Petri Nets: Tutorial and Applications

Jeffrey W. Herrmann
Edward Lin

November 5, 1997

The 32th Annual Symposium of the Washington Operations Research - Management Science Council
Washington, D.C.

INSTITUTE FOR SYSTEMS RESEARCH
A National Science Foundation Engineering Research Center, supported by NSF, the University of Maryland, Harvard University, and Industry

CIM Lab
Institute for Systems Research
University of Maryland
College Park, Maryland

Outline

- Purpose
- Applications
- What is a Petri Net?
- Dynamics
- Basic Constructs
- Properties
- Analysis Methods
- Extensions of Petri Nets
- Resources for Petri Nets
- Summary

Purpose

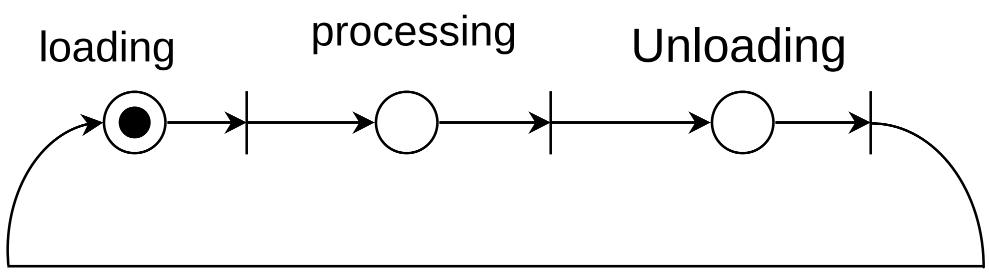
- To describe the fundamentals of Petri nets so that you begin to understand what they are and how they are used.
- To give you resources that you can use to learn more about Petri nets.

Petri Net Applications

- Manufacturing, production, and scheduling systems
- Sequence controllers (Programmable Logic Controller, PLC)
- Communication protocols and networks
- Software -- design, specification, simulation, validation, and implementation

Petri Nets -- Graphic Tool

- A bipartite directed graph containing places (circles), transitions (bars), and directed arcs (places <--> transitions).

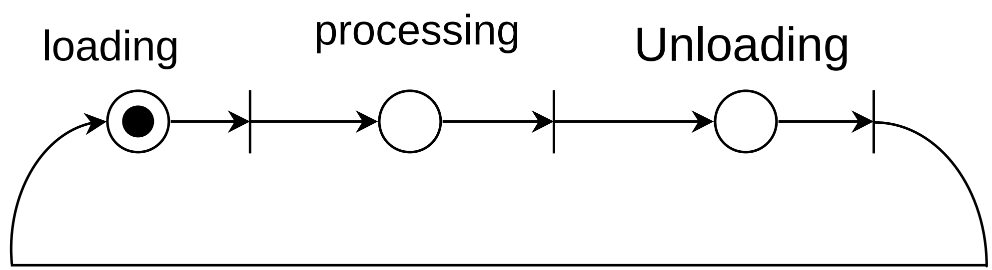


Places -- buffers, locations, states

Transitions -- events, actions

Tokens -- parts

Petri Nets -- Mathematical Models



A Petri net is a four-tuple:

$PN = \langle P, T, I, O \rangle$

P: a finite set of places, $\{p_1, p_2, ..., p_n\}$

T: a finite set of transitions, $\{t_1, t_2, ..., t_s\}$

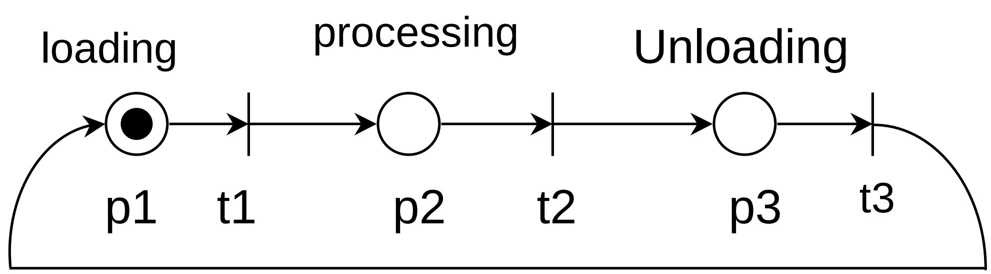
I: an input function, $(T \times P) \longrightarrow \{0, 1\}$

O: an output function, $(T \times P) \longrightarrow \{0, 1\}$

M^0: an initial marking, $P \longrightarrow \mathrm{N}$

$\langle P, T, I, O, M^0 \rangle$ -- a marked Petri net

An Example



- $P = \{p1, p2, p3\}$
- $T = \{t1, t2, t3\}$
- $I =$

$$\begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

- $O =$

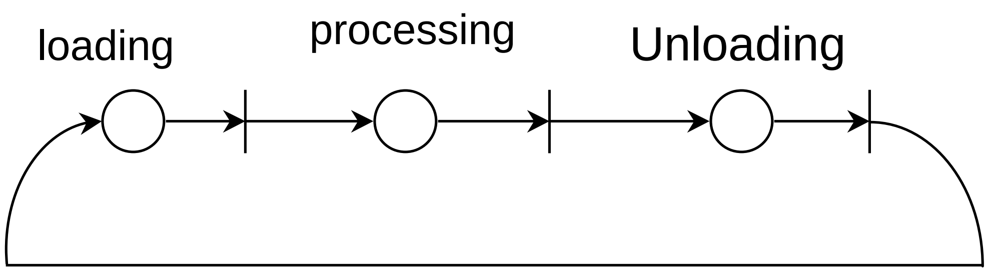
$$\begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

- $M^0 = (1, 0, 0)$

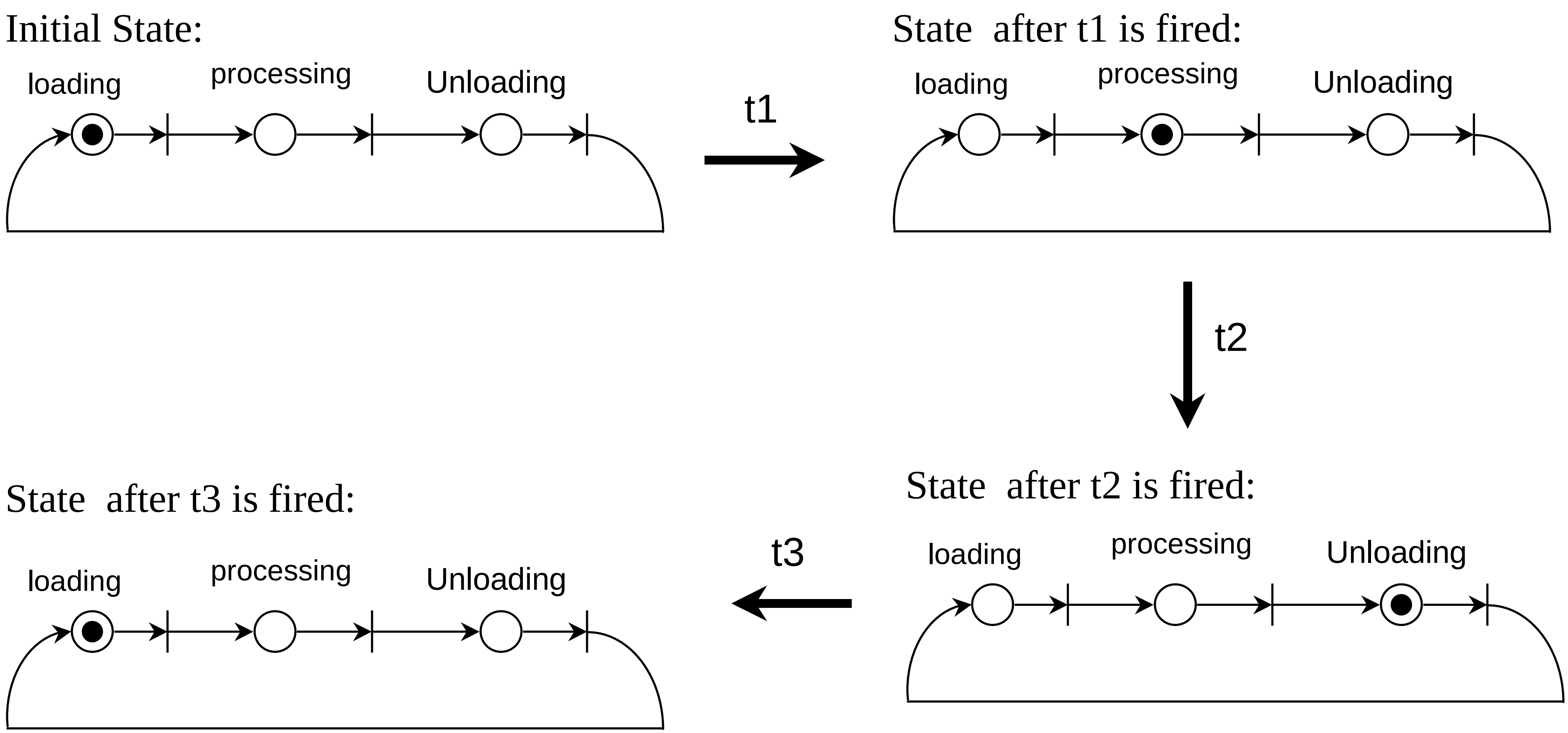
Note:
p1 is the input place of transition t1
p2 is the output place of transition t1

