

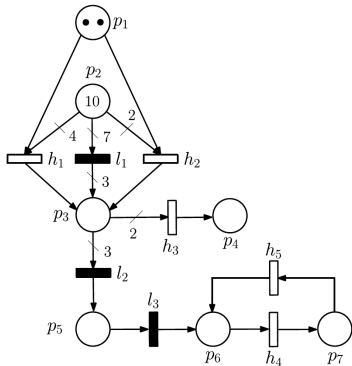
Petri nets and their twofold representation to model DES

Discrete Event Systems and Supervisory Control

Prof. Gianmaria DE TOMMASI
Email: detommas@unina.it

May 2021

- 1 Petri nets systems
 - Basic definitions
 - Language and reachability set
 - State equation
 - Reachability and coverability graphs
- 2 Behavioral properties
- 3 Structural properties
- 4 Labeled Petri net systems
 - Languages of a labeled net system
 - Regular language, Petri net languages and Chomsky grammars
- 5 Linear programming techniques for the analysis of Petri net systems
- 6 Software tools



- Introduced by Carl Adam Petri in his PhD thesis in 1962 (about concurrent programming)
- Tool introduced first in Computer Science → Automatic Control → Operations Research
- In the field of Industrial Automation Petri nets *inspired* the **Grafcet** programming language for Programmable Logic Controllers (PLCs)



M. Silva

Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field
Annual Reviews in Control, 2013



R. Alla

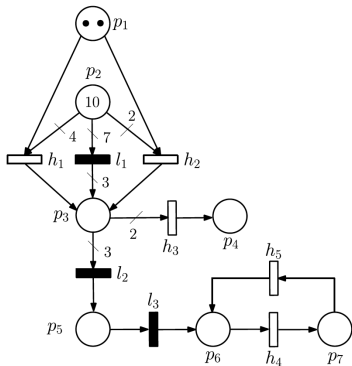
Grafcet: a powerful tool for specification of logic controllers
IEEE Transactions on Control System Technology, 1995

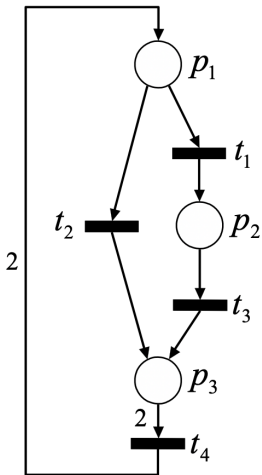
A place/transition (P/T) or Petri net is an oriented bipartite graph. Formally it can be defined as the 4-ple

$$N = (P, T, Pre, Post)$$

where

- P is a set of m places
- T is a set of n transitions
- $Pre : P \times T \mapsto \mathbb{N}$ is the *pre-incidence* function (but we will represent it as a $m \times n$ matrix)
- $Post : P \times T \mapsto \mathbb{N}$ is the *post-incidence* function





$$P = \{p_1, p_2, p_3\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

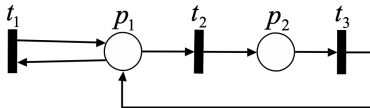
$$\mathbf{Pre} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

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

Given a P/T net $N = \{P, T, \mathbf{Pre}, \mathbf{Post}\}$ with m places and n transitions, the incidence matrix $\mathbf{C} \in \mathbb{Z}^{m \times n}$ is given by

$$\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$$

In general, the incidence matrix does not contain sufficient information to reconstruct the net structure (\rightarrow this will have an impact on the use of the so-called *state equation*)



Given a transition $t \in T$ we denote with

- $\bullet t = \{p \in P \mid \mathbf{Pre}(p, t) > 0\}$, the *preset* of t (*input places*)
- $t^\bullet = \{p \in P \mid \mathbf{Post}(p, t) > 0\}$, the *postset* of t (*output places*)

Given a place $p \in P$ we denote with

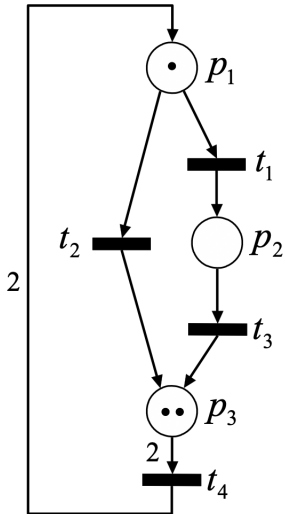
- $\bullet p = \{t \in T \mid \mathbf{Post}(p, t) > 0\}$, the *preset* of p (*input transitions*)
- $p^\bullet = \{t \in T \mid \mathbf{Pre}(p, t) > 0\}$, the *postset* of p (*output transitions*)

- P/T nets are *just* graphs
- In order to use them to model dynamic systems, we need to introduce the concept of *state*
- The **marking** is the way we have to define a discrete state space
- The marking can be defined as a function (other definition can be found in literature, like marking as a multiset)

$$m : P \mapsto \mathbb{N},$$

that assigns to each place a nonnegative integer number of **tokens**

- The marking is usually represented as a vector $\mathbf{m} \in \mathbb{N}^m$
- A P/T net N with its initial marking \mathbf{m}_0 is called **net system**, and is denoted with $\langle N, \mathbf{m}_0 \rangle$



$$m_0 = (1 \ 0 \ 2)^T$$

Enabling condition

A transition t is enabled at the marking m if

$$m \geq \mathbf{Pre}(\cdot, t)$$

i.e., if each place $p \in P$ contains a number of tokens greater than or equal to $\mathbf{Pre}(p, t)$

