Petri Nets

Computational Models for Complex Systems

Paolo Milazzo

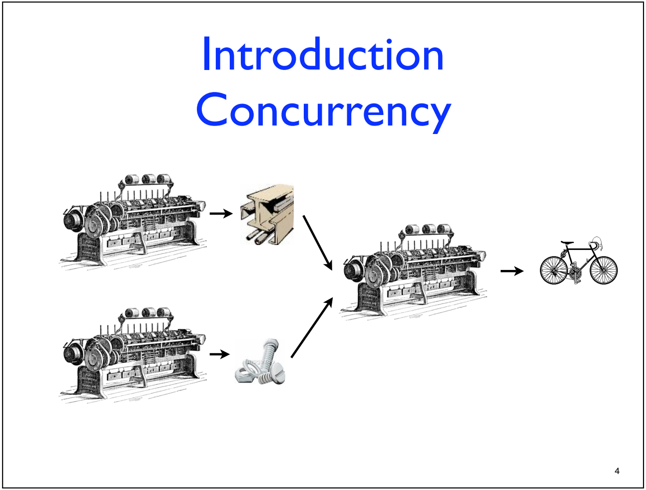
Dipartimento di Informatica, Università di Pisa
http://pages.di.unipi.it/milazzo
milazzo@di.unipi.it

Laurea Magistrale in Informatica
A.Y. 2018/2019

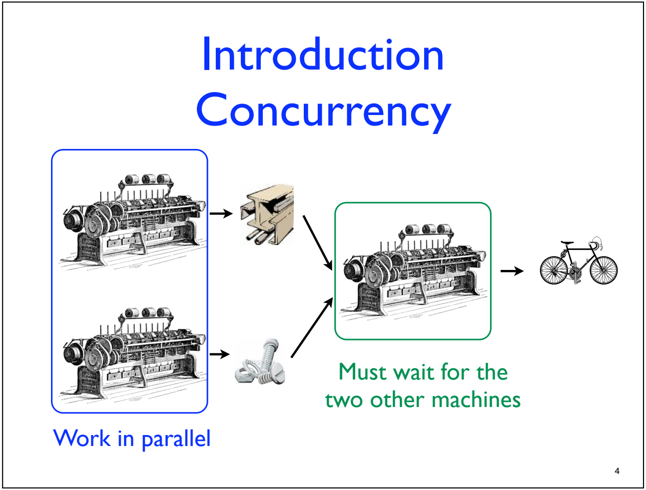
Introduction

- Petri nets have been proposed to model concurrent systems, with applications mainly in manufactury and concurrent programming
- Recent applications are in the context of business processes
  - Have a look at Roberto Bruni's course: `http://didawiki.di.unipi.it/doku.php/magistraleinformaticaeconomia/mpb/start`
- Many books have been written on Petri nets. One of the most famous is
  - Wolfgang Reisig, "Petri Nets. An Introduction". Springer-Verlag.
- This lesson is based on the tutorial by G. Geeraerts available here: `http://di.ulb.ac.be/verif/ggeeraer/Tutorial-Perti-Nets-Geeraerts.pdf`

Introduction

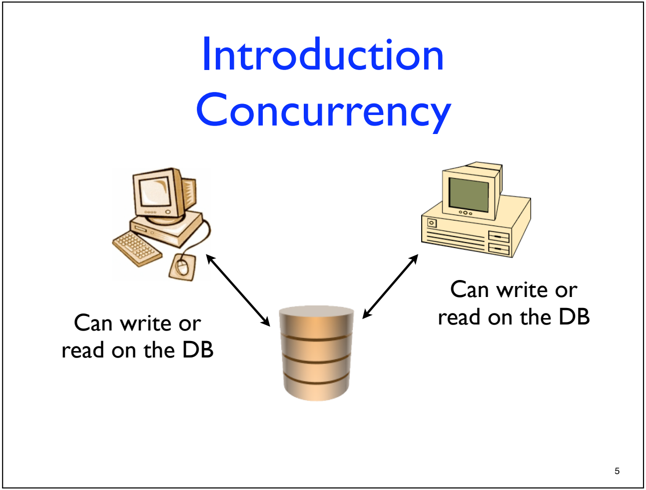


Introduction

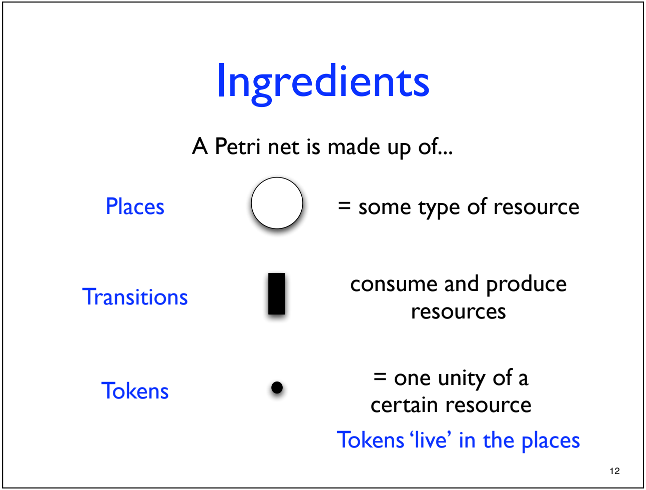


©G. Geeraerts

Introduction



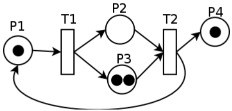
Petri nets



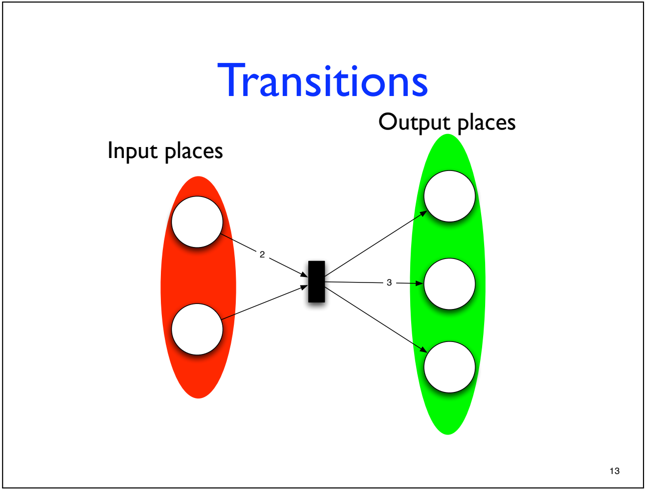
©G. Geeraerts

Places and Transitions are edges of a bipartite graph (the Petri net)

Tokens are inside places



Petri nets



©G. Geeraerts

A Petri net trasition corresponds exactly to a MultiSet Rewriting rule (or a chemical reaction)

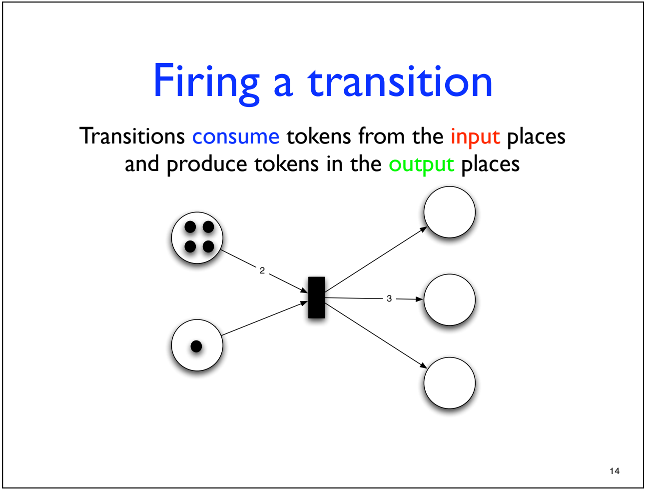
MSR:

$$a^2 b \mapsto c d^3 e$$

Chem:

$$2A + B \rightarrow C + 3D + E$$

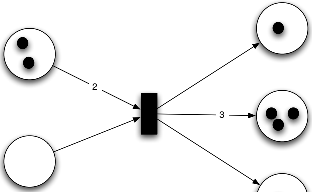
Petri nets


Firing a transition
Transitions consume tokens from the input places
and produce tokens in the output places
2
3
14

Petri nets

Firing a transition

Transitions consume tokens from the input places and produce tokens in the output places



