Industrial Automation

(Automação de Processos Industriais)

DES and Industrial Automation

http://users.isr.ist.utl.pt/~jag/courses/api1819/api1819.html

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2018/2019

Syllabus:

Chap. 7 – Analysis of Discrete Event Systems [2 weeks]

…

Chap. 8 - DESs and Industrial Automation [1 week]

GRAFCET / Petri Nets Relation

Model modification

Tools adaptation

Analysis of industrial automation solutions by analogy with Discrete Event Systems

…

Chap. 9 – Supervision of DESs [1 week]

Some pointers to Discrete Event Systems

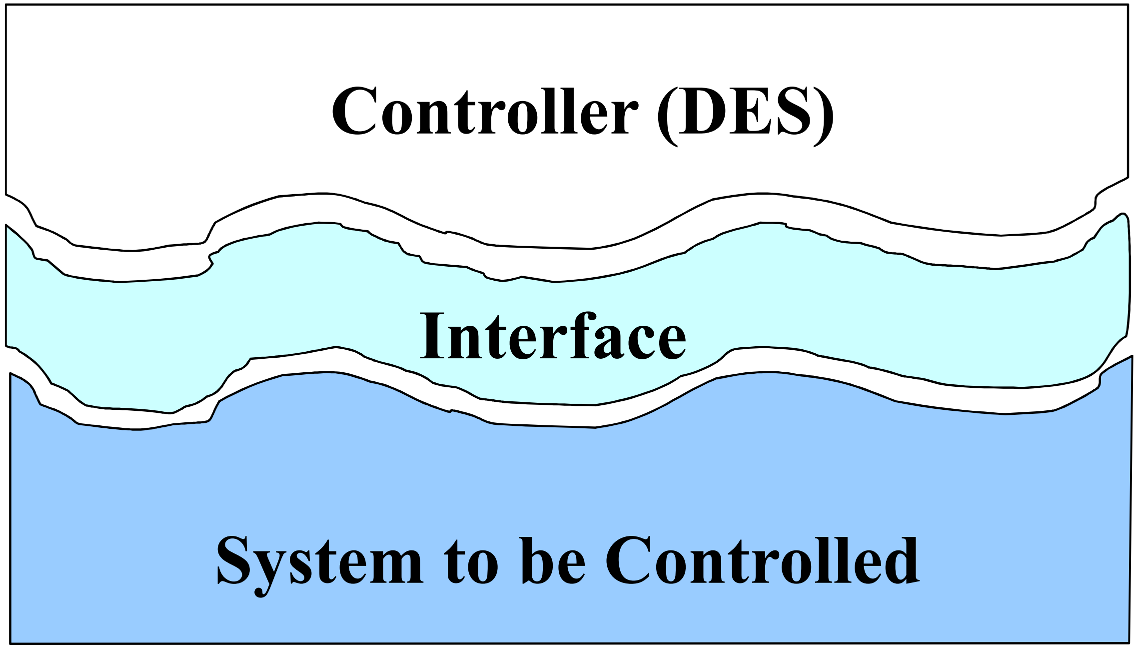
History: http://prosys.changwon.ac.kr/docs/petrinet/1.htm

Tutorial:
http://www.eit.uni-kl.de/litz/ENGLISH/members/frey/VnVSurvey.htm
http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analysers, and Simulators:
http://www.ppgia.pucpr.br/~maziero/petri/arp.html (in Portuguese)
http://wiki.daimi.au.dk:8000/cpntools/cpntools.wiki
http://www.informatik.hu-berlin.de/top/pnk/download.html

Bibliography:
* **Petri Nets and GRAFCET: Tools for Modelling Discrete Event Systems**
R. DAVID, H. ALLA, New York : PRENTICE HALL Editions, 1992

