Chapter 3
Petri nets

Learning objectives :

- Introduce Petri nets
- Dynamic behavior modeling of manufacturing systems using PN
- Analysis of Petri net models

Textbook :

J.-M. Proth and X. Xie, Petri nets: a tool for design and management of manufacturing systems, John Wiley & Sons, 1996

C. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Springer, 2007

Plan

- Introduction to Petri nets
- Formal definitions
- Petri net models of manufacturing system
- Elementary classes of Petri nets
- Properties of PN models
- Analysis methods

Introduction to Petri nets

3

A two-product system

- Two types P1 and P2 of products are produced.
- The production of each product requires two operations.
- The first operation is performed by a shared machine.
- The second operation is performed by a dedicated machine.
- There is at most one product of each type loaded in the system at any time.
- When a product finishes, a new product of the same type is dispatched.

To be modelled using an usual process-resource modelling approach.

A two-product system
Process modeling

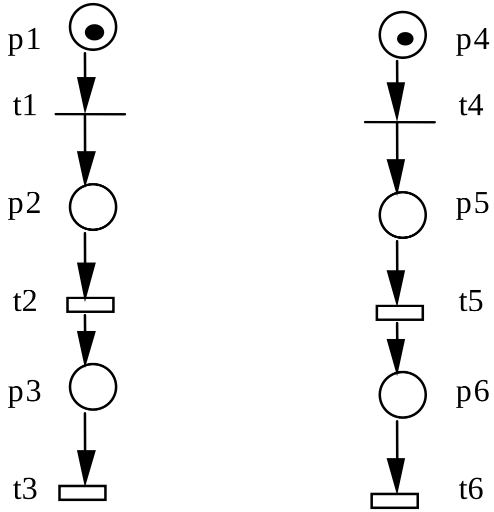
Goal: model the manufacturing process of each product, i.e. all possible states of a product including waiting

- Identify all relevant operations and their precedence constraints.
- Identify all possible waits for shared resources.

wait for shared machine

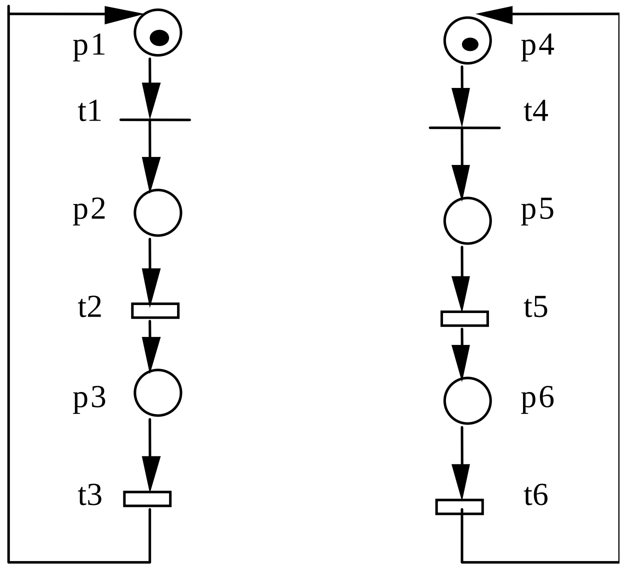
parts under operation 1

parts under operation 2



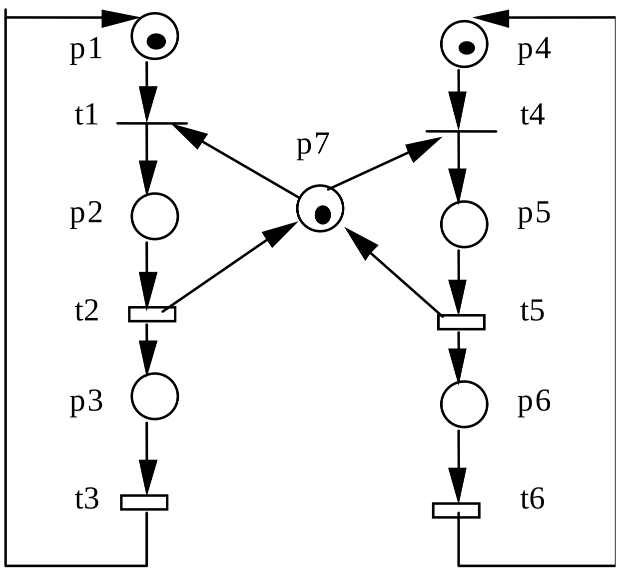
A two-product system
Process modelling

- **Goal**: model the manufacturing process of each product.
- Include eventual constraints related to production control.



A two-product system
Resource modelling

- **Goal**: modelling resource contraint + eventual priority constraints

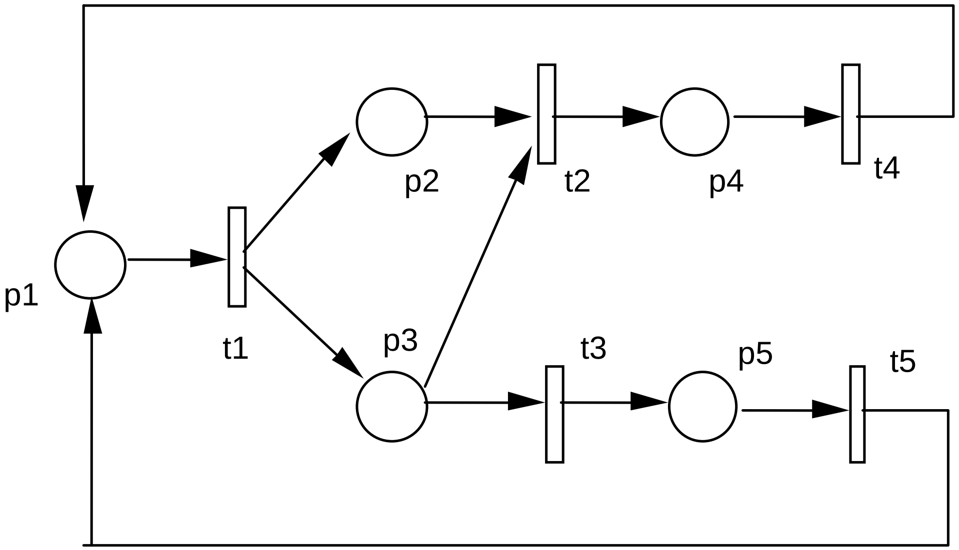


Identifies

- transitions after which the resource is first needed
- transitions after which the resource is no longer needed

Places and transitions

- **A PETRI NET** is a bipartite graph which consists of two types of nodes: **places** and **transitions** connected by directed arcs.
- Place = circle, transition = bar or box.
- An arc connects a place to a transition or a transition to a place.
- No arcs between nodes of the same type.
- **Input and output places** of a transition
- **Input and output transitions** of a place



Token and marking
system state

Each place contains a number of **tokens**.
The distribution of tokens in the Petri net is called the **marking**.

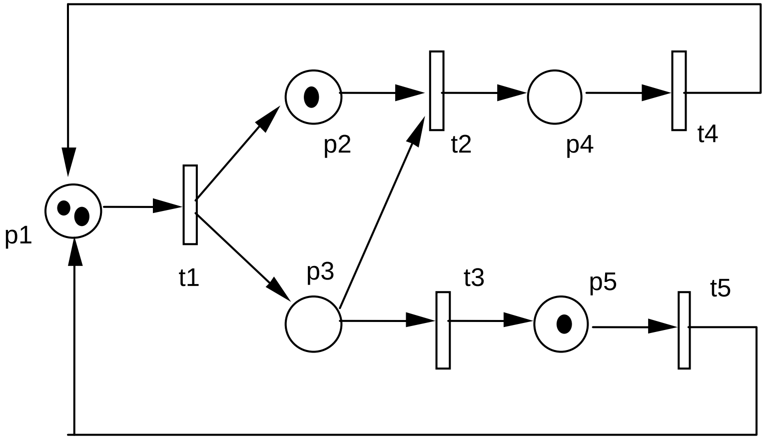
Representations of a marking:

- a vector $M = (m_1, m_2, \ldots, m_n)$ where m_i = nb of tokens in place pi
- a multi-set such as $M = p_1\ 2p_3$

The marking of an PN = state of the corresponding system.

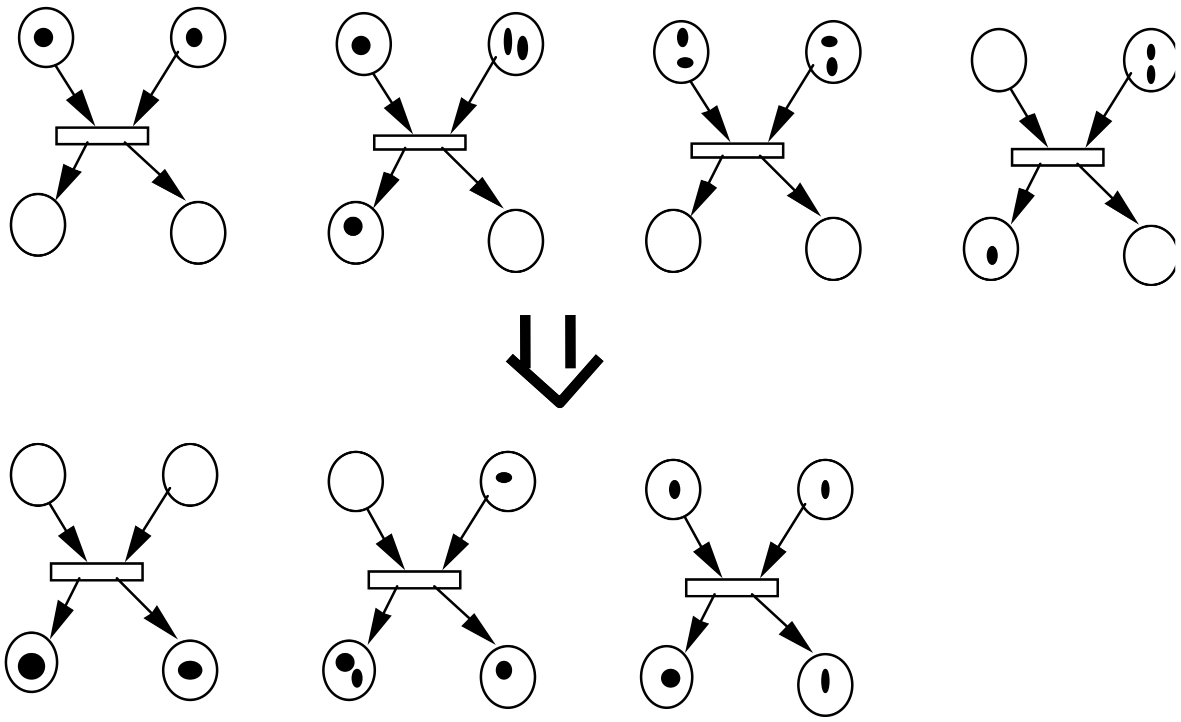
The initial state of the system = the initial marking, denoted as M_0.

Example: M = (???) = ???



System dynamics by transition firing

- A **transition** is said **enabled** (firable) if each of its input places contains at least one token. An enabled transition can fire.
- **Firing a transition** removes a token from each input place and add one token to each ouput place.
- Firing a transition leads to a new marking that enables other transitions.
- The dynamic behavior of the corresponding system = evolution of the marking and transition firings
- Convention: **simultaneous transition firings are forbidden**.