- $m[t\rangle$ denotes that t is enabled at m
- $m\neg[t\rangle$ denotes that t is not enabled at m

Firing of a transition

- A transition t enabled at m can fire
- The firing of t removes $\mathbf{Pre}(p, t)$ tokens from each place $p \in P$ and adds $\mathbf{Post}(p, t)$ tokens in each place $p \in P$
- Hence, the firing of t yields the new marking

$$m' = m - \mathbf{Pre}(\cdot, t) + \mathbf{Post}(\cdot, t) = m + \mathbf{C}(\cdot, t)$$

- $m[t\rangle m'$ denotes that the firing of t from m leads to m'

A firing sequence $\sigma = t^1 t^2 \dots t^r \in T^*$ is enabled at m if and only if

- $m[t^1\rangle$ and $m[t^1\rangle m_1$
- $m_1[t^2\rangle$ and $m_1[t^2\rangle m_2$
- ...
- $m_{r-1}[t^r\rangle$ and $m_{r-1}[t^r\rangle m_r$

$m[\sigma\rangle$ denotes that σ is enabled at m , while $m[\sigma\rangle m'$ denotes that the firing sequence σ starting from m yields the marking m'

The empty sequence

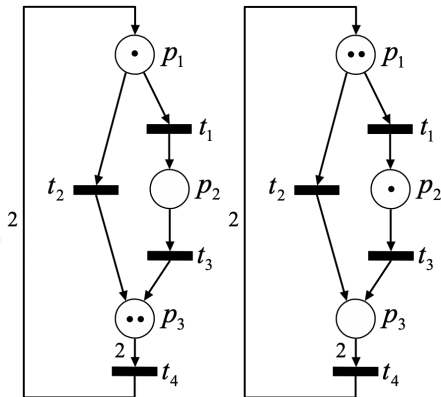
ε is the empty sequence and it is

- $m[\varepsilon\rangle \quad \forall m \in \mathbb{N}^m$
- $m[\varepsilon\rangle m$

Firing sequence – Example

- The firing sequence $\sigma = t_4 t_1$ is enable at $\mathbf{m} = (1 \ 0 \ 2)^T$
- Its firing yields

$$\mathbf{m}' = (2 \ 1 \ 0)^T$$



Given a net system \mathcal{S} and a firing sequence $\sigma \in T^*$, it is possible to introduce the **firing count vector** $\sigma \in \mathbb{N}^n$ whose entry $\sigma(t_i) = \sigma_i$ denotes how many times transition t_i appears in the sequence σ

Example

- $T = \{t_1, t_2, t_3\}$
- $\sigma = t_3 t_1 t_1 t_3 t_2$
- $\sigma = (2 \ 1 \ 2)^T$

- So far we have dealt with the so called **unlabeled** Petri nets
- The set of transitions somehow corresponds to the set of events that drive the system dynamic
- It is then possible to associate a language to $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ defined as follows

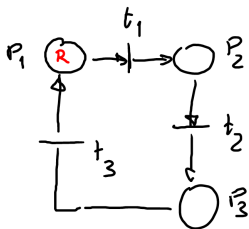
$$L(N, \mathbf{m}_0) = \{ \sigma \in T^* \mid \mathbf{m}_0[\sigma] \}$$

- It readily follows that $\varepsilon \in L(N, \mathbf{m}_0)$

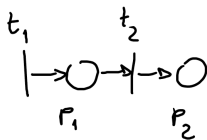
- Given a net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ a marking \mathbf{m} is said to be **reachable** (from \mathbf{m}_0) if there exists a firing sequence $\sigma \in L(N, \mathbf{m}_0)$ such that $\mathbf{m}_0[\sigma\rangle\mathbf{m}$
- It is then possible to introduce the **reachability set** $R(N, \mathbf{m}_0)$ of \mathcal{S} as follows

$$R(N, \mathbf{m}_0) = \{ \mathbf{m} \in \mathbb{N}^m \mid \exists \sigma \in L(N, \mathbf{m}_0) \text{ s.t. } \mathbf{m}_0[\sigma\rangle\mathbf{m} \}$$

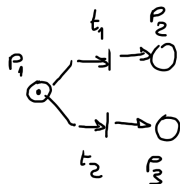
- Obviously it is $\mathbf{m}_0 \in R(N, \mathbf{m}_0)$



$$m_0 = (R \ 0 \ 0)^T$$



$$m_0 = (0 \ 0)^T$$



$$m_0 = (1 \ 0 \ 0)$$

WHAT IS THE CARDINALITY OF $R(N, m_0)$?

Given a net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ and a reachable marking $\mathbf{m} \in R(N, \mathbf{m}_0)$ such that $\mathbf{m}_0 [\sigma) \mathbf{m}$ with $\sigma \in L(N, \mathbf{m}_0)$.