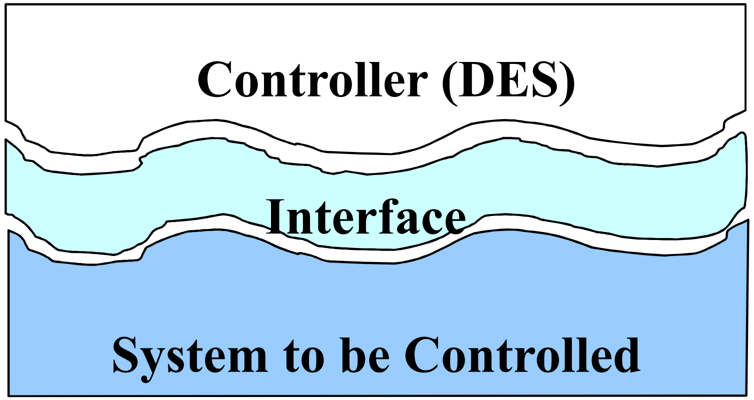
DES Implementation: Models

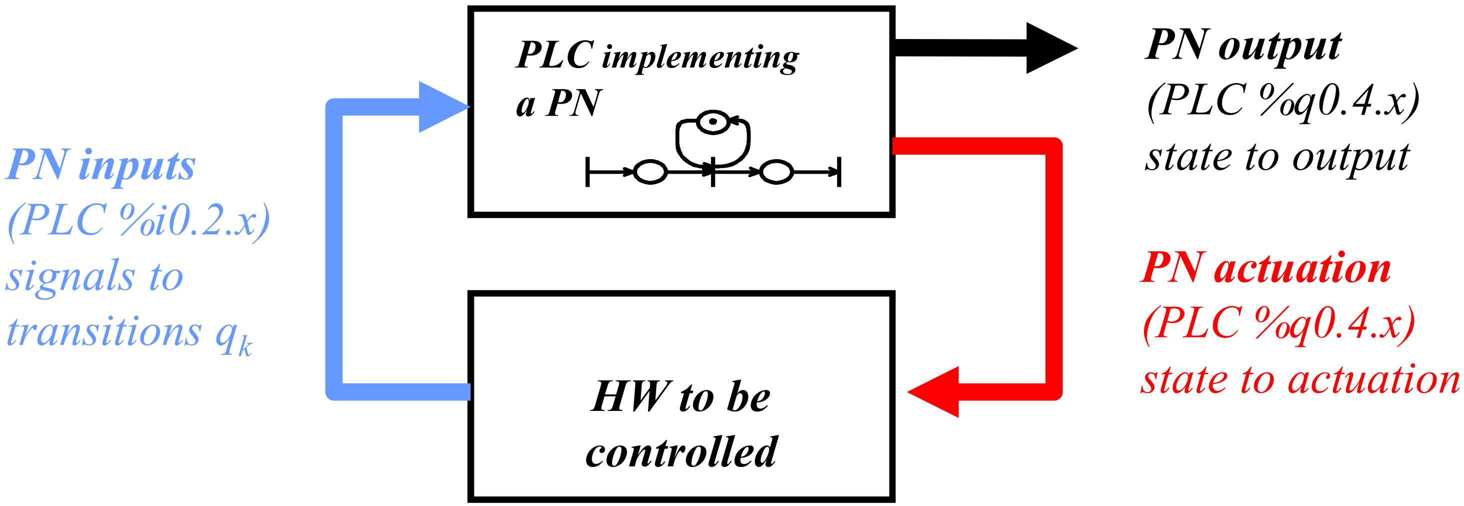


Besides ***modelling the DES*** *it is required to*

design ***models*** *of the* ***System to be controlled*** *and*
of the ***Interface*** *to be used*

DES Implementation: Petri Net and HW (input / output)





Given a Discrete Event System how to implement it?

1. Use a GRAFCET

a) Less modeling ability

b) Implementation in PLCs is straightforward

c) No analysis (or very scarce) methods available

2. Use a Petri Net

a) More modeling capacity

b) No direct implementation in PLCs (therefore indirect or special software solutions required)

c) Classical analysis methods available

(3. Use an Automaton)

Analysis of solutions

GRAFCET and Petri Nets

Similarities to exploit:

a) Steps and Places are similar

b) Transitions compose both tools

c) Places can be used to implement counters (on marking changes) and binary variables (marked vs unmarked place)

d) Logic functions can be rewritten resorting to the firing of transitions

Differences to be taken into account:

a) Firing rules (mutual exclusion)