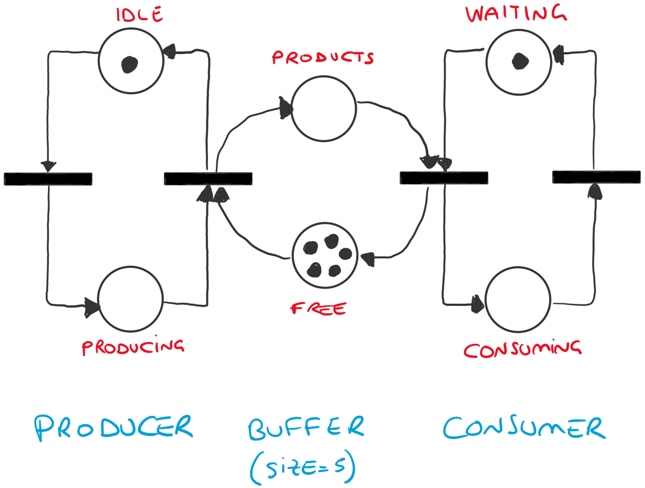
Now, the transition cannot be fired anymore

15

Petri nets

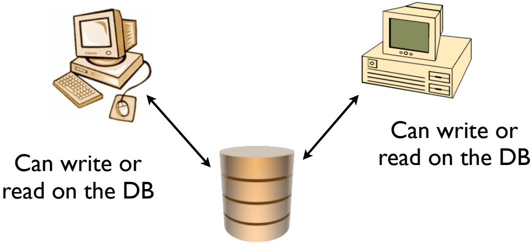
Example: producer/consumer with bounded buffer

22:29



Petri nets

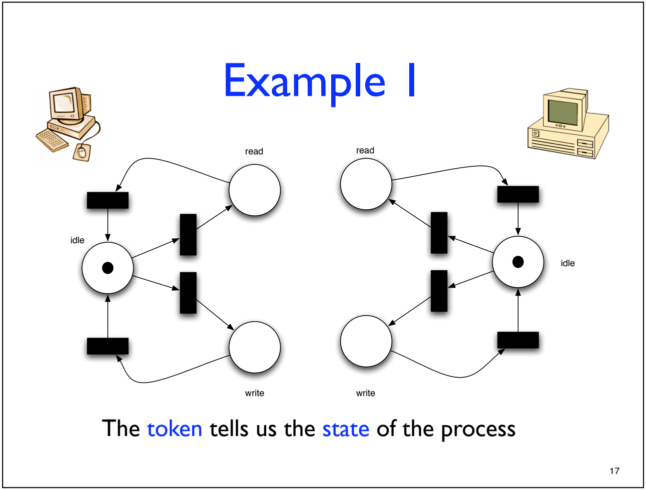
Example I



The two machines cannot write at the same time

16

Petri nets



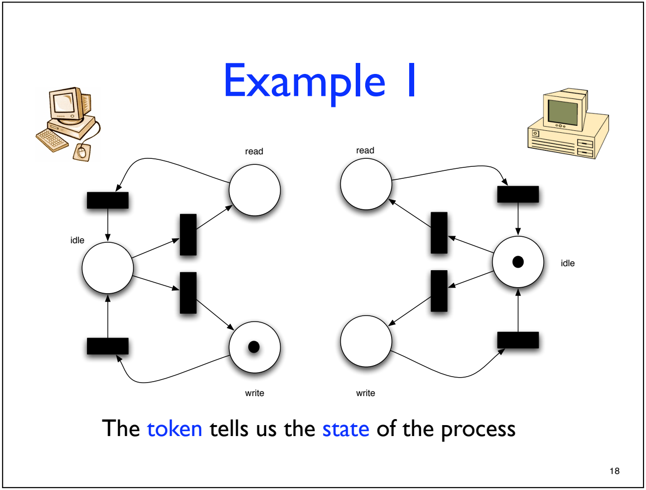
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1$$
$$idle_2 \mapsto read_2$$
$$idle_2 \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2$$

state (multiset):

*idle*1 *idle*2

Petri nets



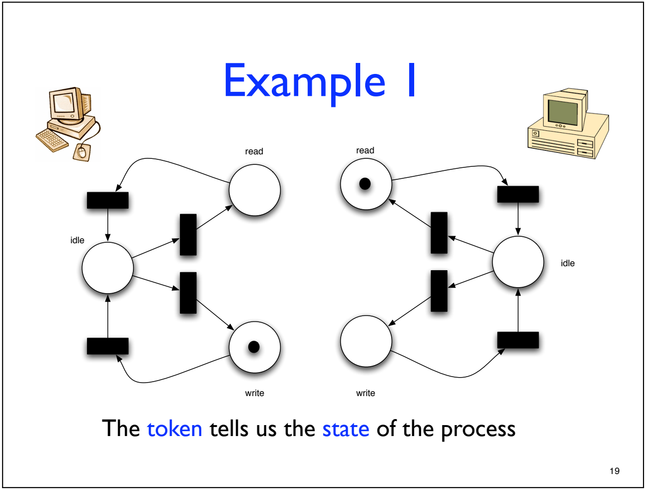
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1$$
$$idle_2 \mapsto read_2$$
$$idle_2 \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2$$

state (multiset):

$$write1 \ idle2$$

Petri nets



©G. Geeraerts

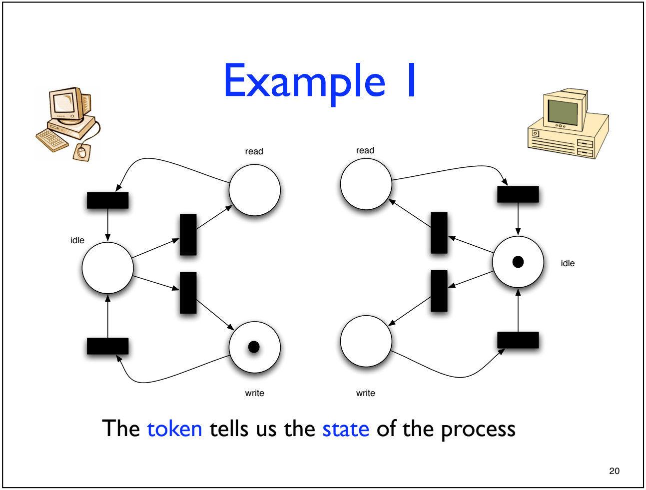
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1$$
$$idle_2 \mapsto read_2$$
$$idle_2 \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2$$

state (multiset):

*write*1 *read*2

Petri nets



©G. Geeraerts

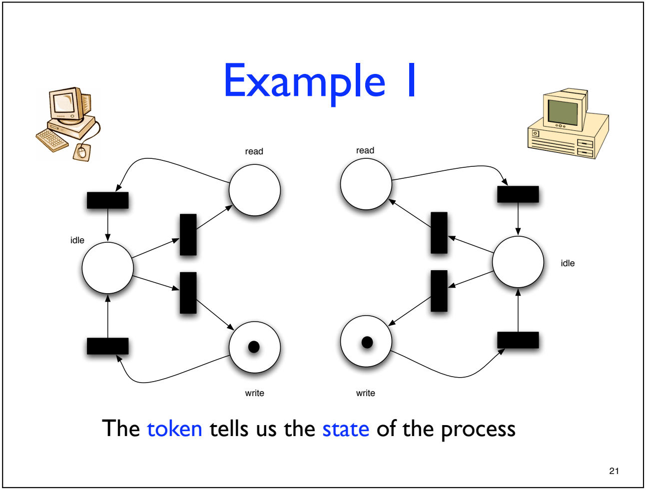
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1$$
$$idle_2 \mapsto read_2$$
$$idle_2 \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2$$

state (multiset):

write1 *idle2*

Petri nets



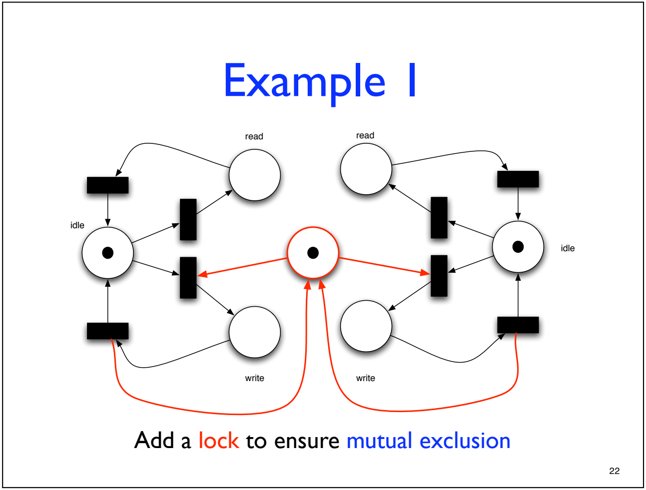
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1$$
$$idle_2 \mapsto read_2$$
$$idle_2 \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2$$

state (multiset):

$write1$ $write2$!!!