If σ is the firing count vector associated to firing sequence σ then the following **state equation** holds

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$$

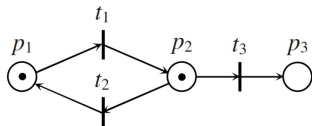
If a net system \mathcal{S} has a reachability set $R(N, \mathbf{m}_0)$ with finite cardinality, then it is possible to explicitly represent the whole state space by means of an automaton, called **reachability graph**

Algorithm 10.2. (Reachability Graph).

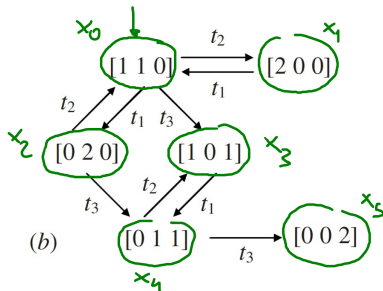
1. The initial node of the graph is the initial marking \mathbf{m}_0 . This node is initially unlabeled.
2. Consider an unlabeled node \mathbf{m} of the graph.
 - a For each transition t enabled at \mathbf{m} , i.e., such that $\mathbf{m} \geq \mathbf{Pre}[\cdot, t]$:
 - i. Compute the marking $\mathbf{m}' = \mathbf{m} + \mathbf{C}[\cdot, t]$ reached from \mathbf{m} firing t .
 - ii. If no node \mathbf{m}' is on the graph, add a new node \mathbf{m}' to the graph.
 - iii. Add an arc t from \mathbf{m} to node \mathbf{m}' .
 - b Label node \mathbf{m} “old”.
3. If there exist nodes with no label, goto Step 2.

Figure: Algorithm taken from Cabasino *et al.*, “Introduction to Petri nets”

Reachability graph – Example



(a)



(b)

THE
REACHABILITY
GRAPH AS AN
AUTOMATA

$$E = \{t_1, t_2, t_3\}$$

- If the cardinality of $R(N, \mathbf{m}_0)$ is infinite, then the reachability graph cannot be constructed
- In order to represent marking whose components can arbitrarily grow, the symbol ω is adopted to denote an arbitrarily large component in a marking vector

Coverability tree

- 1 The root node of the tree is the initial marking \mathbf{m}_0 . This node is initially unlabeled.
- 2 Consider an unlabeled node \mathbf{m} of the tree.
 - a For each transition t enabled at \mathbf{m} , i.e., such that $\mathbf{m} \geq \mathbf{Pre}[\cdot, t]$:
 - i. Compute the marking $\mathbf{m}' = \mathbf{m} + \mathbf{C}[\cdot, t]$ reached from \mathbf{m} firing t .
 - ii. For all markings $\tilde{\mathbf{m}} \preceq \mathbf{m}'$ on the path from the root node \mathbf{m}_0 to node \mathbf{m} and for all $p \in P$,
if $\tilde{\mathbf{m}}[p] < \mathbf{m}'[p]$ then let $\mathbf{m}'[p] = \omega$.
 - iii. Add a new node \mathbf{m}' to the tree.
 - iv. Add an arc t from \mathbf{m} to the new node \mathbf{m}' .
 - v. If there already exists a node \mathbf{m}' in the tree, label the new node \mathbf{m}' “duplicated”.
 - b Label node \mathbf{m} “old”.
- 3 If there exist nodes with no label, goto Step 2.

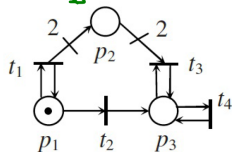
Algorithm 10.4. (Coverability graph).

- 1 *If the tree contains no nodes with label “duplicated” goto Step 4.*
- 2 *Consider a node m of the graph with label “duplicated”.
Such a node has no output arcs but an input arc t from node m' .
Moreover, there surely exists in the graph another node m with label “old”.*
 - a *Remove arc t from node m' to node m “duplicated”.*
 - b *Add an arc t from node m' to node m “old”.*
 - c *Remove node m “duplicated”.*
- 3 *If there still exist nodes with label “duplicated” goto Step 2.*
- 4 *Remove labels from nodes.*

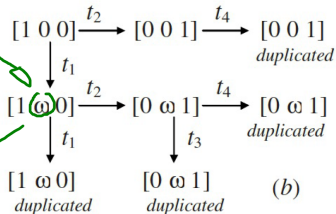
Figure: Algorithm taken from Cabasino *et al.*, “Introduction to Petri nets”

Coverability graph – Example

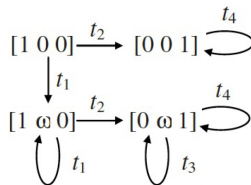
$m(p_2)$ CAN
BE ONLY
EVEN



(a)



(b)



(c)

→ ω DOES NOT STAND FOR "ANY POSITIVE INTEGER"

The behavioral properties of Petri net systems are those ones that depends (also) on the initial marking m_0

- **reachability**
- **boundedness**
- **conservativeness**
- **repetitiveness**
- **reversibility**
- **liveness**

The reachability problem

Given a net systems $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ and a generic marking $\mathbf{m} \in \mathbb{N}^m$, is $\mathbf{m} \in R(N, \mathbf{m}_0)$?

