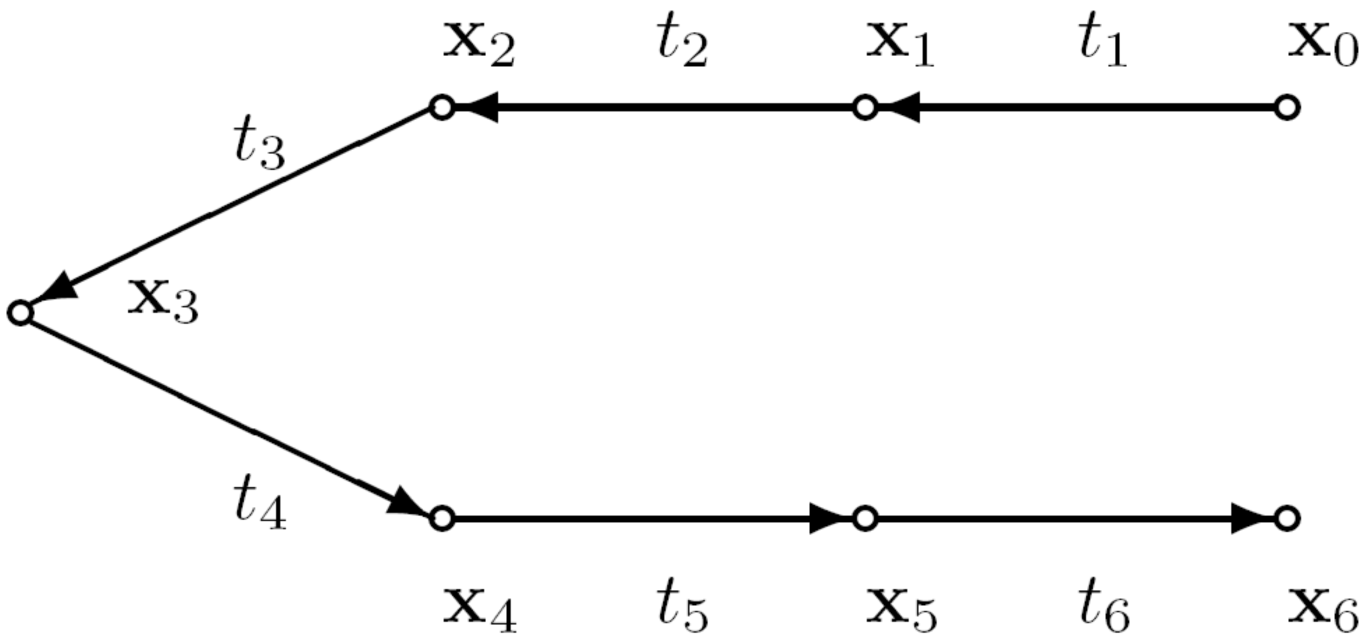
p8 – E is prepared for using

p9 – E is put on A

p10 – C is put on E

p11 – D is put on C

The reachability graph

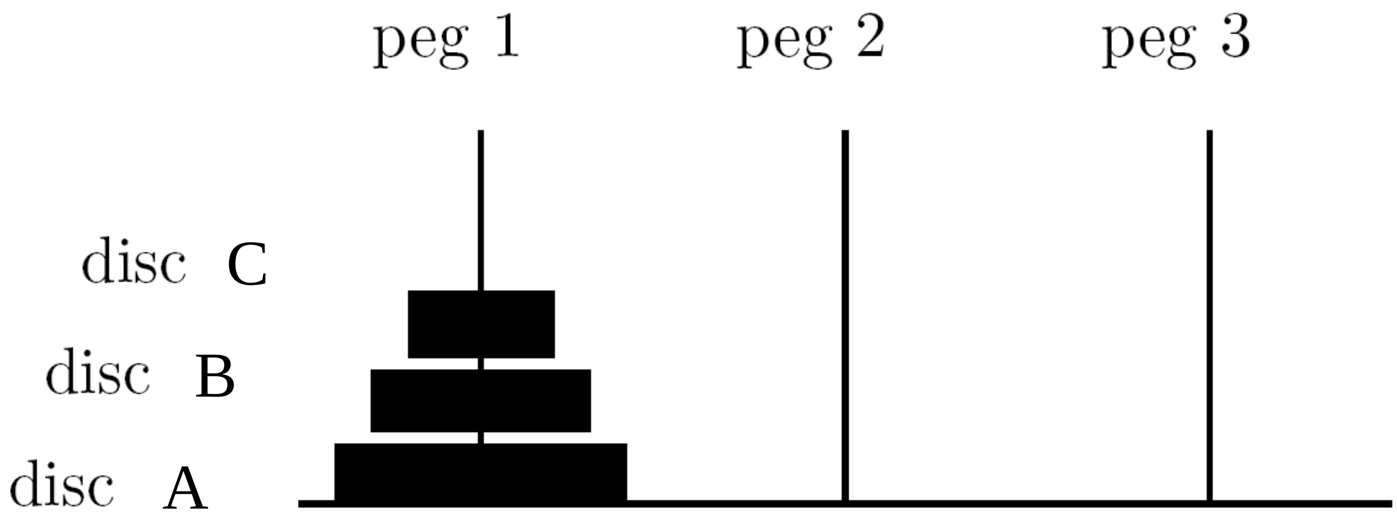


The state space

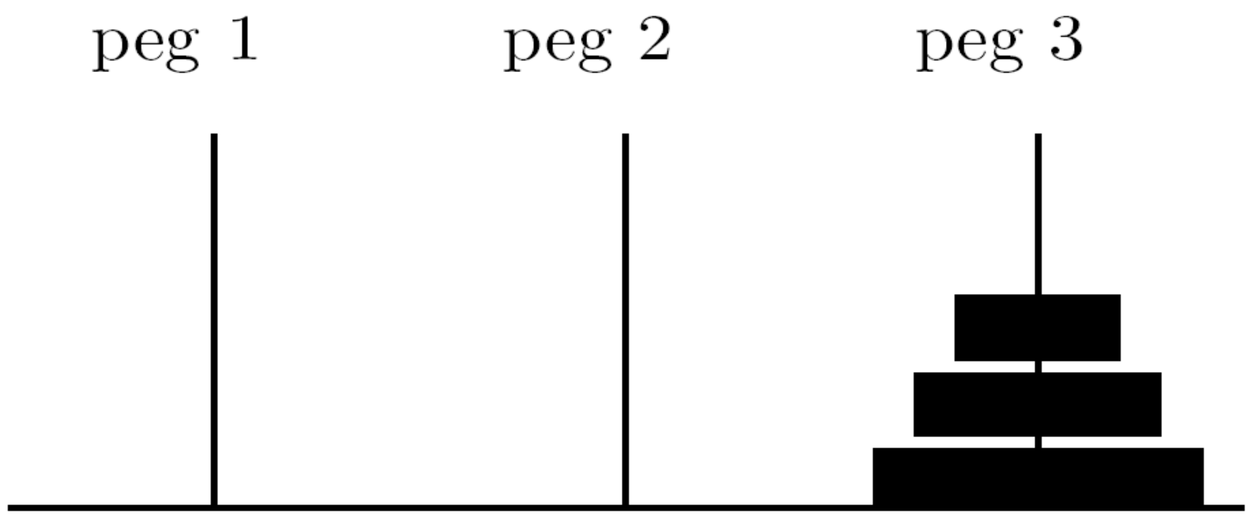
$$\mathbf{X}_{reach}^{blocks} = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}$$

Utilizing the Hanoi tower paradigm

The initial state of the Hanoi tower puzzle



The terminal state of the Hanoi tower puzzle



Usage of this paradigm

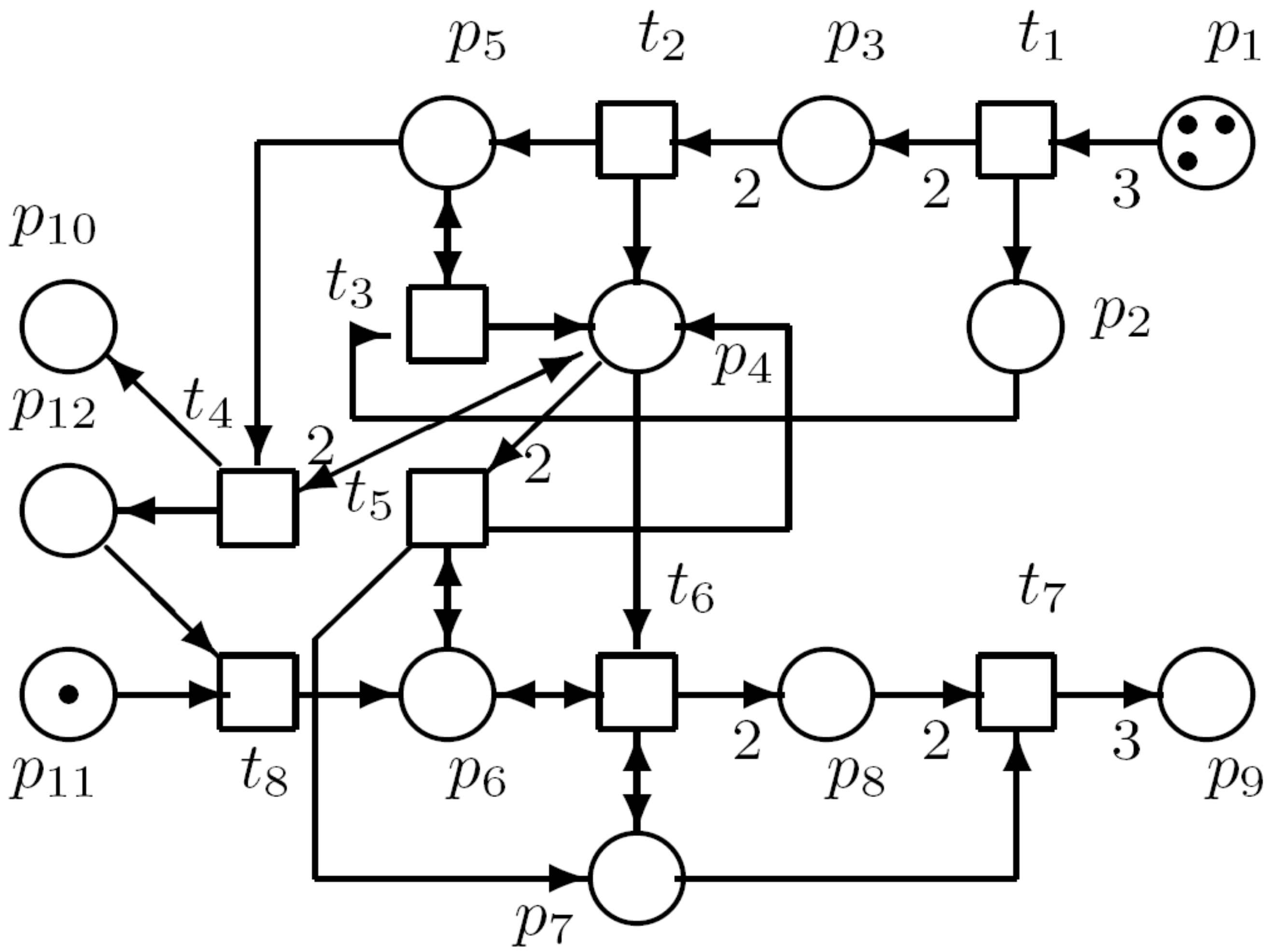
This paradigm can be utilized in assembly when

- the space for putting parts is limited
- the assembled parts have very different mass
- the parts are fragile, etc.

Comparing the puzzle and the assembly process

Step	Original Hanoi Tower			Step	Assembly Process		
1	peg 1	$\xrightarrow{disc\ C}$	peg 3	1	peg 1	$\xrightarrow{bearing\ D}$	peg 3
2	peg 1	$\xrightarrow{disc\ B}$	peg 2	2	peg 1	$\xrightarrow{sleeve\ C}$	peg 2
3	peg 3	$\xrightarrow{disc\ C}$	peg 2	3	peg 3	$\xrightarrow{bearing\ D}$	peg 2
4	peg 1	$\xrightarrow{disc\ A}$	peg 3	4	peg 1	$\xrightarrow{bearing\ B}$	eject bearing B
-	–	-		5	enter bearing E	$\xrightarrow{bearing\ E}$	peg 3
5	peg 2	$\xrightarrow{disc\ C}$	peg 1	6	peg 2	$\xrightarrow{bearing\ D}$	peg 1
6	peg 2	$\xrightarrow{disc\ B}$	peg 3	7	peg 2	$\xrightarrow{sleeve\ C}$	peg 3
7	peg 1	$\xrightarrow{disc\ C}$	peg 3	8	peg 1	$\xrightarrow{bearing\ D}$	peg 3