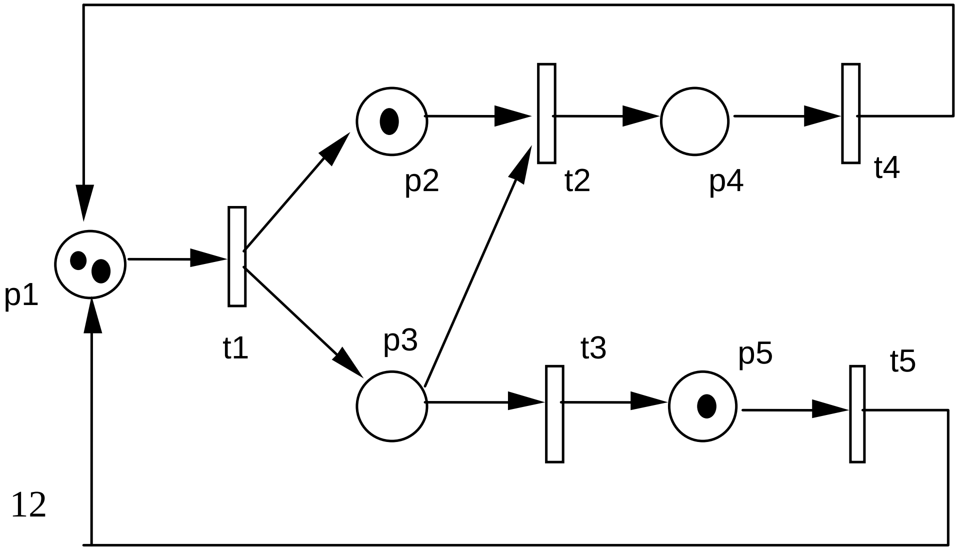
Sequence of transitions

A sequence of transitions that can be fired consecutively starting from the initial marking is said enabled or firable.

The sequence of firable transitions is not unique.

The set of all firable sequences of transitions = PN language

Example: sequence t1t2t1t3



Formal definitions

Petri Nets

A Petri net is a five-tuple $PN = (P, T, A, W, M_0)$ where:

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

$T = \{ t_1, t_2, ..., t_m \}$ is a finite set of transitions

$A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs

$W : A \rightarrow \{ 1, 2, ... \}$ is a weight function

$M_0 : P \rightarrow \{ 0, 1, 2, ... \}$ is the initial marking

$P \cap T = \Phi$ and $P \cap T = \Phi$

PN without the initial marking is denoted by N:

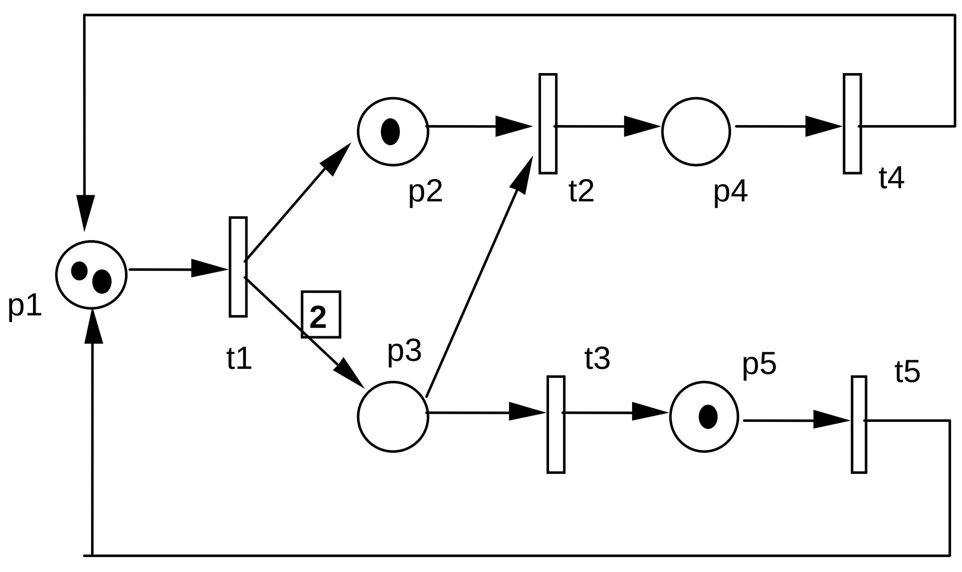
$N = (P, T, A, W)$

$PN = (N, M0)$

A Petri net is said **ordinary** if $w(a) = 1, \forall a \in A$.

Graphic representation

Similar to that of ordinary PN but with default weight of 1 when not explicitly represented.



Transition firing

Rule 1: A **transition** t is **enabled** at a marking M if M (p) $\geq$ w(p, t) for any p $\in$ ^{o}t where ^{o}t is the set of input places of t

Rule 2: An enabled transition may or may not fire.

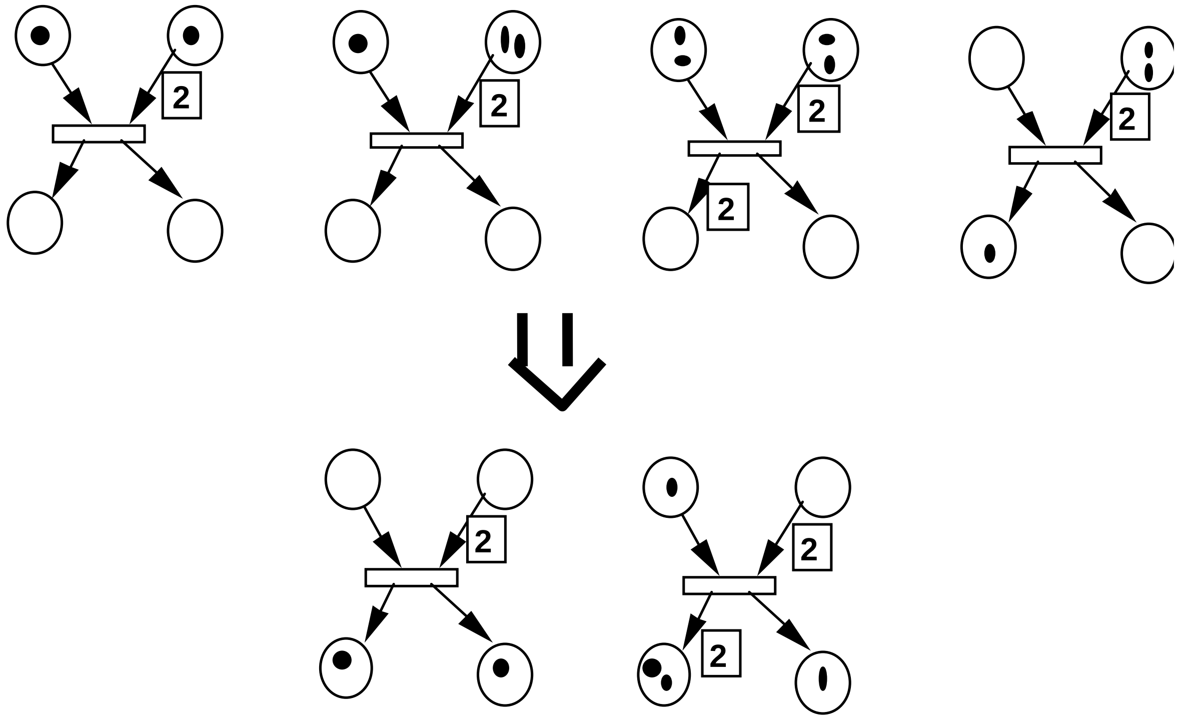
Rule 3: **Firing transition** t results in:

- removing w(p, t) tokens from each p $\in$ ^{o}t
- adding w(t, p) tokens to each p $\in$ t^{o} where t^{o} is the set of output places of t

M(t> M' denotes firing t at marking M with

$$M'(p) = \begin{cases} M(p), & \text{si } (p,t) \notin A \text{ et } (t,p) \notin A, \\ M(p) + W(t,p), & \text{si } (p,t) \notin A \text{ et } (t,p) \in A, \\ M(p) - W(p,t), & \text{si } (p,t) \in A \text{ et } (t,p) \notin A, \\ M(p) + W(t,p) - W(p,t), & \text{si } (p,t) \in A \text{ et } (t,p) \in A, \end{cases}$$

Transition firing



Basic concepts

Source transition: transition without input places, i.e. $^{o}t = \Phi$.

Sink transition: transition without output places, i.e. $t^{o} = \Phi$.

Source place: place without input transitions, i.e. $^{o}p = \Phi$.

Sink place: place without output transitions, i.e. $p^{o} = \Phi$.

Self-loop: a couple (p, t) such that t is both input and output transition of p

Path: a sequence of nodes $\mathbf{s_1 s_2 \ldots s_n}$ such that $\mathbf{s_{i+1}}$ is an output node of $\mathbf{s_i}$.

Circuit: a path such that $\mathbf{s_n} = s_1$.

Online illustration

Incidence matrices

Pre incidence matrix:

$$\mathrm{Pre}(p,t) = \begin{cases} w(p,t), & \text{if } p \in \circ t \\ 0, & \text{otherwise} \end{cases}$$

Post incidence matrix:

$$\mathrm{Post}(p,t) = \begin{cases} w(t,p), & \text{if } p \in t \circ \\ 0, & \text{otherwise} \end{cases}$$

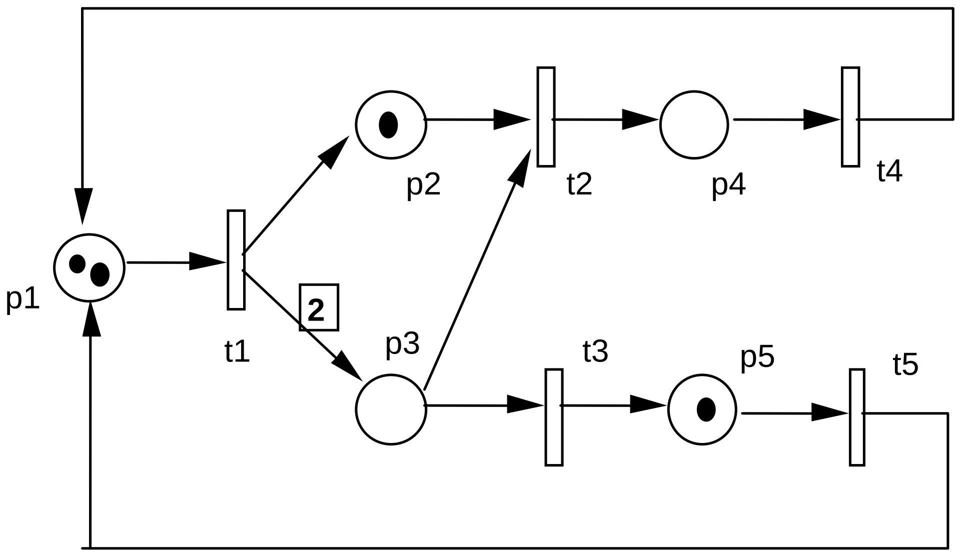
Incidence matrix : C = Post – Pre.