- it is straightforward to see that if $R(N, \mathbf{m}_0)$ has finite cardinality, i.e. the net system is bounded, then the problem is **decidable**. In this case the reachability problem is equivalent to the decidability of regular languages
- it is also straightforward to show that in the general case the problem is at least **semi-decidable**
- In the 1980s it has been proved the the reachability problem is decidable, but the corresponding algorithm has a very high complexity



C. Reutenauer

Aspects Mathématiques des Réseaux
de Petri

Prentice Hall, 1989



k-boundedness

- A place p is k -bounded in the system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ if for all the reachable markings $\mathbf{m} \in R(N, \mathbf{m}_0)$ it holds $\mathbf{m}(p) \leq k$
 - A system \mathcal{S} is k -bounded if all its places are k -bounded
-
- When we are not interested in any specific k , then the system \mathcal{S} is simply called **bounded**
 - Obviously the following holds

\mathcal{S} is bounded $\Leftrightarrow R(N, \mathbf{m}_0)$ is finite

Reversibility

A net system \mathcal{S} is reversible if for all $\mathbf{m} \in R(N, \mathbf{m}_0)$ it holds that $\mathbf{m}_0 \in R(N, \mathbf{m})$

- Reversibility implies that a system \mathcal{S} can always be reinitialized to its initial marking \mathbf{m}_0
- Non reversible systems may exhibit **home states**

Home state

A marking $\tilde{\mathbf{m}} \in R(N, \mathbf{m}_0)$ is a **home state** if for all $\mathbf{m} \in R(N, \mathbf{m}_0)$ it holds that $\tilde{\mathbf{m}} \in R(N, \mathbf{m})$

Liveness of a transition

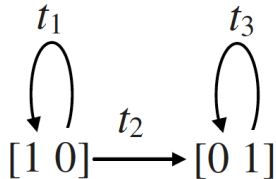
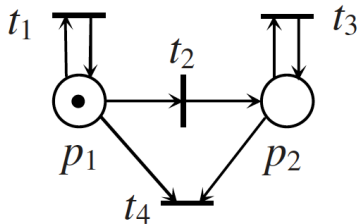
Given a net system \mathcal{S} a transition t is said to be

- **dead** if no reachable marking enables t
- **quasi-live** if t is enabled by some reachable marking, i.e.
 $\exists \mathbf{m} \in R(N, \mathbf{m}_0)$ s.t. $\mathbf{m}[t)$
- **live** if for all reachable markings $\mathbf{m} \in R(N, \mathbf{m}_0)$, t is *quasi-live* in $\langle N, \mathbf{m} \rangle$

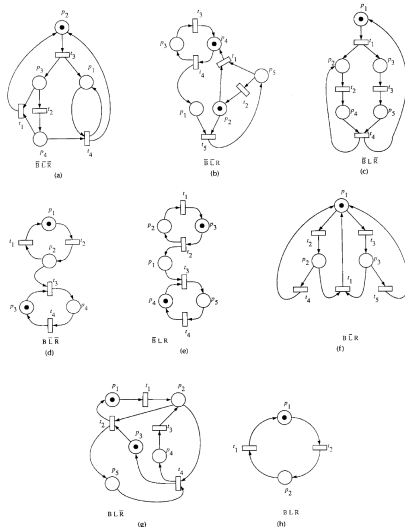
Liveness of a net system \mathcal{S}

- \mathcal{S} is **dead** if all its transitions are dead
- \mathcal{S} is **not quasi-live** if some of its transitions are dead and some quasi-live
- \mathcal{S} is **quasi-live** if all its transitions are quasi-live
- \mathcal{S} is **live** if all its transitions are live

Sometimes a net system is considered live if it has some live transitions (not necessarily all)



Boundedness, liveness and reversibility are independent concepts (image taken from Murata, *Proc. IEEE*, 1989)



Use the graphs to check behavioral properties

- Both the reachability (for bounded systems) and coverability graph (for unbounded systems) can be used to assess the behavioral properties
- Reachability graph permits to formulate necessary and sufficient conditions; there is a link with decidability of regular languages
- Coverability graph permits formulate either necessary or sufficient conditions

The structural properties of Petri net systems are those ones that depends only on the graph and not on the specific initial marking m_0

- **analysis based on the state equation**
- **analysis based on invariants**
- **analysis based on siphons and traps**

- the **satisfaction of the state equation is only a necessary condition for reachability**

- Given a marking $\mathbf{m} \in \mathbb{N}^m$, if $\mathbf{m} \in R(N, \mathbf{m}_0) \Rightarrow \exists \sigma \in \mathbb{N}^n$ such that $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$

- Potentially reachable marking $PR(N, \mathbf{m}_0)$

$$PR(N, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^m \mid \exists \mathbf{y} \in \mathbb{N}^n \text{ s.t. } \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \mathbf{y}\}$$

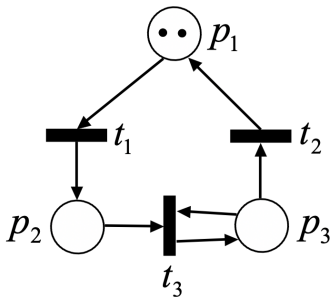
- **It is** $R(N, \mathbf{m}_0) \subseteq PR(N, \mathbf{m}_0)$
- The markings in $PR(N, \mathbf{m}_0) \setminus R(N, \mathbf{m}_0)$ are called **spurious markings**, since they satisfy the state equation but are not reachable

- $m_0 = (2\ 0\ 0)^T$
- If $y = (1\ 0\ 1)^T$ then

$$m = (1\ 0\ 0)^T \in PR(N, m_0)$$

while

$$m = (1\ 0\ 0)^T \notin R(N, m_0)$$



When $PR(N, \mathbf{m}_0) = R(N, \mathbf{m}_0)$?



