Introduction to Petri Nets

Audine SUBIAS

CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Université de Toulouse, INSA, LAAS, F-31400 Toulouse, France

2024

Table des matières

1. Introduction
2. Basic Concepts
3. Composition of Petri Nets
4. Test modelling with Petri Nets
5. Reachability Set
6. Basic Properties
7. Coverability
8. Petri Nets analysis

To start

- Petri nets have been proposed in 1962 by Carl Adam Petri to model parallel and distributed systems (discrete event dynamic systems)
- The basic principle is to describe state changes in a transition system
- Nowadays
  - a powerful key for systems design and analysis
  - a wide field of research and development
  - an important community of academics and industrials

Welcome to the world of Petri Nets!
http://www.informatik.uni-hamburg.de/TGI/PetriNets/

To start (Cont'd)

- Petri Nets are both a graphical and mathematical formalism
- Petri Nets provide a compact way to model very large state spaces
- Petri nets allow to consider complex systems using modular representation and models composition
- Petri nets provide powerful tools for properties verification (*Model Checking*)
- A large spectrum of application fields (Flexible manufacturing systems, communicating systems, biology ...)

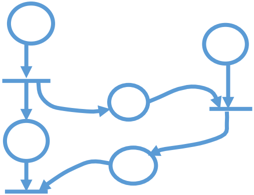
To start (Cont'd)

Informal presentation

- Petri Nets contain places and transitions potentially connected by directed arcs
- Places symbolise states and transitions symbolise actions
- The structure of the net gives the behavioral rules of the modeled system

A place

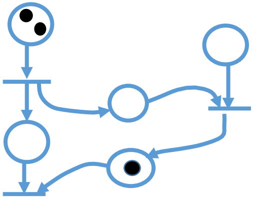
A transition



To start (Cont'd)

Informal presentation

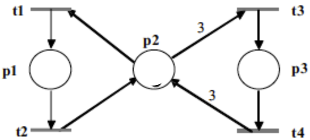
- Places may contain tokens that may move to others places
- The tokens distribution in the net represents the state of the system and transitions symbolize actions
- The state changes are caused by actions
- The execution of an action corresponds to a transition firing
  - Can all the actions be executed given the system state ?
  - if not, what are the allowed actions according to the system state?



Petri Net definition

Place/Transition Petri net
$N = < P, T, Pre, Post >$

- P a finite set of places;
- T a finite set of transitions with $P \cap T = \emptyset$
- Pre: $P \times T \to \mathbb{N}$: precedence function;
  - Pre(p,t) is the weight of the arc from the place p to the transition t
- Post : $P \times T \to \mathbb{N}$: post function;
  - Post(p,t) is the weight of the arc from the transition t to the place p



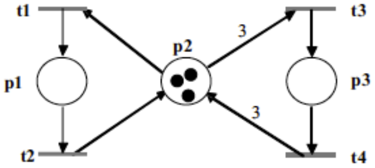
- $P = \{p1, p2, p3\}$
- $T = \{t_1, t_2, t_3, t_4\}$

Petri Net definition

- $Pre = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $Pre(.,t_3) = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$
- $Post = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
- $C = -Pre + Post = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix}$

- Pre(.,t) (resp. Post(.,t) denotes the column of Pre (resp. of Post) relative to t (i.e the set of **input places** (resp. the set of **output places**))
- $Pre(p_2, t_3) = 3$ means that the weight of the arc from the place p_2 to the transition t_3 is equal to 3.

Petri Net definition



Marked Petri net $< N, M_0 >$

- $M : P \to \mathbb{N}$ marking function ; $M \in \mathbb{N}^P$
- $M_0 : P \to \mathbb{N}$ the initial marking
- $M(p)$ denotes the number of tokens in place p in marking M
- Token can be either denoted by black circle or by a number

- Initial marking $M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$
- $M_0(p_1) = 0$; $M_0(p_2) = 3$; $M_0(p_3) = 0$

Petri Net alternative definition

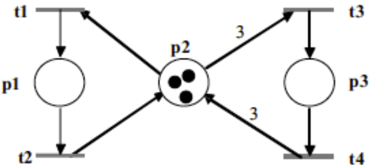
Place/Transition Petri net $N =< P, T, A, W >$

- P a finite set of places
- T a finite set of transitions with $P \cap T = \emptyset$
- A $\subseteq (P \times T) \cup (T \times P)$: the flow relation
- W: A $\rightarrow (\mathbb{N} \setminus \{0\})$: arc weight mapping

Let a $\in$ P $\cup$ T

- The set ${}^{\bullet}a = \{a' | < a', a > \in A\}$ is called the pre-set of a
- $a^{\bullet} = \{a' | < a, a' > \in A\}$ is its post-set.
- A marking is denoted as a set when no place contains more than one token or alternatively by a multiset ($M(p)$ denotes the number of time p is included in the multi-set i.e is marked).

Petri Net alternative definition



- $M_0 = \{p_2, p_2, p_2\}$ or $M_0 = \{3p_2\}$
- $p_1^\bullet = \{t_2\}$, ${}^\bullet p_2 = \{t_2, t_4\}$
- $t_1^\bullet = \{p_1\}$, ${}^\bullet t_3 = \{p2\}$
- $A = \{(t_1, p_1), (p_1, t_2), (t_2, p_2), (p_2, t_1), (p_2, t_3), \cdots (t_4, p_2)\}$
- $W(t_1, p_1) = 1$, $W(p_2, t_3) = 3$

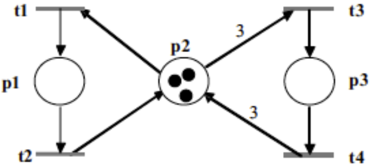
Petri Net alternative definition

Place/Transition Petri net $N =< P, T >$

- P a finite set of places
- T a finite set of transitions with $P \cap T = \emptyset$
- $T \subset \mathbb{N}^P \times \mathbb{N}^P$
- a transition $t \in T$ is denoted by $({}^\bullet t, t^\bullet)$

- With this definition , ${}^\bullet t, t^\bullet, M \in \mathbb{N}^P$

Petri Net alternative definition



- $M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$ $t_1 = (\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix})$
- ${}^\bullet t_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $t_4^\bullet = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} = {}^\bullet\, t_3$

Table des matières

1 Introduction

2 Basic Concepts

3 Composition of Petri Nets

4 Test modelling with Petri Nets

5 Reachability Set

6 Basic Properties

7 Coverability

8 Petri Nets analysis

Basic concepts

- A source transition is a transition without input place i.e $^\bullet t = \emptyset$
- A sink transition is a transition without output place i.e $t^\bullet = \emptyset$
- A source place is a place without input transition i.e $^\bullet p = \emptyset$
- A sink place is a place without output transition i.e $p^\bullet = \emptyset$

Basic concepts

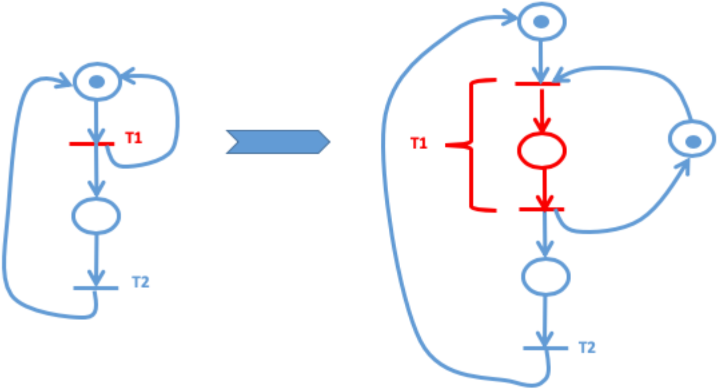
- A self-loop is couple (p, t) s.t t is both input and output transition of p
- A path is a sequence of nodes $n_1 n_2 \cdots n_n$ s.t n_{i+1} is an output node of n_i
- A circuit is a path s.t $n_n = n_1$

Basic concepts

Pure Petri net

A Petri net free of self-loop is said pure i.e $\forall t \in T$, $t^\bullet \cap {}^\bullet t = \emptyset$

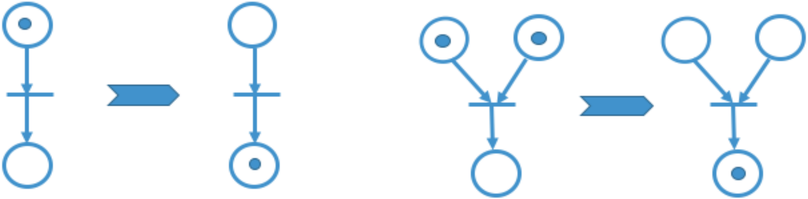
Any not pure Petri net can be transformed into a pure Petri net



Spate Space Notion

- The state of a Petri net is given by the location of the tokens in the places
- The state changes when enabled transitions are fired
- A transition is enabled when each of its input place contains a sufficient number of tokens
- The firing of an enabled transition removes token(s) from each input place (i.e pre-places) and produces token(s)in each output place (i.e post-places)

Evolution of a Petri Net



Enabling

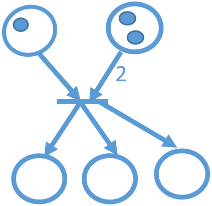
Enabling

A transition t is enabled at a marking M if $\forall p \in P$, $M(p) \geqslant Pre(p,t)$ i.e $M \geqslant Pre(.,t)$.

To denote that t is enabled at M we write $M[t >$ or $M \xrightarrow{t}$

- Pre(.,t) is the minimum marking necessary to fire t
- Post(.,t) is the minimum marking reached after firing t
- Equivalent notation:
  for a transition $t({}^\bullet t, t^\bullet)$, t is enabled for M iff $M \geqslant {}^\bullet t$
- A source transition is always enabled

Enabling



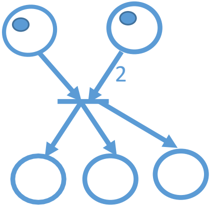
Enabled

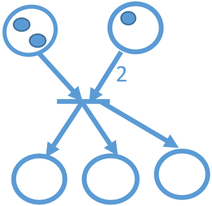
- $M_0 = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and ${}^\bullet t = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $M_0 \geqslant {}^\bullet t$

Enabled

- $M_0 = \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and ${}^\bullet t = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $M_0 \geqslant {}^\bullet t$

Enabling





Not enabled

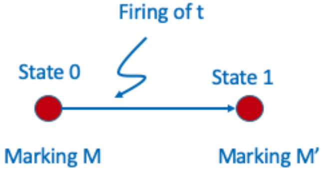
- $M_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and ${}^\bullet t = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $M_0 \ngeq {}^\bullet t$

Not enabled

- $M_0 = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and ${}^\bullet t = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $M_0 \ngeq {}^\bullet t$

Firing

- A transition t enabled at a marking M can be fired.
- The firing of t yields to the successor state i.e successor marking M':