Petri nets



©G. Geeraerts

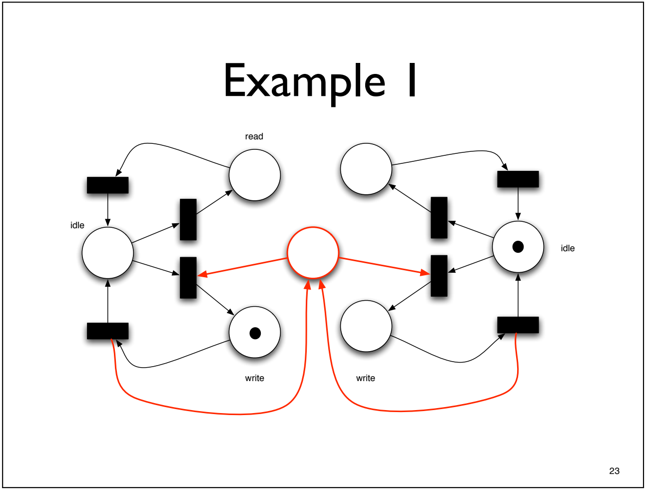
With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1\ M \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1\ M$$
$$idle_2 \mapsto read_2$$
$$idle_2\ M \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2\ M$$

state (multiset):

$$idle1\ idle2\ M$$

Petri nets



©G. Geeraerts

With MSR rules:

$$idle_1 \mapsto read_1$$
$$idle_1 \; M \mapsto write_1$$
$$read_1 \mapsto idle_1$$
$$write_1 \mapsto idle_1 \; M$$
$$idle_2 \mapsto read_2$$
$$idle_2 \; M \mapsto write_2$$
$$read_2 \mapsto idle_2$$
$$write_2 \mapsto idle_2 \; M$$

state (multiset):

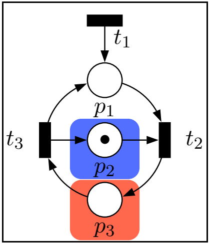
$write1 \; idle2$

Petri nets

Example 2

```
mutex M ;

Process P {
   repeat {
      take M ;
      critical ;
      release M ;
   }
}
```



24

©G. Geeraerts

t_1 represents the start of a new (concurrent) instance of process P

p_1: processes
p_2: free mutex
p_3: critical session

Petri nets

Formal definition

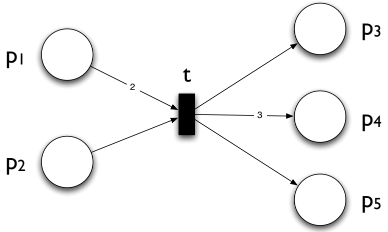
- A Petri net is a tuple ⟨P,T⟩ where:
  - P is the (finite) set of places
  - T is the (finite) set of transitions. Each transition t is a tuple ⟨I, O⟩ where:
    - I: is a function s.t. t consumes I(p) tokens in each place p
    - O is a function s.t. t produces O(p) tokens in each place p

27

Example

$I(p_1)=2$ $I(p_2)=1$ $I(p_3)=0$ $I(p_4)=0$ $I(p_5)=0$

$O(p_1)=0$ $O(p_2)=0$ $O(p_3)=1$ $O(p_4)=3$ $O(p_5)=1$



28

Dynamics of Petri nets (semantics)

Markings

- The distribution of the tokens in the places is formalised by the notion of marking, which can be seen:
  - either as a function m, s.t. m(p) is the number of tokens in place p
  - or as a vector $m = \langle m_1, m_2, \ldots m_n \rangle$ where m_i is the number of tokens in place p_i

29

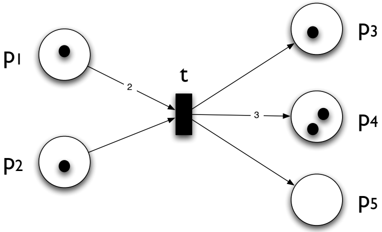
Dynamics of Petri nets (semantics)

Example

$m = \langle 1,1,1,2,0 \rangle$

$m = \langle p_1, p_2, p_3, 2p_4 \rangle$

$m(p_1)=1, m(p_2)=1, m(p_3)=1, m(p_4)=2, m(p_5)=0$



30

Markings are multisets!

Dynamics of Petri nets (semantics)

Firing a transition

- A transition $t = \langle I, O \rangle$ can be fired from m iff for any place p:

$$m(p) \geq I(p)$$

- The firing transforms the marking m into a marking m' s.t. for any place p:

$$m'(p) = m(p) - I(p) + O(p)$$

- Notation: $m \rightarrow m'$
- Notation: $Post(m) = \{m' \mid m \rightarrow m'\}$

31

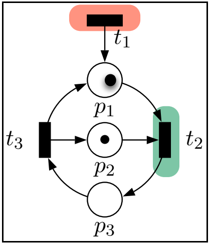
Post(*m*) is the set of markings that can be obtained by firing transitions from *m*

- *Post* corresponds to the transition relation in Transition Systems terminology

Dynamics of Petri nets (semantics)

Example

Post(⟨1, 1, 0⟩)=
{ ⟨2, 1, 0⟩ , ⟨0, 0, 1⟩ }

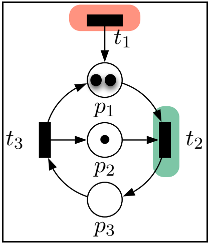


32

Dynamics of Petri nets (semantics)

Example

Post($\langle 1, 1, 0 \rangle$)=
{ $\langle 2, 1, 0 \rangle$, $\langle 0, 0, 1 \rangle$ }

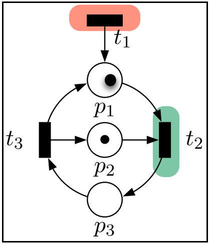


32

Dynamics of Petri nets (semantics)

Example

Post($\langle 1, 1, 0 \rangle$)=
{ $\langle 2, 1, 0 \rangle$, $\langle 0, 0, 1 \rangle$ }



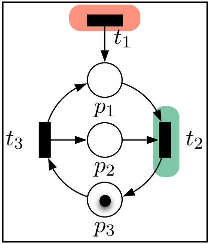
32

©G. Geeraerts

Dynamics of Petri nets (semantics)

Example

Post(⟨1, 1, 0⟩)=
{ ⟨2, 1, 0⟩ , ⟨0, 0, 1⟩ }



32

Dynamics of Petri nets (semantics)

Initial marking
Reachable markings

- All PN are equipped with an initial marking m_0
- If two markings m and m' are s.t.:

  $m \rightarrow m_1 \rightarrow m_2 \rightarrow \cdots \rightarrow m'$

  Then m' is reachable from m
- Let N be a PN with initial marking m_0:

  $\text{Reach}(N) = \{m \text{ reachable from } m_0\}$

  is the set of reachable markings of N.