The PN-based model



The interpretation of the PN-places

p1 – the initial configuration

p2 – D is put on (symbolic) peg 1

p3 – the configuration without D

p4 – the situation on the peg 2

p5 – the configuration without C

p6 – E is put on A

p7 – D is added

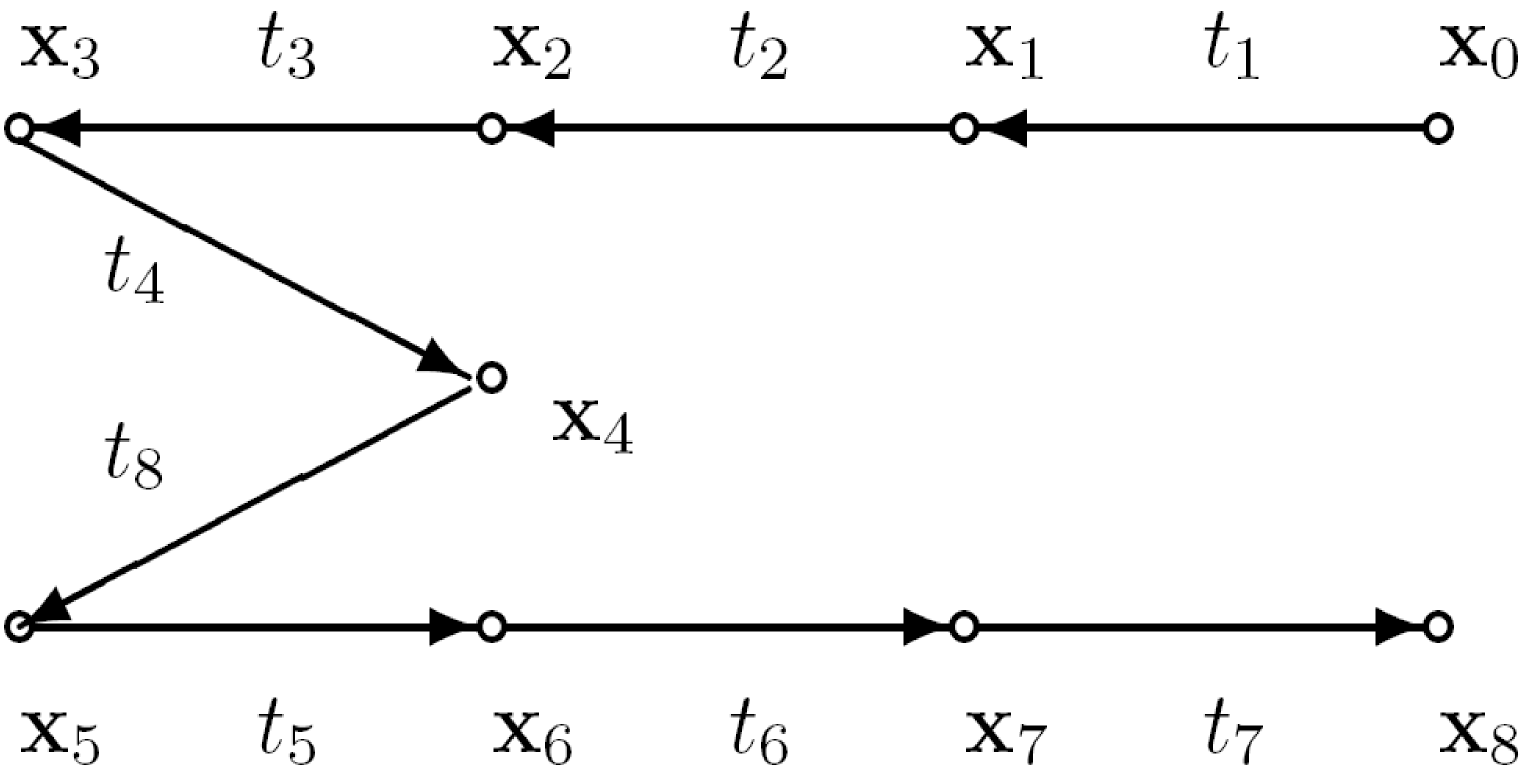
p8 – C is added

p9 – the final configuration

p10 – B is ejected

p11 – E is available

p12 – A is free of parts


x3
t3
x2
t2
x1
t1
x0
t4
x4
t8
x5
t5
x6
t6
x7
t7
x8

The state space

$$\mathbf{X}_{reach}^{Hanoi} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The flexible manufacturing system