b) Conflicts

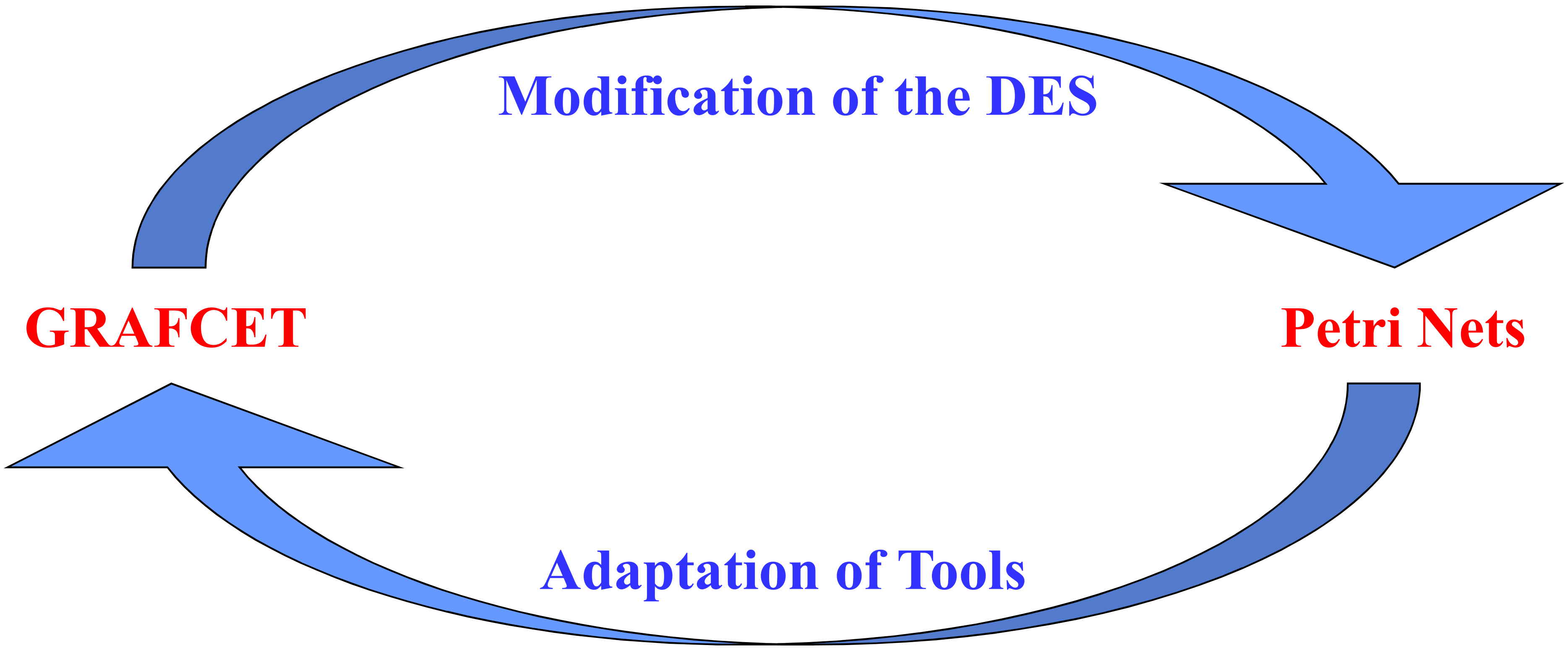
c) Binary activation of stages

d) Interface with the system to be controlled

e) Activation functions

Implementation of DES using GRAFCET

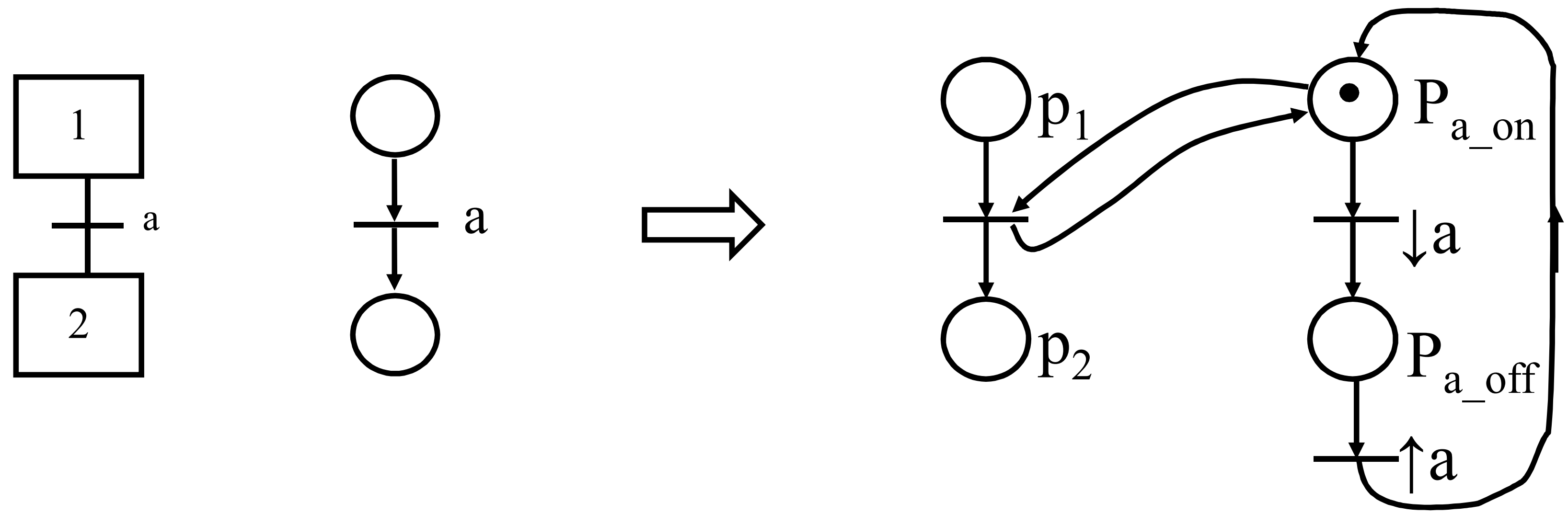
Analysis



Analysis of solutions

GRAFCET → Petri Nets

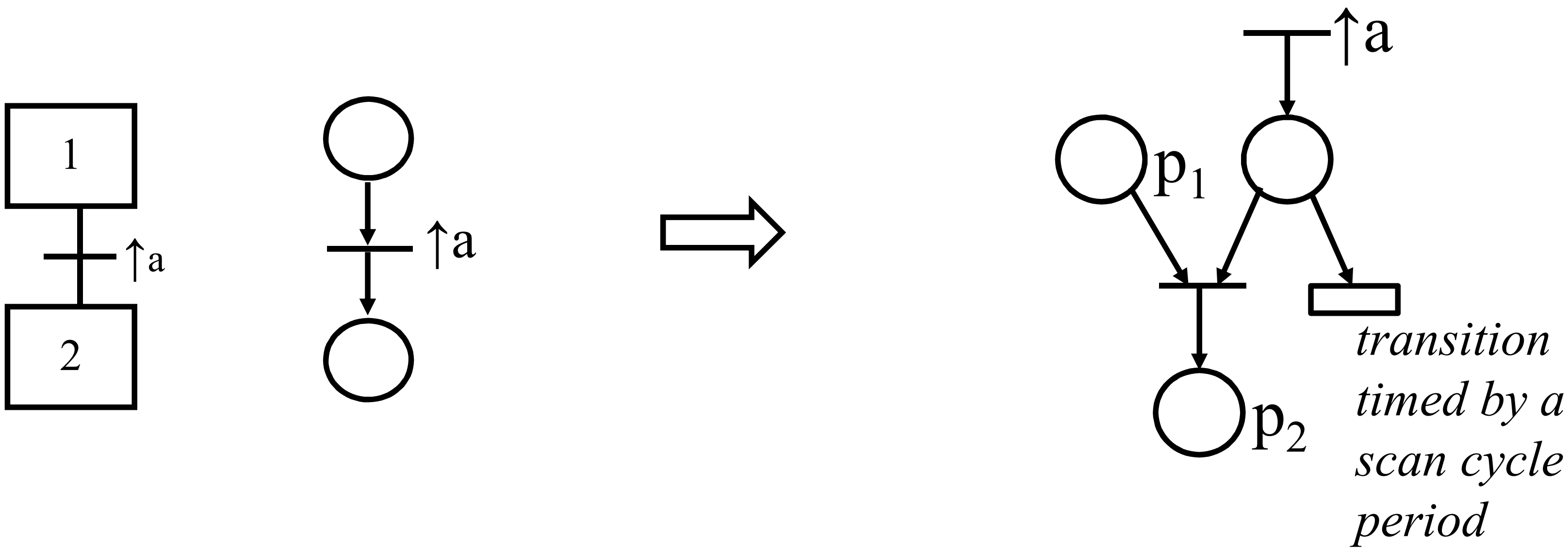
Representation of variables active on level,