- C(., t) = Token flow balance after firing t
- Pre and Post define the Petri net
- For Petri nets without self-loops, i.e. $^{o}t \cap t^{o} = \Phi$, C defines the Petri net with Pre(p,t) = max{0, C(p,t)} and Post(p,t) = max{0, C(p,t)}

Incidence matrices

Example:

Pre = ???, Post = ???, C = ???



Incidence matrices

Enabled transition: A transition t is enabled at a marking M if

$$M \geq Pre(\bullet, t)$$

Transition firing: Firing a transition t at marking M leads to

$$M' = M + C(\bullet, t)$$

Sequence of transitions: Firing a sequence $\sigma = t_1 t_2 \ldots t_n$ of transition starting from marking M leads to:

$$M' = M + C\vec{\sigma} \qquad (1)$$

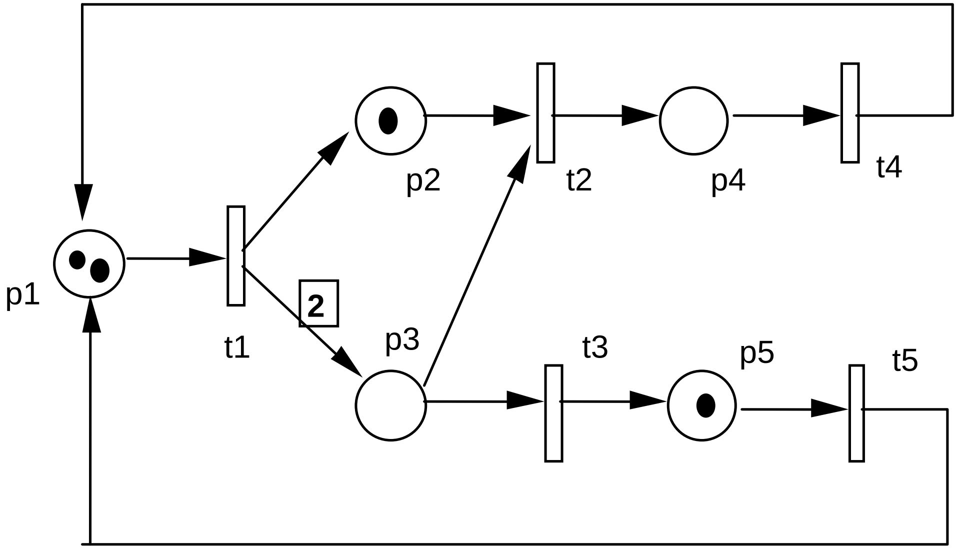
where $\vec{\sigma}$ is the counting vector of the sequence σ. (proof) Equation (1) is also called « **state equation** ».

Question: can this equation be used to checked the feasibility of a sequence and the reachability of a marking?

Incidence matrices

Example:

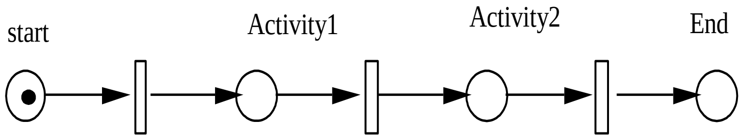
Markings after σ = t1t5t2t3t5



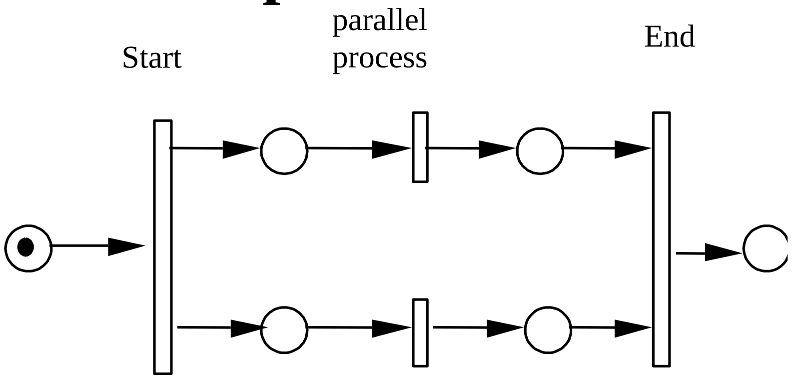
Petri net models of manufacturing systems

PN models of key characteristics

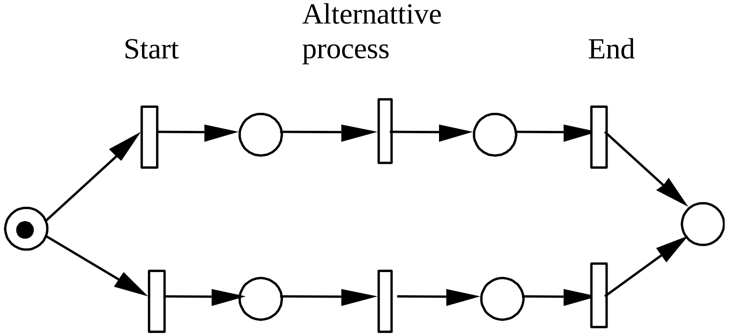
Precedence relation:



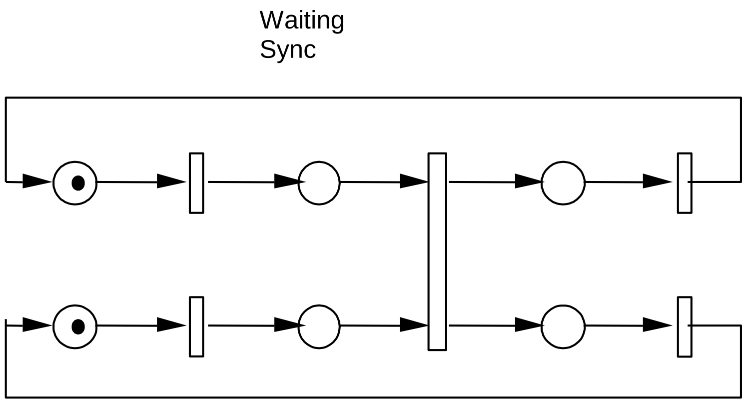
Parallel processes:



Alternative processes:

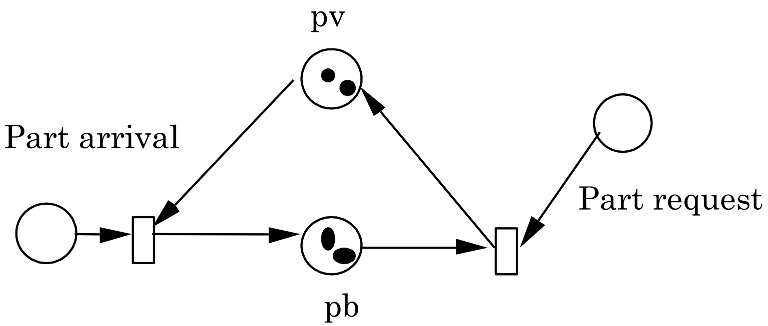


Synchronization:

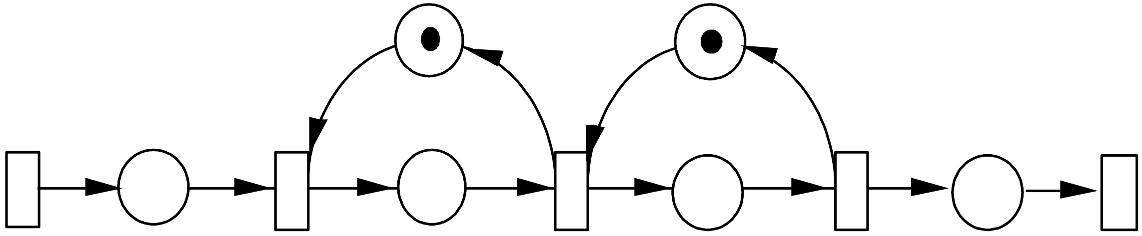


PN models of key characteristics

Buffer of finite capacity (4):

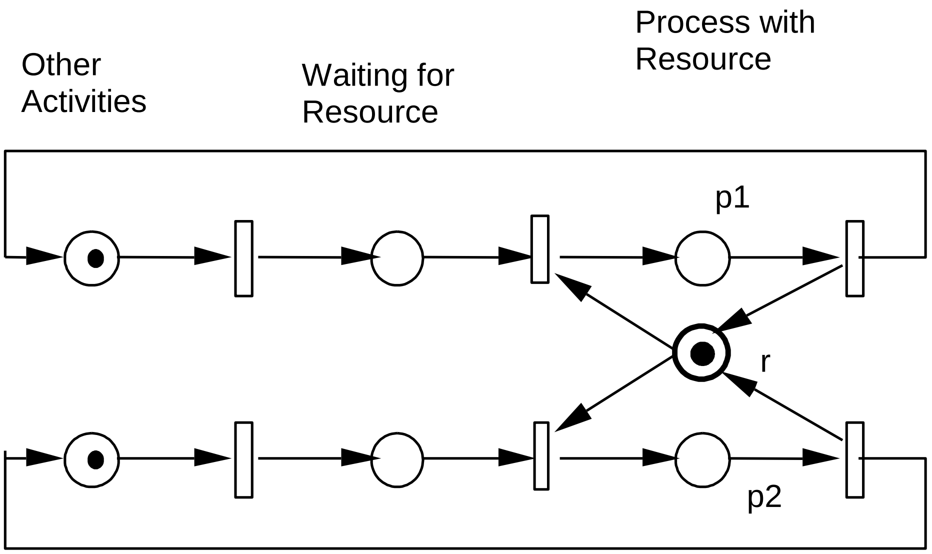


FIFO system:



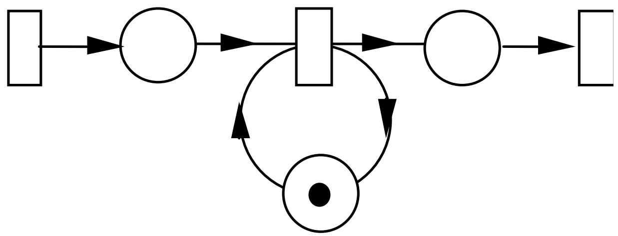
PN models of key characteristics

Shared resources:

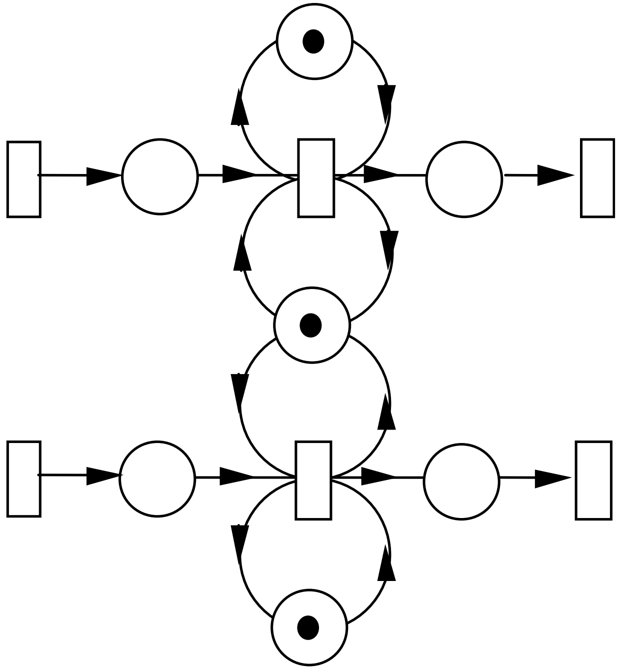


PN models of key characteristics

Dedicated machine:

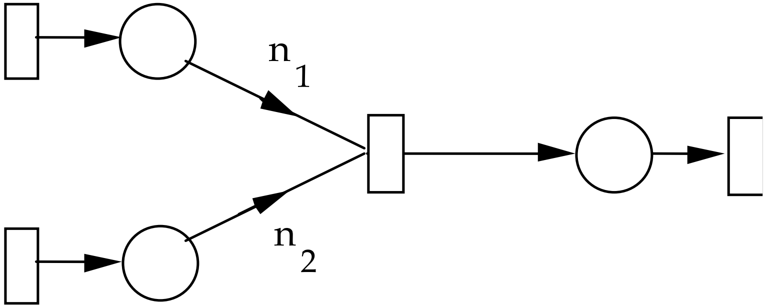


Shared machine:

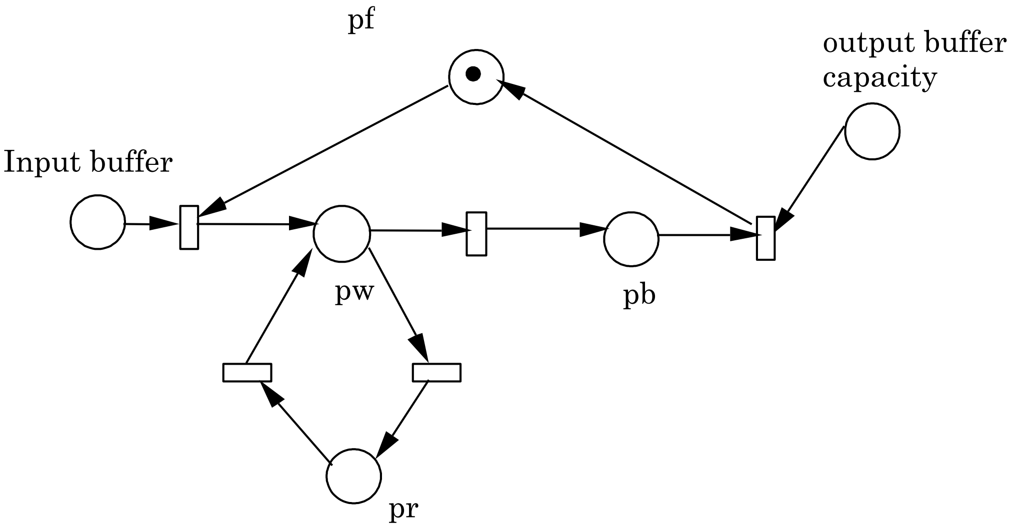


PN models of key characteristics

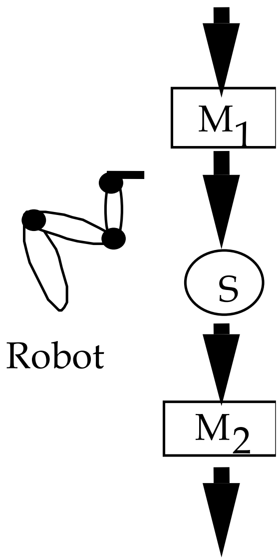
Assembly operation:

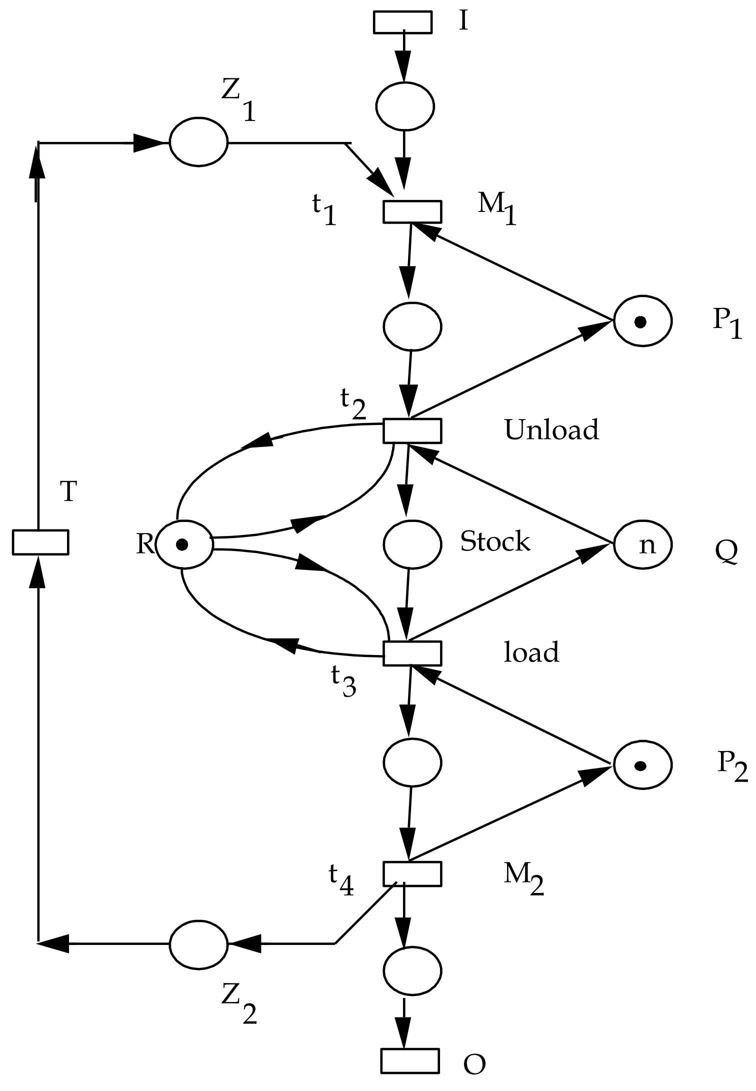


Unreliable machines:



A robotic cell





A two-product system

- Two types P1 and P2 of products are produced.
- The production of each product requires two operations.
- The first operation is performed by a shared machine.
- The second operation is performed by a dedicated machine.
- There is at most one product of each type loaded in the system at any time.
- When a product finishes, a new product of the same type is dispatched.

To be modelled using an usual process-resource modelling approach.

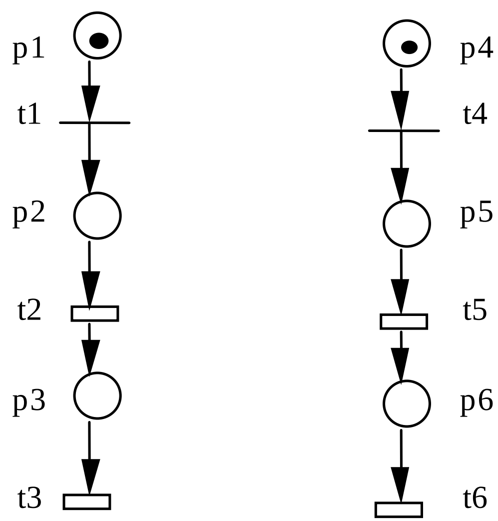
Process modeling

- **Goal**: model the manufacturing process of each product.
- Identify all relevant operations and their precedence constraints.
- Identify all possible waits for shared resources.

wait for shared machine

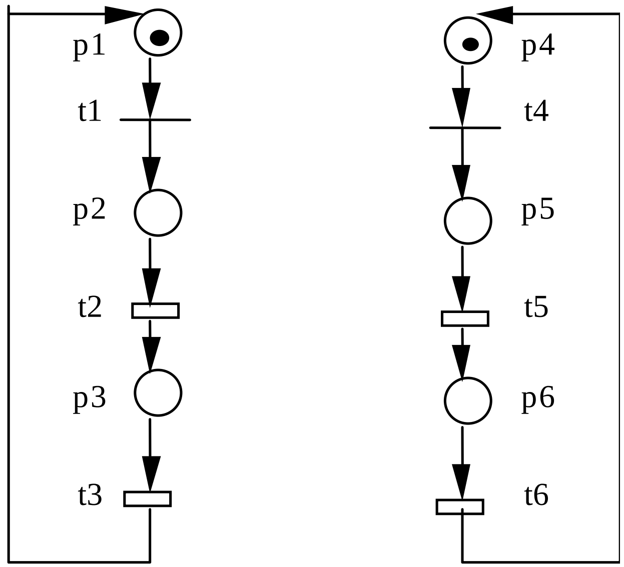
parts under operation 1

parts under operation 2



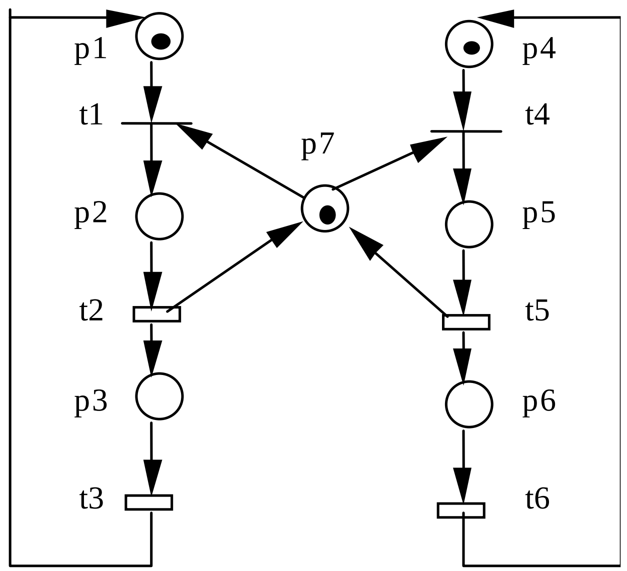
Process modelling

- **Goal**: model the manufacturing process of each product.
- Include eventual constraints related to production control.



Resource modelling

- **Goal**: modelling resource contraint.



Identifies

- transitions after which the resource is first needed

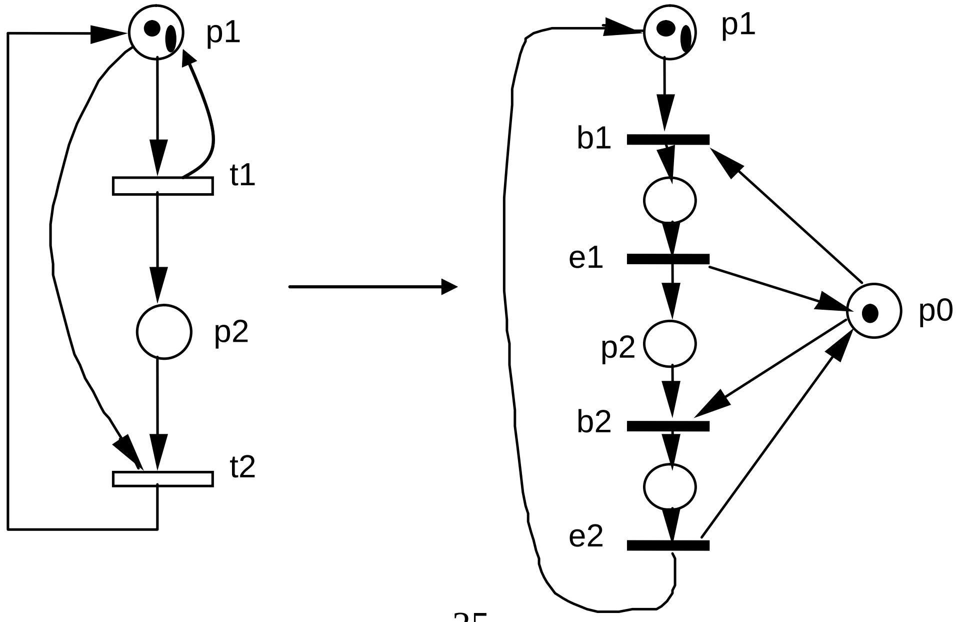
- transitions after which the resource is no longer needed

Elementary classes of Petri nets

Pure Petri nets

Definition: A Petri net free of self loop is said pure, i.e. ${}^{o}t \cap t^{o} = \Phi$.