Dynamics

- Enabling Rule:
  - » A transition *t* is enabled if every input place contains at least one token
- Firing Rule:
  - » Firing an enabled transition
    - – removes one token from each input place of the transition
    - – adds one token to each output place of the transition



Dynamics



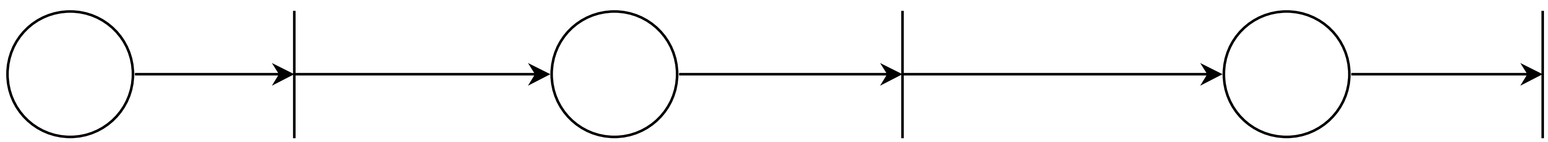
Initial State:
loading
processing
Unloading
t1
State after t1 is fired:
loading
processing
Unloading
t2
State after t2 is fired:
loading
processing
Unloading
t3
State after t3 is fired:
loading
processing
Unloading

Basic Constructs

- Sequential actions
- Dependency
- Conflict (decision, choice)
- Concurrency
- Cycles
- Synchronization - (mutually exclusive actions, resource sharing, communication, queues)

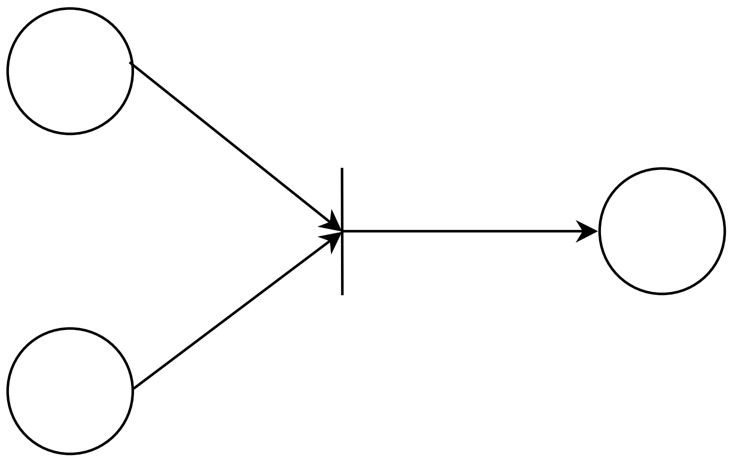
Sequential Actions

Each action is a transition.



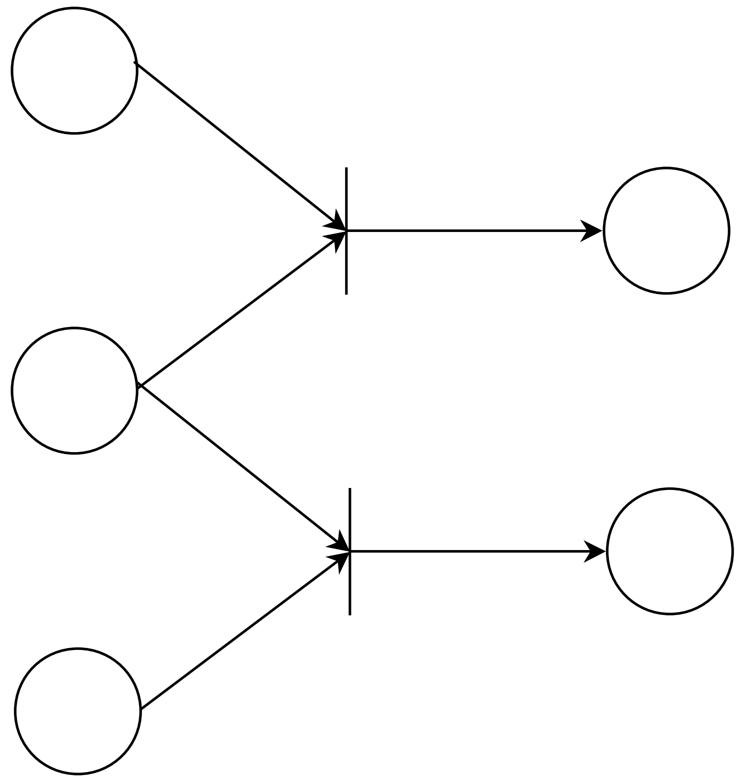
Dependency

A transition requires two inputs.



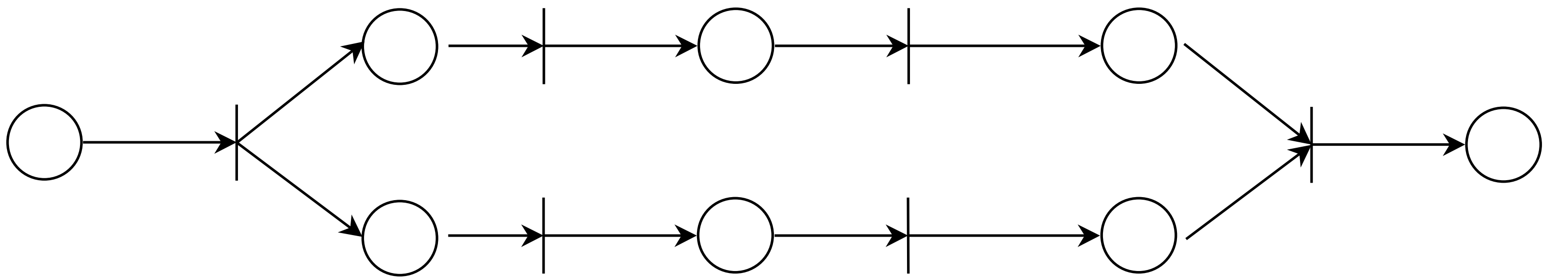
Conflict Construct

Only one of the two transitions can fire.