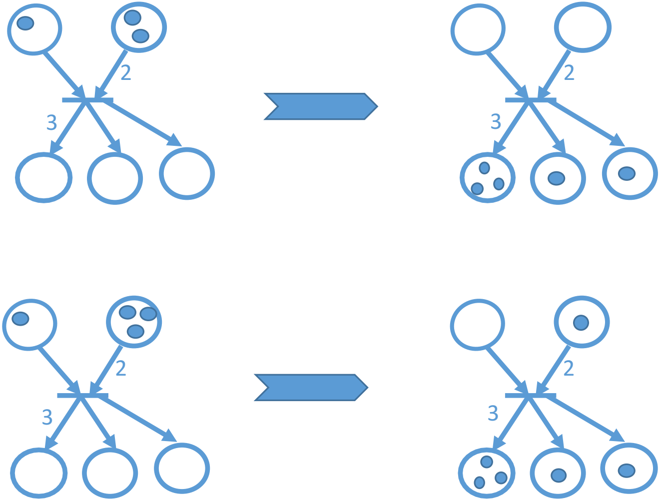
Firing

Firing

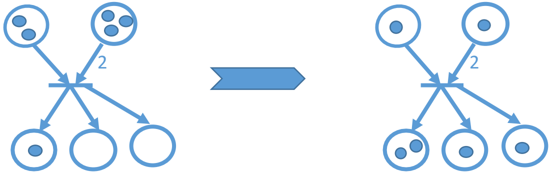
A transition t enabled at a marking M can be fired. The firing of t yields to the successor marking M': $M' = M - Pre(.,t) + Post(.,t) = M + C(.,t)$.
The firing of t is denoted: $M[t > M'$ or $M \xrightarrow{t} M'$

- The firing of a transition is an atomic operation since the removal of tokens from input places and their addition in output places occurs in an indivisible way.
- There is no conservation between the number of tokens removed and the number of tokens produced.
- Equivalent notation:
  for a transition $t(^\bullet t, t^\bullet)$, the successor marking M' is given by:
  $M' = M -^\bullet t + t^\bullet$

Firing



Firing



Firing

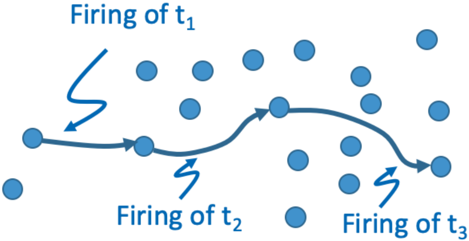
- $M_0 = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ and ${}^\bullet t = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and $\mathrm{t}^\bullet = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$
- $M_0 \xrightarrow{t} M_1$ and $M_1 = M_0 -^\bullet t + t^\bullet = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}$

Firing sequence

One consider successive firing of transitions

Firing sequence

A firing sequence at marking M_0 is a finite string of transitions $s = t_1 t_2 \cdots t_m$ such that $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} M_3 \cdots \xrightarrow{t_m} M_m$



Firing sequence

- To denote that the firing of the sequence s from M_0 leads to the marking M_m we write $M_0 \xrightarrow{s} M_m$ or $M_0[s > M_m$
- A firing sequence s is characterised by its counting vector $\bar{s} \in \mathbb{N}^{\mathbb{T}}$
- $\bar{s}(t)$ is the number of occurrences of transition t in the sequence s
- $s = t_1 t_2 t_1 t_1 t_5$; $T = \{t_1, t_2, t_3, t_4, t_5\}$; $\bar{s} = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
- To one counting vector several sequences can be associated, not only firing sequences i.e feasible sequences. e.g $s = t_1 t_2 t_1 t_1 t_5$; and $s = t_1 t_1 t_5 t_1 t_2$;

Firing sequence

Fundamental State Equation of a Petri net

For a firing sequence s such that $M_0 \xrightarrow{s} M_m$ the reachable marking is given by:

$$M_m = M_0 + C\bar{s}$$

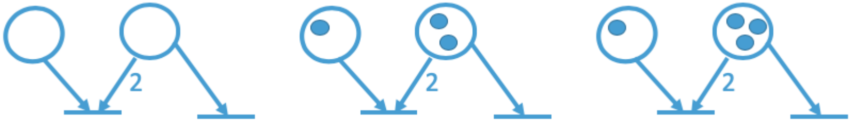
- It is a sufficient condition i.e the marking obtained from the state equation are not necessary reachable i.e the makings obtained are an over-approximation of the actual reachable markings.
- The state equation can be used to show that a marking M is unreachable, if $M_m = M_0 + C\bar{s}$ has no solution for $\bar{s}$ with integer numbers.

Conflicts and Concurrency

Transitions in conflict are both enabled but the marking M does not contain enough tokens to allow the firing of both transitions.

Conflict

Two transitions t and t' are in conflict if there exists a place p with a pre arc to both t and t'. Given a marking m, we say that transitions t and t' are in behavioral conflict if $M \geqslant Pre(.,t)$ and $M \geqslant Pre(.,t')$ but $M \ngeqslant Pre(.,t) + Pre(.,t')$.

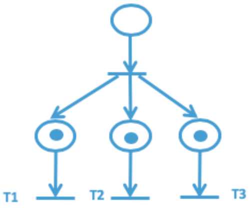


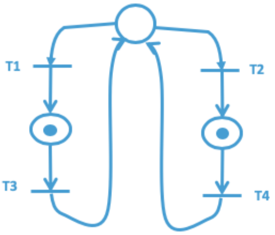
Conflicts and Concurrency

The firings of the transitions are independent.

Concurrency

Two transitions t and t' are in concurrency if $Pre(.,t)^T * Pre(.,t') = 0$ i.e there exists no place p with a pre arc to both t and t'. Given a marking M, we say that transitions t and t' are in behavioral concurrency or parallelism if $M \geqslant Pre(.,t)$ and $M \geqslant Pre(.,t')$.





Conflicts and Concurrency

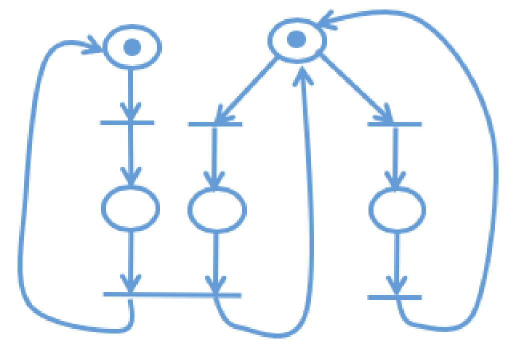
Free-Choice net

- In a free-choice Petri net every arc from a place p to a transition t is either the only arc from that place (i.e t is the only output transition) or the only arc to that transition (i.e p is the only input place of t), $\forall p \in P : (|p^\bullet| \leq 1) \vee ({}^\bullet(p^\bullet) = \{p\})$
- In a free-choice Petri net there can be both concurrency and conflict, but not at the same time.
- In a free-choice net a transition in conflict with an enabled transition is also enabled.

Choices are free: the result of the choice does not depend on the rest of the system

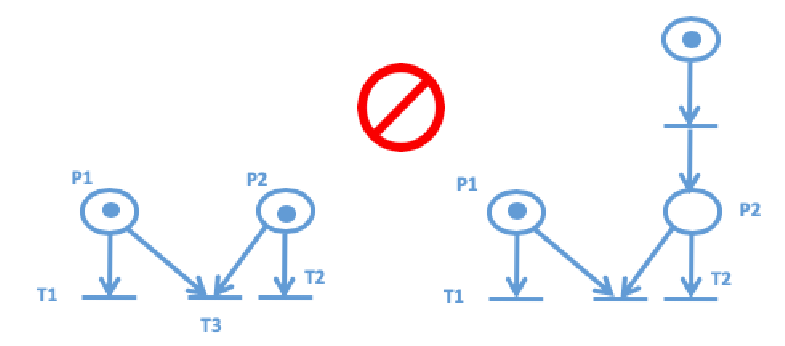
Conflicts and Concurrency

Free-Choice net



Conflicts and Concurrency

Free-Choice net



Conflicts and Concurrency

State Machine

- A State machine is a Petri Net s.t every transition has one input arc, and one output arc, and all markings have exactly one token.
  $\forall t \in T : |t^{\bullet}| = |{}^{\bullet}t| = 1$
- no concurrency - conflict is possible

Event graph

- An Event Graph or Marked Graph (MG) s a Petri Net s.t every place has one input arc and one output arc. $\forall p \in P : |p^{\bullet}| = |{}^{\bullet}p| = 1$
- no conflict - concurrency is possible

Table des matières

1 Introduction

2 Basic Concepts

3 Composition of Petri Nets

4 Test modelling with Petri Nets

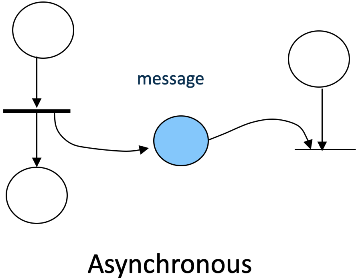
5 Reachability Set

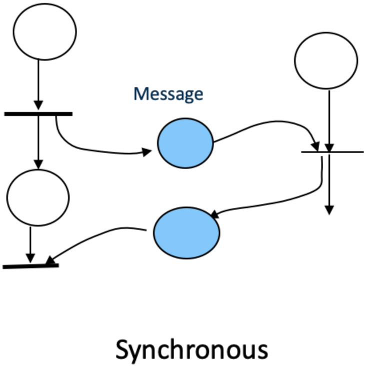
6 Basic Properties

7 Coverability

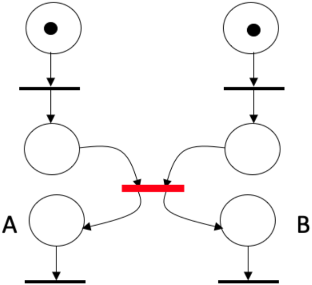
8 Petri Nets analysis

Communication

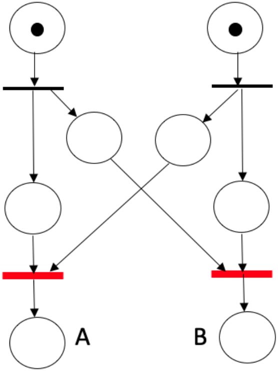




Communication



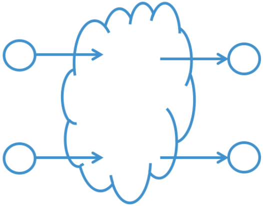
Rendez-Vous

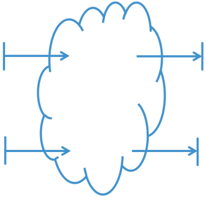


Rendez-Vous
with message

Communication and composition

Input/output of a process may be modeled by either external places or external transitions





Place Fusion

Processes share some places of communication
To model asynchronous communications

Transition fusion

Processes share some transitions
To model synchronous communications

Communication

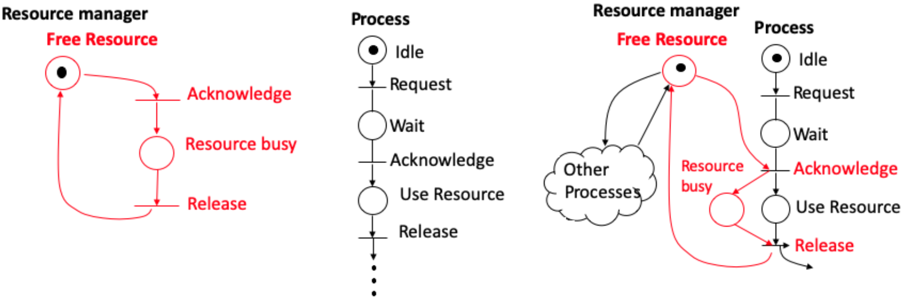


Table des matières

1 Introduction

2 Basic Concepts

3 Composition of Petri Nets

4 Test modelling with Petri Nets

5 Reachability Set

6 Basic Properties

7 Coverability

8 Petri Nets analysis

Test modelling

- Mechanisms may need to be put in place to allow for a read and a comparison of values with respect to a threshold, without altering the values.
- The expression of tests by ordinary Petri net is not easy it is a limitation to the expression power
- Extensions to the basic model allow this aspect to be taken into account.