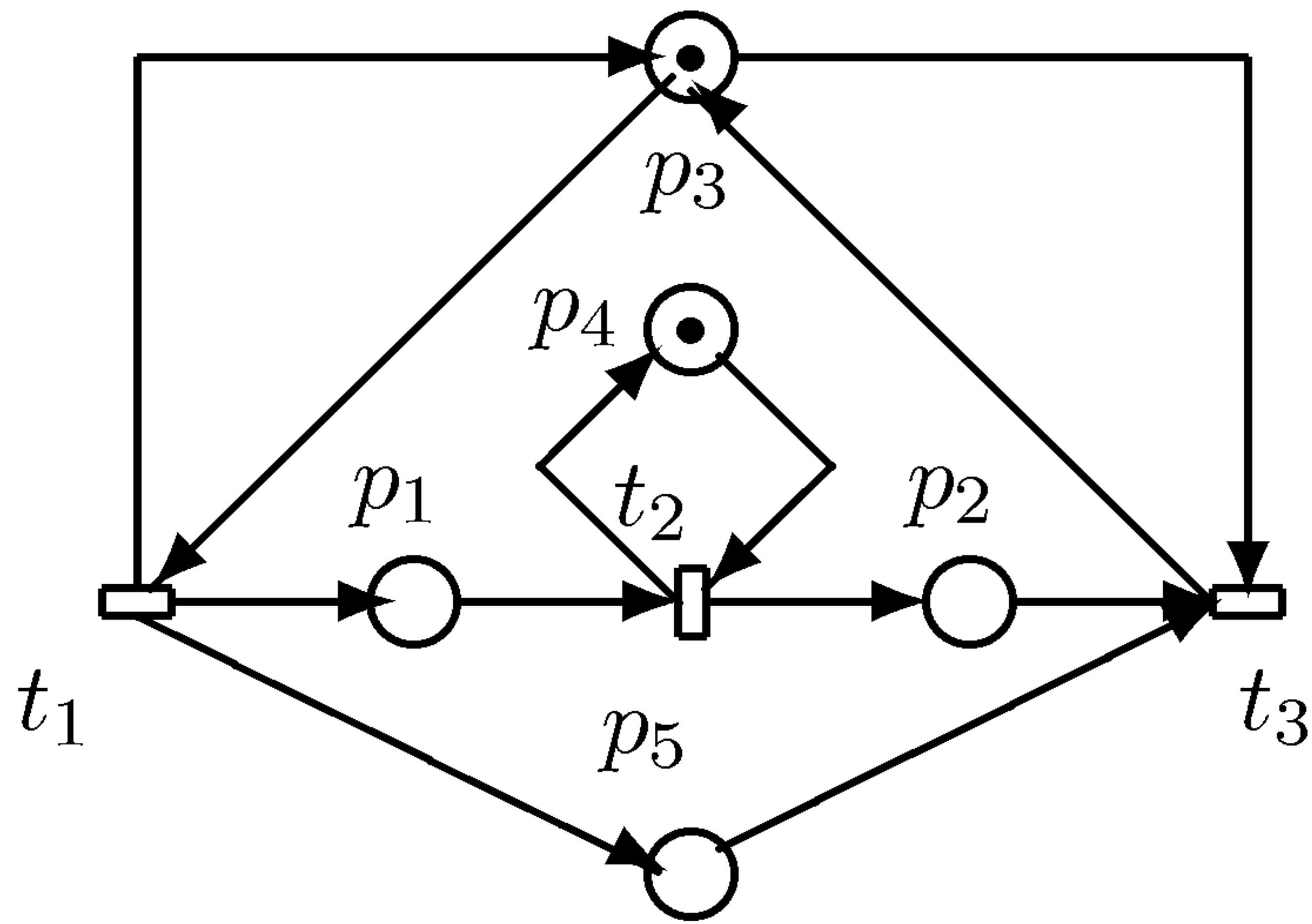
Consider the robotic cell with two conveyors C1, C2, the NC-machine M, with the buffer B (having the input part B1 and the output part B2), and the robot R.

Defining the PN places and transitions:

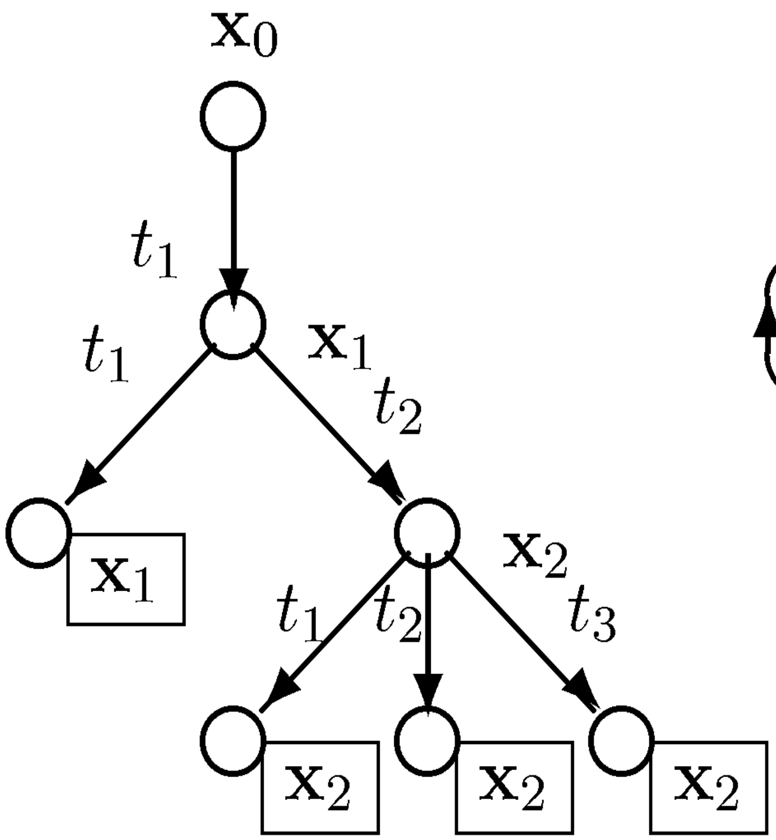
p1 = waiting the input parts
p2 = waiting the output parts
p3 = R is available
p4 = M is available
p5 = contents of B

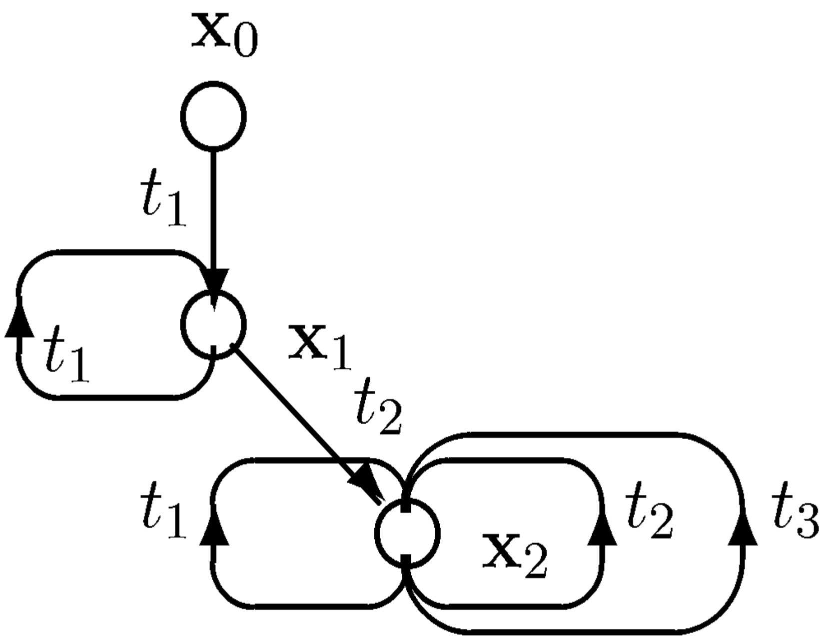
t1 = taking from C1 by R
t2 = machining by M
t3 = putting on C2 by R

The PN-based model of the FMS



The reachability tree and reachability graph





The model parameters

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_0 = (0,\ 0,\ 1,\ 1,\ 0)^T$$