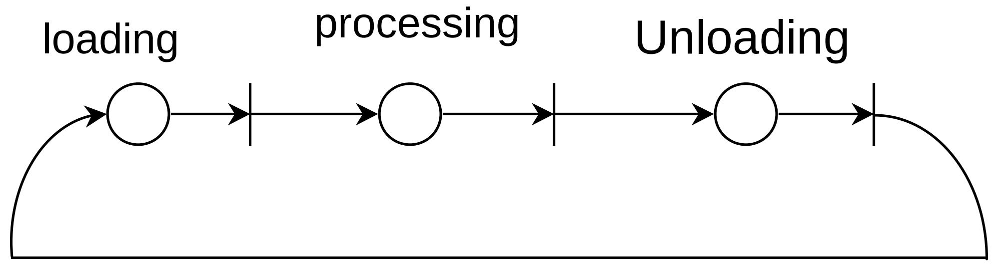


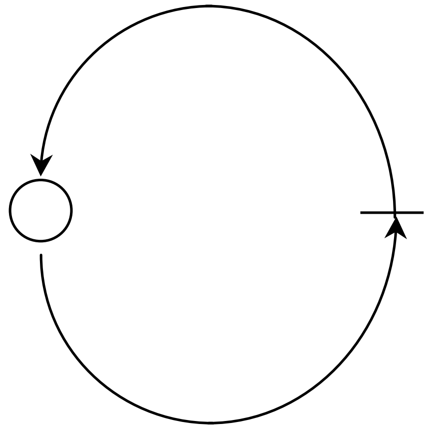
Concurrency Construct

These two sequences can occur simultaneously.



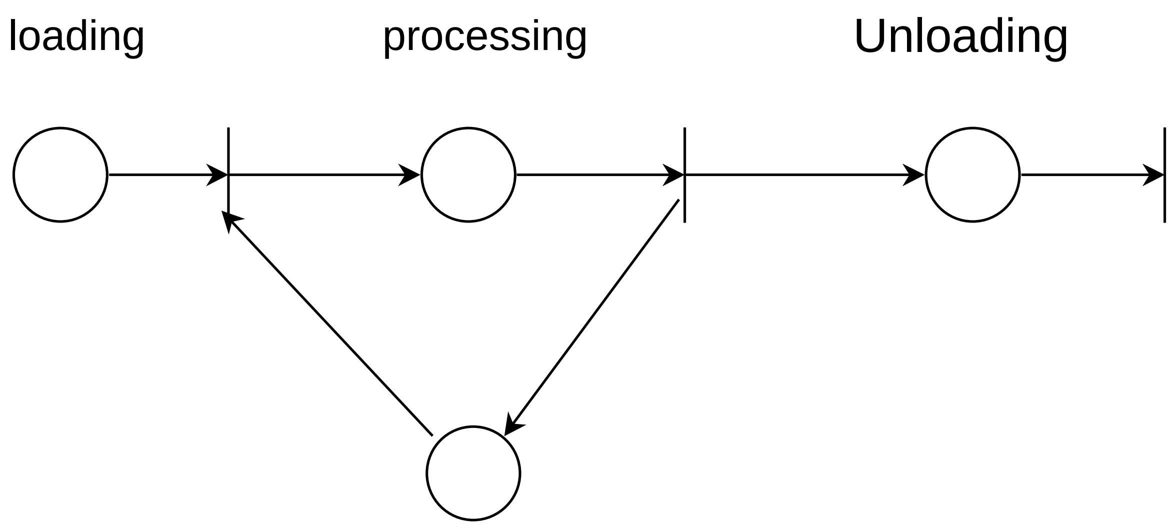
Cycles



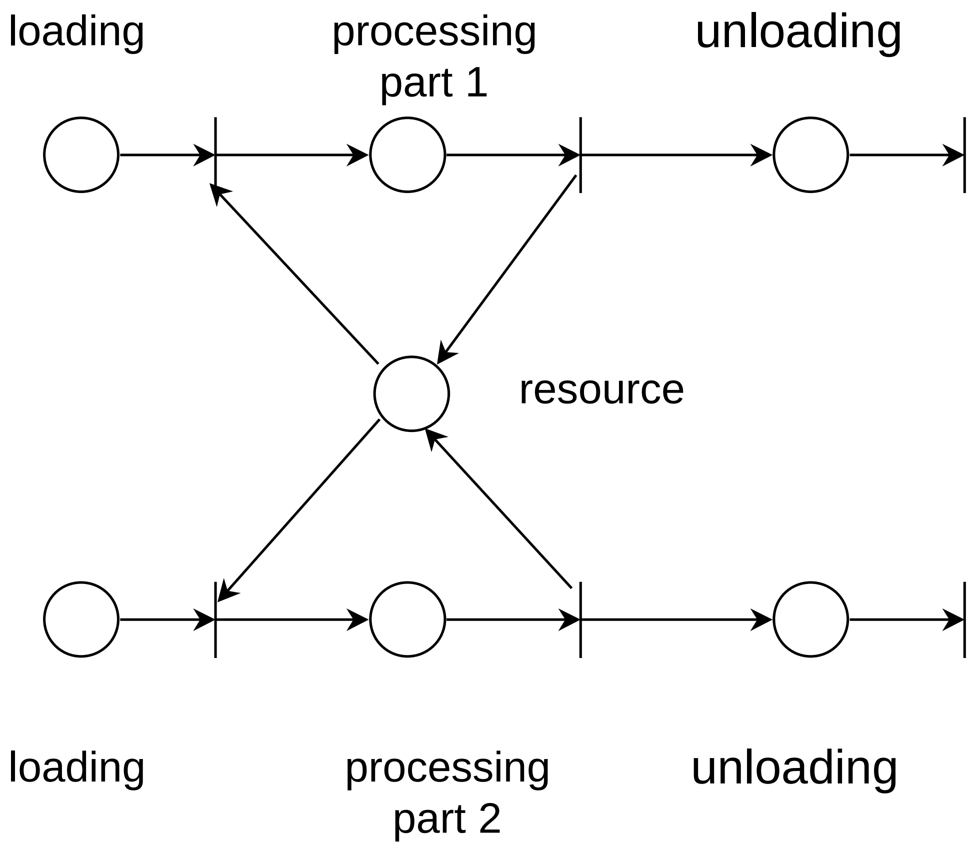


Synchronization

Machine can process one part at once.