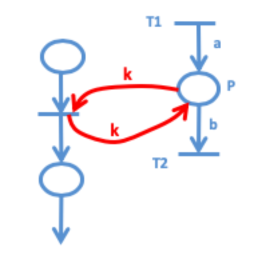
Enabling test

- In a Petri net the only existing test mechanism is the comparison of the number of token from a place with the weight of the arcs form the input place. This is the enabling test of a transition.
- If we consider that a place is a variable and that the number of tokens it contains is the value of this variable, this mechanism leads to an alteration of the data as M is modified by the firing

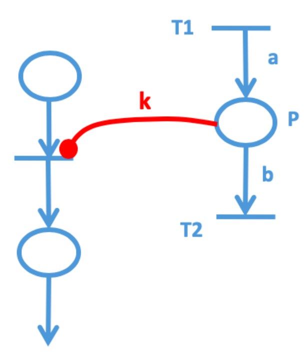
Test by elementary loop

- the test is not visible in the incidence matrix C
- the problem of data alteration is still present
- one solution: a read arc

Elementary loop



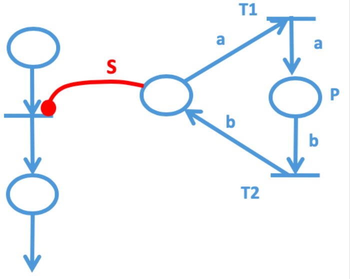
Read arc



Test of an empty place

- Given a place bounded p s.t $M(p) \leqslant S$
- Use of a complementary place p'
- with $M(p) + M(p') = S \; \forall M \in R(N, M_0)$

Test by complementary place



- Another solution: an inhibitor arc

Inhibitor arc

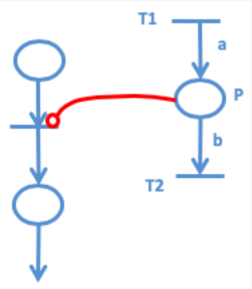


Table des matières

1 Introduction

2 Basic Concepts

3 Composition of Petri Nets

4 Test modelling with Petri Nets

5 Reachability Set

6 Basic Properties

7 Coverability

8 Petri Nets analysis

Reachability set

- For a net N, the set of reachable markings from the initial marking M_0 (the reachability set) is the set of markings that can be reached from the marking M_0 i.e the set $R(N, M_0) = \{M_i \in \mathbb{N}^{\mathbb{P}}, \exists s / M_0 \xrightarrow{s} M_i\}$
- If the reachability set is finite, an exhaustive enumeration is possible and the reachability graph (or the reachability tree) of the net is constructed.
- If the reachability set is not finite, a finite coverability graph can still be constructed. The coverability graph provides a larger approximation of the reachable state space.

Reachability set

Reachability graph

The reachability graph of a Petri net $N =< P, T, Pre, Post, M_0 >$ is a rooted directed graph $G =< V, E, v_0 >$ with

- $v_0 = M_0$ the root node i.e the initial marking,
- $V = R(N, M_0)$ the vertices i.e the set of all reachable markings
- $E = \{(M, t, M')/M \in V$ the edges and $M \xrightarrow{t} M'\}$ i.e there is an edge from each marking (vertex) M to each of its successor markings and the edge is labelled with the firing transition.

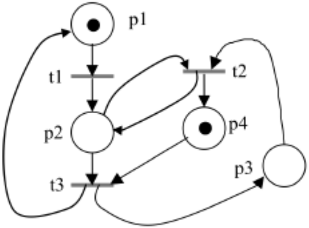
The reachability tree is a graph without loops and without duplicated nodes

Reachability graph

Algorithm

1. The initial marking is the root node of the graph; this node is labelled `new`
2. While `new` markings exits do
    - Select a marking M
    - For each transition t enabled at M (i.e $M \geqslant Pre(.,t)$)
        - Compute the marking $M' = M + C(.,t)$ reached from M firing t
        - If M' does not appear in the Graph, add M' and tag it `new`
        - Draw an arc with label t from M to M'
    - Label the node M `old`

Reachability graph



Reachability Graph

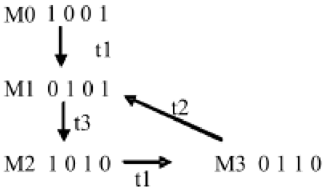


Table des matières

1 Introduction

2 Basic Concepts

3 Composition of Petri Nets

4 Test modelling with Petri Nets

5 Reachability Set

6 Basic Properties

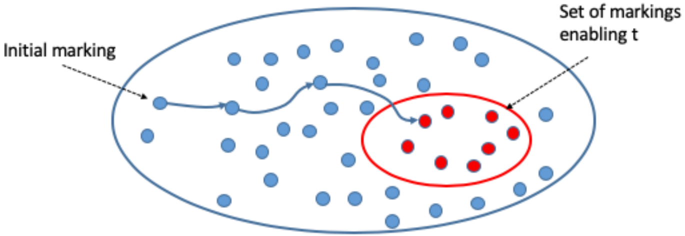
7 Coverability

8 Petri Nets analysis

Basic properties: intuitive view

Given a marked Petri Net N with an initial marking M_0

- (N, M_0) is bounded: the number of tokens in each place (i.e in the net) is limited
- (N, M_0) is live : any transition can be fired from any marking with a suitable firing sequence
- (N, M_0) is reversible: from any marking there is path to reach the initial state
- (N, M_0) is deadlock-free: from any marking there is a path to evolve.
- (N, M_0) is infinitely active: it exists one infinite path to evolve.

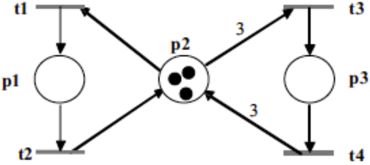


Boundedness

- A place $p \in P$ is k-bounded if and only if $\forall M \in R(N, M_0), M(p) \leqslant k \in \mathbb{N}$
- If $k = 1$ the place is safe
- A net (N, M_0) is k-bounded if and only if all the places are k-bounded
- A net (N, M_0) is safe if and only if for any $M \in R(N, M_0)$ and any $p \in P : M(p) \leqslant 1$

(N, M_0) is bounded if and only if there is a $k \in \mathbb{N}$ s.t for any $M \in R(N, M_0)$ and any $p \in P : M(p) \leqslant k$

Boundedness

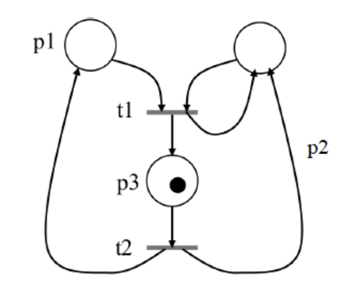


For $M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$ $P3$ is safe; $P1$ and $P2$ are 3-bounded, the net is bounded

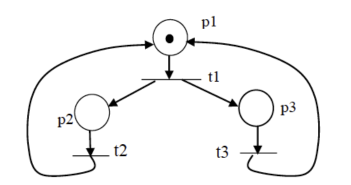
For $M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}'$ the net is safe

Boundedness

Unbounded



Unbounded



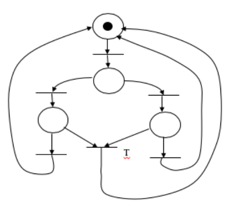
Liveness

- A transition $t \in T$ is 1-live (1L) iff there exists $M \in R(N, M_0)$ and $M' \in R(N, M)$ s.t $M'[t >$
- A transition $t \in T$ is live iff **for any** $M \in R(N, M_0)$ there exists $M' \in R(N, M)$ s.t $M'[t >$ (i.e $\forall M \in R(N, M_0) \exists s/M[s.t >$)
- A net (N, M_0) is 1-live iff all the transitions are 1-Live
- A net (N, M_0) is live iff all the transitions are live

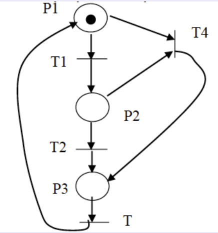
(N, M_0) is live iff $\forall t \in T$ and $\forall M \in R(N, M_0)$ $\exists M' \in R(N, M)$ s.t $M'[t >$

Liveness

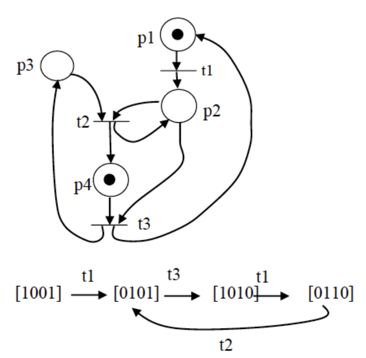
Not 1-live,Not live



Not 1-live, not live



live



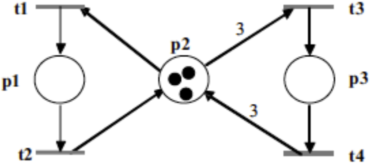
Reversible and Home Marking

- A marking M' is a Home Marking if for any $M \in R(N, M_0)$: $M' \in R(N, M)$
- A net is reversible if and only if for any $M \in R(N, M_0) : M_0 \in R(N, M)$
- A net is reversible if and only if M_0 is a *Home Marking*

(N, M_0) is reversible if and only if for any $M \in R(N, M_0) : M_0 \in R(N, M)$

A reversible and 1-Live net is live

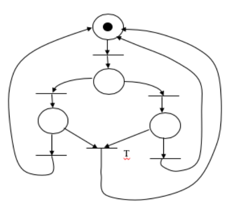
Reversible and Home Marking



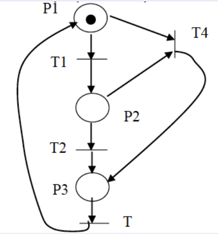
- For $M_0 = [030]'$ all the transitons are 1-Live, the net is 1-Live
- For $M_0 = [030]'$ all the transitons are Live, the net is Live
- For $M_0 = [030]'$ the net is reversible
- For $M_0 = [010]'$ the net is reversible t_1 and t_2 are live

Reversible

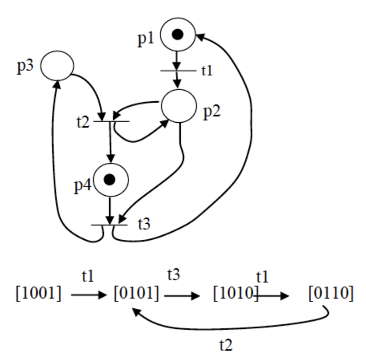
Reversible



Reversible



Not reversible



[0101]' is a Home marking if $M_0 = [1010]'$: net is reversible

Deadlock-free

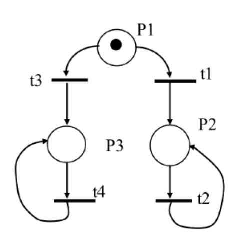
- A marking $M \in R(N, M_0)$ is a deadlock marking if $\nexists t \in T$ s.t $M[t >$
- A net is deadlock-free if no reachable marking is a deadlock
- A deadlock marking implies that the net is not live
- A net can be deadlock-free with no live transitions
- If one transition is live the net is deadlock-free

(N, M_0) is deadlock-free if and only if for any $M \in R(N, M_0)$ there exists a transition t s.t $(N, M)[t >$

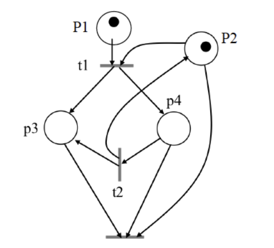
- **A reversible net is deadlock-free**
- **A live net is deadlock-free**