Theorem : All impure Petri nets can be transformed into pure Petri nets.



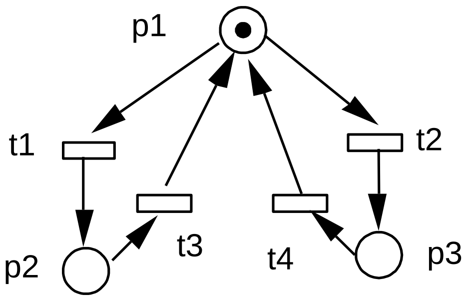
Ordinary Petri nets

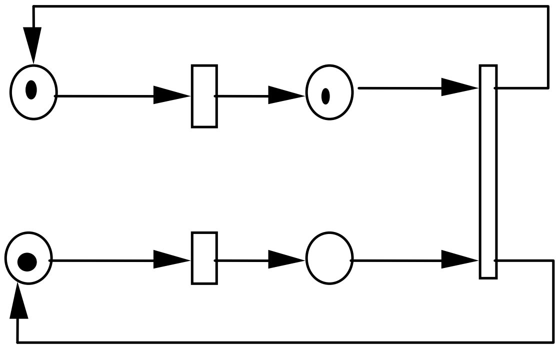
STATE MACHINES
Each transition has exactly one input place and one output place.

Property: The total number of token is constant.

EVENT GRAPHS (OR MARKED GRAPHS)
Each place has exactly one input and one output transition.

Property: The total number of tokens in each elementary circuit is constant

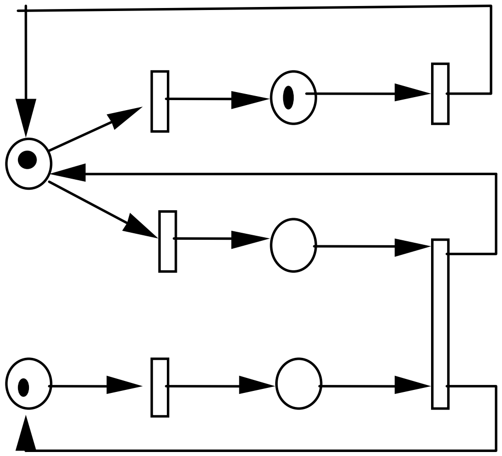




Ordinary Petri nets

FREE-CHOICE NETS

$\text{card}(p°) > 1 \Rightarrow °(p°) = \{p\}, \forall\, p \in P.$

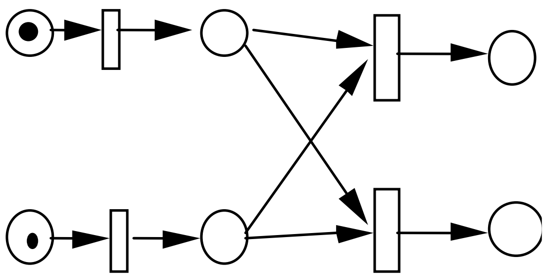


Property : For any free-choice net, a **t'** in conflict with an enalbed transition **t** , i.e. $\bullet t' \cap \bullet t \neq \Phi$, is also enabled.

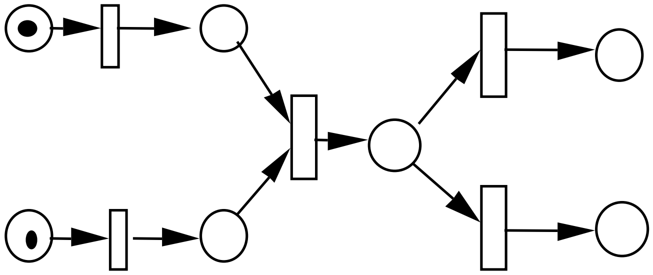
EXTENDED FREE-CHOICE NETS

$p1° \cap p2° \neq \Phi \Rightarrow p1° = p2°, \forall p1, p2 \in P$

An extended free-choice net can always be transformed into a free-choice net.



$\longrightarrow$

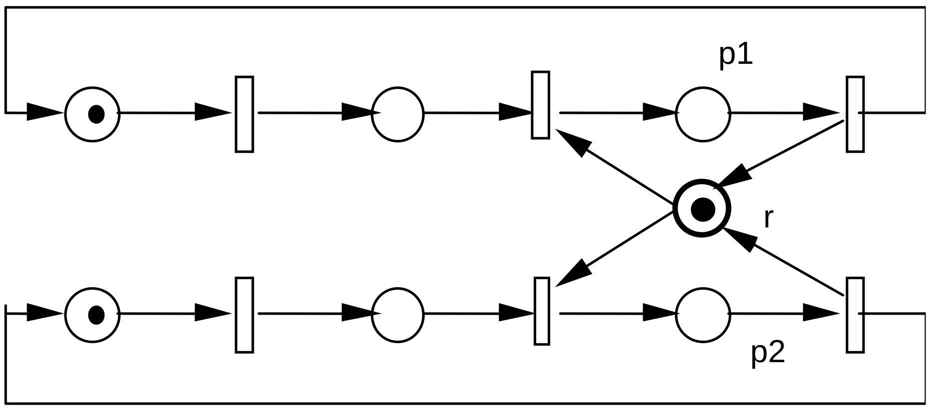


Ordinary Petri nets

ASYMMETRIC CHOICE NETS

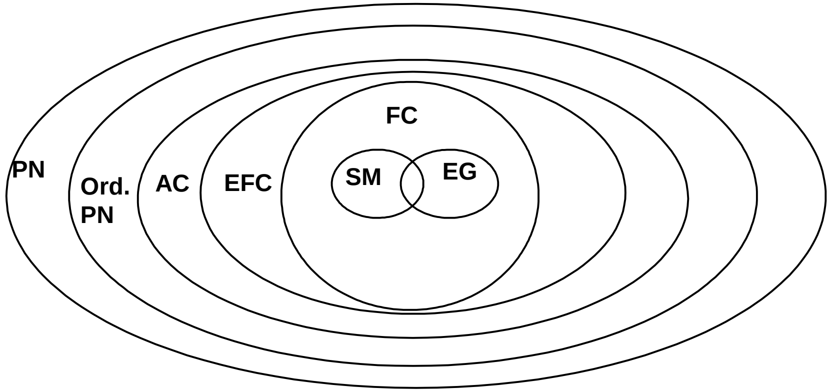
$p1° \cap p2° \neq \Phi \Rightarrow \; p1° \subseteq p2° \text{ or } p2° \subseteq p1°, \forall\, p1, p2 \in P$

Theorem : For any asymmetric choice net, the set {p1, p2, …, pk} of input places of any transition can be renumbered such that $p1° \subseteq p2° \subseteq \ldots \subseteq pk°$.

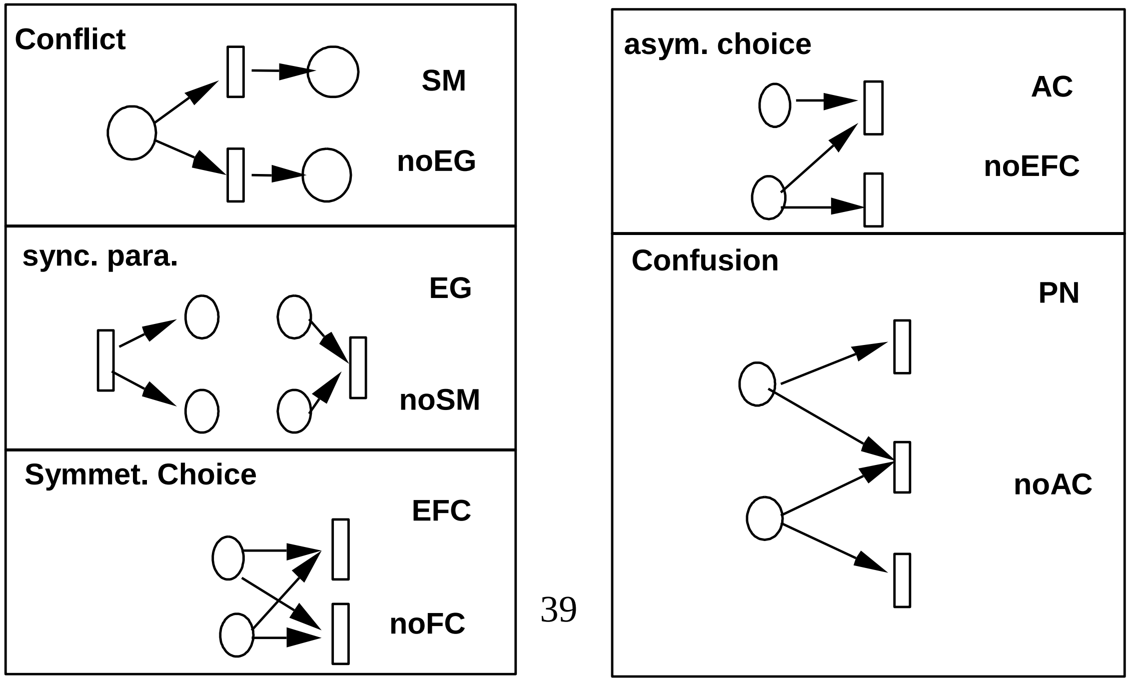


Relations between different classes

PN = Petri Net
AC = Assymmetric choice
EFC = Extended Free Choice
FC = Free Choice
SM = State Machine
EG = Event Graph



Modeling power



Properties of PN models

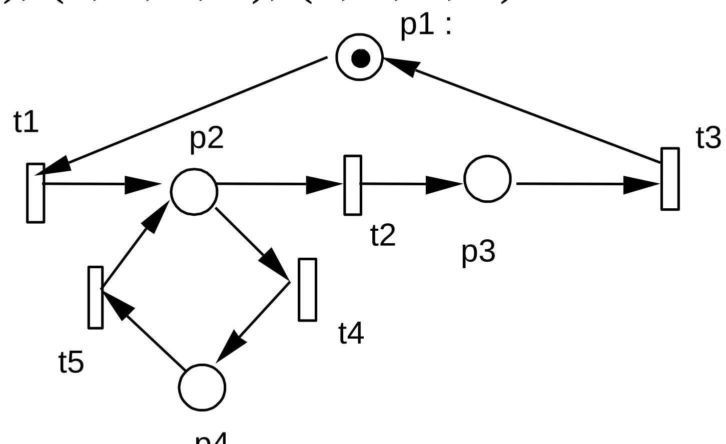
Reachability

A marking M is said reachable from another marking M' if there exists a seqence σ of transitions such that M'(σ >M.