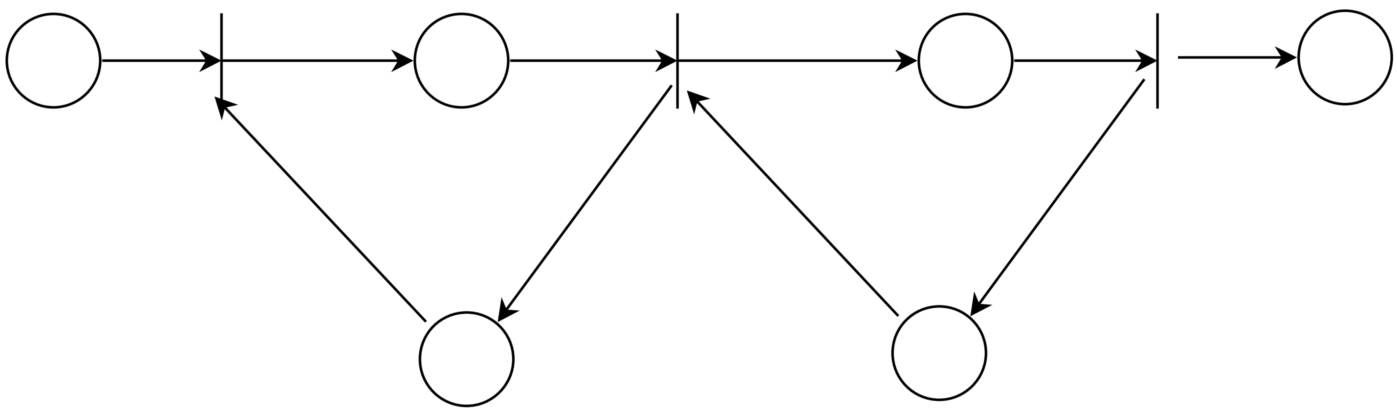
Resource Sharing



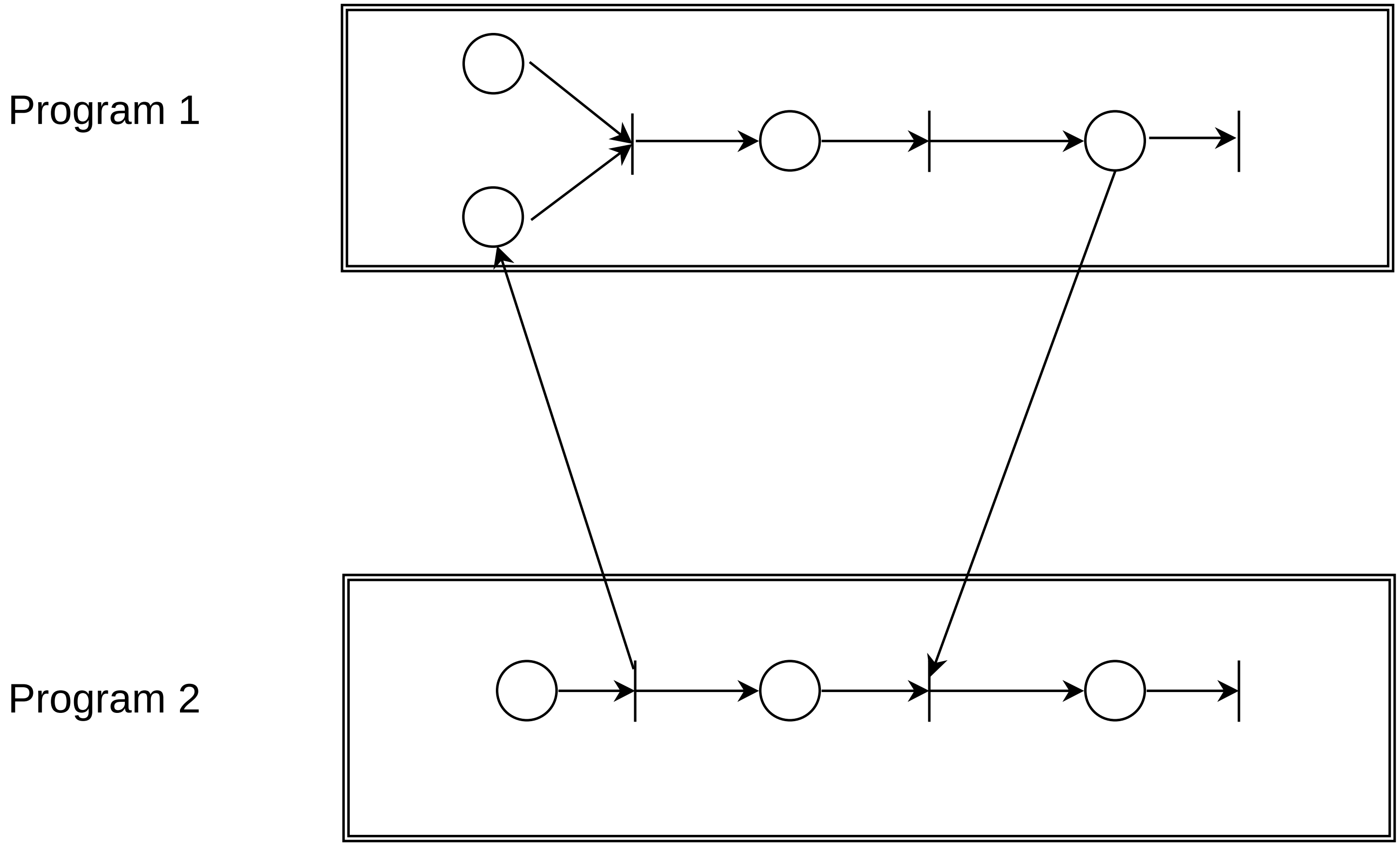
One worker for two machines.
The worker can work at one machine at a time.

Buffer (Queue)

The buffer can hold a limited number of parts.

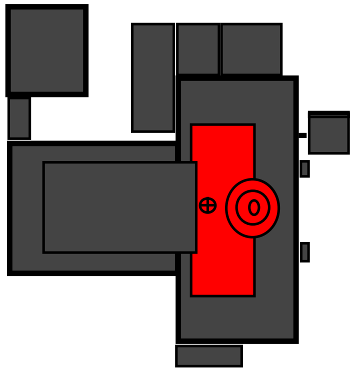


Communication

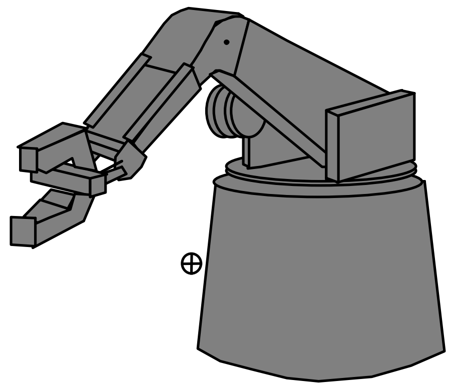


An Example

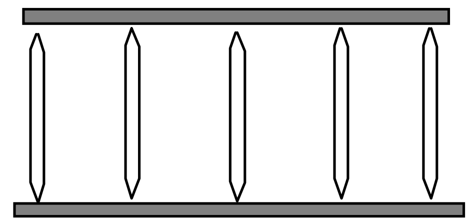
Machine 1



Robot

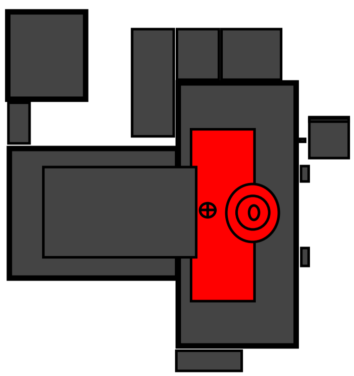


Buffer



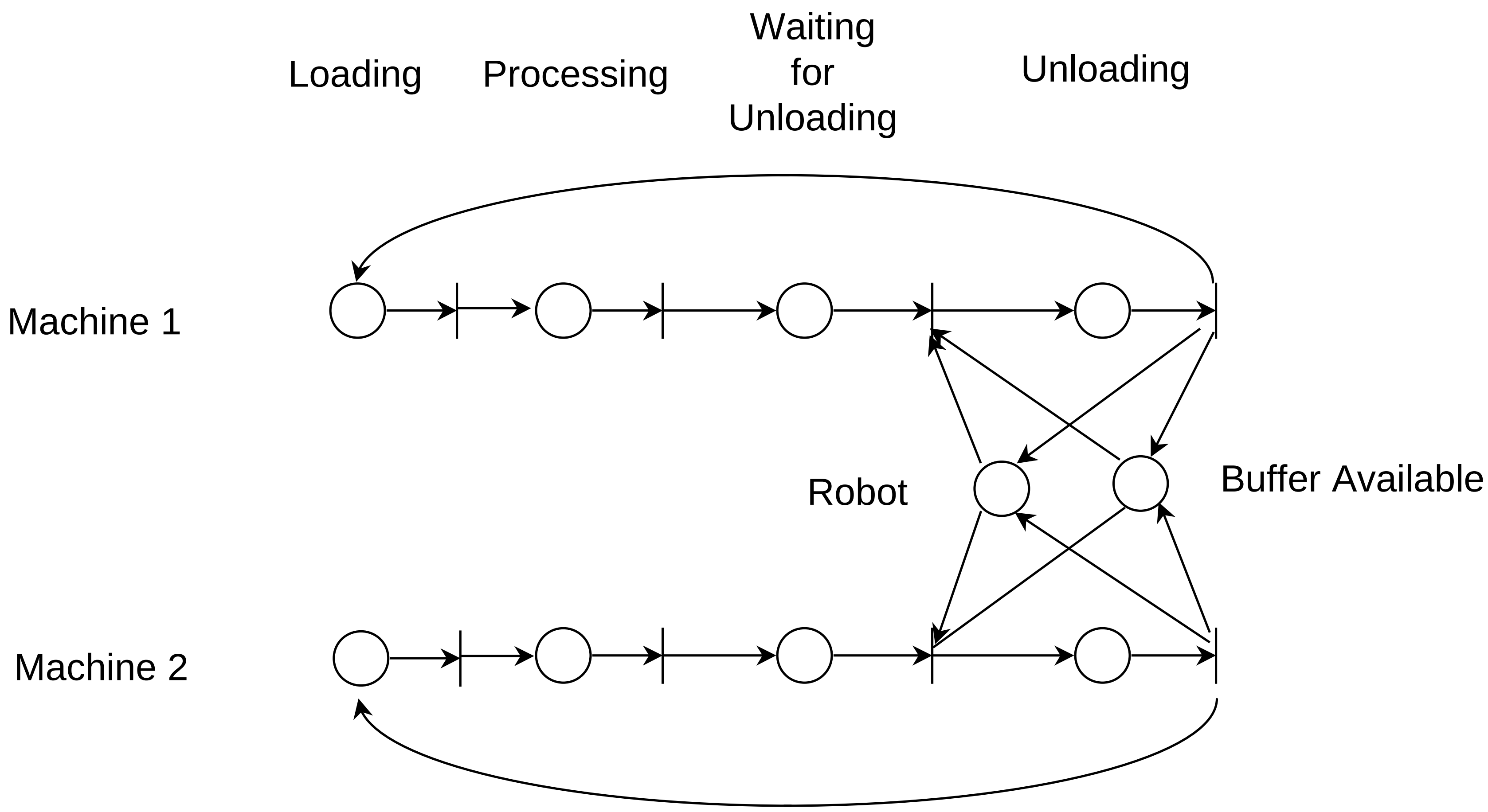
Machine States:
Loading
Processing
Waiting for unloading
Unloading