33

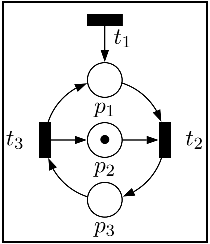
Analogous to initial state and reachable states in Transition Systems terminology

Dynamics of Petri nets (semantics)

Example

$$\text{Reach}(\mathcal{N}) = \{\langle i,1,0\rangle \mid i \in \mathbb{N}\} \cup \{\langle i,0,1\rangle \mid i \in \mathbb{N}\}$$

This set allows us to prove that the mutual exclusion is indeed enforced



34

Dynamics of Petri nets (semantics)

Ordering on markings

- Markings can be compared thanks to $\preccurlyeq$:

  $m \preccurlyeq m'$ iff for any place p: $m(p) \leqslant m'(p)$

  $m \prec m'$ iff $m \preccurlyeq m'$ and $m \neq m'$

- Examples:
  - $\langle 1, 0, 0\rangle \prec \langle 1, 1, 0\rangle \preccurlyeq \langle 1, 1, 0\rangle \preccurlyeq \langle 5, 7, 2\rangle$
  - $\langle 1, 0, 0\rangle$ is not comparable to $\langle 0, 1, 0\rangle$

35

Structural/Dynamical properties

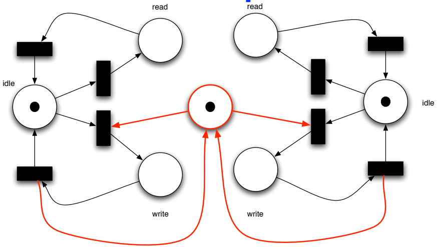
Questions on PN

- Meaningful questions about PN include:
  - Boundedness: is the number of reachable markings bounded ?
  - Place boundedness: is there a bound on the maximal number of tokens that can be created in a given place ?
  - Semi-liveness: is there a reachable marking from which a given transition can fire ?
  - Coverability

36

Structural/Dynamical properties

Example



Bounded PN All the places are bounded

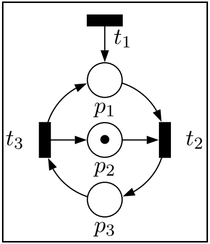
All the transitions are semi-live

37

Structural/Dynamical properties

Example

- Unbounded PN
- p_2 and p_3 are bounded
- p_1 is unbounded
- All the transitions are semi-live



38

Reachability graph

Reachability graph

- Idea: build a node for each reachable marking and add an edge from m to m' if some transition transforms m into m'
  - remark: now, if we meet the same marking twice, we do not create a new node, but re-use the previously created node.

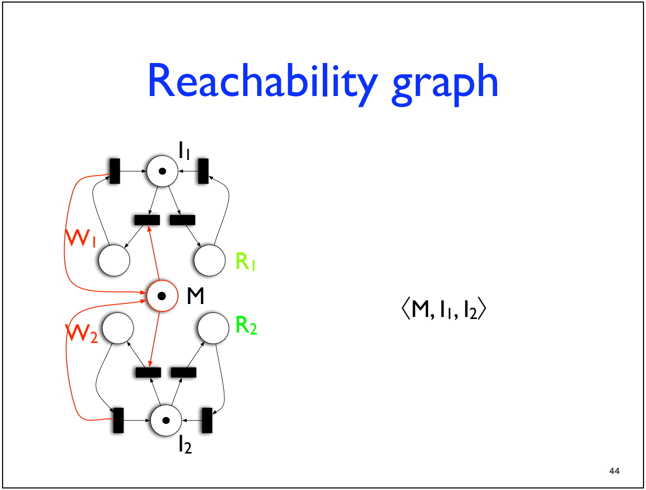
43

©G. Geeraerts

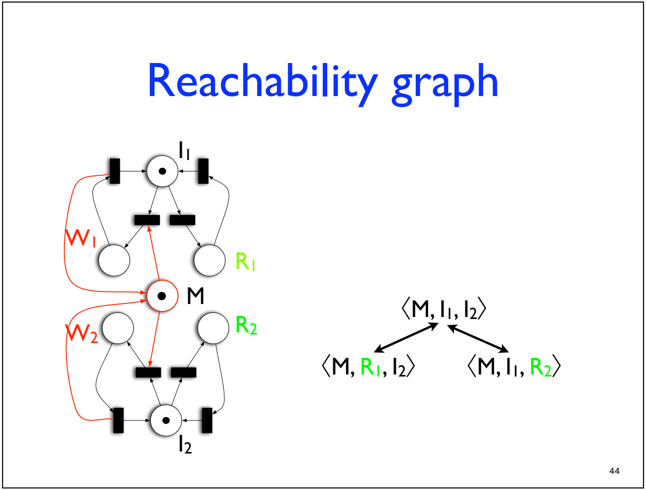
It is a Transition System

- Exactly the same Transition System that would be obtained from the MultiSet Rewriting representation of the Petri net

Reachability graph

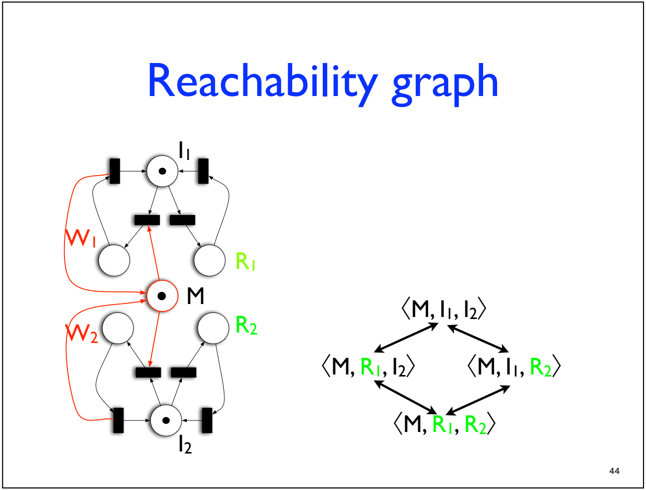


Reachability graph

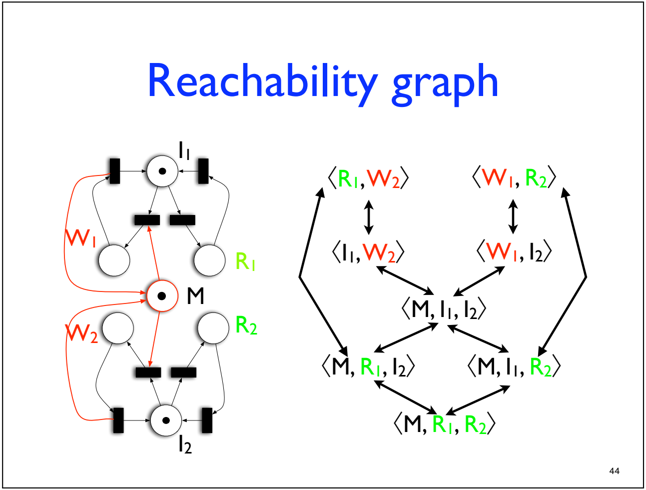


©G. Geeraerts

Reachability graph

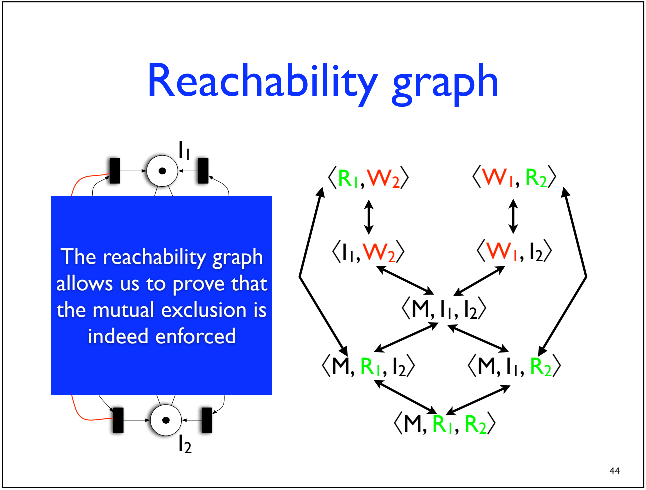


Reachability graph



©G. Geeraerts

Reachability graph



Reachability graph

Reachability graph

- The reachability graph of a PN contains all the necessary information to decide:
  - boundedness
  - place boundedness
  - semi-liveness
  - ...

45

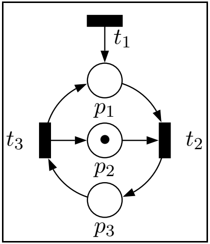
©G. Geeraerts

Reachability graph

Reachability graph

- Unfortunately...

  $\langle p_2 \rangle$

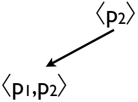


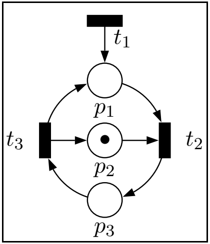
46

Reachability graph

Reachability graph

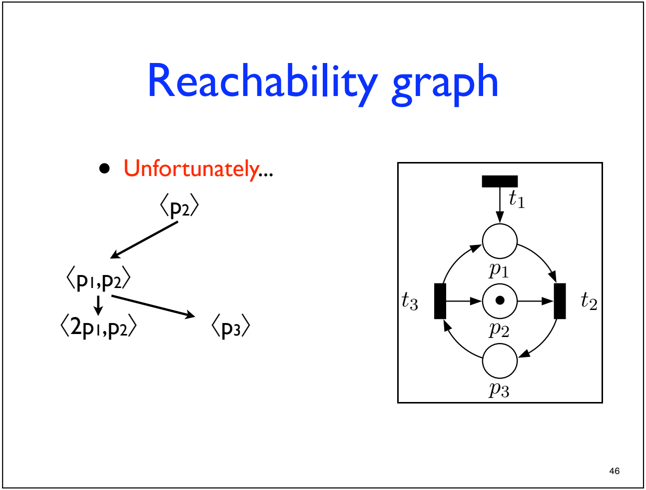
- Unfortunately...