$R(M_0)$ = set of markings reachable from the initial marking M0.

Reachability is important for verification of the reachability of some desired (proper termination) or undesired markings (deadlock).

Example: R(M0) = {(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1)} and (1, 0, 1, 0) not reachable.



Reachability

Theorem1 (monotonicity) : Any sequence s of transitions firable starting from a marking M0 is also firable starting from M0' such that $M0' \geq M_0$.

Theorem2 (necessary condition) : The equation system $\mathbf{CY = M - M_0}$ with $Y \geq 0$ has a solution for all reachable marking M.

Theorem3 (Acyclic PN) : For any PN free of cycles, a marking M is reachable iff the equation system $C\,Y = M - M_0$ with $Y \geq 0$ has a solution.

Ex: Find a PN and a marking that is not reachable but for which condition of Theorem 2 holds.

Boundedness

A place p is said **k-bounded** if the number of tokens in p never exceed k, i.e. $M(p) \leq k$, $\forall M \in R(M0)$.

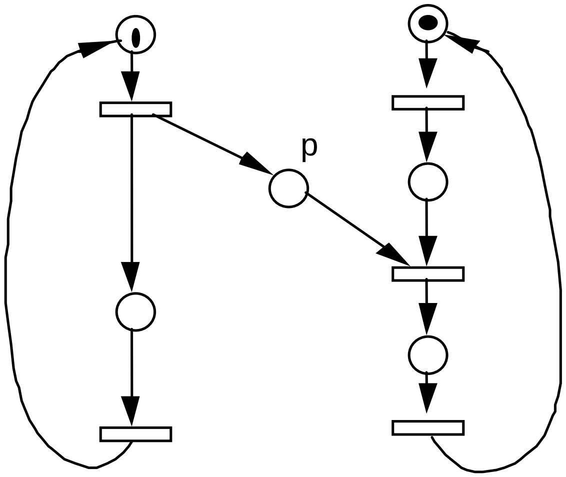
A Petri net is said k-bounded if all places are k-bounded, i.e. $M(p) \leq k$, $\forall p$ and $M \in R(M0)$.

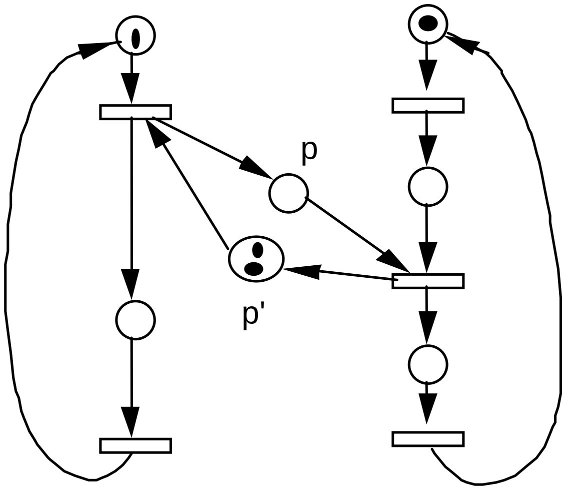
A Petri net is said **bounded** if it is k-bounded for some $k > 0$.

A Petri net is said **safe** if it is 1-bounded, $M(p) \leq 1$, $\forall p$ and $M \in R(M0)$.

Boundedness is often needed for a well-designed system as, without this property, goods could accumulated without limit, which is often a design error.

Boundedness





Boundedness

Theorem (monotonicity) : If (N, M0) is bounded, then (N, M0’) such that M0' $\leq$ M0 is bounded.

Theorem (necessary condition) : A Petri net (N, M0) is k-bounded if M(p) $\leq$ k, $\forall$p and $\forall$M such that M = M0 + CY for some Y $\geq$ 0.

Liveness

A **transition t is said live** if it can always be made enabled starting from any reachable marking, i.e. $\forall M \in R(M0)$, $\exists M' \in R(M)$ such that M‘(t>.

A **Petri net is said live** if all transitions are live.

A **transition is said quasi live** if it can be fired at least once, i.e. $\exists M \in R(M0)$ such that M(t>.

A **Petri net is said quasi live** if all transitions are quasi live.

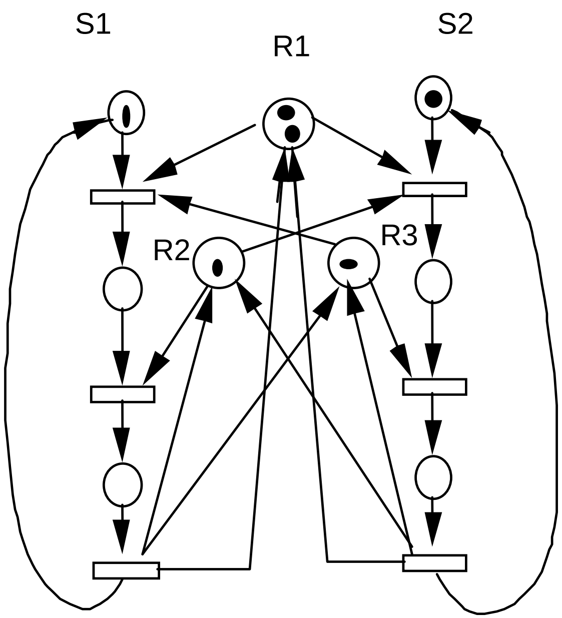
A **marking M is said a deadlock** or dead marking if no transition is enabled at M.

A **Petri net is said deadlock-free** if it does not contain any deadlock.

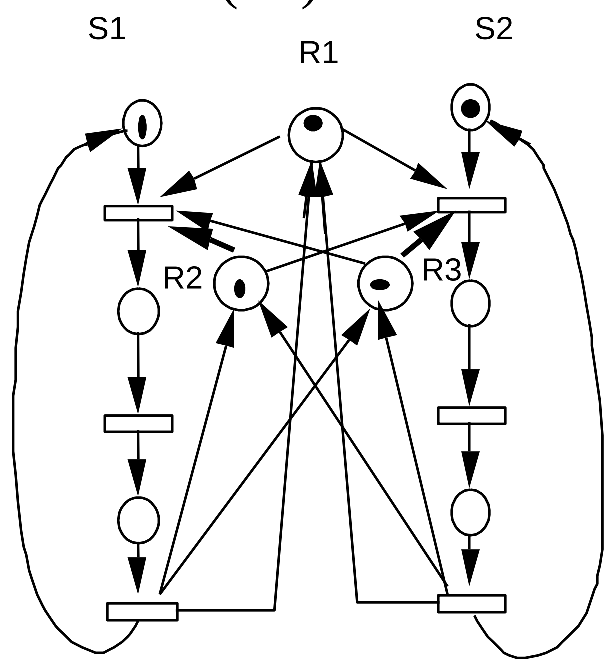
Liveness

- **Liveness implies the absence of total or partial deadlock** and is often required for well-designed systems. But the reverse is not true.
- Deadlock often results from resource sharing and synchronization of parallel processes.
- **No monotonicity of liveness** as the Petri net below is not live if M0(R1) = 0, live if M0(R1) = 1, and not live if M0(R1) = 2.

PN1



PN2



Reversibility

A Petri net (N, M0) is said **reversible** if the initial marking remains reachable from any reachable marking, i.e. $M0 \in \mathcal{R}(M), \forall M \in \mathcal{R}(M0)$

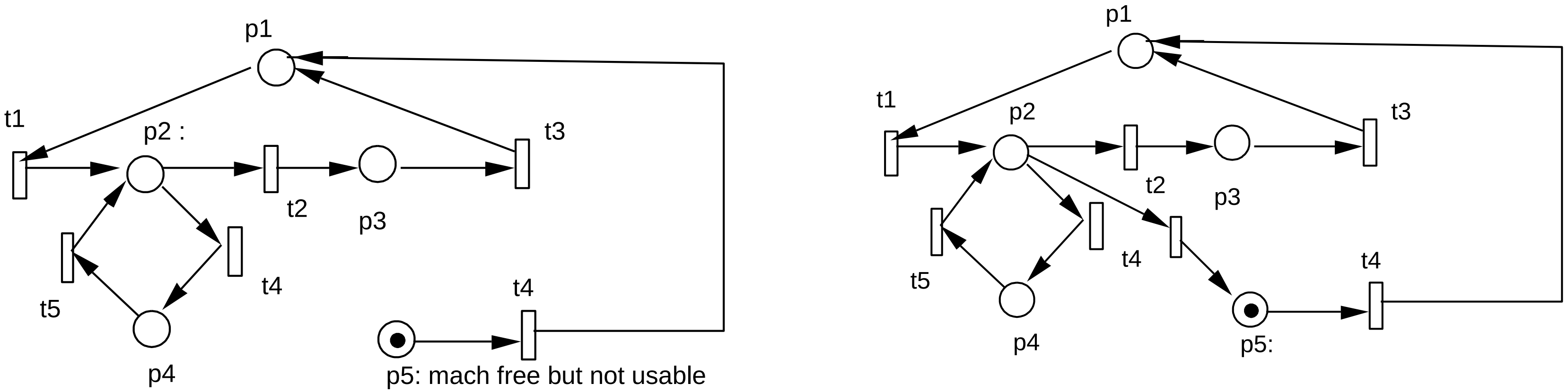
A marking M* is said a **home state** if it is reachable from all reachable markings, i.e. $M^* \in \mathcal{R}(M), \forall M \in \mathcal{R}(M0)$.

Existence of the reversibility ensures that the system can always recover the normal behavior and is important for systems subject to failures.

Existence of home state is important for systems requiring proper termination.

Reversiblity implies existence of home states but the reverse is not true.

Reversibility



Reversibility, liveness and boundedness are independent

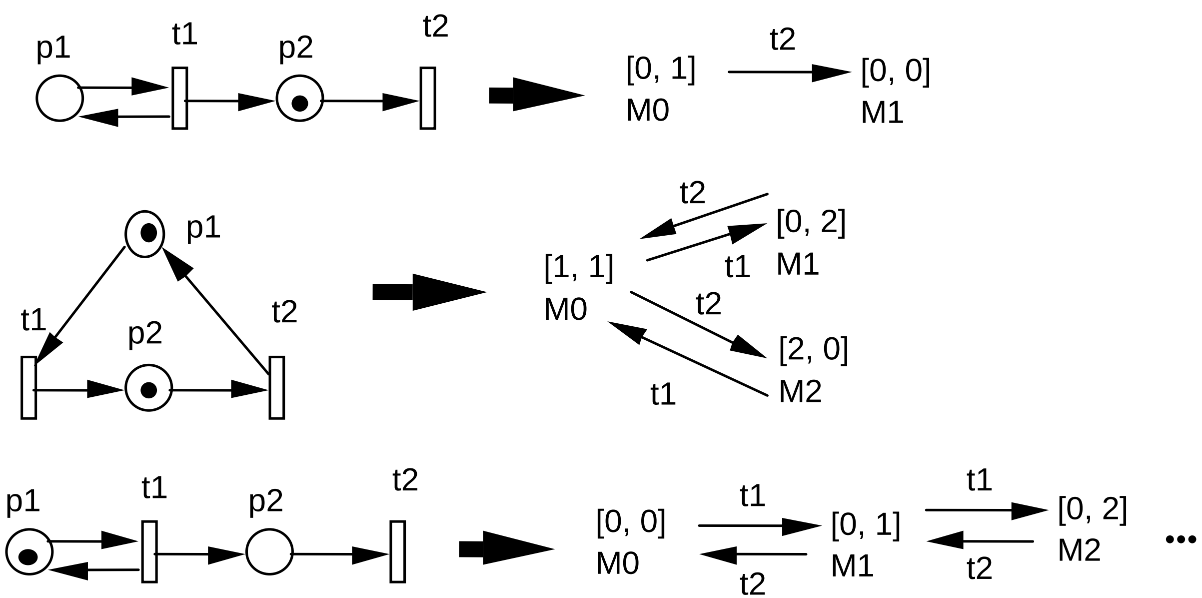
Analysis methods

Reachability tree

Definition: The reachability tree, also called marking graph, of a Petri net (N, M0) is a graph in which

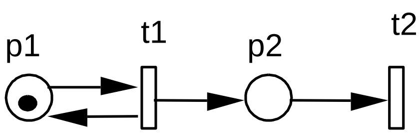
- nodes corresponds to reachable markings
- arcs correpond to feasible transitions.