Deadlock-free

Deadlock-free but not live



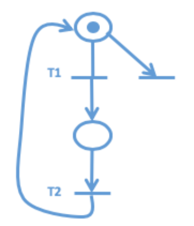
Not deadlock-free then not live ($M_0[t_1t_2 > M_2 = [0120]'$)



Infinitely active

Not infinitely active

Infinitely active but not deadlock free



(N, M_0) is infinitely active iff $\forall k \in \mathbb{N}, \exists M \in R(N, M_0), \exists \sigma \in T^k$ s.t $M[\sigma >$

- **An unbounded net is infinitely active**
- **A deadlock-free net is infinitely active**

Basic properties

Given a marked Petri Net N with an initial marking M_0

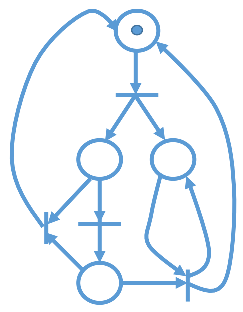
- (N, M_0) is bounded iff $\exists k \in \mathbb{N}$ s.t $\forall M \in R(N, M_0)$ and $\forall p \in P : M(p) \leqslant k$
- (N, M_0) is live iff $\forall t \in T$, $\forall M \in R(N, M_0), \exists M' \in R(N, M)$ s.t $M'[t >$
- (N, M_0) is reversible iff $\forall M \in R(N, M_0)$: $M_0 \in R(N, M)$
- (N, M_0) is deadlock-free iff $\forall M \in R(N, M_0) \exists t$ s.t $M[t >$
- (N, M_0) is infinitely active iff $\forall k \in \mathbb{N}, \exists M \in R(N, M_0), \exists \sigma \in T^k$ s.t $M[\sigma >$

Lets try !!

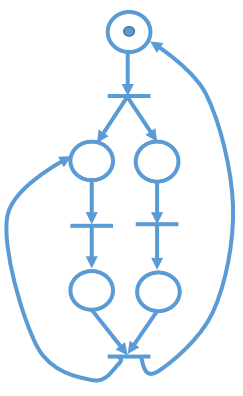
Quizz ...

For each net indicate if it is Bounded, Reversible, Live, Deadlock-free, L1-Live, Infinitely Active.

Net 1

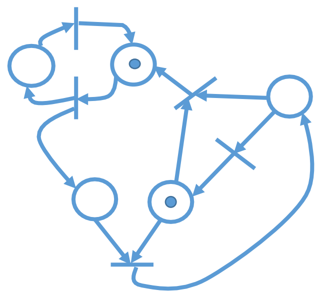


Net 2

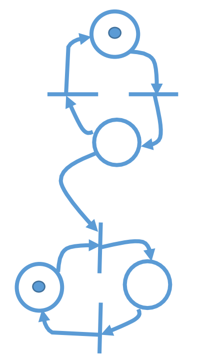


Quizz ...

Net 3

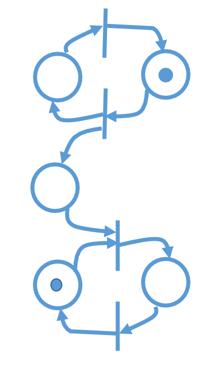


Net 4

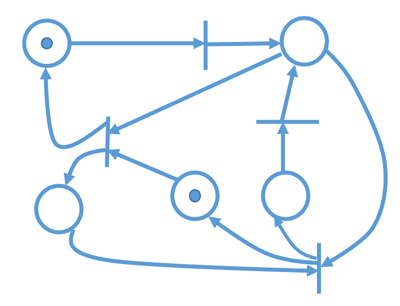


Quizz ...

Net 5

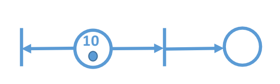


Net 6

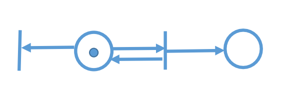


Quizz ...🙂 🙂

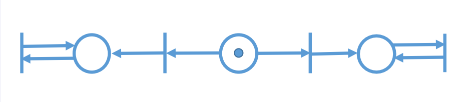
Net 7



Net 8

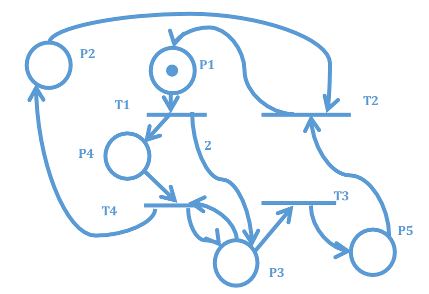


Net 9



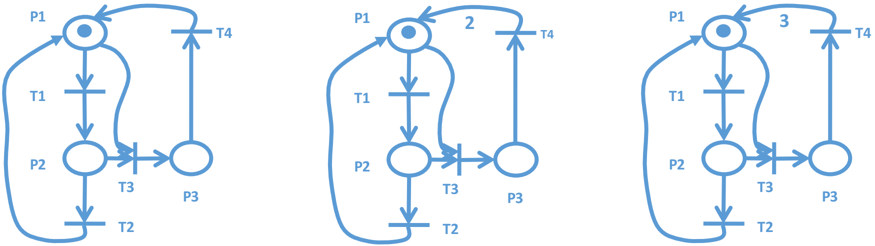
Quizz ...

Net 10



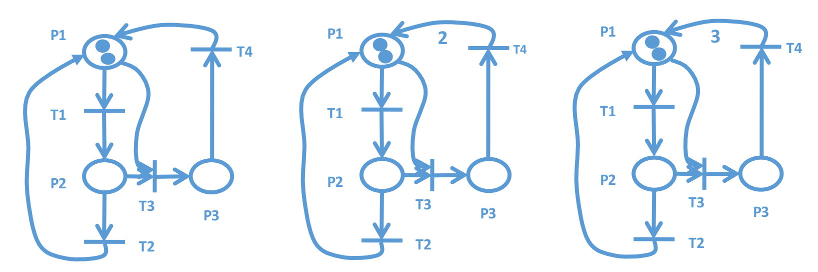
Quizz ...

Net(s) 11



Quizz ...☺ ☺

Net(s) 12



Quizz ...☺ ☺

Results

Net	**B**	**L**	**R**	**D**	**L1-L**	**IA**
Net 1						
Net 2						
Net 3						
Net 4						
Net 5						
Net 6						
Net 7						

B(Bounded) - L(Live) - R(Reversible) - D (Deadlock) - L1-L (L1-live) -IA (Infinitely active)

Quizz ...☺ ☺

see Wooclap

Results

Net	**B**	**L**	**R**	**D**	**L1-L**	**IA**
Net 8						
Net 9						
Net 10						
Net 11a						
Net 11b						
Net 11c						
Net 12a						
Net 12b						
Net 12c						

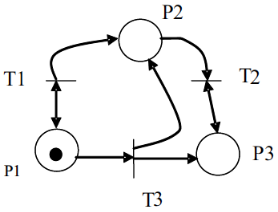
B(Bounded) - L(Live) - R(Reversible) - D (Deadlock) - L1-L (L1-live) -IA (Infinitely active)

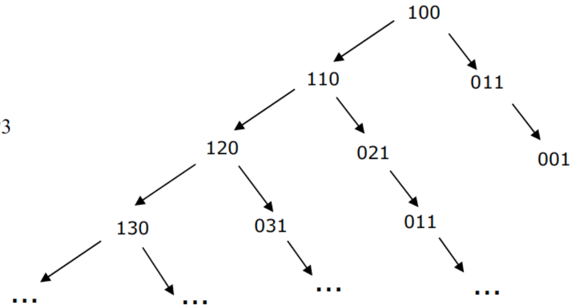
Table des matières

1. Introduction
2. Basic Concepts
3. Composition of Petri Nets
4. Test modelling with Petri Nets
5. Reachability Set
6. Basic Properties
7. Coverability
8. Petri Nets analysis

Coverability notion

What happens when the reachability set i.e state space is infinite ?





Coverability Graph

- If the reachability set is infinite, an exhaustive enumeration is not possible
- The building of the reachability graph is not possible any longer
- Another way to represent the state space is considered based on the coverability notion

Coverability

A marking M' covers a marking M (noted $M' \geqslant M$) Iff $\forall p \in P, M'(p) \geqslant M(p)$

- $\begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}$ covers $\begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}$
- $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ does not cover $\begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}$

Coverability

Let us consider M and M' s.t $M' \geqslant M$
Then it holds: $\forall t \in T$, if $M[t >$ then $M'[t >$
If M' has at least as many tokens as M has (on each place) then M' enables at least the same transitions as M does.
This can be extended to sequences of transitions: $\forall \sigma$, if $M[\sigma >$ then $M'[\sigma >$

Assume $\exists M$ and σ s.t, $M[\sigma > M'$ and $M' \geqslant M$
then there is a marking M'' s.t $M'[\sigma > M''$
Let $\Delta M = M' - M$
Then $M'' - M' = C\bar{\sigma} = M' - M$ and $M'' = M' + \Delta M = M + 2\Delta M$
$M'' \geqslant M$ then $M''[\sigma >$

The transition sequence σ can be fired indefinitely

Coverability and Repetitive sequence

Repetitiveness of a sequence of transitions ensures that the sequence can occur indefinitely.

A sequence of transitons σ is said repetitive if it can be fired an infinite number of times from M: $\forall M$ and $M' \in R(N, M_0)$, $M[\sigma > M'$ and $M' \geqslant M$

A marked net is repetitive if there exists a repetitive sequence.

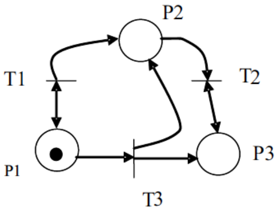
Coverability and repetitive sequence

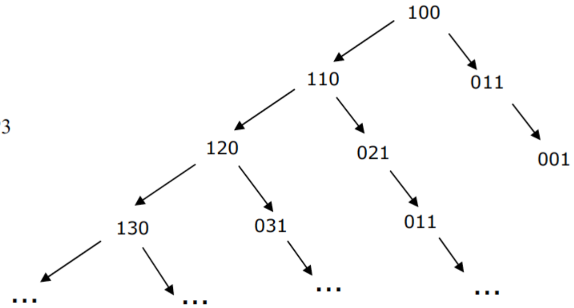
- A **repetitive sequence** σ said stationary if $M[\sigma > M$
- A **repetitive sequence** σ is said increasing if $M[\sigma > M'$ and $M' > M$
- A **repetitive sequence** σ is said complete if $\forall t \in T$, $\bar{\sigma}(t) \neq 0$ (with $\bar{\sigma}$ the counting vector of σ)

A marked net is bounded if and only if it does not admit increasing repetitive sequences.

Coverability Tree/Graph

An increasing repetitive sequence induces an infinite state space





Coverability Tree/Graph

- If the reachability set is infinite, an exhaustive enumeration is not possible
- A finite graph called the coverability graph can still be constructed.
- The coverability graph provides a larger approximation of the reachable state space.

Coverability Tree/Graph

- The coverability graph is based on the notion of *ω-marking*
- The symbol ω represents an arbitrarily huge quantity of tokens

  - $\omega + n = \omega$
  - $\omega - n = \omega$
  - $n < \omega$
  - $\omega \leqslant \omega$

- In an *ω-marking* each place p will either have $n \in \mathbb{N}$ tokens or ω tokens
- *ω-marking* : $P \rightarrow \mathbb{N} \cup \omega$

Coverability Tree/Graph

- The coverability graph needs the construction of the coverability tree
- The coverability tree is a graph with no loops and where duplicated nodes may exist.
- The coverability tree algorithm is almost exactly the same as the reachability tree algorithm but the ending conditions are different.