- There are cases where the satisfaction of the state equation becomes necessary **and sufficient** for reachability, i.e. $PR(N, \mathbf{m}_0) = R(N, \mathbf{m}_0)$
 - Acyclic nets
 - Some classes of ordinary nets (nets with all arcs that have unitary multiplicity)
 - State machines (only choices and convergences)
 - Marked graphs (only parallelisms and synchronizations)
- When $PR(N, \mathbf{m}_0) = R(N, \mathbf{m}_0)$, the following feasibility problem with integer unknowns and linear constraints can be solved to assess reachability
 $\mathbf{m} \in \mathbb{N}^m$ is reachable **if and only if** it is possible to find a $\boldsymbol{\sigma} \in \mathbb{N}^n$ such that

$$\mathbf{C} \cdot \boldsymbol{\sigma} = \mathbf{m} - \mathbf{m}_0$$

which is NP-hard

P-invariants

Given a net N a vector $\mathbf{x} \in \mathbb{N}^m$ with $\mathbf{x} \neq \mathbf{0}$ is called a **P-invariant** if

$$\mathbf{x}^T \cdot \mathbf{C} = \mathbf{0}^T$$

T-invariants

Given a net N a vector $\mathbf{y} \in \mathbb{N}^n$ with $\mathbf{y} \neq \mathbf{0}$ is called a **T-invariant** if

$$\mathbf{C} \cdot \mathbf{y} = \mathbf{0}$$

- If a marking m is reachable then it satisfies the state equation

$$m = m_0 + C \cdot \sigma$$

- by multiplying by a P-vector x

$$x^T \cdot m = x^T \cdot m_0 + x^T \cdot C \cdot \sigma$$

hence, if $m \in R(N, m_0)$ then

$$x^T \cdot m = x^T \cdot m_0$$

- Let σ be a firing sequence such that $m_0 \langle \sigma \rangle m$. If the corresponding firing count vector σ is a T-invariant, then it is $m = m_0$
- All sequences σ whose corresponding firing count vector σ is a T-invariant are **repetitive and stationary** sequences

Siphons and traps are usually introduced for *ordinary nets*, i.e. **nets with all arcs with unitary multiplicity**

Siphons

A siphon of an ordinary net N is a set of places $\mathbb{S} \subseteq P$ such that **the set of input transitions of \mathbb{S} is included in the set of output transitions of \mathbb{S}** , that is

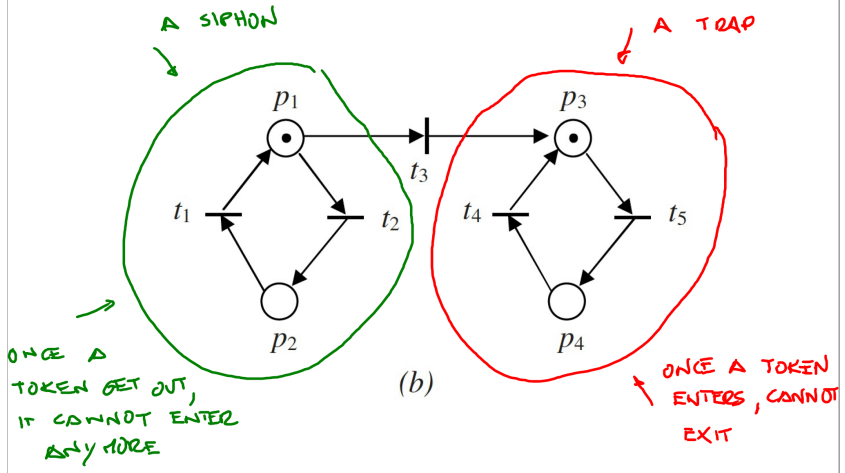
$$\bigcup_{p \in \mathbb{S}} \bullet p \subseteq \bigcup_{p \in \mathbb{S}} p \bullet$$

Traps

A trap of an ordinary net N is a set of places $\mathbb{T} \subseteq P$ such that **the set of output transitions of \mathbb{T} is included in the set of input transitions of \mathbb{T}** , that is

$$\bigcup_{p \in \mathbb{T}} p \bullet \subseteq \bigcup_{p \in \mathbb{T}} \bullet p$$

Siphon and traps – Example



The X -invariant set $I_X(N, \mathbf{m}_0)$

Let $S = \langle N, \mathbf{m}_0 \rangle$ be a net system with m places and consider the matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\} \in \mathbb{N}^{m \times k}$, whose generic column \mathbf{x}_i is a P-invariant of N . The X -invariant set of S is

$$I_X(N, \mathbf{m}_0) = \left\{ \mathbf{m} \in \mathbb{N}^m \mid \mathbf{X}^T \cdot \mathbf{m} = \mathbf{X}^T \cdot \mathbf{m}_0 \right\}$$