i.e. give a marking representation to an input active as a level



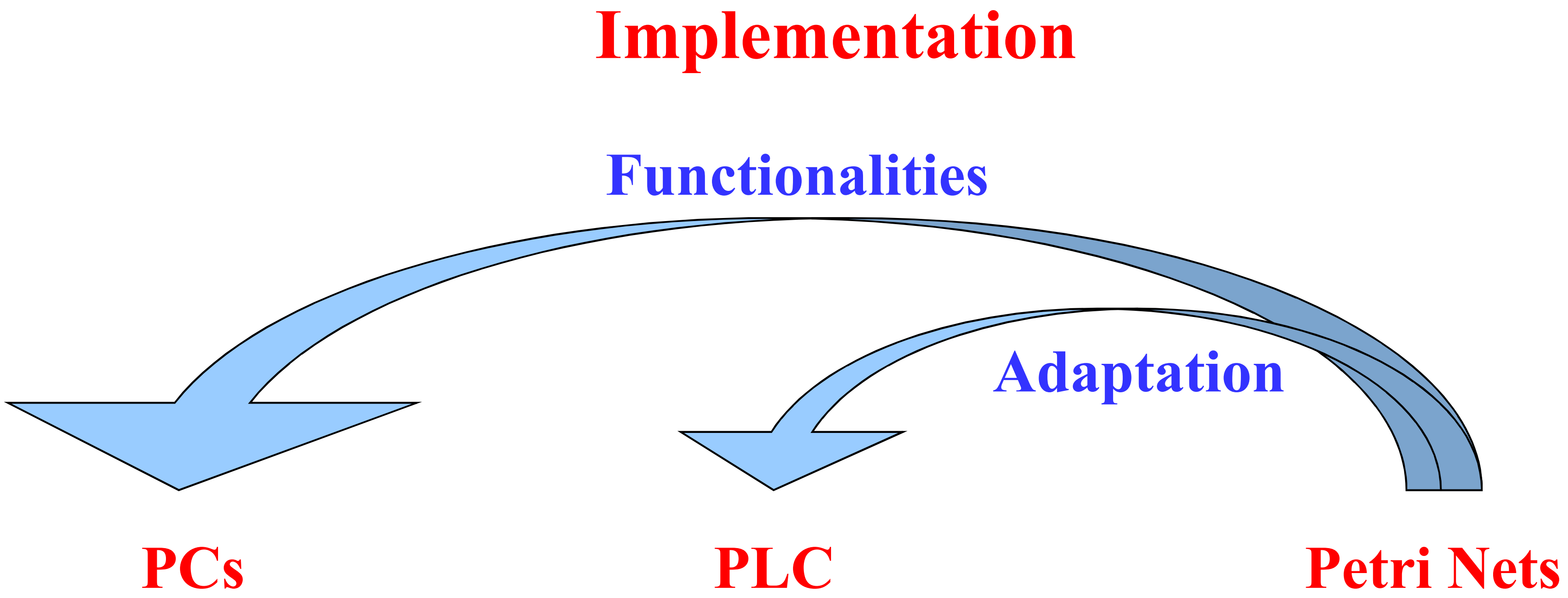
Analysis of solutions

GRAFCET → Petri Nets

Representation of variables active at edge,
i.e. give a marking representation to an input active as a edge trigger



Implementation of DES using Petri Nets



Both solutions are valid
(but out of the scope of this course).