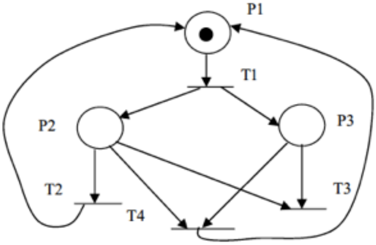
Coverability Tree/Graph

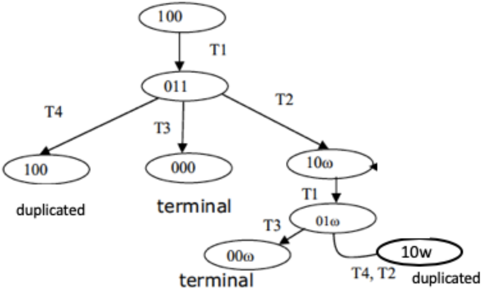
Coverability tree Algorithm

1. The initial marking is the root node of the tree; this node is labelled `new`
2. While `new` nodes exits do
   - Select a marking M
   - For each transition t enabled at M (i.e $M \geqslant Pre(.,t)$)
     - Compute the marking $M' = M + C(.,t)$ reached from M firing t
     - For all markings $M'' \leqslant M'$ on the path from the root node M_0 to the node M and for all $p \in P$, if $M''(p) \leqslant M'(p)$ then $M'(p) = \omega$
     - add M' and tag it `new`
     - Draw an arc with label t from M to M'
     - If there already exists a node M', label the new node `duplicated`
   - Label the node M `old`

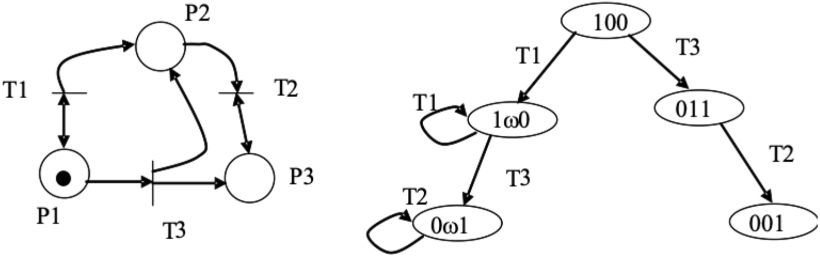
Coverability Graph

The coverability graph is obtained from the coverability tree by merging the duplicated nodes and by redirecting the arcs.

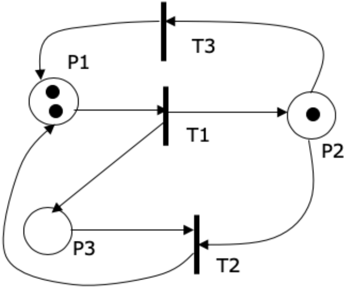


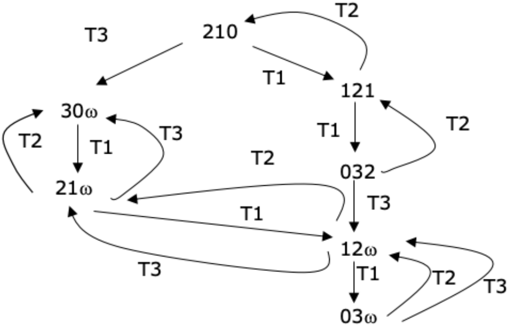


Coverability Graph



Coverability Graph



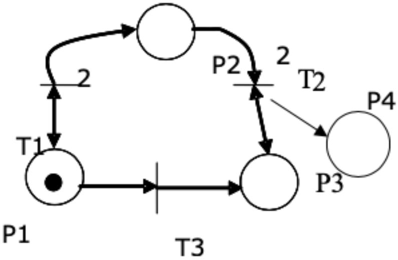


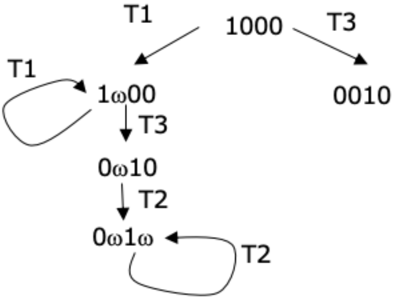
Coverability Graph

The coverability tree/graph gives an over-approximation.

- Coverability property: $CS(R, M_0)$ *covers* $R(N, M_0)$: $\forall M \in R(N, M_0)$, $\exists M' \in CS(R, M_0)$ s.t $M \leqslant M'$
- **If a marking M of net N is reachable from M_0, then M is covered by some vertex of the coverability graph of N**
- **A marking that is covered by some vertex of the coverability graph is not necessarily reachable**

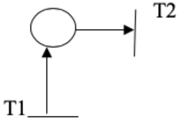
Coverability Graph

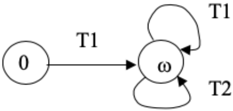




$M = [0315]'$ is covered but not reachable and $M = [0411]'$ is covered and reachable by $T1T1T1T3T2$

Coverability Graph





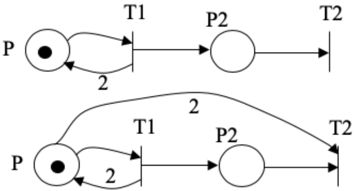
The path labeled $T1T2T2$ appears in the *Coverability Graph* but the corresponding transition sequence does not exist in the net.

Coverability Graph

- The lemne of Karp and Miller proved that the Algorithm always terminates in a finite number of states
- A marked Petri net is bounded if and only if the corresponding coverability tree/graph contains only ω-free markings.
- The coverability graph and reachability graph are identical if the marked Petri net is bounded
- Different Petri nets may have the same coverability tree/graph.

Coverability Graph

Same coverability graph but two different nets



The sequence $T1T2$ leads to different markings ([20]' vs [00]')

Table des matières

1. Introduction
2. Basic Concepts
3. Composition of Petri Nets
4. Test modelling with Petri Nets
5. Reachability Set
6. Basic Properties
7. Coverability
8. Petri Nets analysis
   - Enumerative Approach
   - Structural Approach

Properties Analysis

- Enumerative approaches : based on **reachability space** ✔
  ➽ properties depending on initial marking
- Structural approaches : based on **Invariants** ✔ or **traps** & **siphons** ✔
  ➽ properties are `true` whatever the initial marking
- Reduction approaches : based on an **equivalent net** obtained from the initial by **reduction rules** ✔

✔ seen in this lecture ✔ not seen in this lecture

Enumerative approaches

Properties Analysis: why and how ?

Reachability property

Goal?
- Does one particular situation can be reached?

Application example?
- To check that a bad situation never occurs e.g use of a transition observer to check mutual exclusion property
- To check that a problem has a solution

How?
-Based on the reachability graph an exact answer is obtained.
-Based on the coverability graph only the absence of a bad state can be certain if no vertex covers the bad state.

Properties Analysis: why and how ?

Enabledness property/Live transitions

Goal?

- Does one particular transition can be enabled ?

Application example?

- To check that a bad transition i.e action never occurs
- To detect that a good transition is 1-Live
- To check a good action can be infinitely repeated if needed
- To check that a system can be reinitialised

How?

-Based on the reachability graph an exact answer is obtained looking for a the liveness properties

-Based on the coverability graph only results on existing dead transitions or deadlocks can be obtained

Properties Analysis: why and how ?

Boundedness

Goal?
- Does a particular place is bounded ?

Application example?
- To check the use of limited resources

How?
-Based on the reachability graph an exact answer is obtained looking for the markings.
-Based on the coverability graph unbounded places and bounded places can be determined.

A wolf story

A man is travelling with a wolf, a goat, and a cabbage. The four come to a river that they must cross. There is a boat available for crossing the river, but it can carry only the man and at most one other object. The wolf may eat the goat when the man is not around, and the goat may eat the cabbage when unattended.

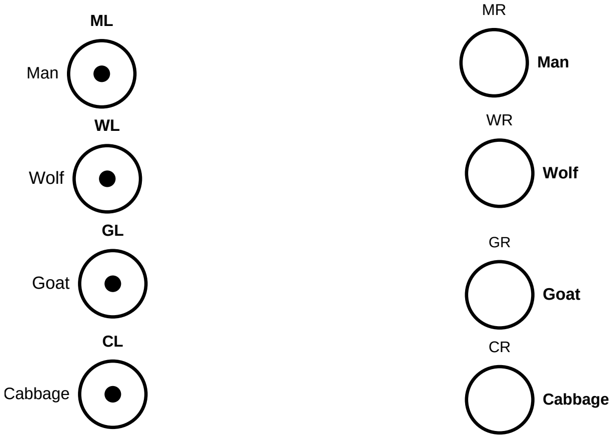
- Can the man bring everyone across the river without endangering the goat or the cabbage?
- And if so, how?

A wolf story

- Several objects associated to places : *Man*, *Wolf*, *Goat*, *Cabbage*, *Boat*
- Each object can be on either side of the river
- note that the *boat* and the *man* can be merged as they are always on the same side of the river
- several actions in this problem: crossing the river, the wolf eats the goat, the goat eats the cabbage
- Actions are modeled by transitions

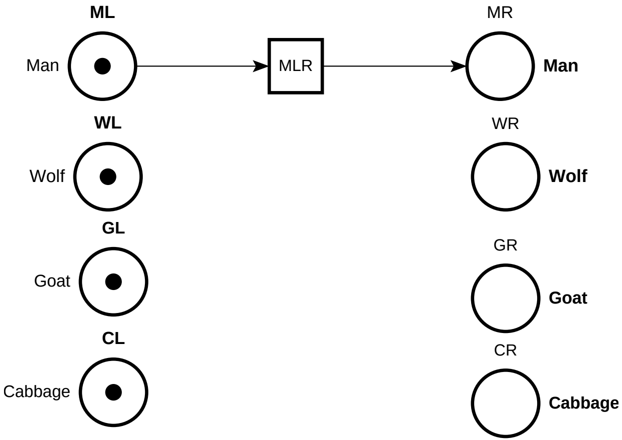
- 4 places to indicate the left position of each object: ML,WL,GL and CL
- 4 places to indicate the right position of each object: MR,WR,GR and CR
- Initially all the objects are on the left river side