Remark: the reachability tree of an unbounded PN is unlimited.

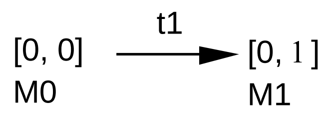


Coverability tree

Symbol "ω" implying « as great as possible » with the following properties:
ω > n, ω ± n = ω, for all integer n and ω ≥ ω.

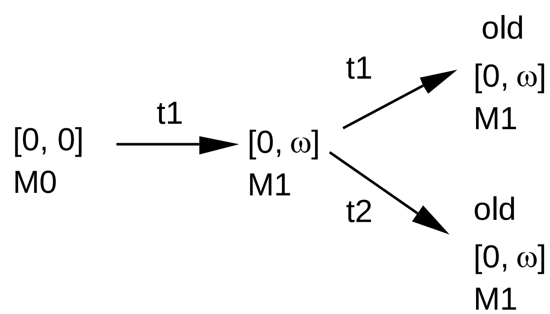


Step1

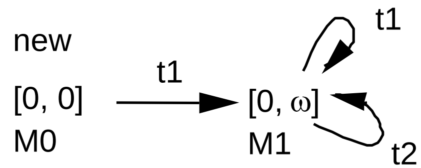


- M1 covers M0
- Repeat t1 leads to ω tokens in p2.
- Replace M1 by [0, ω]

Step2



Step3



Coverability tree

Algorithm of coverability tree

1. Initiate the tree by a root node labeled M0 and marked as "new".

2. While there exists "new" nodes :

   2.1. Select a "new" node A. Let M be its marking.

   2.2. If there exists a node B with marking M on the path from the root to A, then mark A as "old" and go to 2.

   2.3. If M is a dead marking, then mark A"dead-end" and go to 2.

   2.4. Otherwise, for each transition t enabled at M,

      2.4.1. Add a node C, an arc from A to C with label t, mark C "new".

      2.4.2. Determine the marking M' of node C.

      2.4.3. If, on the path from the root to node C, there exists a node D with marking M'' such that $M' \geq M''$ & $M'(p) > M''(p)$ for some p, then $M'(p) = \omega$ for all p such that $M'(p) > M''(p)$.

   2.5. Go to 2.

Coverability tree

Theorem (boundedness) :
A Petri net (N, M0) is bounded iff the symbol ω does not appear in the coverability tree.

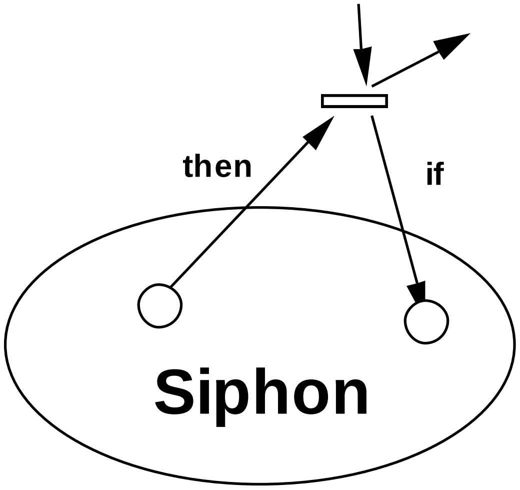
Theorem (bounded PN) :
For a bounded Petri net, it is deadlock-free iff any node of the reachability tree has a successor. It is reversible iff the reachability tree is strongly connected. A transition t is live iff it appears a all strongly connected components that do not have arcs going out.

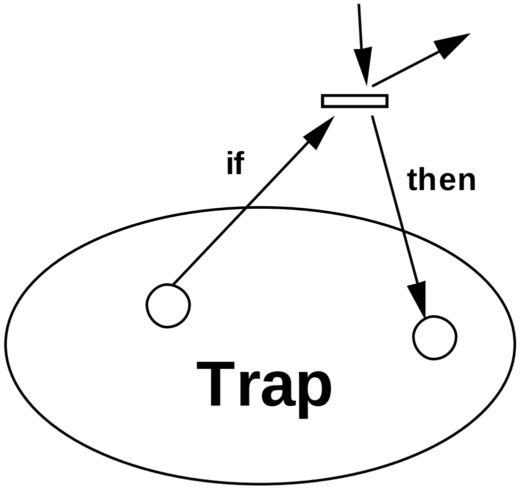
Remark:
Liveness and reversibility of unbounded PN cannot be checked with coverability trees.

Siphons and traps

A siphon is a subset of places such that any input transition of a place is an output transition of some other place.

A trap is a subset of places such that any output transition of a place is an output transition of some other place.

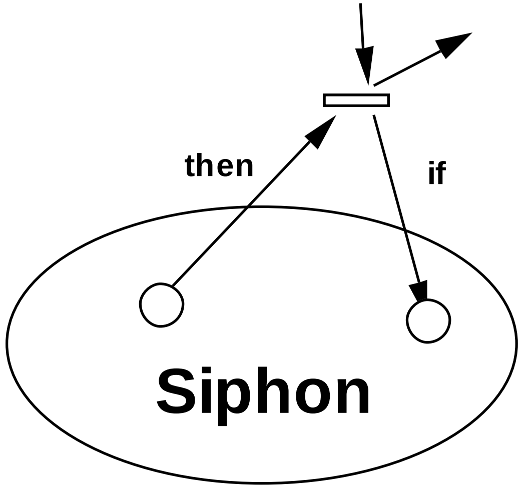


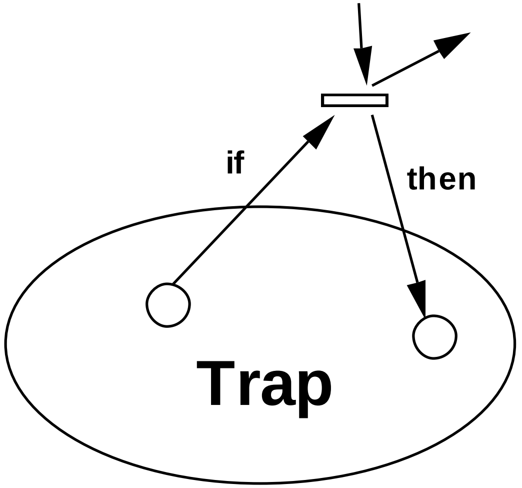


Siphons and traps

Theorem: For any ordinary PN,

- A siphon free of tokens at a marking remains token-free
- A trap marked by a marking remains marked
- The empty places of a dead marking form a siphon for any marking such that no transition is enabled.
- A Petri net is deadlock-free if no siphon eventually becomes empty.





Siphons and traps

Theorem: **A connected event graph** (N, M0) is live iff every circuit contains a token. A live event graph is reversible. A connex event graph is bounded iff it is strongly connected.

Theorem: **A connected state machine** is always bounded. It is live and reversible iff it is strongly connected.

Theorem : **A free-choice (extended or not) (N, M0)** is live iff all siphon contains a trap marked at M0.

Theorem : **An assymetric net (N, M0) is live** iff no siphon can become unmarked.

Remarks:

- Whether all siphons remain marked can be checked by integer programming.
- For usual manufacturing systems, both liveness and reversibility are ensured if no siphon can become unmarked

Siphons and traps

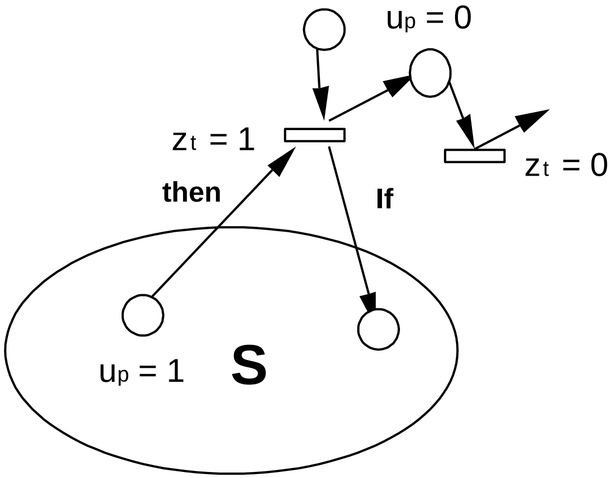
Theorem: A Petri net (N, M_0) is deadlock-free if $G = 0$ where

$$G = \max \sum_{p \in P} u_p$$

such that

- S is a siphon, i.e.

$$z_t \leq \sum_{p \in {}^\bullet t} u_p, \ \forall t \in T$$
$$u_p \leq z_t, \ \forall\ t, p \ /\ t \in {}^\bullet p$$
$$u_p\ ,\ z_t \in \{0, 1\}$$



- S can become unmarked:

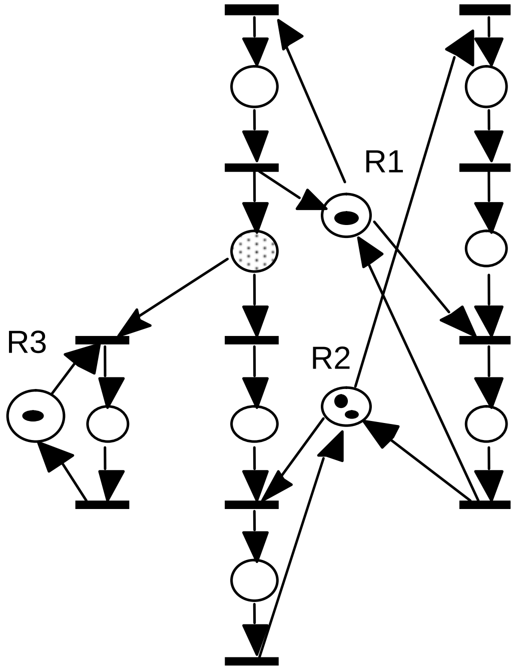
$$\mathbf{1\{M(p)\} + u_p \ \leq 1}\ ,\ \forall p \in P \qquad \mathbf{(NL)}$$
$$M = M_0 + CY$$
$$M \geq 0,\ Y \geq 0.$$

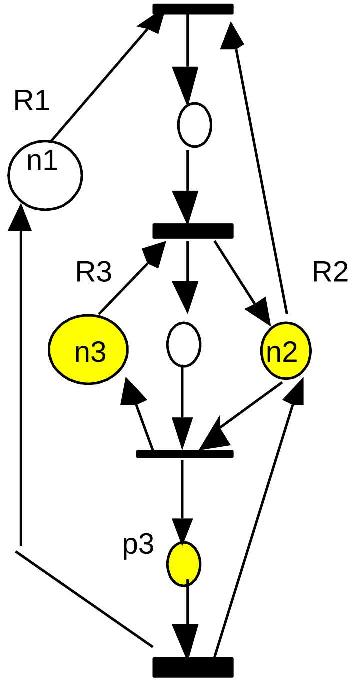
The nonlinear constraint (NL) can be replaced by

(NL) $\quad \Leftarrow\!\!\Rightarrow \quad$ $\mathbf{M(p) / SB(p) + u_p \ \leq 1}$

where SB(p) is the upper bound of the marking of place p.