The RT and state space

$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & (1,2,3) \end{pmatrix} \quad \mathbf{\Delta}_k = \begin{pmatrix} 0 & 0 & 0 \\ t_1 & t_1 & 0 \\ 0 & t_2 & (t_1, t_2, t_3) \end{pmatrix}$$

$$\mathbf{X}_{reach} = \begin{pmatrix} 0 & \omega & \omega \\ 0 & 0 & \omega \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & \omega & \omega \end{pmatrix}$$

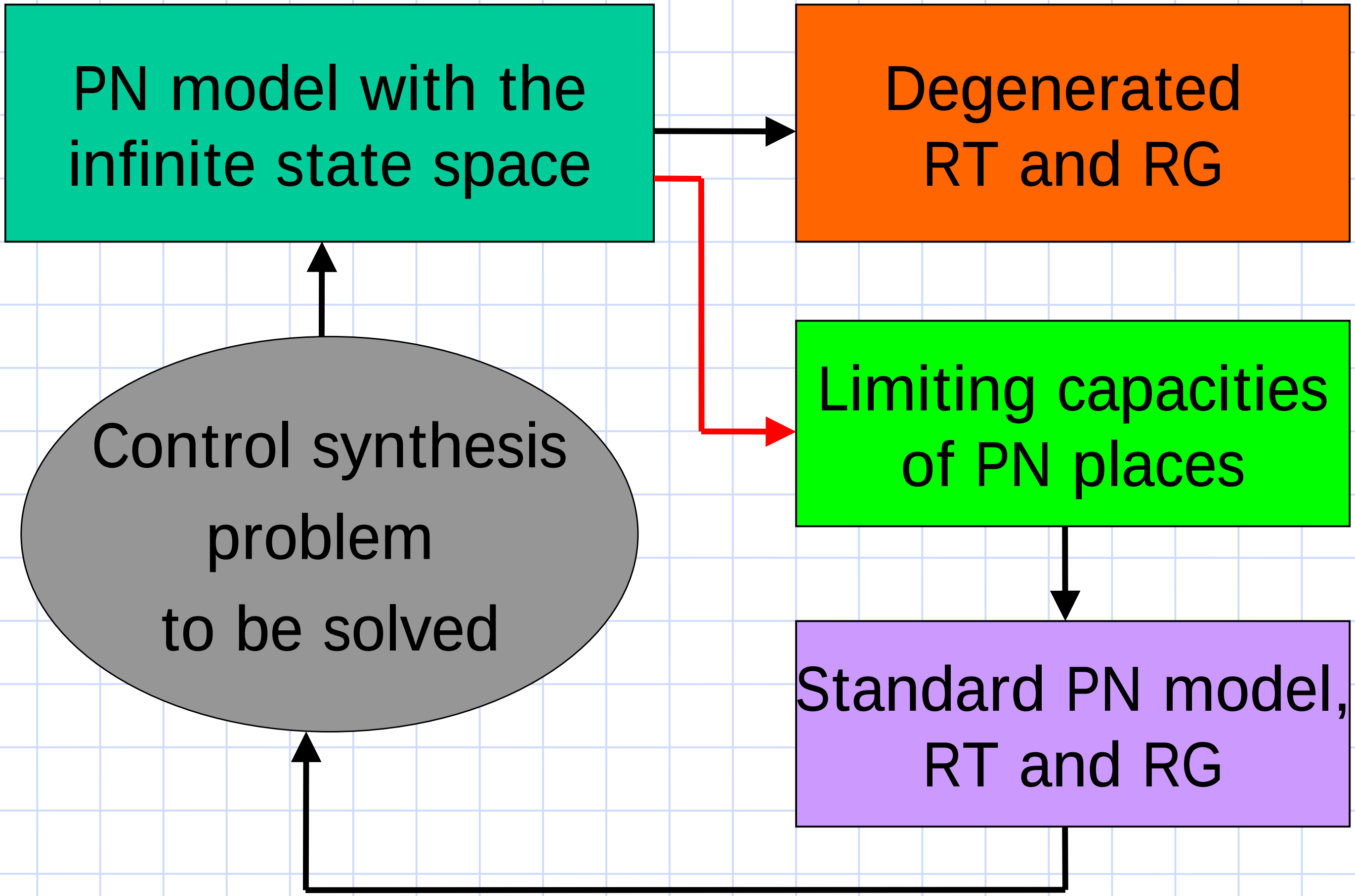
The ambiguity and how to deal with it

When the capacities of places are infinite the ambiguity occurs in the matrix A.

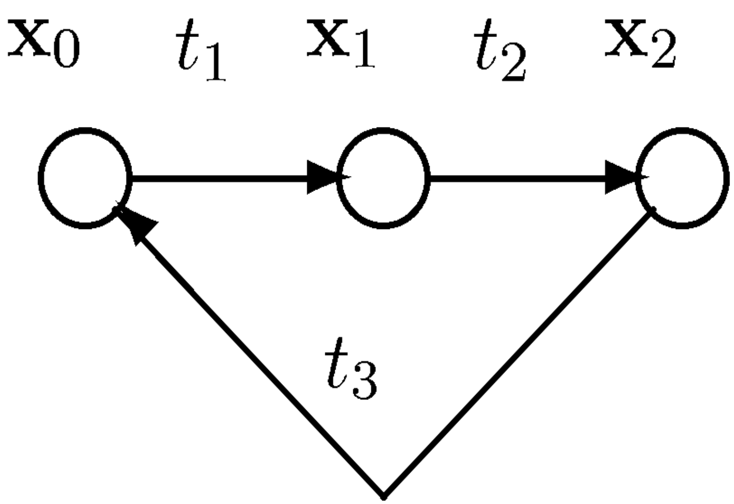
Namely, the cycles engender in the RT and RG. The state space of the reachable states is infinite. Infinity is expressed by the symbol ω

Hence, in order to find a reasonable solution, the finite capacities of the PN places have to be determined.