A wolf story


ML
Man
WL
Wolf
GL
Goat
CL
Cabbage
MR
Man
WR
Wolf
GR
Goat
CR
Cabbage

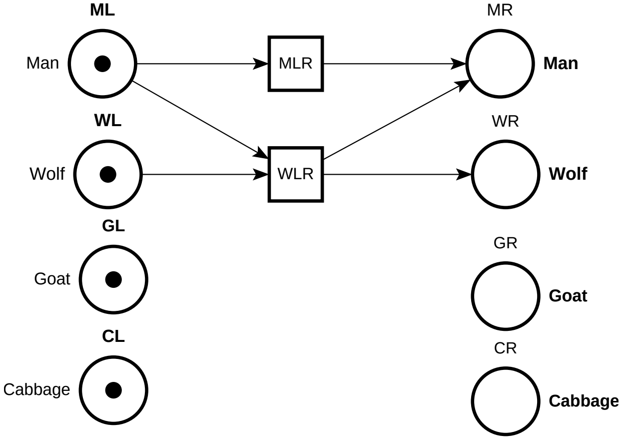
A wolf story

Crossing the river: left to right

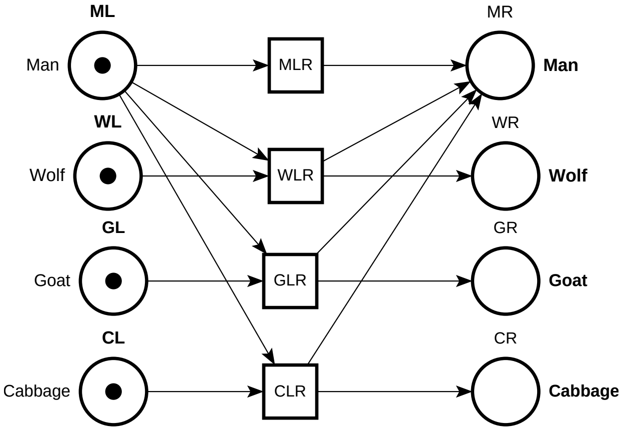


A wolf story

Crossing the river: left to right

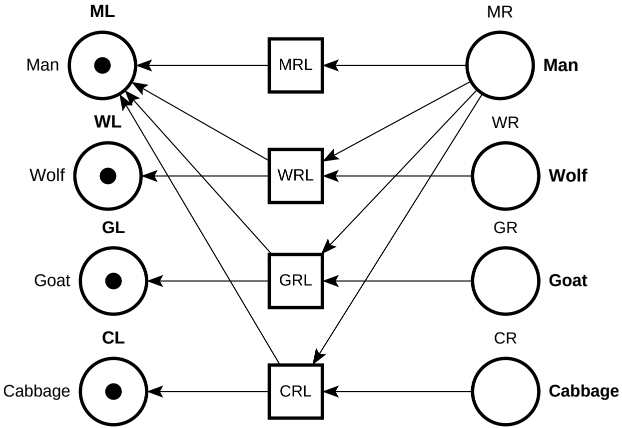


Crossing the river: left to right



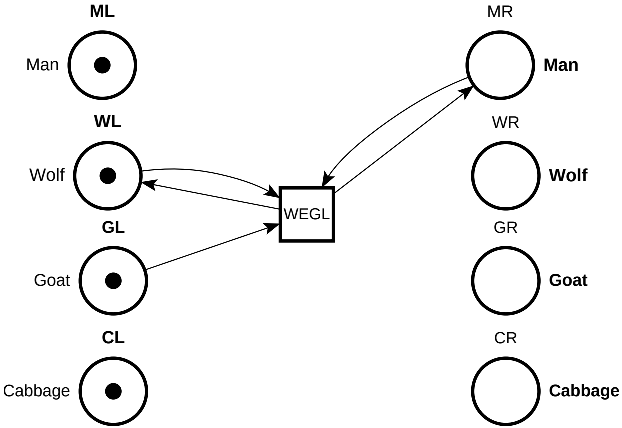
A wolf story

Crossing the river: right to left



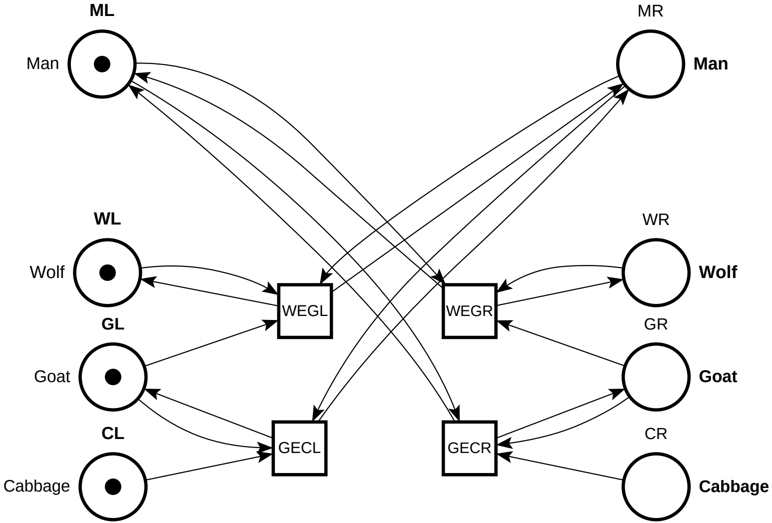
A wolf story

Wolf eats goat !



A wolf story

Wolf eats goat and goat eats cabbage !!



A wolf story

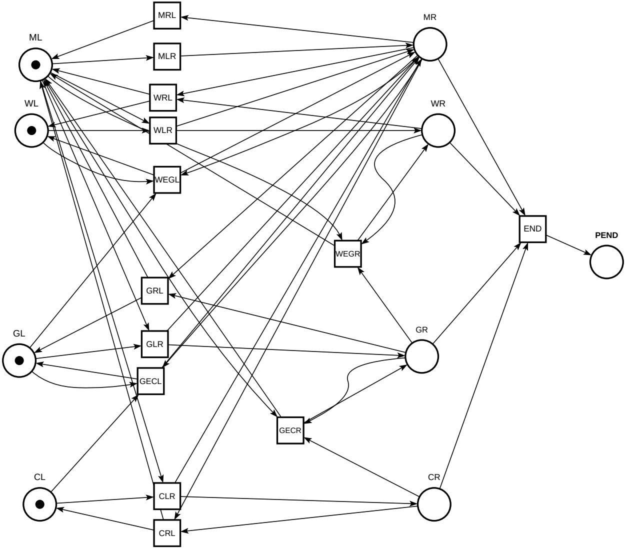
- Can the man bring everyone across the river ?
- Is the marking $M = \{MR, WR, CR, GR\}$ reachable from the initial marking $M_0 = \{ML, WL, CL, GL\}$

- Can the man bring everyone across the river without endangering the goat or the cabbage?
- One must avoid dangerous marking i.e. marking enabling eating transitions

- A transtion *end* and a sink place (Pend) : $Post(Pend, end) = 1$ and $Pre(Pend, .) = 0$; $Pre(MR, end) = Pret(WR, end) = PreCR, end) = Pre(GR, end)$
- $M(Pend = 1) =>$ the man brought everyone across the river

A wolf story

Global Model



A wolf story

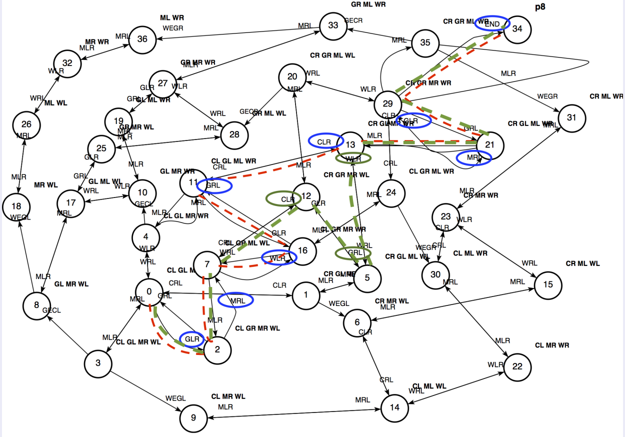
Results

```
REACHABILITY ANALYSIS -------------------------------------------

bounded

37 marking(s), 89 transition(s)

MARKINGS:

0 : CL GL ML WL
1 : CR GL MR WL
2 : CL GR MR WL
3 : CL GL MR WL
4 : CL GL MR WR
5 : CR GL ML WL
6 : CR MR WL
7 : CL GR ML WL
8 : GL MR WL
9 : CL MR WL
10 : GL MR WR
11 : CL GL ML WR
12 : CR GR MR WL
13 : CR GL MR WR
14 : CL ML WL
15 : CR ML WL
16 : CL GR MR WR
17 : GL ML WL
18 : MR WL
19 : GL ML WR
20 : CR GR ML WL
21 : CR GL ML WR
22 : CL MR WR
23 : CR MR WR
24 : CL GR ML WR
25 : GR MR WL
26 : ML WL
27 : GR MR WR
28 : GR ML WL
29 : CR GR MR WR
30 : CL ML WR
31 : CR ML WR
32 : MR WR
33 : GR ML WR
34 : p8
35 : CR GR ML WR
36 : ML WR
```

```
LIVENESS ANALYSIS -----------------------------------------

not live
not reversible

1 dead marking(s), 13 live marking(s)
0 dead transition(s), 0 live transition(s)

dead marking(s): 34
```

A wolf story

Results



A wolf story

2 solutions

1. $GLR \rightarrow MRL \rightarrow CLR \rightarrow GRL \rightarrow WLR \rightarrow MRL \rightarrow GLR \rightarrow END$
2. $GLR \rightarrow MRL \rightarrow WLR \rightarrow GRL \rightarrow CLR \rightarrow MRL \rightarrow GLR \rightarrow END$

Modeling examples

The swimming pool

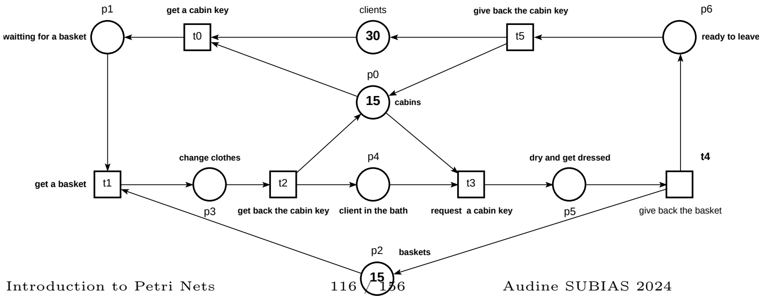
One wants to model the behavior of a swimming pool. In this system, users behave using the following protocol:

1. enter in the swimming pool building
2. get a cabin key,
3. get a basket,
4. enter the cabin and change their clothes against a swimming suit,
5. drop the basket with the clothes in the office and get back the cabin key
6. get into the swimming pool, have fun and swim,
7. request their cabin key
8. get back to the cabin, dry and get dressed,
9. give back the basket in the office,
10. give back the cabin key
11. exit the building.

The swimming pool

- $T0$: get a cabin key
- $T1$: get a basket
- $T2$: get back the cabin key
- $T3$: request a cabin key
- $T4$: give back the basket
- $T5$: give back the cabin key

- $P0$: cabin counter
- $P1$: wait for a basket
- $P2$: basket counter
- $P3$: change clothes
- $P4$: client (e.g user) in the bath
- $P5$: dry and get dressed
- $P6$: ready to leave and *clients*: client counter



The swimming pool

- Does this protocol really effective ?
- How many cabins (denoted c) and baskets (denoted b) are required for 3 clients (i.e 3 users denoted u) ? for 10 clients ?

The swimming pool

for c=b=2 and u=10 (u $\geq$ 4)

- 32 states end 57 transitions
- 1dead marking: $M(Client) = 6$ and $M(P1) = M(P4) = 2$
- 2 clients are blocked in the bath !!!

for c=b=10 and u=20

- Combinatory explosion !
- 3003 states and 12012 transitions
- Always one dead marking: 10 clients are blocked in the bath !!!

for c=b=15 and u=30

- Combinatory explosion !
- 38759 states and 178703 transitions
- Always one dead marking: 15 clients are blocked in the bath !!!