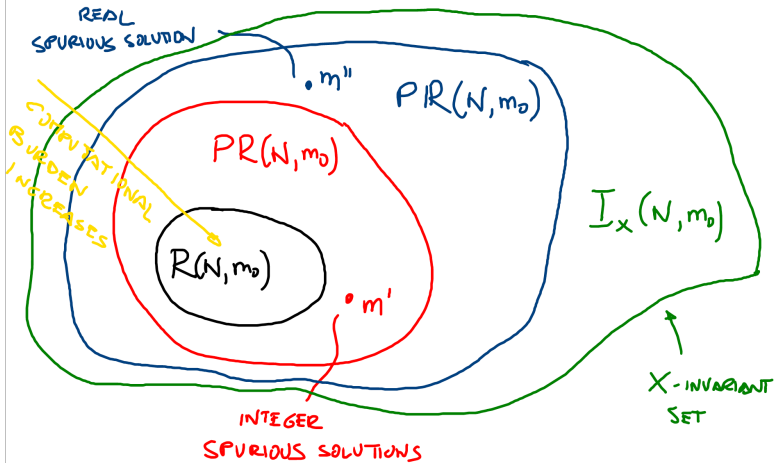
- It can be easily proved that

$$R(N, \mathbf{m}_0) \subseteq PR(N, \mathbf{m}_0) \subseteq I_X(N, \mathbf{m}_0)$$

Estimates of the reachability set $R(N, m_0)$



$$PR(N, m_0) = \{m \in N^m \mid \exists \sigma \in \mathbb{R}^n, \sigma \geq 0 \text{ and } m = m_0 + C\sigma\}$$



Given an **alphabet (of events)** E , a **labeled Petri net system** is the 4-ple

$$\mathcal{S}_\ell = (N, \ell, \mathbf{m}_0, F)$$

where

- $\ell : T \mapsto E \cup \{\varepsilon\}$ is the labeling function that associates an event $e \in E \cup \{\varepsilon\}$ to every transition $t \in T$
- F is the set of **final** markings

NOTE: an event can be associated to more than one transition
→ source of nondeterminism

Three different types of labeling functions can be considered

- **free labeling** – all transitions are distinctly labeled and none is labeled as the silent event ε ; the labeling function does not add any relevant information, i.e. we can consider $T = E$
- **ε -free labeling** (usually referred to also as λ -free)– no transition is labeled with the silent event ε (source of nondeterminism)
- **arbitrarily labeling** – no restriction posed on the labeling function (further source of nondeterminism)

The labeling function $\ell(\cdot)$ can be recursively extended to firing sequences σ so to associate them words w

- $\ell : T^* \mapsto E^*$ with
 - $\ell(\varepsilon) = \varepsilon$
 - $\ell(\sigma t) = \ell(\sigma)\ell(t)$

- The adopted notation will be
 - $e = \ell(t)$, with $t \in T$ and $e \in E$
 - $w = \ell(\sigma)$, with $\sigma \in T^*$ and $w \in E^*$

Similarly to what has been done in the case of automata, also in the case of labeled Petri net system, the following two different languages can be introduced

- The generated language (sometimes referred to as the *prefix language*)

$$\mathcal{L}(\mathcal{S}_\ell) = \{ \mathbf{w} = \ell(\sigma) \in E^* \text{ s.t. } \mathbf{m}_0[\sigma] \}$$

- The marked language (sometimes referred to as the *terminal language*)

$$\mathcal{L}_m(\mathcal{S}_\ell) = \{ \mathbf{w} = \ell(\sigma) \in E^* \text{ s.t. } \mathbf{m}_0[\sigma] \mathbf{m}_f \in F \}$$

- $\mathcal{L}(\mathcal{S}_\ell)$ and $\mathcal{L}_m(\mathcal{S}_\ell)$ are used as in the automata context for supervisory control based on language specifications



Let $\mathcal{S}_1 = (N_1, \ell_1, \mathbf{m}_{0_1}, F_1)$ and $\mathcal{S}_2 = (N_2, \ell_2, \mathbf{m}_{0_2}, F_2)$ two labeled systems. Their **parallel composition** (concurrent composition) $\mathcal{S} = \mathcal{S}_1 \parallel \mathcal{S}_2$ that generates the language

$$\mathcal{L}(\mathcal{S}) = \mathcal{L}(\mathcal{S}_1) \parallel \mathcal{L}(\mathcal{S}_2)$$

and marks the language

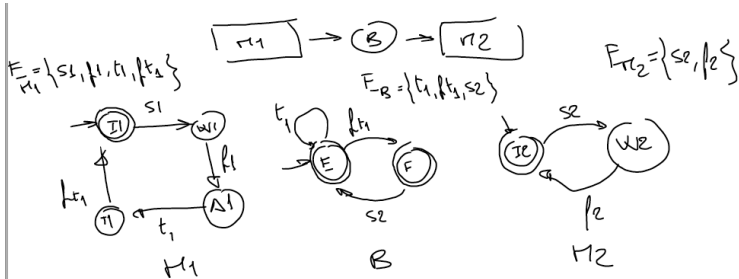
$$\mathcal{L}_m(\mathcal{S}) = \mathcal{L}_m(\mathcal{S}_1) \parallel \mathcal{L}_m(\mathcal{S}_2)$$

and his given by the following algorithm.