Buffer State:
Space availability



Machine 2

Put It Together



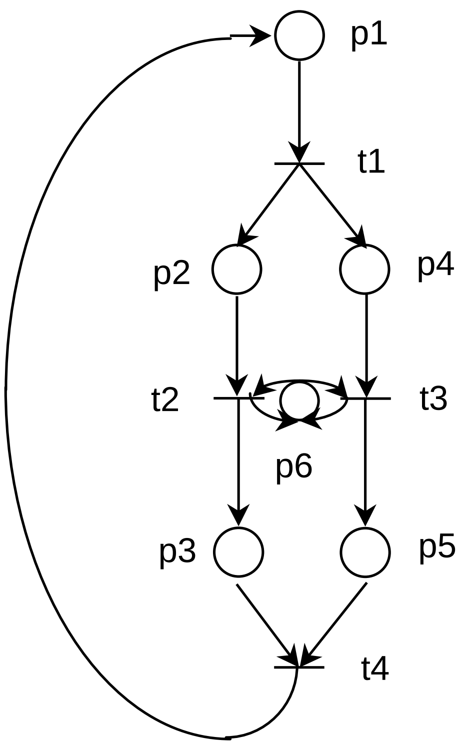
Properties (Questions)

Property	Example
Boundedness - the number of tokens in a place is bounded	Work-in-process
Safeness - the number of tokens in a place never exceeds one	Hardware devices
Deadlock-free - none of markings in $R(PN, M^0)$ is a deadlock	Resources competing
Reachability - find $R(PN, M^0)$	Messages delivery

Analysis Methods

- Enumeration
  - » Reachability Tree
  - » Coverability Tree
- Linear Algebraic Technique
  - » State Matrix Equation
  - » Invariant Analysis: P-Invariant and T-invariant
- Simulation

Reachability Tree (1)



Initialization: M0=(1,0,0,0,0,1)

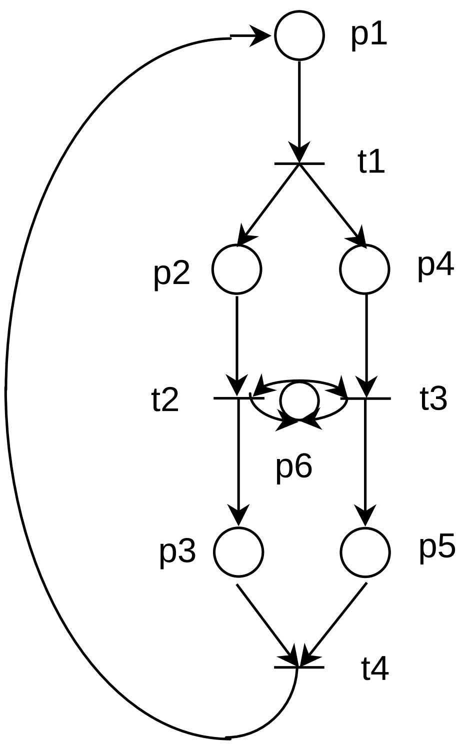
Step 1:

M0 = (1,0,0,0,0,1)
$\downarrow$ t1
M1 = (0,1,0,1,0,1)

Step 2:

M0 = (1,0,0,0,0,1)
$\downarrow$ t1
M1 = (0,1,0,1,0,1)
t2 $\swarrow$ $\searrow$ t3
M2 = (0,0,1,1,0,1)   M4 = (0,1,0,0,1,1)

Reachability Tree (2)



Step 3:

M0 = (1,0,0,0,0,1)

$\downarrow$ t1

M1 = (0,1,0,1,0,1)

t2 t3

M2 = (0,0,1,1,0,1) M4 = (0,1,0,0,1,1)

t3

M3 = (0,0,1,0,1,1)

Step 4:

M0 = (1,0,0,0,0,1)

$\downarrow$ t1

M1 = (0,1,0,1,0,1)

t2 t3

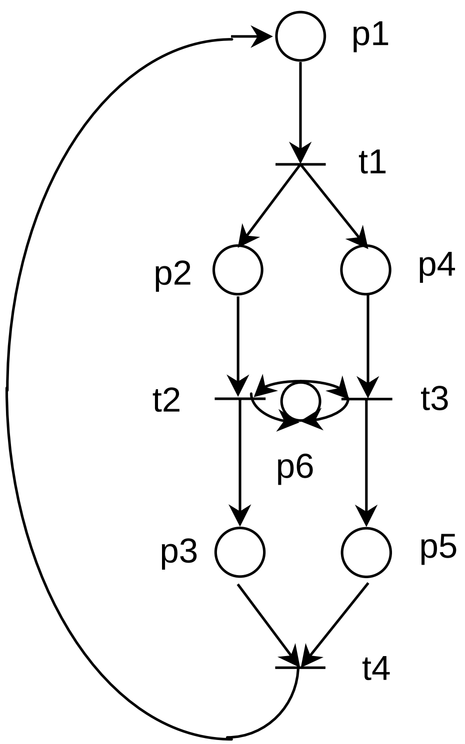
M2 = (0,0,1,1,0,1) M4 = (0,1,0,0,1,1)

t3 t2

M3 = (0,0,1,0,1,1) M3

Reachability Tree (3)

Step 5:



M0 = (1,0,0,0,0,1)

↓ t1

M1 = (0,1,0,1,0,1)

t2 ↙ ↘ t3

M2 = (0,0,1,1,0,1)   M4 = (0,1,0,0,1,1)