Analysis of solutions

Petri Nets → GRAFCET

Adaptation of Tools:

1) Reachability Tree → Reachability Graph

2) Method of the Matrix Equations
to describe the state evolution

Petri Nets → GRAFCET

Reachability Graph

Is a graph containing the reachable markings.
Is composed by two types of nodes:

- terminal
- interior

The duplicated nodes are not represented. *They become connected to the respective copies.*

The symbol infinity (ω) is not used in GRAFCET. *While in Petri nets ω is necessary to obtain finite trees, when a marking covers other(s), in GRAFCET the finite number of states renders ω unnecessary.*

Petri Nets → GRAFCET

Reachability Graph

Theorem - If a reachability graph has terminal nodes then the corresponding GRAFCET has deadlocks.

This reachability graph will be used to study the properties introduced in Chapter 6.

Petri Nets → GRAFCET

Reachable Set

Given the GRAFCET G=(S, T, I, O, μ_0) with initial marking μ_0 ,
the set of all markings that are reachable is the reachable set R(G, μ_0).

Property usage: $\mu \in R(G, \mu_0)$

Remark: the Reachable Set ***is not infinite!***
Given a GRAFCET with m steps it has at most 2^m nodes.
In some cases, m steps imply just m nodes.

Petri Nets → GRAFCET

Safeness, Boundedness and Limitation

The GRAFCET G=(S, T, I, O, μ_0) is always safe

The GRAFCET may become not safe if one uses some auxiliary elements such as counters or buffers.

If one uses these auxiliary elements, the analysis methods studied for Petri Nets can be used to assert specific safeness, boundedness and limitation properties.

Petri Nets → GRAFCET

Conservation

A GRAFCET G=(S, T, I, O, μ_0) is **strictly conservative** if for all $\mu' \in R(C, \mu)$

$$\sum_{p_i \in P} \mu'(p_i) = \sum_{p_i \in P} \mu(p_i).$$

A GRAFCET G=(S, T, I, O, μ_0) is **conservative** if there exist a weight vector w without null elements, for all $\mu' \in R(C, \mu)$ such that

$$\sum_{p_i \in P} w(p_i)\, \mu(p_i) = \mathit{constant}$$

Petri Nets → GRAFCET

Liveness of transitions: The transition t_j is live of

Level 0 - it can never be fired (dead transition).

Level 1 - if it is potentially firable, e.g. if there exist m' ∈ R(C, μ) such that t_j is enabled in μ'.

Level 2 - if, for each positive n, there exist a sequence of firings where occurs n firings of t_j .

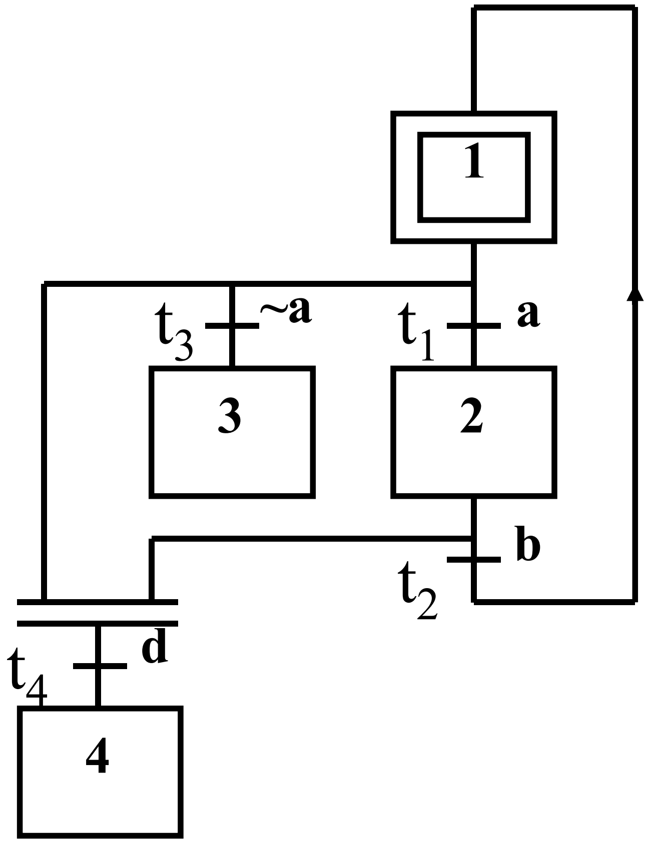
Level 3 - if there exist a sequence of firings where an infinite number of firings of t_j occurs .

Level 4 - if for each μ' ∈ R(C, μ) there exist a sequence s that enables the firing of t_j (live transition).