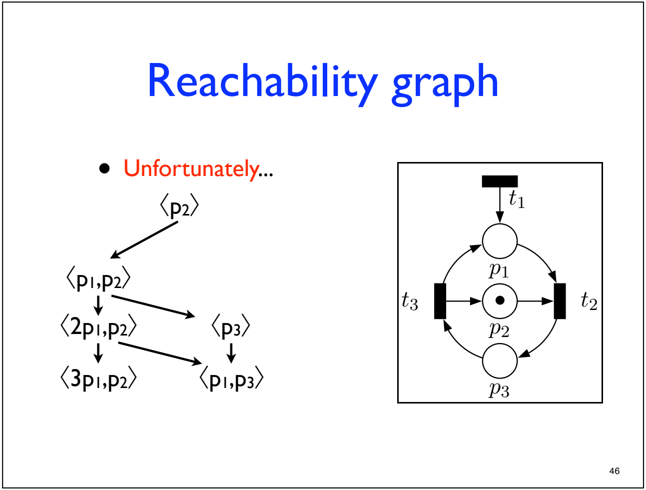
46

Reachability graph

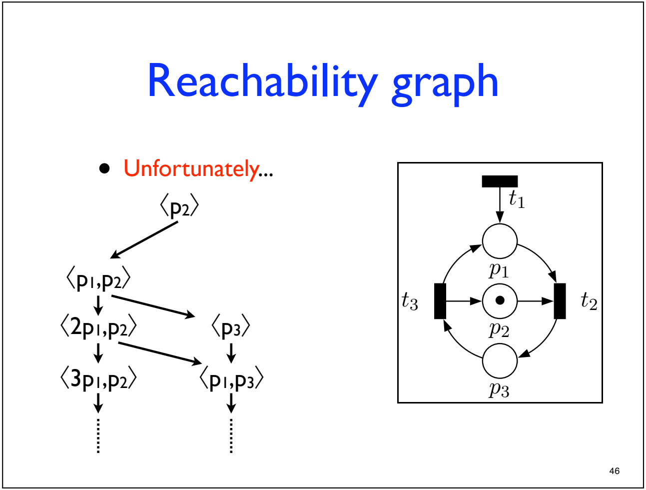


©G. Geeraerts

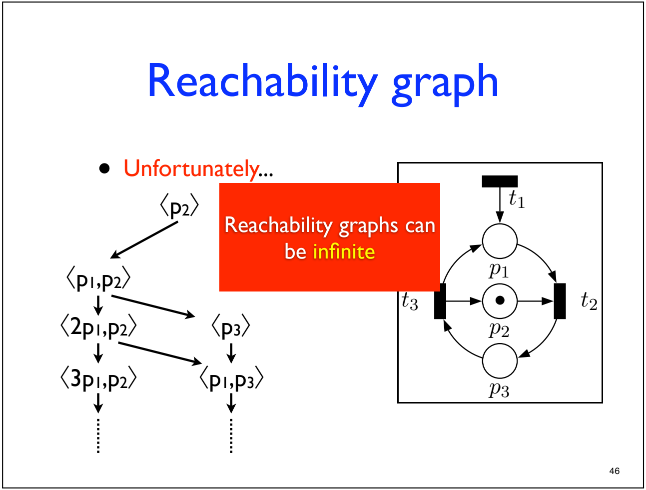
Reachability graph



Reachability graph



Reachability graph



Infinite, but decidable!

The reachability graph of a Petri net (aka the Transition System of a MultiSet Rewriting system) can be infinite

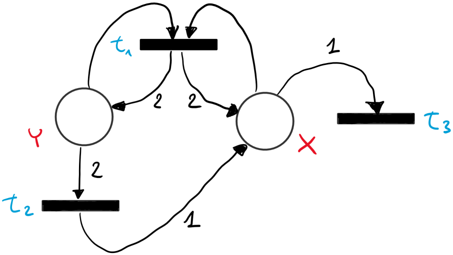
But, the reachability property (i.e. is a given state/marking reachable?) is DECIDABLE

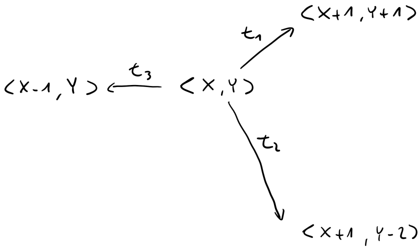
- Its computation has been proven to require EXPONENTIAL time

The reason for decidability is that the reachability graph has a regular structure!

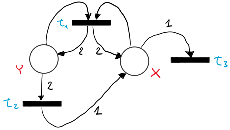
Infinite, but decidable!

For example:

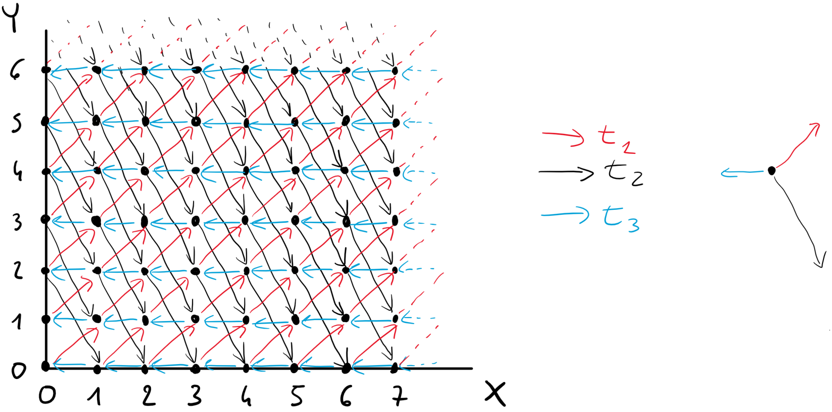




Infinite, but decidable!



The reachability graph, plotted on the cartesian plane:



Infinite, but very regular!

Decidable, but exponential!

Reachability of a marking M is decidable, but exponential...

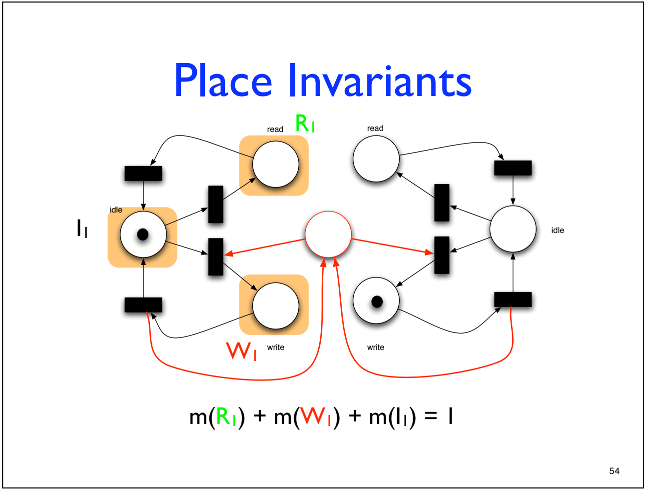
SOLUTION 1: consider overapproximations of the set of reachable states

1. based on Place Invariants
2. based on Karp and Miller tree

If M does not belong to the overapproximation it is not reachable (this is a sufficient condition: if M belongs, nothing can be said...)