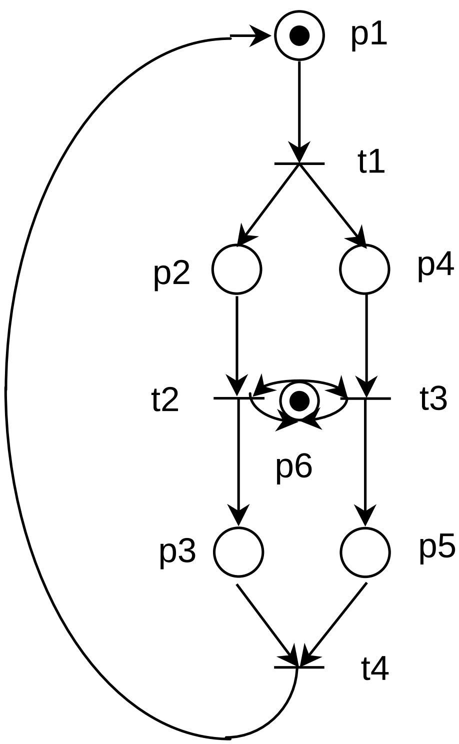
t3 ↓   ↓ t2

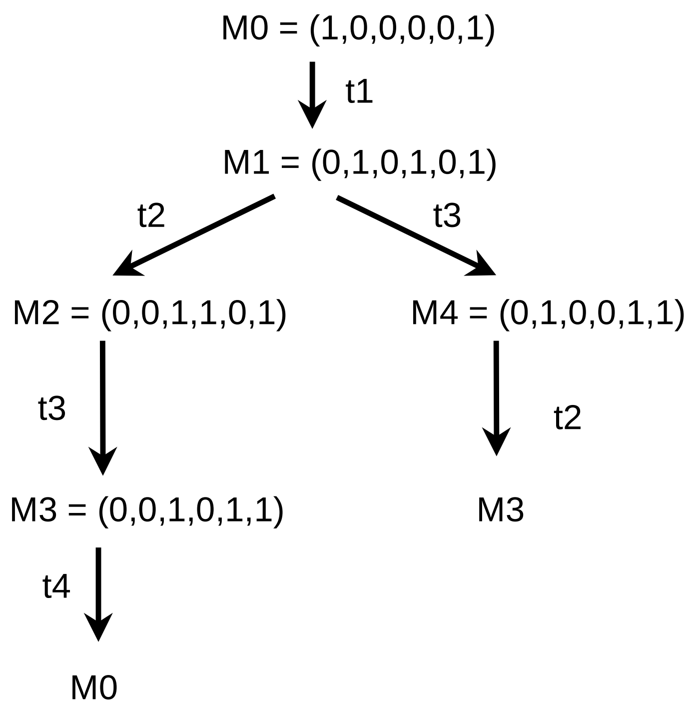
M3 = (0,0,1,0,1,1)   M3

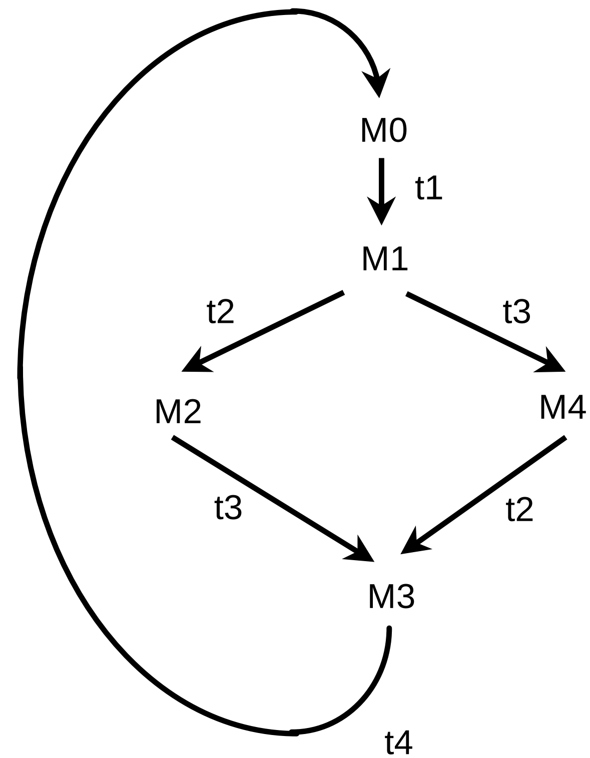
t4 ↓

M0

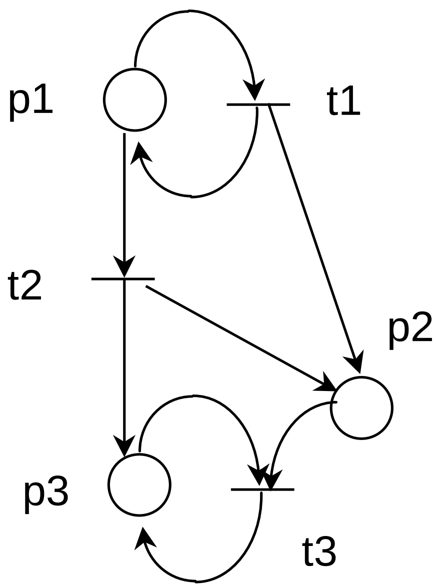
Reachability Tree/Graph

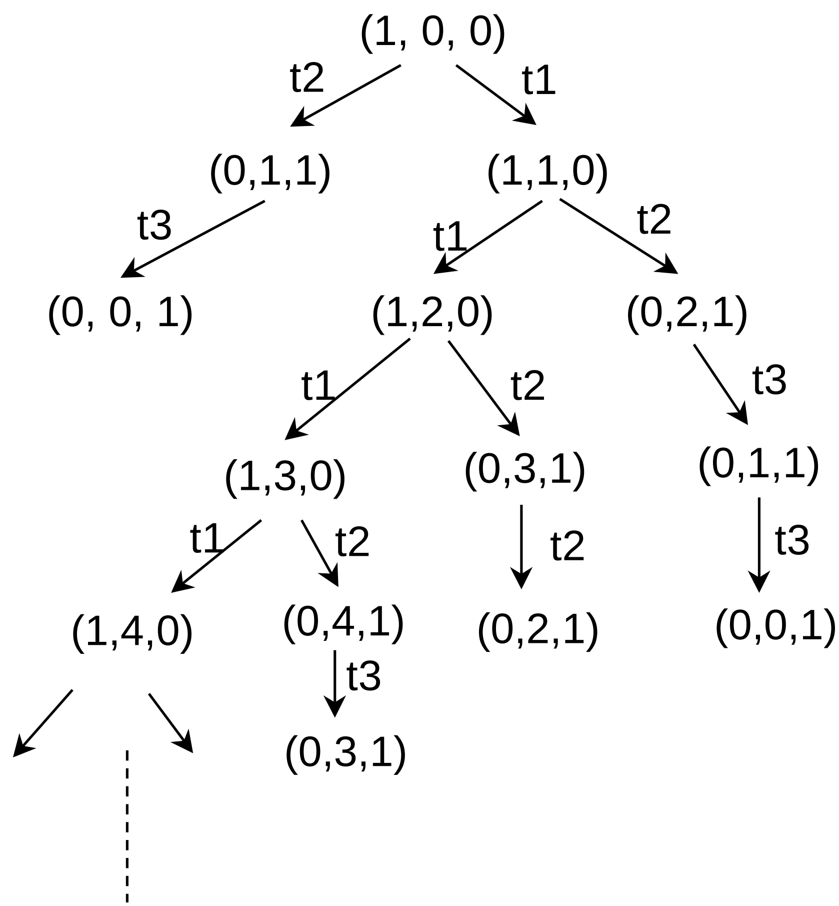




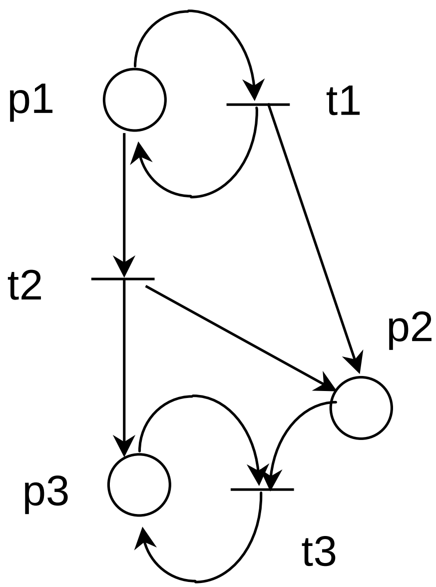


Reachability Tree





Coverability Tree (1)



Initialization: M0 = (1,0,0) new

Step 1:

(1,0,0) new

t1 → new m1 = (1,ω,0)

t2 → m2 = (0,1,1) new

m1 = (1,1,0) >= (1,0,0)

Step 2 (m1):

(1,0,0) new

t1 → new m1 = (1,ω ,0)