Petri Nets → GRAFCET

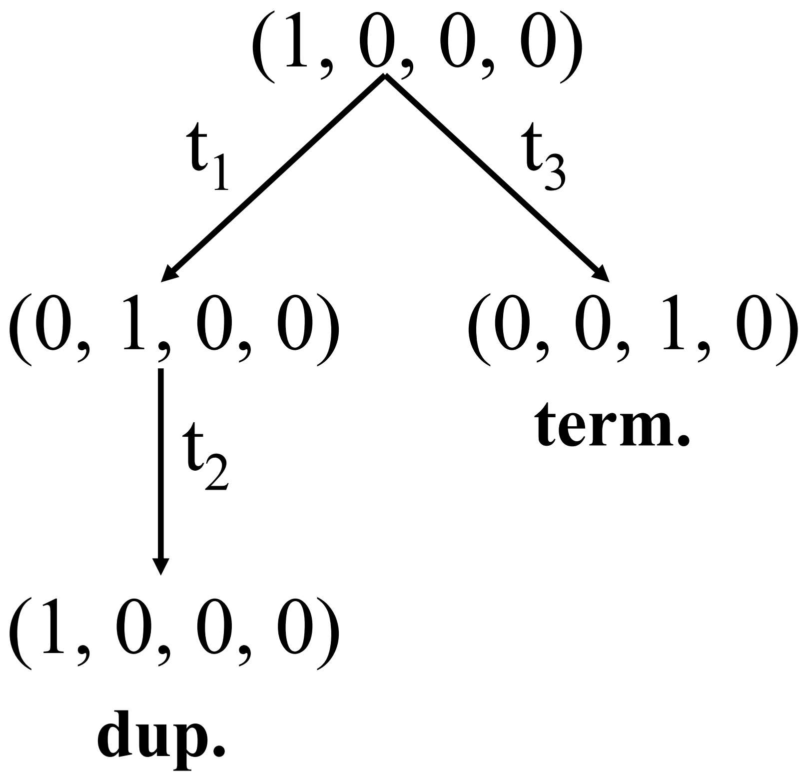
Example of GRAFCET

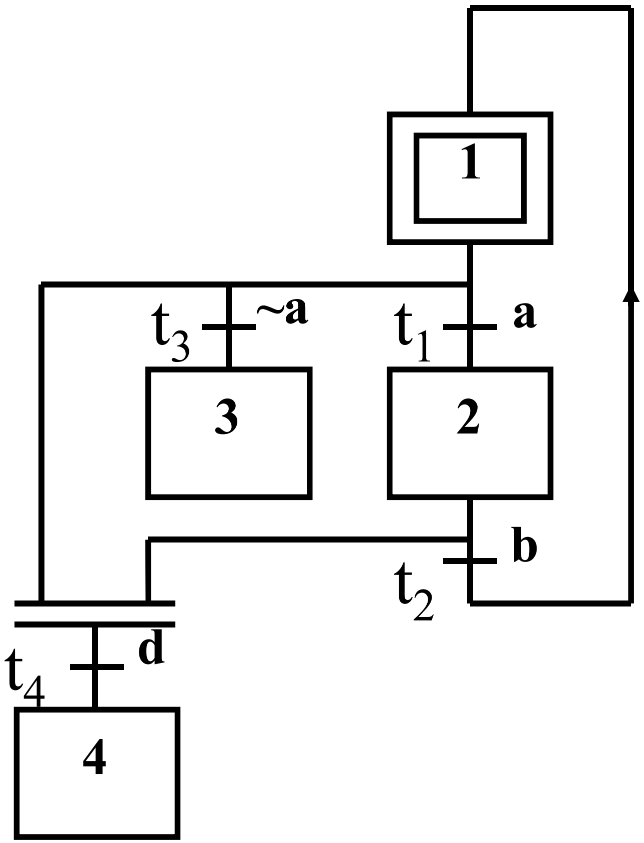
- t_1 and t_2 are level 3.
- t_3 is level 1.
- t4 is level 0.