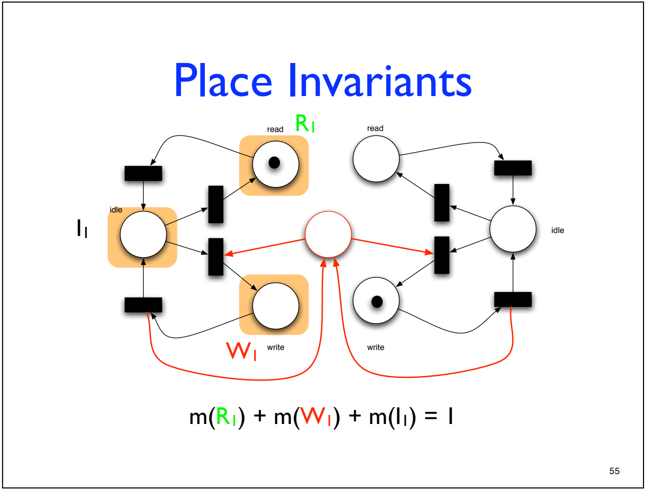
SOLUTION 2: consider coverability instead of reachability (weaker than reachability, but meaningful in the context of Petri Nets and easy to compute)

Place Invariants



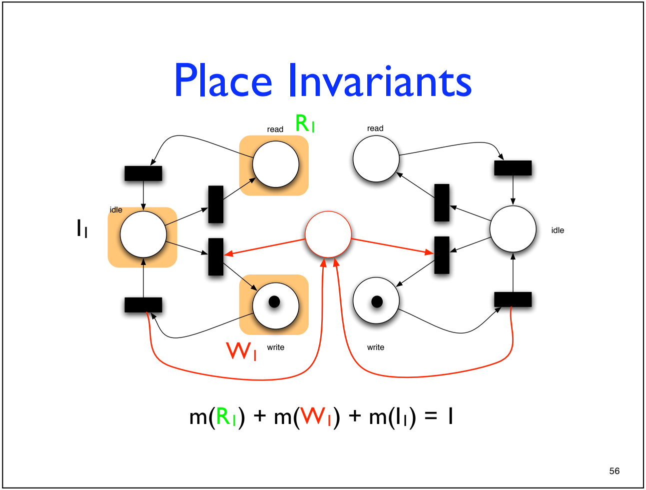
The idea is to identify groups of places whose overall number tokens is (more or less) constant