Siphons and traps





Live as it is an AC net and any siphon contain a trap marked at M0

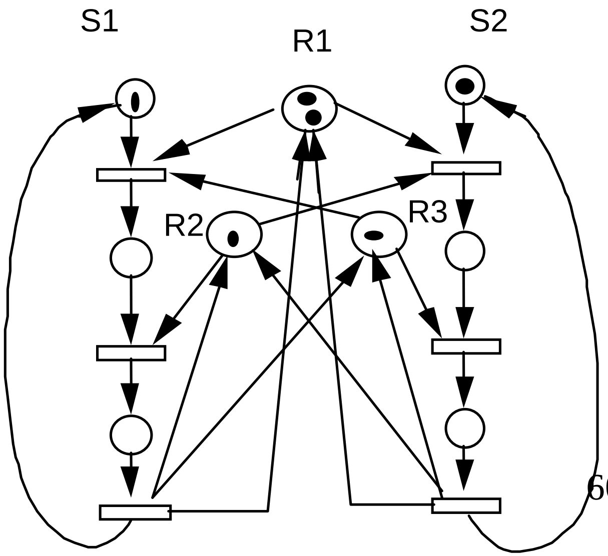
- {R2, R3, p3} = siphon that can be unmarked
- The AC net is life iff $n_1 < n_2+n_3$.

p-invariants

Definition:

- A integer vector $X \geq 0$ of dimension $n = |P|$ is a p-invariant if $X^t C = 0$.
- The set of places pi with Xi > 0 is called the support of the p-invariant and is denoted $||X||$.
- A p-invariant X is said minimal if there does not exist another p-invariant X' such that $X' \neq X$ and $X' \leq X$.

Exampel:

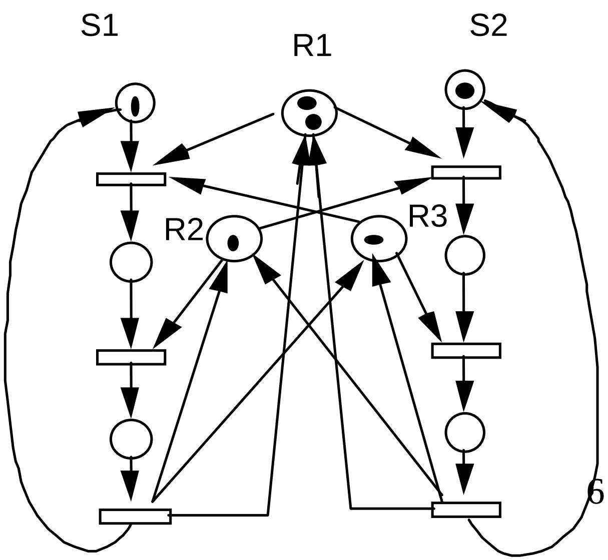


p-invariants

Theorem: X is a p-invariant iff, for all M0, $X^t M = X^t M0$, $\forall M \in R(M0)$.
Theorem : Any linear combination of p-invariants is a p-invariant.
Theorem : All p-invariant is a non negative linear combination of minimal p-invariants.

Remark : For PN models of real systems, a minimal p-invariant has clear physical significance (resource, production control strategies, ...) and can be derived by inspection of resources and processes.

Exampe:

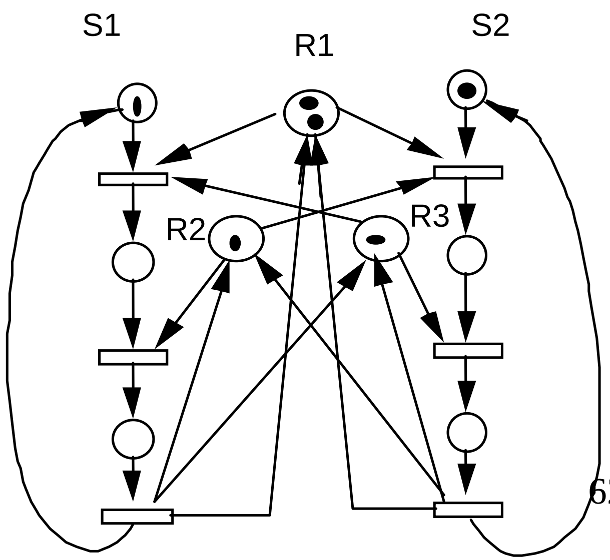


t-invariants

Definition:

- A integer vector $Y \geq 0$ of dimension $m = |T|$ is a t-invariant if $CY = 0$.
- The set of transitions ti with $Y_i > 0$ is called the support of the t-invariant and is denoted $||Y||$.
- A t-invariant Y is said minimal if there does not exist another t-invariant Y' such that $Y' \neq Y$ and $Y' \leq Y$.

Exampel:

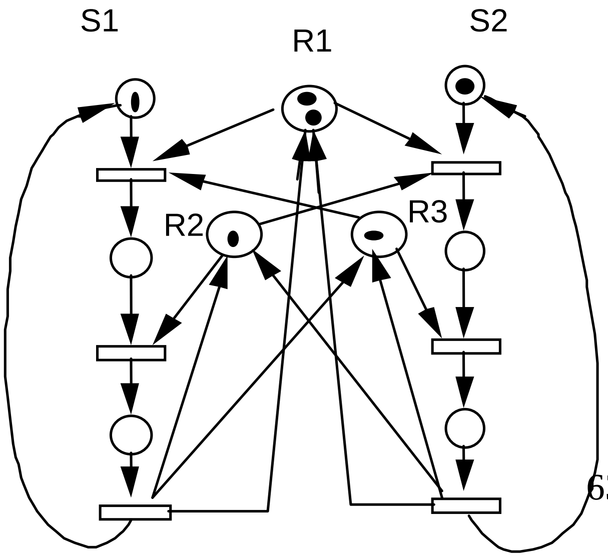


t-invariants

Theorem: Let s be a sequence of transitions tranforming M0 into M and Y its counting vector. Then M = M0 iffY is an t-invariant.
Theorem : Any linear combination of t-invariants is a t-invariant.
Theorem : All t-invariant is a non negative linear combination of minimal t-invariants.

Remark : In general, a minimal t-invariant corresponds to a process that can be repeat for ever. They can be identified by neglecting resources.

Exampe:



Structural properties

STRUCTURAL BOUNDEDNESS

A Petri net N is **structurally bounded** if it is bounded starting from any M0.
Criterion : N is structurally bounded $\Leftrightarrow \exists\, \mathbf{X} > 0,\ \mathbf{X}^T\mathbf{C} \leq 0$.

Theorem: (N, M0) is bounded if it is structurally bounded.

CONSERVATIVENESS

A Petri net N is **conservative** if there exists a vector $\mathbf{X} > 0$ associated with places such that $\mathbf{X}^T\mathbf{M} = \mathbf{X}^T\mathbf{M0}$, $\forall M0$, $\forall M \in R(M0)$.
Criterion : N is conservative $\Leftrightarrow \exists\, \mathbf{X} > 0,\ \mathbf{X}^T\mathbf{C} = 0$.

Theorem:

- (N, M0) is bounded if it is conservative.
- A Petri net is conservative if all places are covered by some p-invariant.

Structural properties

REPETITIVENESS

A Petri net N is **repetitive** if there exists M0 and a feasible firing sequence such that each transition appears infinitely often.

Criterion : N is repetitive $\Leftrightarrow \exists$ $\mathbf{Y} > 0$, $\mathbf{CY} \geq 0$.

Theorem: A live Petri net (N, M0) is repetitive.

CONSISTENCY

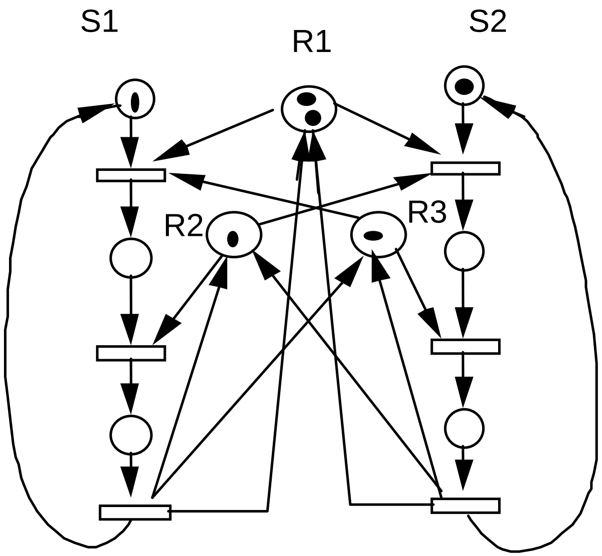
A Petri net N is **consistent** if there exist an initial marking M0 and a firing sequence σ such that > 0 and M0 [σ >M0.

Criterion : N is consistent $\Leftrightarrow \exists$ $\mathbf{Y} > 0$, $\mathbf{CY} = 0$.

Theorem :

- A live Petri net (N, M0) with a home state is consistent.
- A live and bounded Petri net (N, M0) is consistent. It is also conservative if it is live and structurally bounded.

Structural properties



In practice, boundedness reduces to conservativeness.

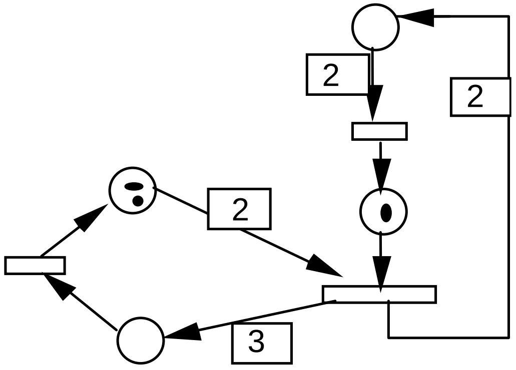
Consistency and conservativeness provide necessary conditions for liveness and resersibility.

Unfortunately, liveness and resersibility remain difficult to check.

Determination of p- and t-invariants

Algorithm of minimal p-invariants

1. Set $A = I_{n \times n}$ with $n = |P|$ and $B = C$ (incidence matrix). Construct matrix [A | B].
2. For each transition tj:
   2.1. Add to [A | B] non negative linear combination of any two lines that zeros the entry of column tj
   2.2. Remove in the matrix [A | B] all lines i such that the entry (i, j) is not zero.
3. p-invariants correspond to lines of matrix A.



The algorithm of t-invariants is similar with C replaced by C^T.

Topics not addressed in Chapters 2-3

Supervisory control with automata theory

Timed Petri nets

Color Petri nets

Petri net controls

Petri net models synthesis