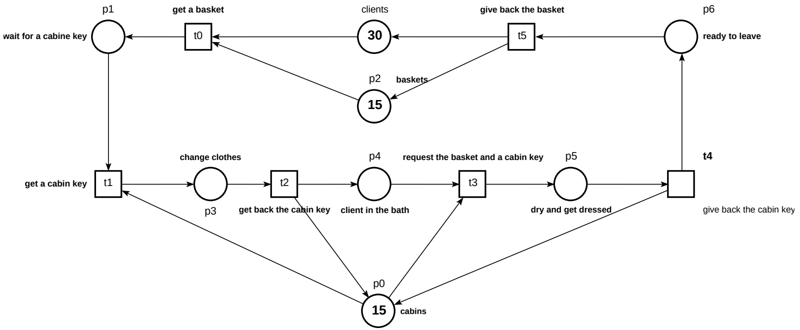
The swimming pool

A new protocol

To avoid the previous limitation, a client must behave as following:

1. enter in the swimming pool building
2. get a basket for his clothes, it will keep until he gets out,
3. get the cabin key and enter the cabin and change his clothes for his swimming suit,
4. drop the basket with his clothes in the office,
5. get into the swimming pool, have fun and swim,
6. request for the bag containing his clothes,
7. get back to a cabin, dry and get dressed,
8. give back the cabin key
9. give back the basket to the office
10. exit the building.

The swimming pool



for c=b=15 and u=30

- 15504 states and 69768 transitions
- The net is live !!!

Properties Analysis based on strong connected components

Backgrounds:

- A directed graph is strongly connected if there is a directed path from any vertex to every other vertex.
- A Connected Component of a graph is a subgraph such that there is a path between any pair of vertices of that subgraph.
- A Strongly Connected Component is a maximal connected subgraph of G. Each vertex belongs to exactly one connected component, as does each edge.
- An output edge of a strongly connected component is an edge which has as its origin in the component and its end out of that component.
- Several algorithms exist for decomposing a graph into Strong Connected Components

Properties Analysis based on strong connected components

Bounded Net (reachability Graph)

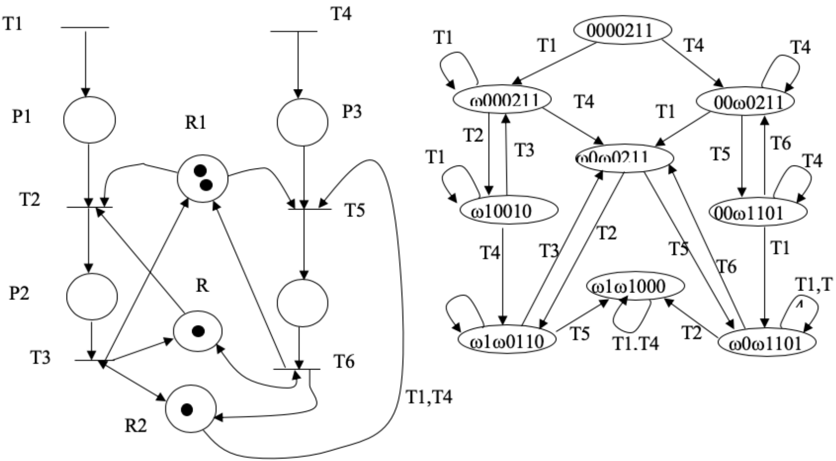
- A bounded net is live iff each SCC of its reachability graph without any output edge includes at least one edge labeled by each transition
- A bounded net is not dead-lock free if its reachability graph includes at least one trivial SCC (i.e a SCC reduced to one point without loop) without any output edge.
- A bounded net is reversible iff its reachability graph is strongly connected
- A bounded net is infinitely active iff its reachability graph has at least one non trivial SCC.
- A bounded net has a Home Marking iff its reachability graph has one and only one SCG without any output edge; the vertices of this SCG give the set of Home Markings.

Properties Analysis based on strong connected components

UnBounded Net (Coverability Graph)

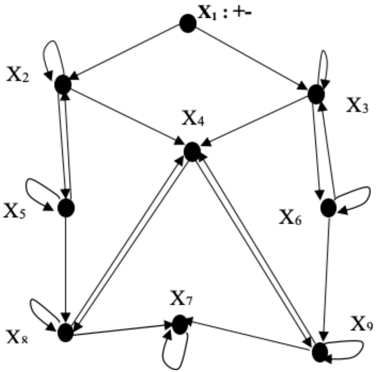
- A transition t of an unbounded net is not live if its Coverability Graph has a SCC without any output edge that does not include any edge labeled by t.
- An unbounded net is not live if its Coverability Graph has at least one SCC without any output edge in which at least one transition does not appear as an edge label.
- An unbounded net is not dead-lock free if its Coverability Graph has at least one trivial SCC (i.e a SCC reduced to one point without loop) without any output edge.

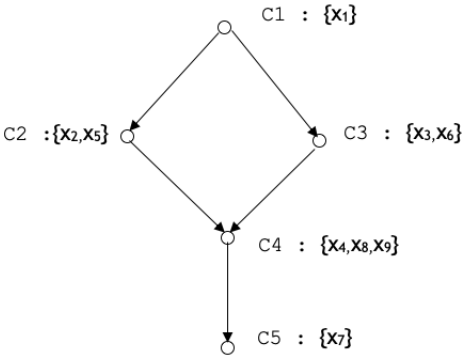
Properties Analysis based on strong connected components



The net is unbounded and deadlock free

Properties Analysis based on strong connected components





One SCC component without output edge ($C5$) including only $T1$ and $T4$ - not live

Structural Analysis

Structural analysis

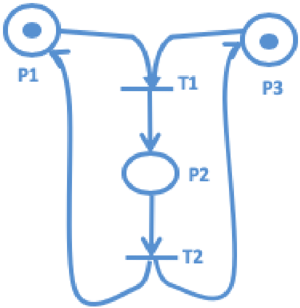
- The idea is to express properties deduced only from the structure of the net
- Structural analysis makes it possible to prove some properties without constructing the reachability (or coverability) graph
- A convenient way to tackle the combinatory explosion of the state space
- An automatic method (based on linear algebra) but the results need to be interpreted

Structural analysis

2 types of invariants:

- Place invariants or P-invariants: a P-invariant indicates that the number of tokens in all reachable markings satisfies some linear invariant. It allows to verify boundness properties.
- Transition invariants or T-invariants: a T-invariant indicates some possible loops in the net. It allows to verify (un)liveness properties

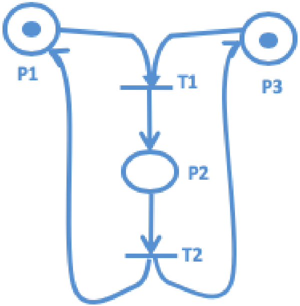
Place invariants: intuitive view



- For $M_0 = [101]'$
  $M_0(P_2) + M_0(P3) = 0 + 1 = 1$
- After the firing of $T1$,
  $M(P_2) + M(P3) = 1 + 0 = 1$
- After the firing of $T2$, $M = M_0$

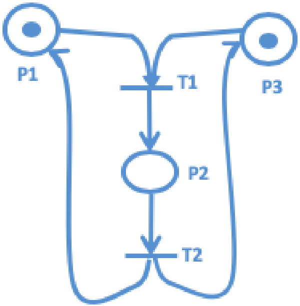
- All reachable markings M satisfy
  $M(P_2) + M(P3) = 1$
- The summ $M(P_2) + M(P3)$ is invariant
- $\forall M \in R(N, M_0)$,
  $M(P_2) + M(P3) = constant$
- This invariance result is independent of the initial marking

Place invariants: intuitive view



- The set $B = \{P2, P3\}$ is a conservative component or the support of the invariant
- The linear relation of markings $f_2 * M(P_2) + f_3 * M(P3)$ is the place invariant or P-invariant
- The support of the invariant is noted $\|f\| = \{P2, P3\}$
- The ponderation vector $f = [f_1 f_2 f_3]' = [011]'$ is the P-(semi)flow
- The invariant result is called the invariant constant
- The number of tokens is constant on the sub-net associated to B

Place invariants: intuitive view



- For $M_0 = [101]'$ $f = [011]'$ and $f'.M = 1$
- For $M_0 = [111]'$ $f = [011]'$ and $f'.M = 2$
- The value of the invariant constant depends on the initial marking
- But, $\forall M \in R(N, M_0)$, $f'.M = constant$ so $f'.M = f'.M_0$

Place invariants: intuitive view

Fundamental State Equation of a Petri net

For a firing sequence s such that $M_0 \xrightarrow{s} M$ the reachable marking is given by:

$$M = M_0 + C\bar{s}$$

- From the fundamental equation $f'.M = f'.M_0 + f'.C\bar{s}$
- To obtain a place invariant independent of the firing sequence $(\bar{s})$, $f'.C\bar{s} = 0$
- The place invariant is obtained if $f'.C = 0$

Place invariants

Definition

The linear relation of markings f'.M , $f \geqslant 0$, is a P-invariant iff $f'.C = 0$

Definition

$B \subseteq P$ is a conservative component iff $\exists\ f \geqslant 0$ s.t $\|f\| = B$ and $f'.C = 0$ with $\|f\| = \{p_i \in P : f_i \neq 0\}$

- The vector $f \in \mathbb{Q}^P$ that satisfies $f'.C = 0$ is called a P- flow
- The integer vector $f \in \mathbb{N}^P$ that satisfies $f'.C = 0$ is called a P-semi flow

Place invariants

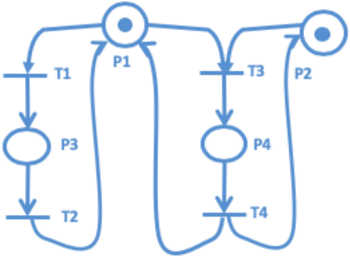
- A P-invariant (of support f_x) is said minimal if there does not exist another P-invariant (of support f_y) s.t $f_x \neq f_y$ and $f_x \geqslant f_y$
- Any linear combination of a P-invariant is a P-invariant (possible infinity of invariants)
- All P-invariant is a non negative linear combination of minimal P-invariants

Place invariants

- A basis of the set of all invariants can be computed using linear algebra
- Gauss algorithm to compute a basis of P-flows. The size of the basis is given by: $|P| - rank(C)$
- Farkas Algorithm to compute the set of minimal P-semi-flows

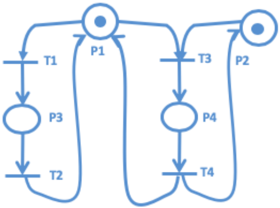
- In this lecture manual computation will be used...