Dealing with the ambiguity



The finite capacity $c_{p_5} = 1$



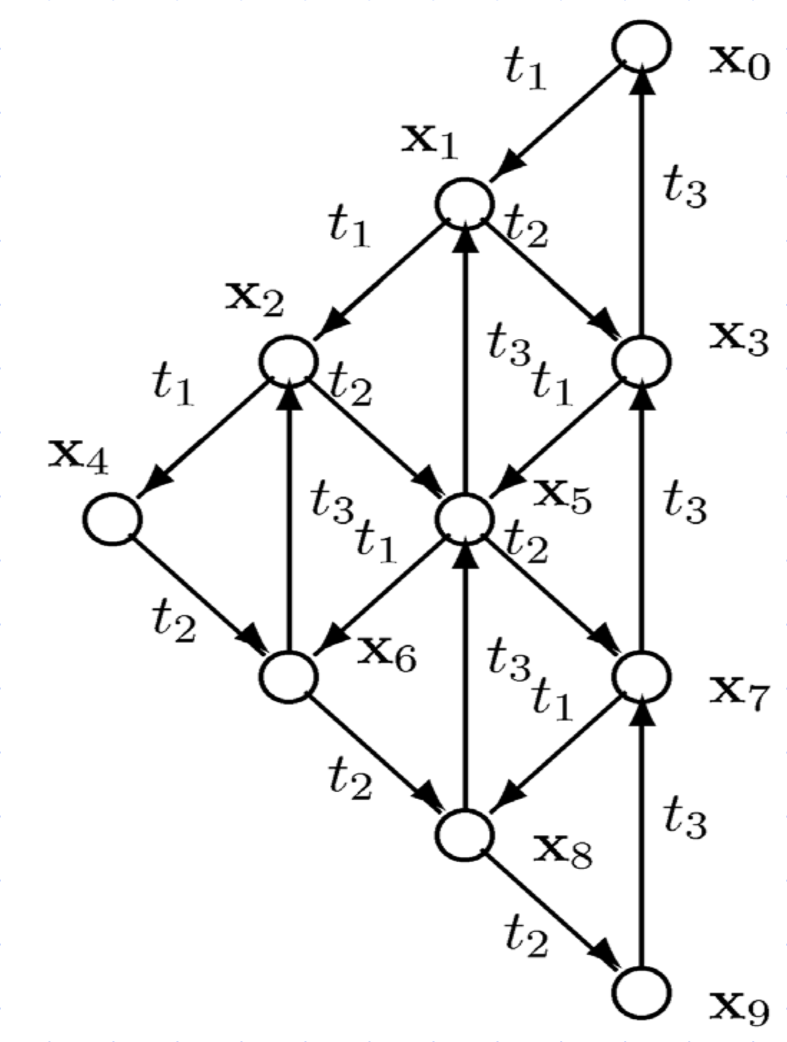
$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 3 & 0 & 0 \end{pmatrix} \quad \mathbf{X}_{reach} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The finite capacities

$$c_{p_i} = 3, \ i = 1, \ 2, \ 5$$

$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \end{pmatrix}$$

The reachability graph



The state space of reachable states

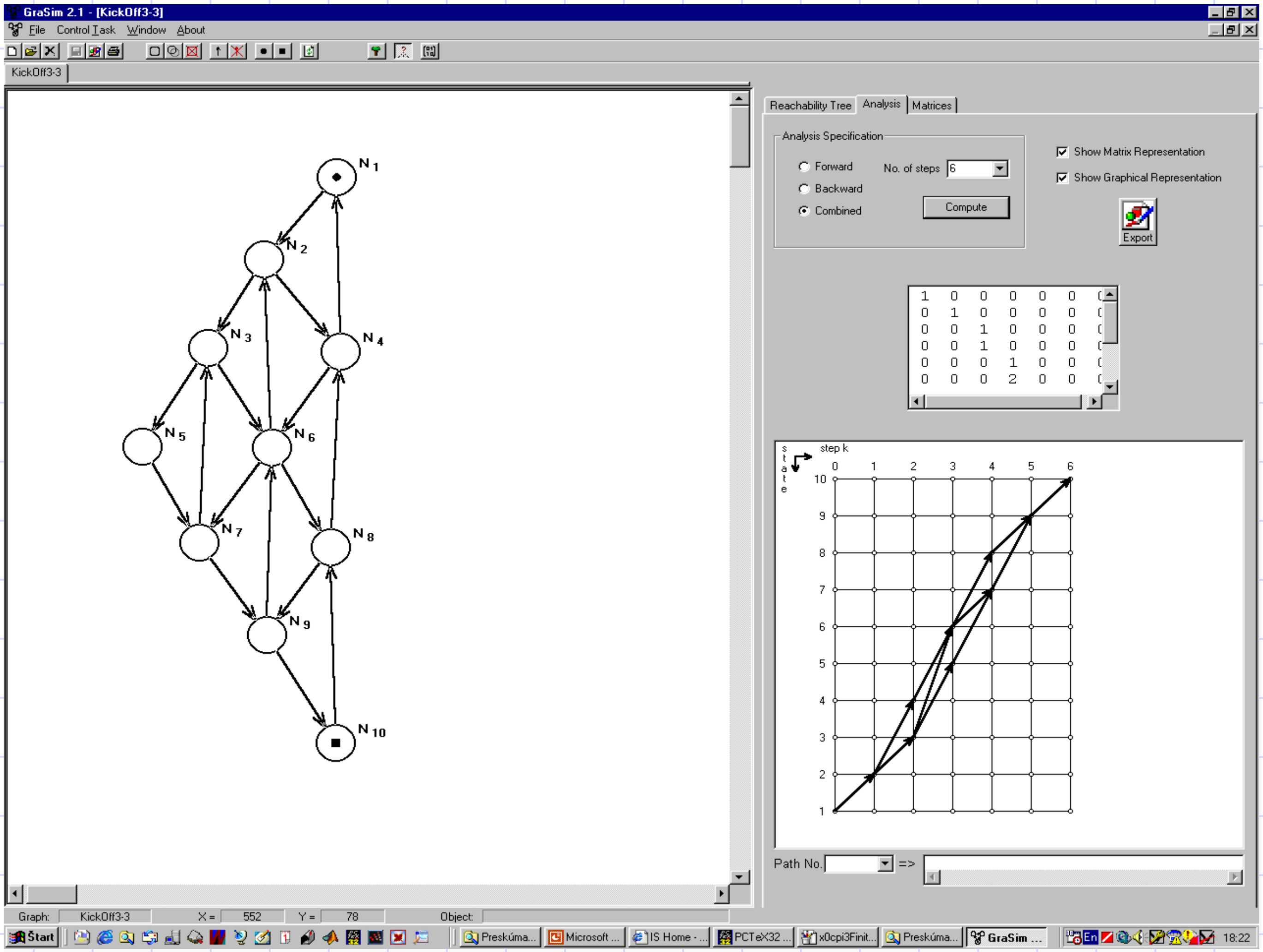
$$\mathbf{X}_{reach} = \begin{pmatrix} 0 & 1 & 2 & 0 & 3 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 3 & 2 & 3 & 2 & 3 & 3 \end{pmatrix}$$