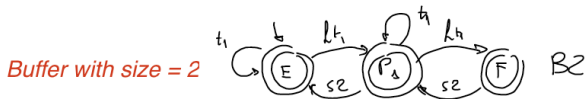
Algorithm for parallel composition of labeled PNs

Let P_i , T_i and E_i ($i = 1, 2$) be the place set, transition set, and the alphabet of S_i .

- The place set P of N is the union of the place sets of N_1 and N_2 , i.e.
 $P = P_1 \cup P_2$
- The transition set T of N and the corresponding labels are computed as follows
 - For each transition $t \in T_1 \cup T_2$ labeled ε , a transition with the same $\bullet t$ and t^\bullet and labeled ε belongs to T
 - For each transition $t \in T_1 \cup T_2$ labeled $e \in (E_1 \setminus E_2) \cup (E_2 \setminus E_1)$, a transition with the same $\bullet t$ and t^\bullet and labeled e belongs to T
 - Consider a symbol $e \in E_1 \cap E_2$ and assume it labels μ_1 transitions $T_{e,1} \subseteq T_1$ and μ_2 transitions $T_{e,2} \subseteq T_2$. Then $\mu_1 \times \mu_2$ transitions labeled e belong to T . The input (output) bag of each of these transitions is the sum of the input (output) bags of one transition in $T_{e,1}$ and of one transition in $T_{e,2}$
- $m_0 = (m_{0_1}^T \ m_{0_2}^T)^T$
- $F = \left\{ (m_1^T \ m_2^T)^T \mid m_1 \in F_1 \text{ and } m_2 \in F_2 \right\}$

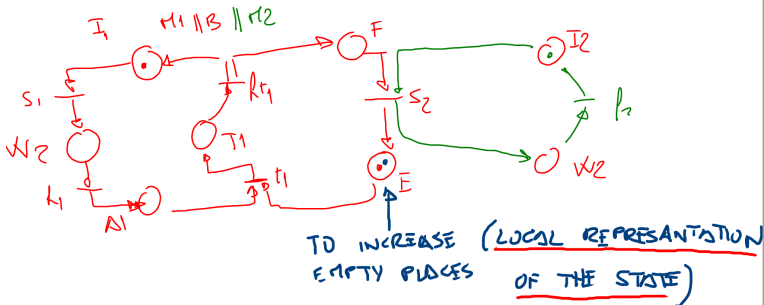
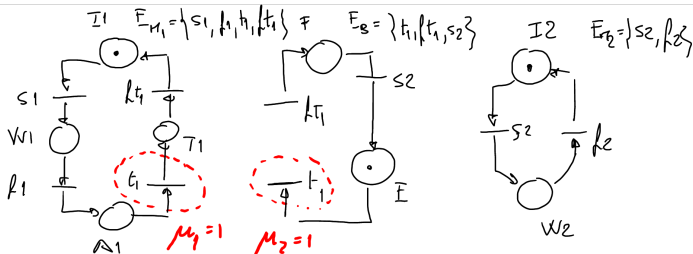


$P_{PLANT} = H1 \parallel B \parallel \tau_2 \quad \text{card}(X) = 14$



$P_{PLANT} = \tau_1 \parallel B2 \parallel \tau_2 \quad \text{card}(X) = 22$

The FMS example with PNs

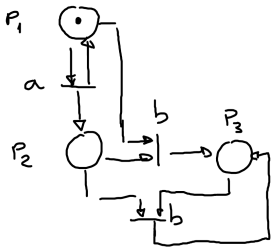


What are the Petri net languages?

- The languages recognized by the finite state automata are the regular languages (Kleene theorem)
- What about *finite* Petri nets, i.e. Petri nets with a finite number of places and transitions?

$$E = \{a, b\} \quad L = \{w \in E^* \mid w = a^n b^n \text{ with } n > 0\}$$

↑ THIS IS NOT A REGULAR LANGUAGE



$$m_0 = (1 \ 0 \ 0)^T$$

FINAL MARKINGS

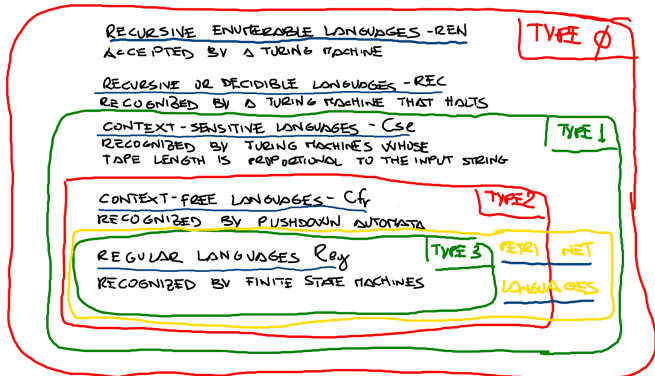
$$F = \{(0 \ 0 \ 1)^T\}$$

$$\leftarrow \mathcal{L}_m(\mathcal{S}_1) = L$$

Can Petri nets recognize any language?