Place Invariants

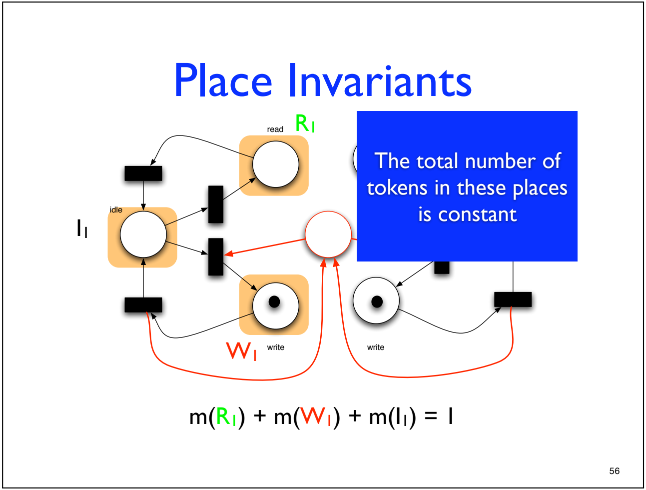


©G. Geeraerts

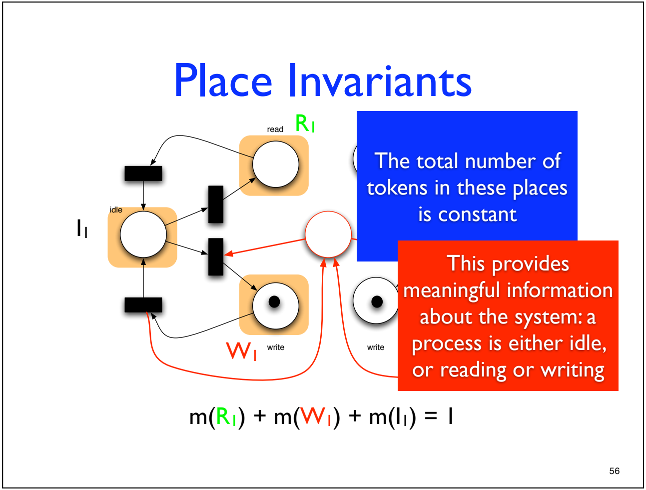
Place Invariants



Place Invariants

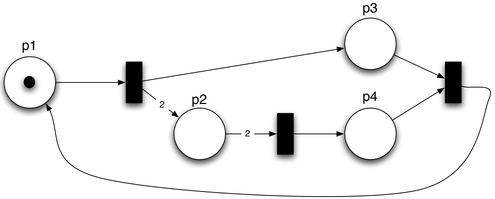


Place Invariants



Place Invariants

Place Invariants

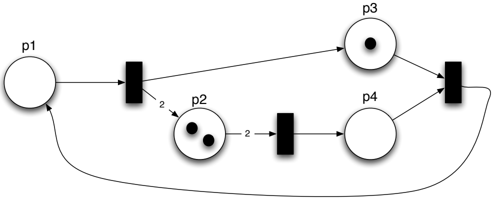


$m(p_1) + m(p_2) + m(p_3) + m(p_4) = 1$

57

Place Invariants

Place Invariants

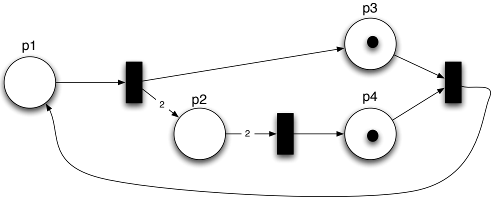


$$m(p_1) + m(p_2) + m(p_3) + m(p_4) = 3$$

58

Place Invariants

Place Invariants

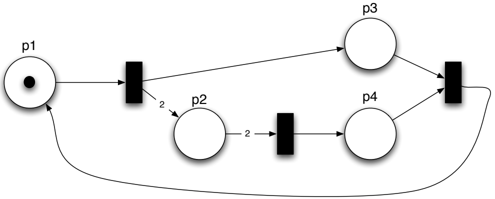


$$m(p_1) + m(p_2) + m(p_3) + m(p_4) = 2$$

59

Place Invariants

Place Invariants

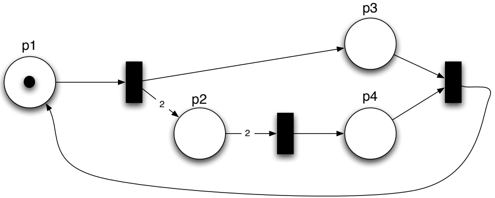


$$m(p_1) + m(p_2) + m(p_3) + m(p_4) = 1$$

60

Place Invariants

Place Invariants

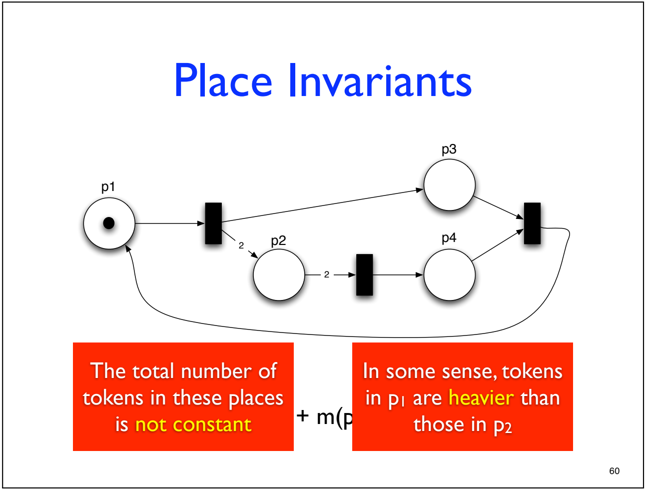


The total number of tokens in these places is not constant

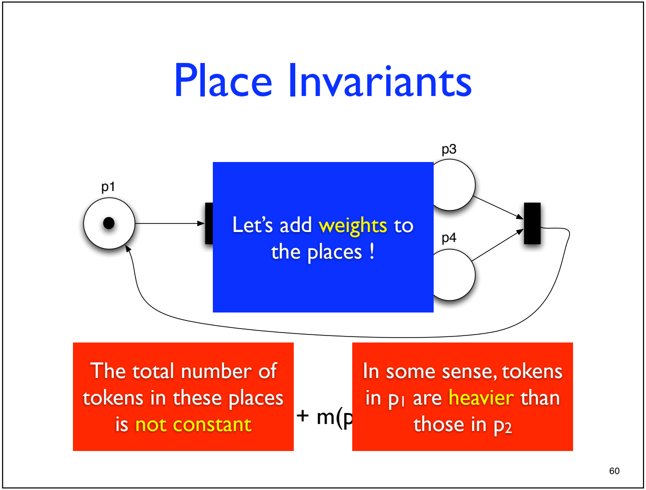
+ m(p3) + m(p4) = 1

60

Place Invariants



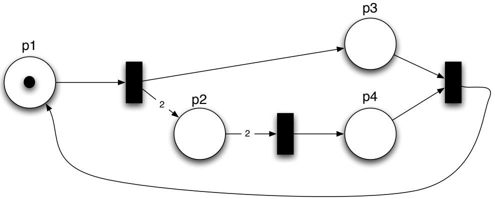
Place Invariants



©G. Geeraerts

Place Invariants

Place Invariants

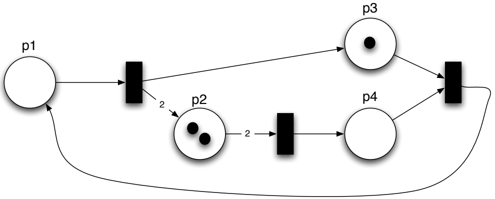


$$3\, m(p_1) + m(p_2) + m(p_3) + 2\, m(p_4) = 3$$

61

Place Invariants

Place Invariants

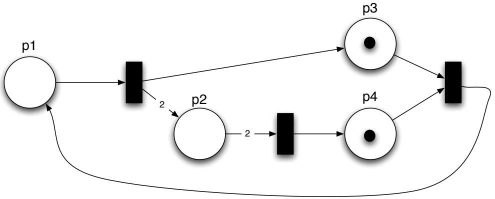


$$3\ m(p_1) + m(p_2) + m(p_3) + 2\ m(p_4) = 3$$

62

Place Invariants

Place Invariants



$$3\, m(p_1) + m(p_2) + m(p_3) + 2\, m(p_4) = 3$$

63

Place Invariants

Place invariant: Definition

- Definition: a place-invariant (or p-semiflow) is a vector i of natural numbers s.t. for any reachable marking m:

$$\sum_{p \in P} i(p) \times m(p) = \sum_{p \in P} i(p) \times m_0(p)$$

remark: there exists a trivial invariant i = $\langle 0, 0, .., 0 \rangle$

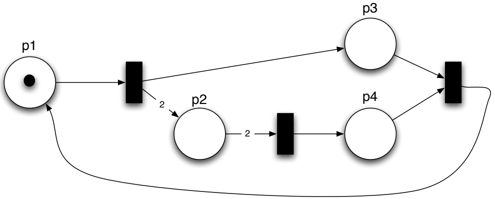
64

Corresponds to the notion of mass conservation in (bio)chemistry

"Matter can never be created, nor destroyed..."

Place Invariants

Example: other invariants



$$m(p_1) + m(p_3) = 1$$

$$2\, m(p_1) + m(p_2) + 2\, m(p_4) = 2$$

65

Place Invariants

Invariants as over-approximations

- A place-invariant expresses a constraint on the reachable markings.
  - If m is reachable and i is an invariant, then:

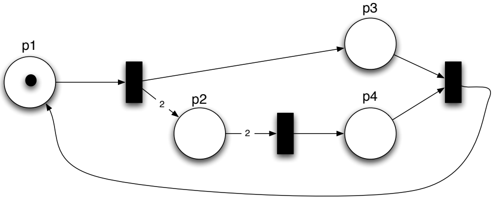
$$\sum_{p \in P} i(p) \times m(p) = \sum_{p \in P} i(p) \times m_0(p)$$

- The reverse is not true !

66

Place Invariants

Example



$m(p_1) + m(p_3) = 1$

is an invariant

but $\langle 1, 25, 0, 234 \rangle$ is not reachable

67

Place Invariants

Invariants as over-approximations

- Theorem: For any Petri net N:

$$\text{Reach}(N) \subseteq \{m \mid m \text{ respects some invariant of } N\}$$

68

©G. Geeraerts

So, every marking that does not respect the invariant is not reachable!

We do not need to explore the reachability graph!

Place Invariants

Place invariant and boundedness

- Theorem: If there exists a place invariant i and a place p s.t. i(p)>0 then p is bounded.
- Remark: the reverse is not true.
  - One can find a bounded net that doesn't have a place invariant i with i(p)>0 for each place.

69

Place Invariants

Place invariant

- Question: how do we compute them ?

70

Matrix representation of Petri nets

Matrix characterisation

- The negative effect (consumption) of all the transitions on all the places can be summarised in one matrix:

$$W^- = \begin{pmatrix} I_1(p_1) & I_2(p_1) & \cdots & I_k(p_1) \\ I_1(p_2) & I_2(p_2) & \cdots & I_k(p_2) \\ \vdots & \vdots & \ddots & \vdots \\ I_1(p_n) & I_2(p_n) & \cdots & I_k(p_n) \end{pmatrix}$$

neg. eff. on p_1

neg. eff. on p_2

where, for any i: $t_i = \langle I_i, O_i \rangle$

71

Matrix characterisation

- The same can be done with the positive effects:

$$W^+ = \begin{pmatrix} O_1(p_1) & O_2(p_1) & \cdots & O_k(p_1) \\ O_1(p_2) & O_2(p_2) & \cdots & O_k(p_2) \\ \vdots & \vdots & \ddots & \vdots \\ O_1(p_n) & O_2(p_n) & \cdots & O_k(p_n) \end{pmatrix}$$

pos. eff. on p_1

pos. eff. on p_2

where, for any i: $t_i = \langle I_i, O_i \rangle$

72

Matrix representation of Petri nets

Incidence Matrix

- The global effect of every transition can be summarised as a single matrix:

$$W = W^{+} - W^{-}$$

W is called the incidence matrix of the net

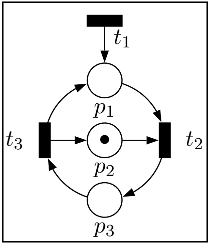
73

Matrix representation of Petri nets

Example

$$W^+ = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad W^- = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$



74

©G. Geeraerts

If you think the Petri net as a graph, the incidence matrix is a standard representation...

... more or less, since rows and columns should be defined for all graph nodes (6x6 matrix, in the example)...

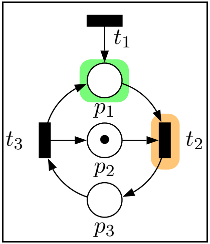
... but a Petri net is a bipartite graph

Matrix representation of Petri nets

Example

$$W^+ = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad W^- = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

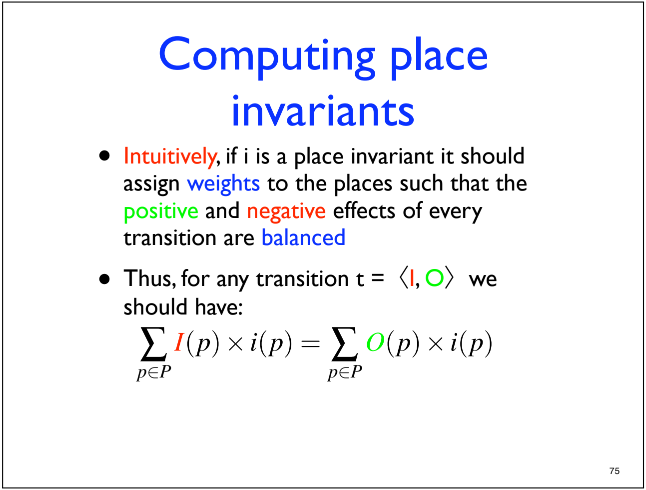
$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$