The control synthesis

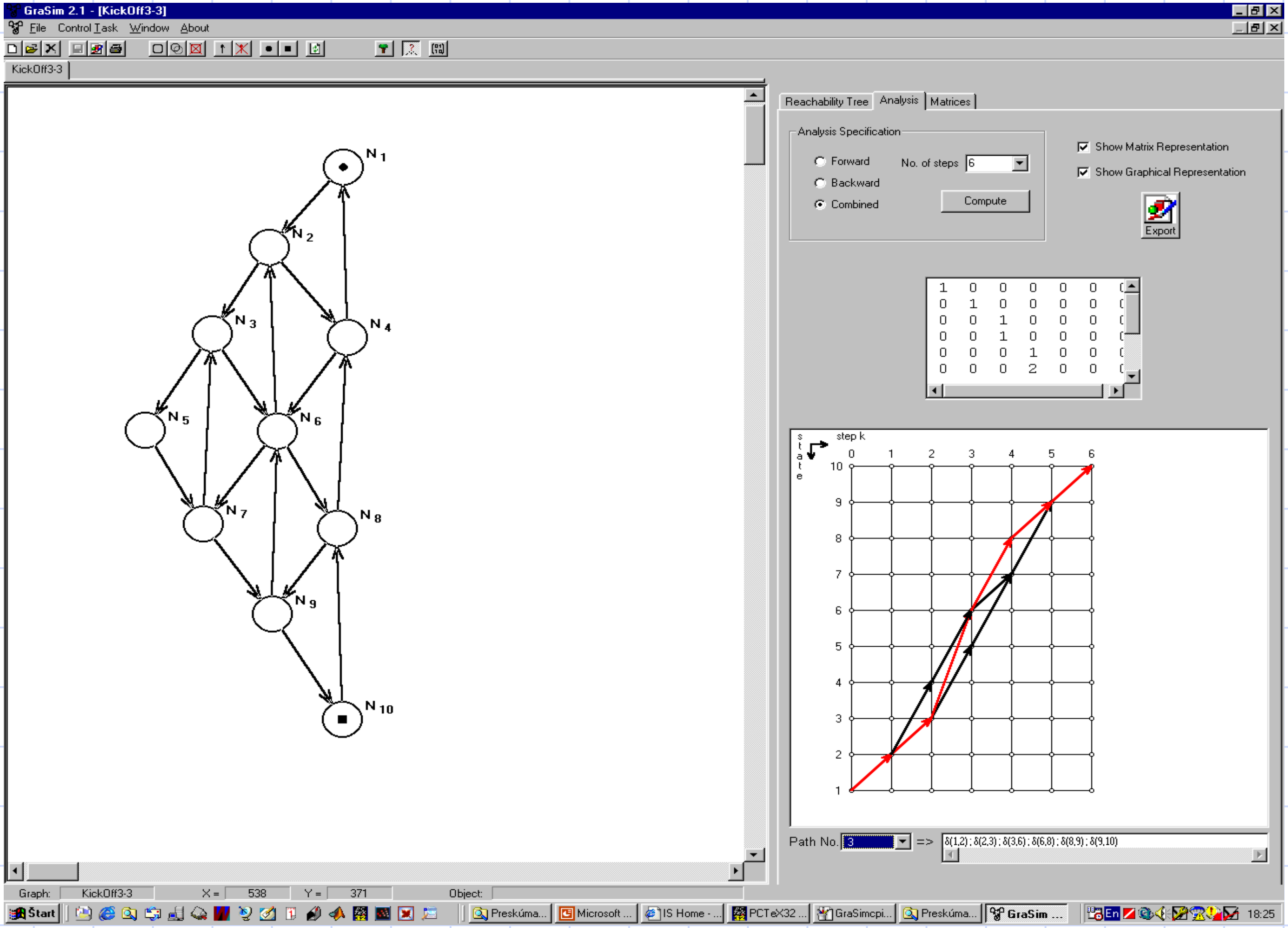
$$\mathbf{x}_0 = (0,\ 0,\ 1,\ 1,\ 0)^T \qquad \mathbf{x}_9 = (0,\ 3,\ 1,\ 1,\ 3)^T$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