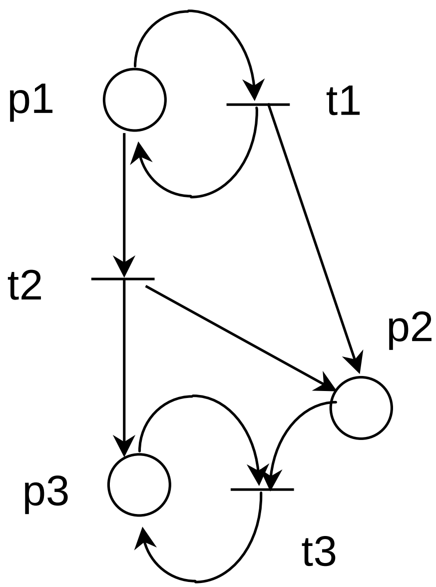
t2 → m2 = (0,1,1) new

new m1 = (1,ω ,0):

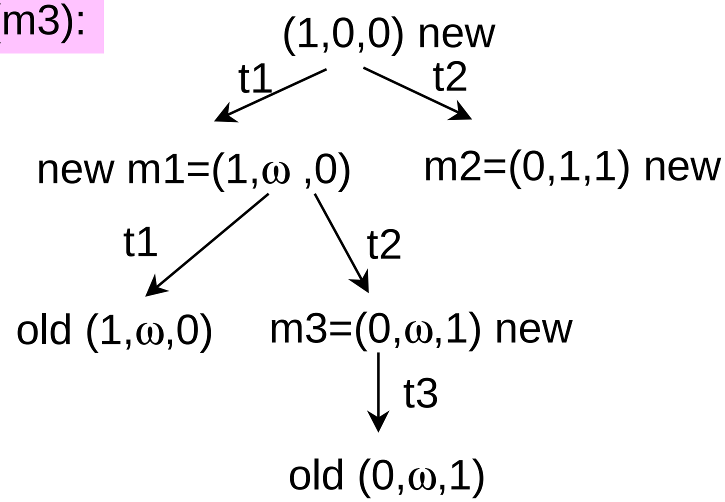
t1 → old (1,ω,0)

t2 → new m3=(0,ω,1)

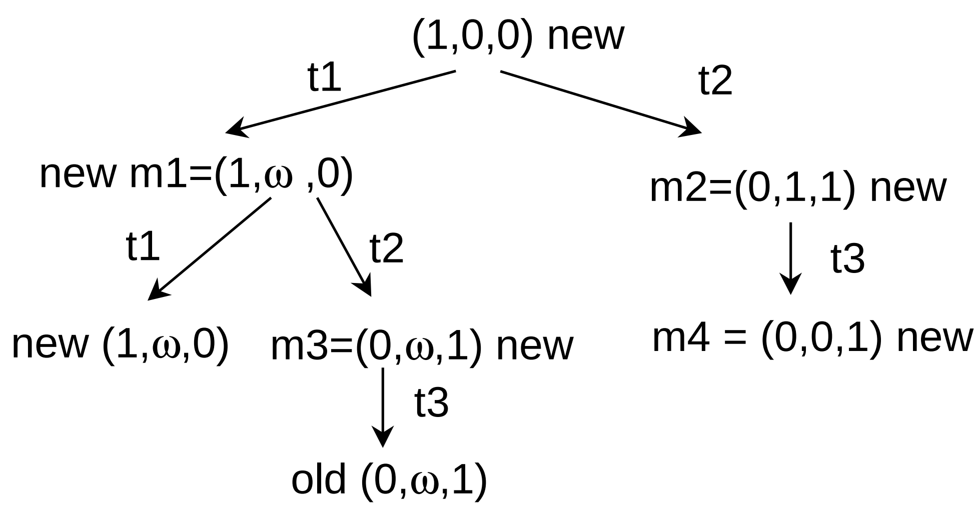
Coverability Tree (2)



Step 3 (m3):

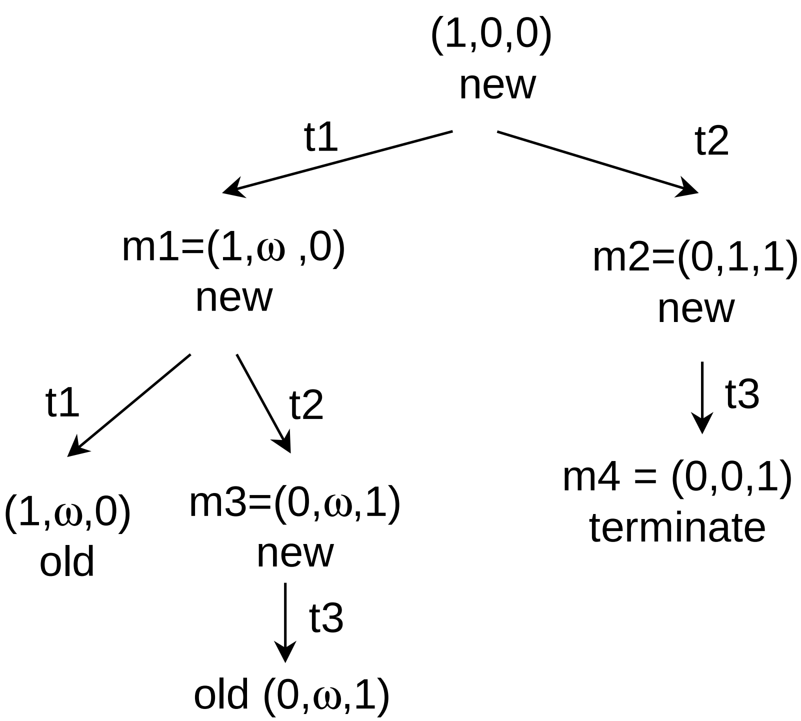


Step 4 (m2):

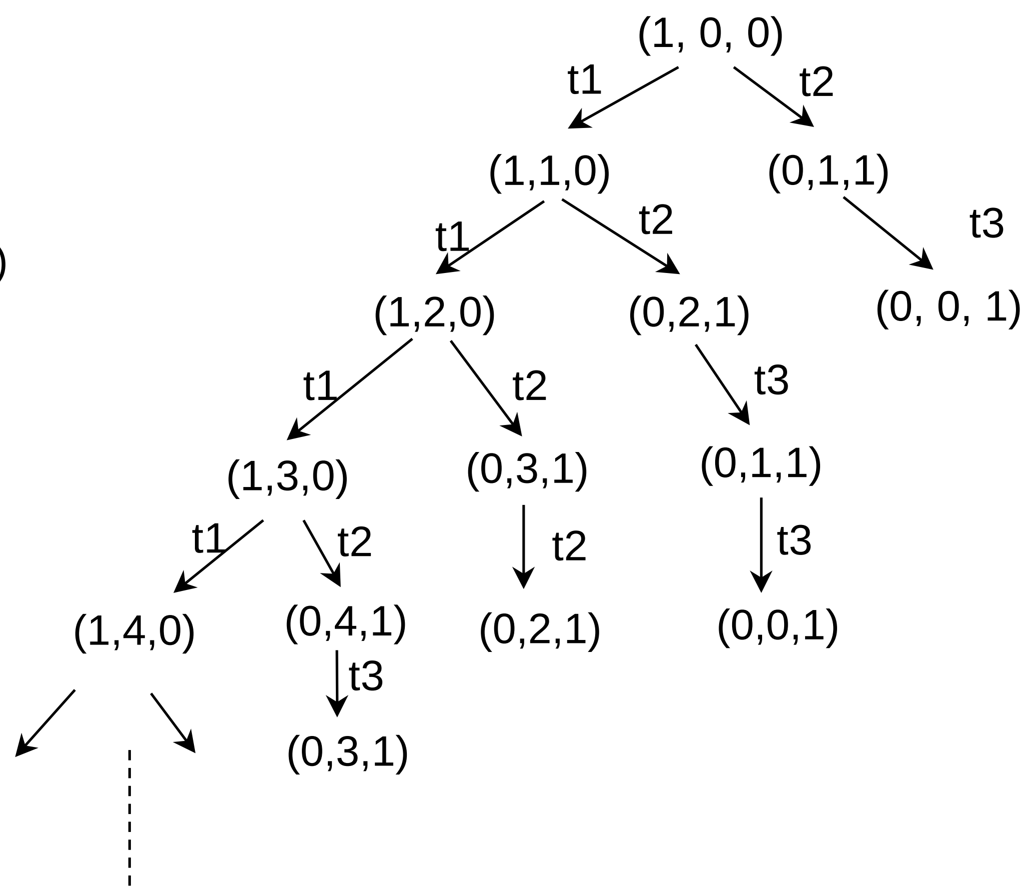


Coverability Tree (3)