LANGUAGE HIERARCHY BASED ON CHOMSKY GRAMMARS



- ① $Reg \subsetneq Cfr \subsetneq Cse \subsetneq REC \subsetneq REN \neq \Sigma^*$
- ② $Reg \subset \text{PETRI NET LANGUAGES} \subset Cse$
- ③ $(\text{PETRI NET LANGUAGES}) \cap Cfr \neq \emptyset$

- Given their twofold representation – **both graphical and algebraic** – **linear programming techniques** can be used to assess both structural and behavioral properties in Petri nets systems
- This is one of the main advantages when compared to automata

Example – Reachability

Given a net system $\mathcal{S} = \langle N, m_0 \rangle$ with m places, and a marking $m \in \mathbb{N}^m$, if the following problem is infeasible, then m is unreachable, i.e. $m \notin R(N, m_0)$

$$\begin{aligned} & \max \mathbf{0}^T \cdot \sigma \\ & \text{s.t.} \\ & \mathbf{C} \cdot \sigma = m - m_0 \\ & \sigma \geq \mathbf{0} \\ & \sigma \in \mathbb{R}^n \end{aligned}$$



M. Silva, E. Teruel and J. M. Colom

Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems

Lectures notes in computer science, 1998

Given a net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ with m places and $p \in P$, if the solution of the following ILP problem (**NP-hard**)

$$sb(p) = \max \mathbf{m}(p)$$

s.t.

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma}$$

$$\mathbf{m} \geq \mathbf{0}$$

$$\boldsymbol{\sigma} \geq \mathbf{0}$$

$$\mathbf{m} \in \mathbb{N}^m$$

$$\boldsymbol{\sigma} \in \mathbb{N}^n$$

is such that

$$sb(p) \leq k$$

then the place p is k -bounded

Example of linear programming relaxation

If the solution of the following LP problem (**polynomial time**)

$$sb(p) = \max \mathbf{m}(p)$$

s.t.

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$$

$$\mathbf{m} \geq \mathbf{0}$$

$$\sigma \geq \mathbf{0}$$

$$\mathbf{m} \in \mathbb{R}^m$$

$$\sigma \in \mathbb{R}^n$$

is such that

$$\lfloor sb(p) \rfloor \leq k$$

then the place p is k -bounded

Theorem

Given a net N , the following statements are equivalent

- N is structurally bounded, i.e. every place is bounded for every initial marking m_0
- There exists $\mathbf{x} > \mathbf{0}$ such that $\mathbf{x}^T \cdot \mathbf{C} \leq \mathbf{0}$ (integer feasibility problem with linear constraints)
- There does not exist any $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{C} \cdot \mathbf{y} \geq \mathbf{0}$ (integer feasibility problem with linear constraints)

Theorem – sufficient condition

Given a net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$, if the following feasibility problem **does not** admit any solution $\mathbf{m} \in \mathbb{N}^m, \sigma \in \mathbb{N}^n$

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$$

$$\sigma \geq \mathbf{0}$$

$$m(p) \geq 1$$

$$m(p') \geq 1$$

then the places p and p' are in *mutual exclusion*

Theorem – sufficient condition

Given a net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$, if the following feasibility problem **does not** admit any solution $\mathbf{m} \in \mathbb{N}^m, \sigma \in \mathbb{N}^n$

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$$

$$\mathbf{m} \geq \mathbf{0}$$

$$\sigma \geq \mathbf{0}$$

$$\bigvee_{p \in \bullet t} \mathbf{m}(p) < \mathbf{Pre}(p, t), \quad \forall t \in T$$

then \mathcal{S} is *deadlock free*

- Many tools
- Maybe even more than for automata (due to the infinite number of PN *subspecies*)
- A database is maintained at **Petri Nets World website**
<http://www.informatik.uni-hamburg.de/TGI/PetriNets/index.php>
- Many tools have been developed for *model-based verification and validation*
- Some tools
 - TINA (LAAS/CNR)
 - Petri Net Toolbox (University “Gh. Asachi” of Iasi)
 - SNAKES (Python toolkit for coloured Petri nets)
 - ...

- If the algebraic approach is used, tools to solve ILP problems are needed
 - Solvers: CPLEX, XPRESS, GLPK (free, <http://glpkmex.sourceforge.net/>),...
 - **YALMIP**: a useful Matlab parser for optimization problems
<https://yalmip.github.io/>

Petri nets vs automata – Pros...

- More powerful expressive power – not just regular languages
- More compact and modular representation
- Twofold representation that permits to use linear programming techniques

...and cons

- Lack of necessary and sufficient conditions to due practical semi-decidability of the reachability problem
- Many proposed algorithms are NP-hard ...which nowadays does not necessarily represent a problem

-  M. P. Cabasino, A. Giua, C. Seatzu
Introduction to Petri nets
in *Control of Discrete-Event Systems*
Springer, 2013
-  M. P. Cabasino, A. Giua, C. Seatzu
Structural analysis of Petri nets
in *Control of Discrete-Event Systems*
Springer, 2013

Petri nets and their twofold representation to model DES

Discrete Event Systems and Supervisory Control

Prof. Gianmaria DE TOMMASI
Email: detommas@unina.it

May 2021