Place invariants



- $P1$: Paul is at home
- $P2$: The teacher is available
- $P3$: Paul's walking around
- $P4$: Paul is in class

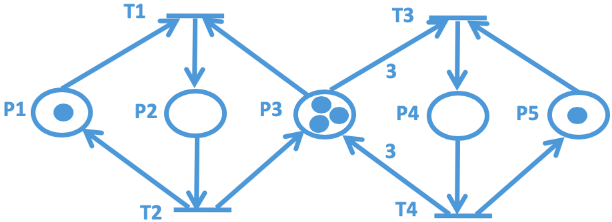
Place invariants



- $P1$: Paul is at home
- $P2$: The teacher is available
- $P3$: Paul's walking around
- $P4$: Paul is in class

- $C = \begin{bmatrix} -1 & 1 & -1 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$
- $-f_1 + f_3 = 0$ and $f_1 + f_2 = f_4$
- $f_{B1} = [1011]'$ and $f_{B2} = [0101]'$
- $B_1 = \{P1, P3, P4\}$ and $B_2 = \{P2, P4\}$
- $(f_{B1})'M_0 = 1$ and $(f_{B2})'M_0 = 1$
- $M(P1) + M(P3) + M(P4) = 1$ Paul is at home $\oplus$(Xor) in class $\oplus$ walking around
- $M(P2) + M(P4) = 1$ The teacher is available $\oplus$ in class with Paul
- P-invariant can be used to check mutal exclusion property (if the constant $=$1)

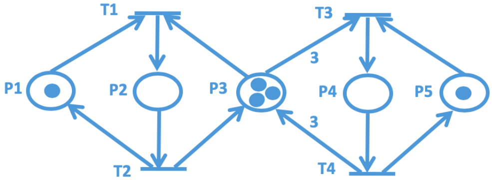
Place invariants



$$C = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad M0 = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 1 \end{bmatrix} \quad f'C = 0 \Leftrightarrow \begin{cases} -f_1 & +f_2 & -f_3 & = 0 \\ f_1 & +f_2 & -f_3 & = 0 \\ -3f_1 & +f_4 & -f_5 & = 0 \\ 3f_3 & -f_4 & +f_5 & = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} f_2 & = & f_1 & +f_3 \\ f_4 & = & 3f_3 & +f_5 \end{cases}$$

Place invariants



$$\Leftrightarrow \begin{cases} f_2 &= f_1 &+f_3 \\ f_4 &= 3f_3 &+f_5 \end{cases}$$

3 P-invariants

$f_1 = 1 f_3 = f_5 = 0 \rightarrow f_2 = 1,\ f_4 = 0$
$M(P1) + M(P2) = f'M_0 = 1$

$f_3 = 1 f_1 = f_5 = 0 \rightarrow f_2 = 1,\ f_4 = 3$
$M(P2) + M(P2) + 3M(P4) = f'M_0 = 3$

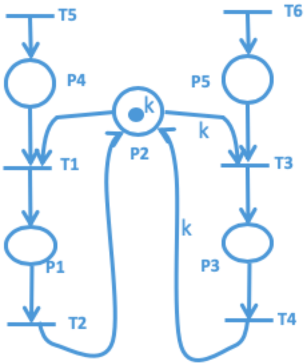
$f_5 = 1 f_1 = f_3 = 0 \rightarrow f_2 = 0,\ f_4 = 1$
$M(P4) + M(P5) = f'M_0 = 1$

Conservativeness

- A net R is conservative iff $\exists$ a P-semi flow f s.t $\|f\| = P$ i.e R is conservative if all places are covered by some P-invariant
- A net R is structurally bounded iff $\exists$ f>0 $s.t f'C \leqslant 0$ i.e R is bounded from any M_0

- Any place that belongs to at least one conservative component is bounded
- A conservative net is bounded
- A structurally bounded net is bounded

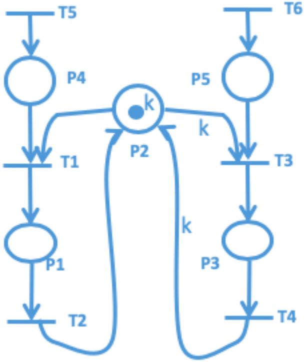
Readers-Writers Problem



Readers do not modify the object - Writers modify the object - An infinite number of reader and writers

1. Readers can share access but no more than k access simultaneously
2. No more than 1 writing simultaneously
3. No writing and reading operations simultaneously

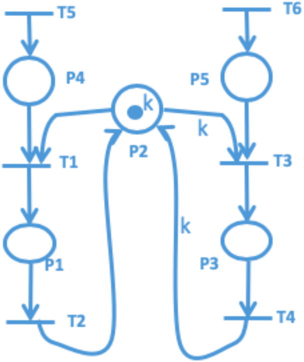
Readers-Writers Problem



3 conditions to verify w.r.t specifications

1. P1 k-bounded ? $M(P1) \leqslant k$
2. P3 safe ? $M(P3) \leqslant 1$
3. Mutual exclusion between P1 and P3 ? $M(P1) \times M(P3) = 0$

Readers-Writers Problem



$$C = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -k & k & 0 & 0 \\ 0 & 0 & 1 & {}_{1} & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad f'C = 0 \Leftrightarrow \begin{cases} f_2 & = & f_1 & \\ f_3 & = & kf_2 & \\ f_4 & = & f_5 & = 0 \end{cases}$$

Readers-Writers Problem

Place Invariant

$f = \begin{bmatrix} 1 & 1 & k & 0 & 0 \end{bmatrix}'$, $B = \{P1, P2, P3\}$, $f'M_0 = k$

$M(P1) + M(P2) + k.M(P3) = k$

Checking properties

1. $M(P1) + M(P2) + k.M(P3) = k$ then as $M \in \mathbb{N}^P$ $M(P1) \leqslant k$
2. $$M(P3) \leqslant \frac{k}{k}$$
   i.e $M(P3) \leqslant 1$ as $k > 0$
3. Reduction to the absurd: $M(P1) > 0$ and $M(P3) > 0$ then $M(P1) + k.M(P3) \geqslant k + 1$ which contradicts the P-invariant then condition ③ is verified.

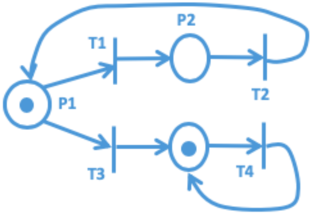
Readers-Writers Problem

if k=0 ?

- Reading is not possible as $M_0(P2) = 0$
- The place $P3$ is unbounded and then simultaneous writings (possibly infinite) are possible
- Only the condition ① can be verified by place invariant as $M(P1) + M(P2) = 0$, P1 is always empty !

Transition invariants: intuitive view

Dual notion of Place invariant



- For $M_0 = [101]'$ $M_0 \xrightarrow{T1T2} M_0$
- The set $S_1 = \{T1, T2\}$ is a repetitive component or the invariant support noted $\|\bar{s}\|$
- The firing sequence $T1T2$ is a transition invariant or T-invariant
- The sequence $T2T1$ is not an invariant as it is not a firing sequence
- $S_2 = \{T4\}$ is another repetitive component

- For $M_0 = [011]'$ $M_0 \xrightarrow{T2T1} M_0$
- S={T1,T2} is a repetitive component
- $T2T1$ is a T-invariant for $M_0 = [011]'$
- $T1T2$ is not an invariant

Transition invariants: intuitive view

Dual notion of Place invariant

- The invariant existence depends on the initial markings as it supposes the existence of a firing sequence
- Any T-invariant satisfies thought its counting vector the fundamental equation. $M = M_0 + C\overline{s}$
- From the fundamental equation : $f'.M = f'.M_0 + f'.C\overline{s}$
- A transition invariant induces $f'.M = f'.M_0$ then $C\overline{s} = 0$
- The counting vector of the invariant is obtained by solving $C\overline{s} = 0$

Transition invariants

Definition

The firing sequence $s \in T^*$ is a T-invariant iff $C\bar{s} = 0$

Definition

$S \subseteq T$ is a repetitive component iff $\exists\ \bar{s} \geqslant 0$ s.t $\|\bar{s}\| = S$ and $C\bar{s} = 0$ with $\|\bar{s}\| = \{t_i \in T : s_i \neq 0\}$

- The counting vector $\bar{s} \in \mathbb{Q}^T$ of a firing sequence s.t $C\bar{s} = 0$ is called a T-flow
- The integer counting vector $\bar{s} \in \mathbb{N}^T$ of a firing sequence s.t $C\bar{s} = 0$ is called a T-semi flow

Transition invariants

- A T-invariant (of counting vector $\bar{s}_x$) is said minimal if there does not exist another T-invariant (of counting vector $\bar{s}_y$) s.t $\bar{s}_x \neq \bar{s}_y$ and $\bar{s}_x \geqslant \bar{s}_y$
- Any linear combination of a T-invariant is a T-invariant (possible infinity of invariants)
- All T-invariant is a non negative linear combination of minimal T-invariants

Transition invariants

- A basis of the set of all invariants can be computed using linear algebra
- Gauss algorithm to compute a basis of T-flows. The size of the basis is given by: $|T| - rank(C)$
- Farkas Algorithm to compute the set of minimal T-semi-flows

- In this lecture manual computation will be used...

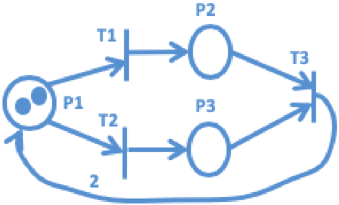
Repetitiveness

- A net R is consistent or stationary repetitive iff $\exists s$ s.t $C\bar{s} = 0$ and $\|\bar{s}\| = T$ i.e R is consistent if $\exists s$ s.t $M_0[s > M_0$ and $\|\bar{s}\| = T$

- A net R is repetitive iff $\exists\ \bar{s} > 0$ s.t $C\bar{s} \leqslant 0$ i.e R is repetitive if all transitions are covered by some T-invariants

- Any transition that does not belong to a repetitive component is not live
- A live Petri net is repetitive
- A live Petri with home states is consistent.
- A live and bounded Petri net is consistent

Repetitiveness

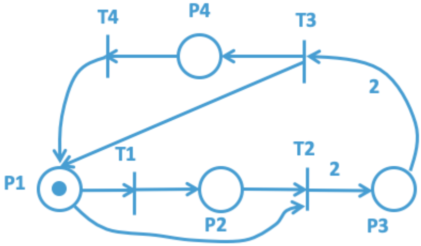


$$C = \begin{bmatrix} -1 & -1 & 2 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \quad C\bar{s} = 0 \Leftrightarrow \begin{cases} -\bar{s}_1 & -\bar{s}_2 & 2\bar{s}_3 & = 0 \\ \bar{s}_1 & -\bar{s}_3 & = 0 & \\ \bar{s}_2 & -\bar{s}_3 & = 0 & \end{cases} \Leftrightarrow s = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$S = \{T1, T2, T3\} = T = \|\bar{s}\|$: the net is consistent

$T1T2T3$ and $T2T1T3$ are two T-invariants for M_0
$T2T3T1$, $T3T1T2$, $T1T3T2$ are not a T-invariants.

The net is not live (dead lock with $T2T2$)



Set of markings E1

From the reachabilitby set :

$$E1 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

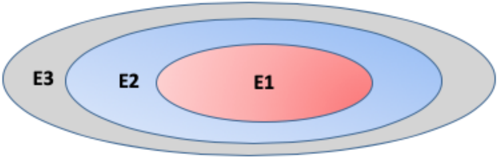
Set of markings E2

From the state equation with

$$\bar{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha \\ \alpha \\ \alpha \\ \alpha \end{bmatrix} \text{ and } \alpha > 0$$

$$E2 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Markings characterisation



Set of markings E3

From place invariants. $f'C = 0 \Leftrightarrow f = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

$$E3 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Some References I

G.W. Brams.
Réseaux de Petri: théorie et pratique.
Réseaux de Petri, théorie et pratique. Masson, 1983.

MP. Cabasino, A. Giua, and C. Seatzu.
Introduction to Petri Nets, pages 191–211.
Springer London, 2013.

R. David and H. Alla.
Du Grafcet aux réseaux de Petri.
Série automatique. Hermès, 1992.

M. Diaz.
Les réseaux de Petri : modèles fondamentaux / sous la direction de Michel Diaz.
IC2, Information, commande, communication Informatique et systèmes d'information. Hermès, Paris, 2001.

Some References II

W. Reisig.
Petri Nets: An Introduction, volume 4 of *EATCS Monographs on Theoretical Computer Science*.
Springer, 1985.

C. Reutenauer.
The Mathematics of Petri Nets.
Prentice-Hall, Inc., USA, 1990.