Petri Nets → GRAFCET

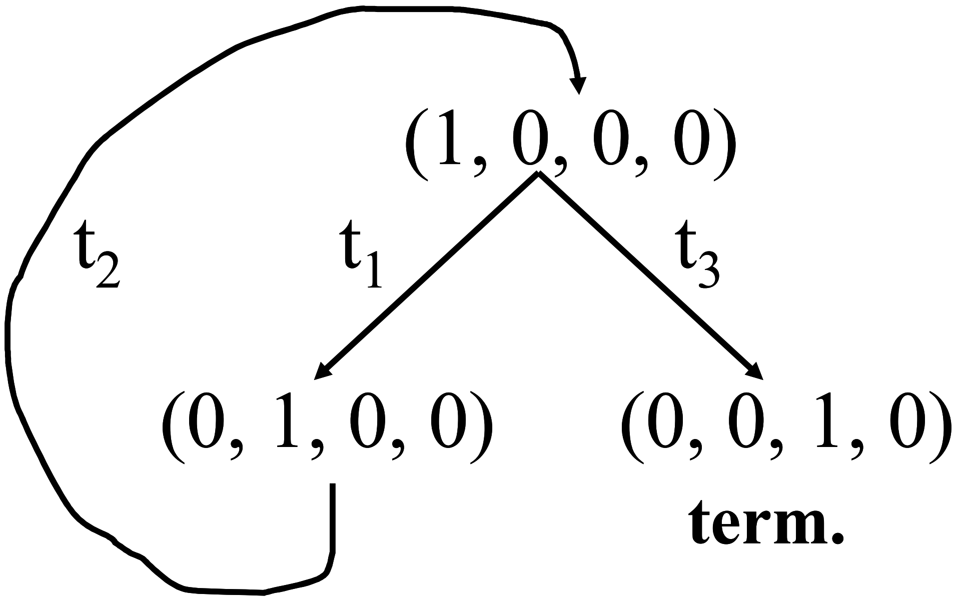
Example of GRAFCET

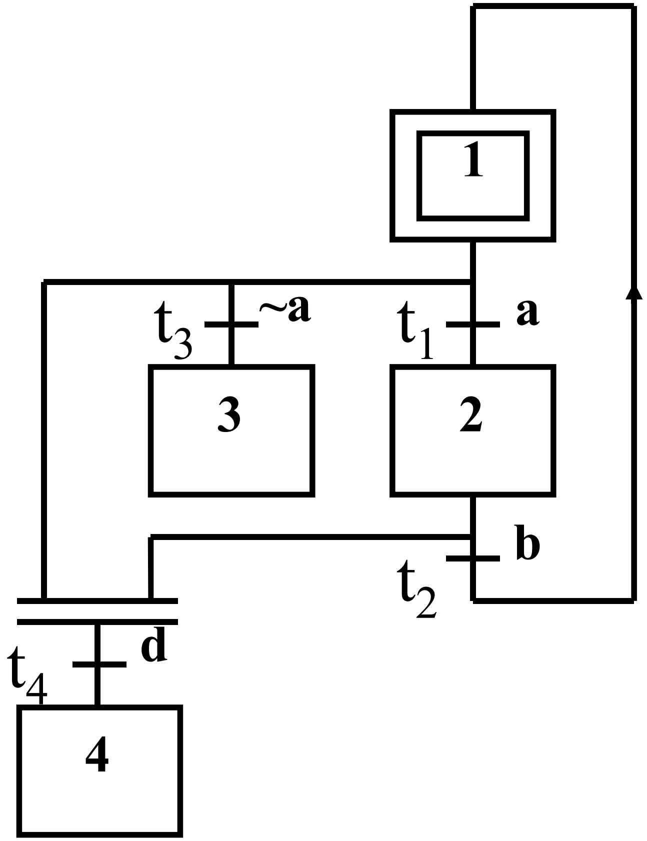




Petri Nets ⟶ GRAFCET

Example of GRAFCET





Strictly conservative.

Petri Nets → GRAFCET

Method of Matrix Equation (for the state evolution)

The evolution of a GRAFCET can be written in compact form as:

$$\mu' = \mu + Dq$$

where:

μ’	- Desired marking vector (column vector)
μ	- Initial marking
q	- Transition firing vector (column vector)
D	- Incidence matrix. Accounts for the token evolution as a consequence of transitions firing.

Petri Nets → GRAFCET

Additional problems that can be addressed resorting to the Method of Matrix Equations:

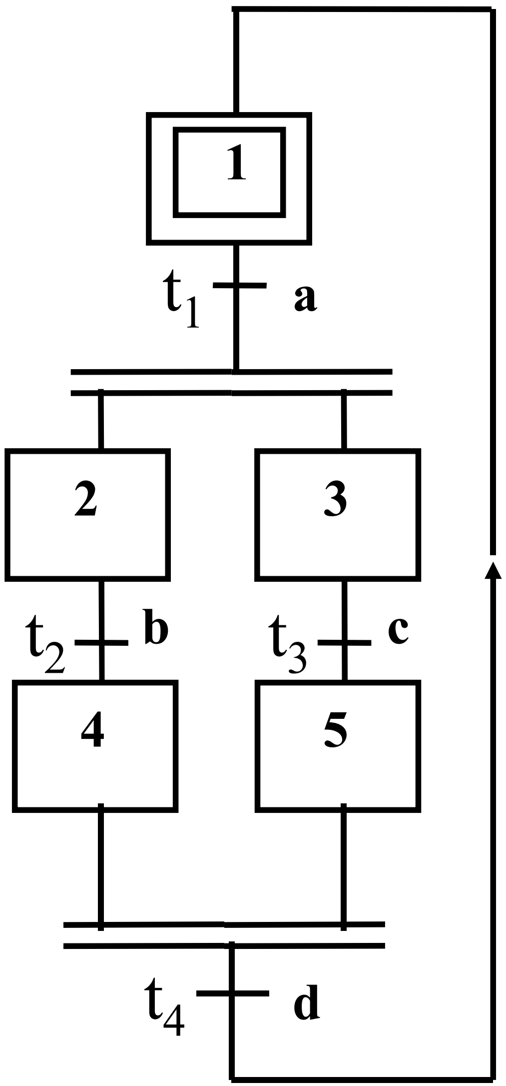
• Reachability (sufficient condition)

> **Theorem** – if the problem of finding the vector of firings, for a GRAFCET without conflicts, from the state μ to the state μ' has no solution using the Method of Matrix Equations, then the problem of reachability of m' is impossible.