Step 5 (m4): Coverability Tree

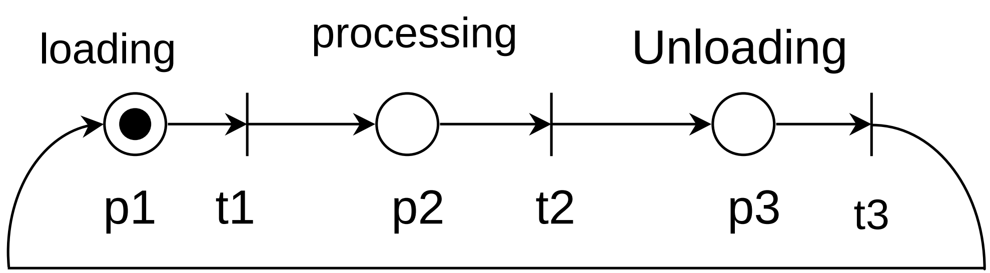


Reachability Tree



Linear Algebraic Technique

State Equation: $M = M^0 + \mu A$, where μ is a vector with s elements



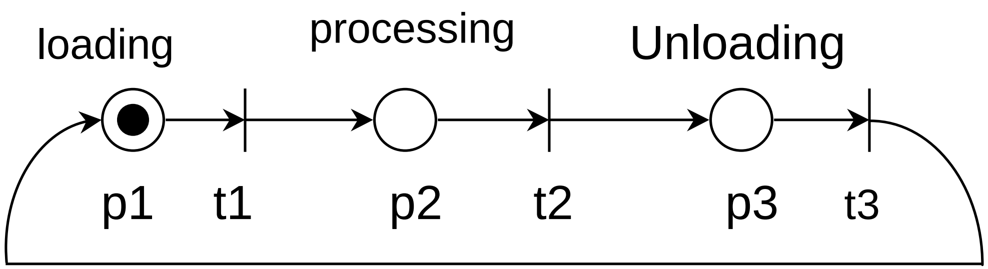
- $O = \begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$
- $I = \begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$
- $M^0 = (1, 0, 0)$

Incidence Matrix

- $A = O - I$

$= \begin{array}{c} \\ t1 \\ t2 \\ t3 \end{array} \begin{array}{c} \begin{array}{ccc} p1 & p2 & p3 \end{array} \\ \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \end{array}$

Linear Algebraic Technique



t1 fired

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

t1, t2 fired

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

t1 t2 t3 fired

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

T-Invariant

T-Invariant: $YA = 0$, where Y is a s element vector
Y is the number of transition firings

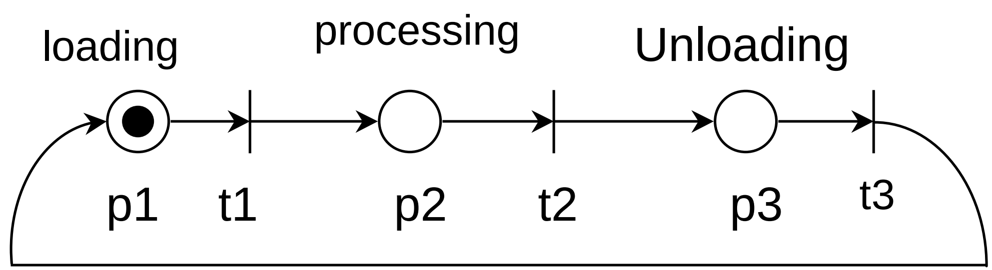
$$\begin{bmatrix} y1 & y2 & y3 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} = 0$$

$$\begin{aligned} -y1 \quad\quad + y3 &= 0 \\ y1 - y2 \quad\quad &= 0 \\ y2 - y3 &= 0 \end{aligned}$$

$$y1 = y2 = y3 \quad \Longrightarrow \quad M^0 \xrightarrow{y_q} M^0$$

minimum t-invariant = (1, 1, 1)

T-Invariant



(1,0,0)

↓

(0,1,0)

↓

(0,0,1)

↓

(1,0,0)

$$(1,0,0) \xrightarrow{(1,1,1)} (1,0,0)$$

P-Invariant

P-Invariant: $AX^T = 0$, where X is a n element vector,
X is the weight of each place

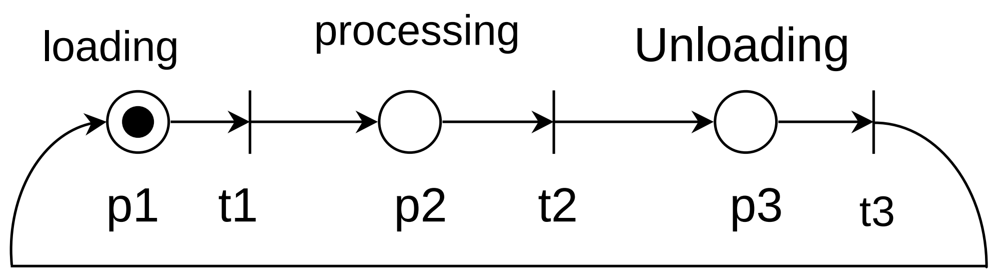
$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = 0$$

$$\begin{aligned} -x1 + x2 \quad &= 0 \\ -x2 + x3 &= 0 \\ x1 \quad - x3 &= 0 \end{aligned}$$

$x1 = x2 = x3$ $\longrightarrow$ The quantity S = x1 M(p1) + x2 M(p2) + x3 M(p3)

minimum p-invariant = (1, 1, 1)

P-Invariant



(1,0,0)
↓
(0,1,0)
↓
(0,0,1)
↓
(1,0,0)

The quantity S =
x1 M(p1) + x2 M(p2) + x3 M(p3)

1 = 1 M(p1) + 1 M(p2) + 1 M(p3)

Simulation

- Discrete event simulation
- Same model for simulation and analysis
- Need rules to resolve conflicts
- Useful for validation and visualization

Extensions of Petri Nets

- Event Graph (marked graph, decision-free)
  - » Each place has exactly one input transition and exactly one output transition
- Deterministic Timed Petri Nets
  - » Deterministic time delays with transitions
- Stochastic Timed Petri Nets
  - » Stochastic time delays with transitions
- Color Petri Nets
  - » Tokens with different colors
- Hybrid Nets
  - » Combine object-oriented concept into Petri nets

Further Readings

- Petri nets home page: http://www.daimi.aau.dk/%7Epetrinet/
- Petri nets mailing list: PetriNets@daimi.aau.dk
- Coloured Petri nets: http://www.daimi.aau.dk/designCPN/
- Petri nets standard: http://www.daimi.aau.dk/%7Epetrinet/standard/

- *Petri Net Theory and the Modeling of Systems,* by J. L. Peterson, Prentice-Hall, 1981.
- *Petri Nets: An Introduction,* by W. Reisig, Springer-Verlag,1985
- *Petri Nets: a Tool for Design and Management of Manufacturing Systems,* by J.-M. Proth, X. Xie, Wiley, 1996

- Computer Integrated Laboratory(CIM Lab) page: http://www.isr.umd.edu/Labs/CIM/

Summary

- A graphical and mathematical tool
- Applications
- Constructs
- Properties: Boundedness, Safeness, Deadlock-free, liveness, Reachability
- Analysis Techniques:
  - » Reachability trees
  - » Coverability trees
  - » Linear algebraic techniques
  - » Simulation
- Extensions
- Resources