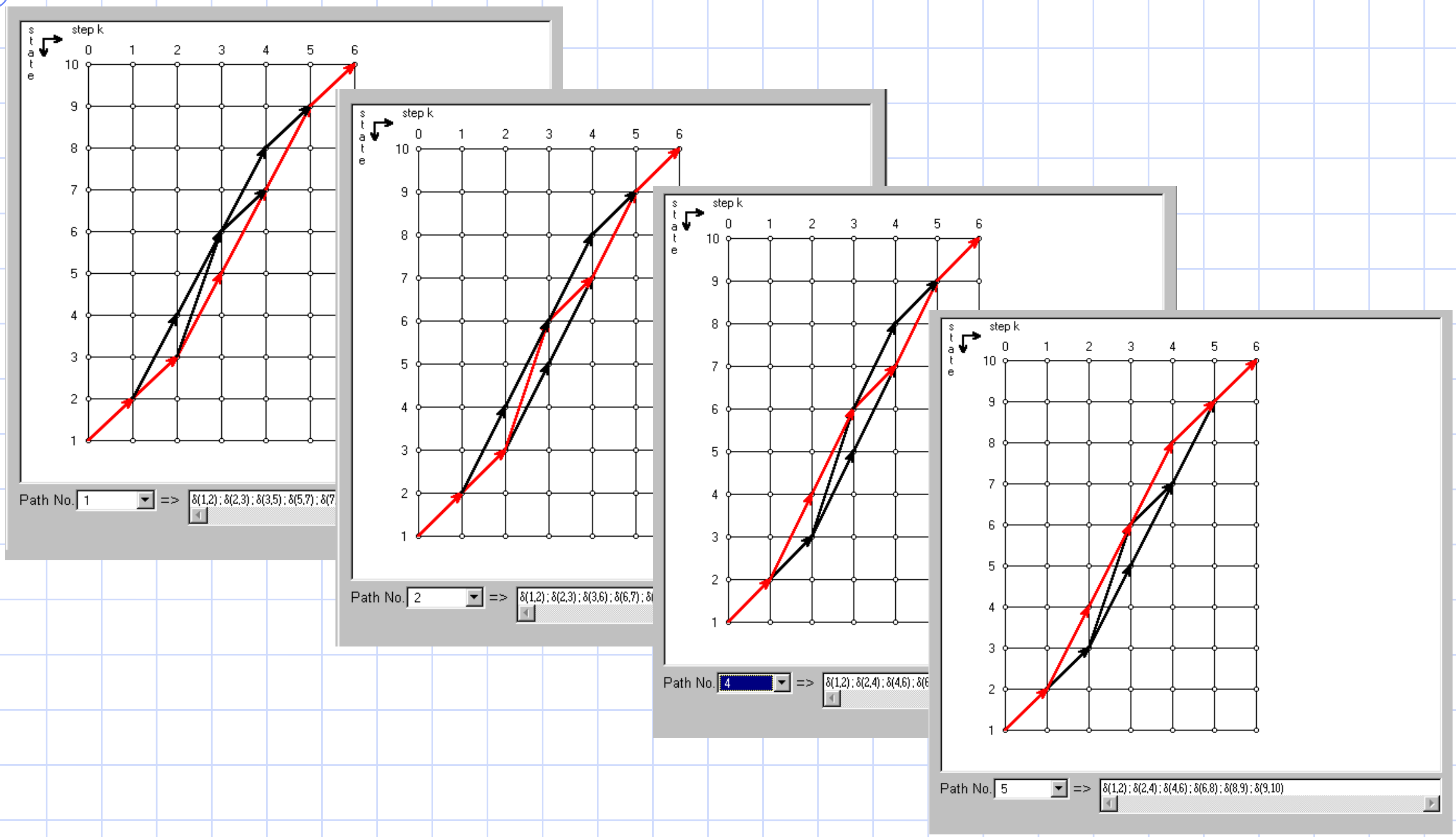
Using the system GraSim



The trajectory No. 3



The other trajectories



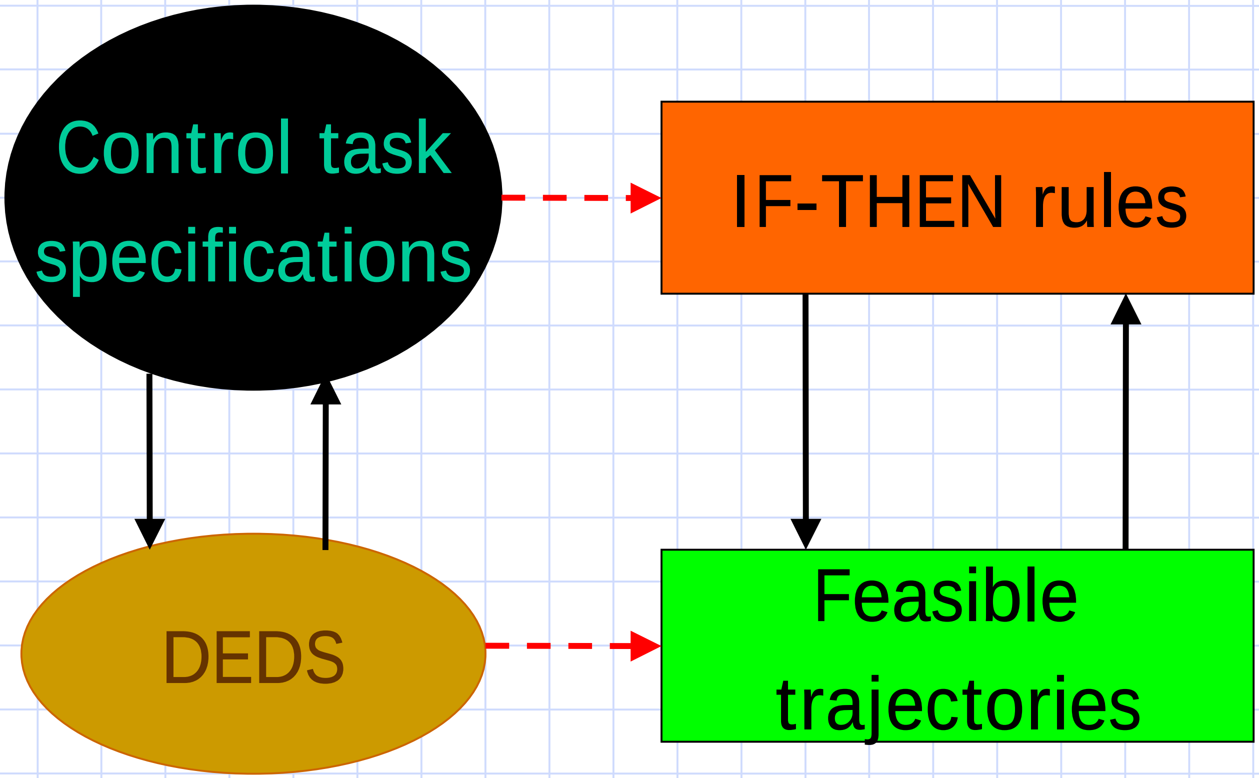
Intelligent control synthesis

DEDS control task specifications are usually given in nonanalytical terms, often only verbally.

To choose the most suitable trajectory knowledge-based approaches must be used.

The knowledge base (KB) expressing the control task specifications in the form of IF-THEN rules can be modelled by means of the logical and/or fuzzy PN. Thus, the KB can be expressed in analytical terms.

Knowledge-based choice of the trajectory



Conclusions

- Simple general method of DEDS modelling, analyzing and control synthesis was presen-ted
- Its applicability to different kinds of systems was demonstrated
- A simple general method for agent-based problem solving was presented
- Its applicability to solving the problem of the DES control synthesis was demonstrated

- Three different approaches to the control synthesis of the same DES were illustrated on the example:
  - Mutual intersection of autonomous solutions of the elementary agents
  - Solving the global problem in the whole
  - Utilizing the invariants of Petri net-based model
- The approaches were compared and evaluated.

and finally,

- Several engineering applications were presented to illustrate the applicability of the approach

Future work on this way

- To innovate presented methods permanently
- To extend its reasonable applicability for larger and larger class of DES able to be modelled by Petri nets
- To find new methods, procedures and tools for DEDS modelling, analyzing and control