• Conservation – the conservation vector can be computed automatically.

• Temporal invariance – cycles of operation can be found.

Example of GRAFCET



$$\mu' = \mu + Dq$$

$$D = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Conservation $w^T D = 0$

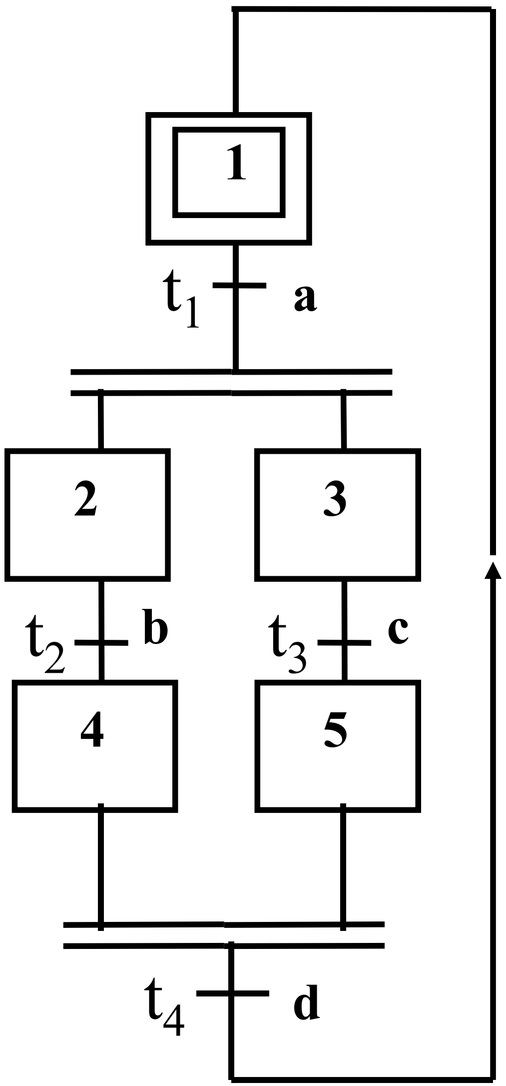
$$\begin{cases} -w_1 + w_2 + w_3 = 0 \\ -w_2 + w_4 = 0 \\ -w_3 + w_5 = 0 \\ w_1 - w_4 - w_5 = 0 \end{cases}$$

$$w_1 = w_3 + w_4$$
$$w_1 = w_2 + w_5$$
$$w_2 + w_3 = w_4 + w_5$$

Solution:
Undetermined
set of equations

$$w = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Example of GRAFCET



$$\mu' = \mu + Dq$$

$$D = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad q = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{bmatrix}$$

Temporal invariance $Dq = 0$

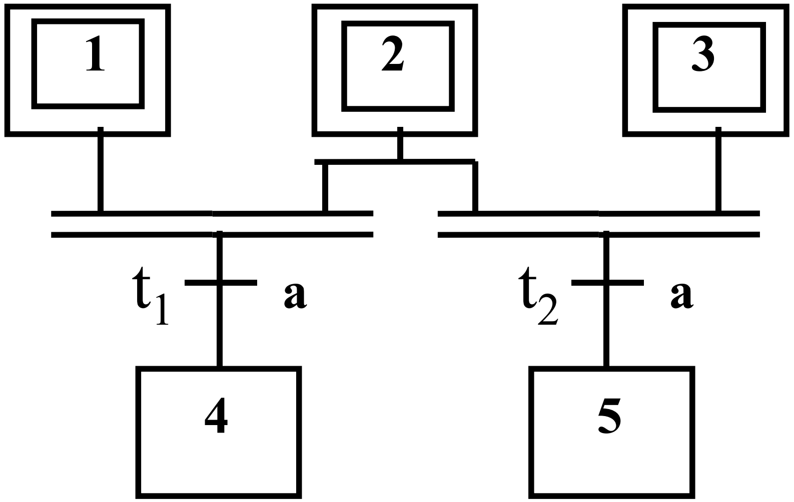
Solution:
Set of equations
with solution

$$\begin{cases} -\sigma_1 + \sigma_4 = 0 \\ \sigma_1 - \sigma_2 = 0 \\ \sigma_1 - \sigma_3 = 0 \\ \sigma_2 - \sigma_4 = 0 \\ \sigma_3 - \sigma_4 = 0 \end{cases}$$

$$\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 1.$$

Example of GRAFCET

$$\mu' = \mu + Dq$$



$$\mu' = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \mu = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 0 \\ -1 & -1 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad q = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}$$

Set of Equations **is impossible** therefore the marking **is not reachable**...

WRONG !

$$\begin{cases} 0 = 1 - \sigma_1 \\ 0 = 1 - \sigma_1 - \sigma_2 \\ 0 = 1 - \sigma_2 \\ 1 = \sigma_1 \\ 1 = \sigma_2 \end{cases}$$

The method fails if there are conflicts!