74

©G. Geeraerts

Place invariants

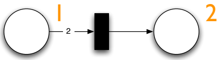


Place invariants

Computing place invariants

- Intuitively, if i is a place invariant it should assign weights to the places such that the positive and negative effects of every transition are balanced
- Thus, for any transition t = ⟨I, O⟩ we should have:

$$\sum_{p \in P} I(p) \times i(p) = \sum_{p \in P} O(p) \times i(p)$$



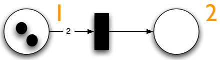
75

Place invariants

Computing place invariants

- Intuitively, if i is a place invariant it should assign weights to the places such that the positive and negative effects of every transition are balanced
- Thus, for any transition t = ⟨I, O⟩ we should have:

$$\sum_{p \in P} I(p) \times i(p) = \sum_{p \in P} O(p) \times i(p)$$



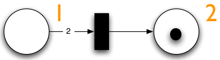
75

Place invariants

Computing place invariants

- Intuitively, if i is a place invariant it should assign weights to the places such that the positive and negative effects of every transition are balanced
- Thus, for any transition t = $\langle I, O \rangle$ we should have:

$$\sum_{p \in P} I(p) \times i(p) = \sum_{p \in P} O(p) \times i(p)$$



75

©G. Geeraerts

Place invariants

Computing place invariants

$$\sum_{p \in P} I(p) \times i(p) = \sum_{p \in P} O(p) \times i(p)$$

means

$$\sum_{p \in P} \big(O(p) - I(p)\big) \times i(p) = 0$$

76

Place invariants

Computing place invariants

$$\sum_{p \in P} I(p) \times i(p) = \sum_{p \in P} O(p) \times i(p)$$

means

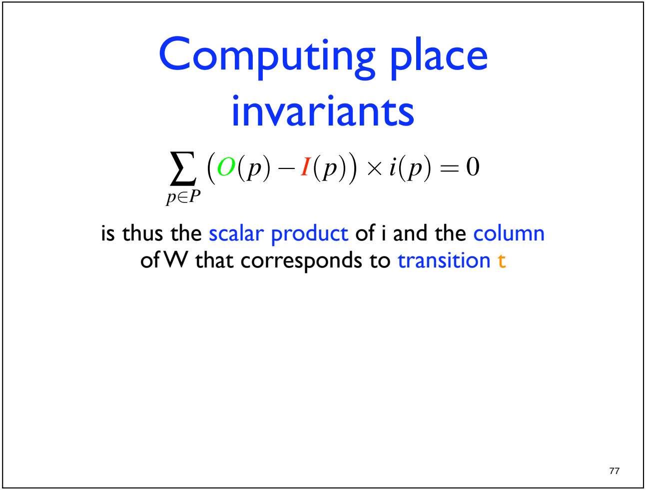
$$\sum_{p \in P} \big(O(p) - I(p)\big) \times i(p) = 0$$

t = $\langle$I, O$\rangle$

$$W = \begin{pmatrix} \cdots & O(p_1) - I(p_1) & \cdots \\ \cdots & O(p_2) - I(p_2) & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & O(p_n) - I(p_n) & \cdots \end{pmatrix}$$

76

Place invariants



Place invariants

Computing place invariants

$$\sum_{p \in P} \big(O(p) - I(p) \big) \times i(p) = 0$$

is thus the scalar product of i and the column of W that corresponds to transition t

Since this must hold for any t, we obtain:

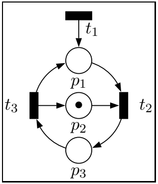
Theorem: any solution i to the following system of equations is a place-invariant:

$$i \times W = 0$$

77

Place invariants

Example

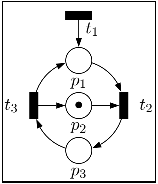


$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

78

Place invariants

Example



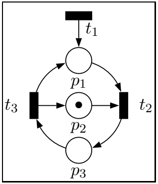
$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

$$\langle i_1, i_2, i_3 \rangle \times W = 0$$

78

Place invariants

Example



$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

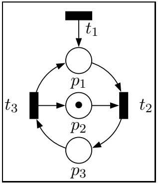
$$\langle i_1, i_2, i_3 \rangle \times W = 0$$

$$\begin{cases} i_1 & = 0 \\ -i_1 - i_2 + i_3 & = 0 \\ i_1 + i_2 - i_3 & = 0 \end{cases}$$

78

Place invariants

Example



$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

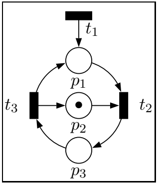
$$\langle i_1, i_2, i_3 \rangle \times W = 0$$

$$\begin{cases} i_1 = 0 \\ -i_1 - i_2 + i_3 = 0 \\ i_1 + i_2 - i_3 = 0 \end{cases} \qquad \begin{cases} i_1 = 0 \\ -i_2 + i_3 = 0 \\ +i_2 - i_3 = 0 \end{cases}$$

78

Place invariants

Example



$$W = \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ & & -1 \end{pmatrix}$$

Any vector of the form
$\langle 0, i, i \rangle$
is a place invariant

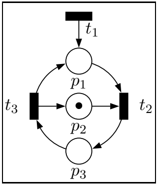
$$\langle i_1, i_2, i_3 \rangle \times W = 0$$

$$\begin{cases} i_1 = 0 \\ -i_1 - i_2 + i_3 = 0 \\ i_1 + i_2 - i_3 = 0 \end{cases} \qquad \begin{cases} i_1 = 0 \\ -i_2 + i_3 = 0 \\ +i_2 - i_3 = 0 \end{cases}$$

78

Place invariants

Proving properties



Let us choose $\langle 0, 1, 1 \rangle$ as place-invariant

This means that p_2 and p_3 are bounded !

For any reachable marking m:

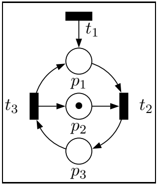
$$0\, m(p_1) + 1\, m(p_2) + 1\, m(p_3) = 0\, m_0(p_1) + 1\, m_0(p_2) + 1\, m_0(p_3)$$

$$m(p_2) + m(p_3) = 1$$

79

Place invariants

Proving properties



Let us choose $\langle 0, 1, 1 \rangle$ as place-invariant

This means that p_2 and p_3 are bounded !

For any reachable marking m:

$$0\, m(p_1) + 1\, m(p_2) + 1\, m(p_3) = 0\, m_0(p_1) + 1\, m_0(p_2) + 1\, m_0(p_3)$$

$$m(p_2) + m(p_3) = 1$$

Hence, mutual exclusion is enforced !

79

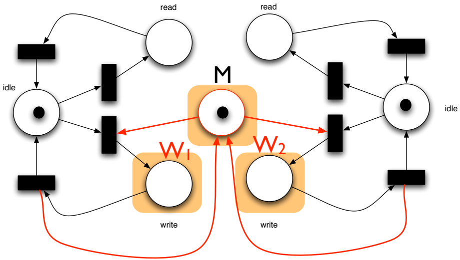
The marking

$$m = \langle p_2, p_3 \rangle$$

is not reachable

Place invariants

Proving properties

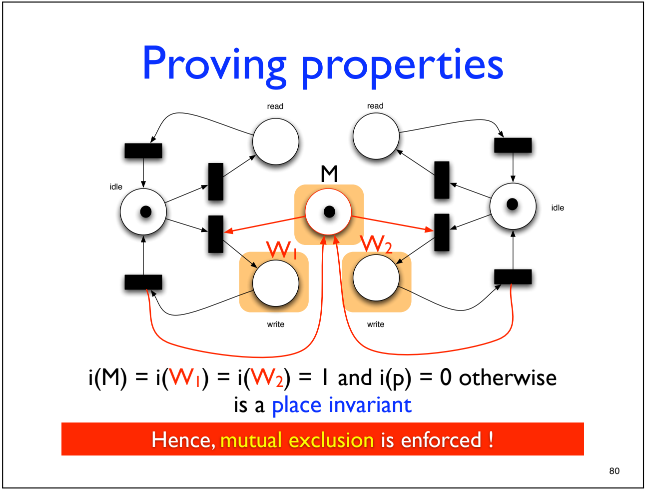


$i(M) = i(W_1) = i(W_2) = 1$ and $i(p) = 0$ otherwise
is a place invariant

80

©G. Geeraerts

Place invariants



©G. Geeraerts

The marking

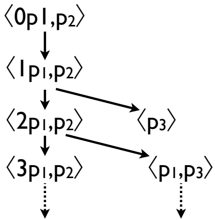
$$m = \langle W_1, W_2 \rangle$$

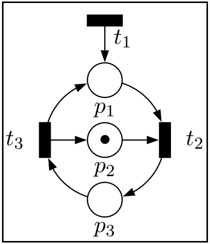
is not reachable

Karp and Miller tree

The reachability tree revisited

- Reminder: reachability trees can be infinite



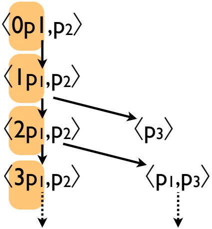


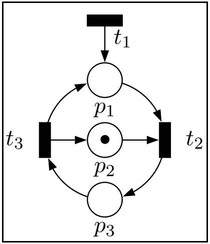
82

Karp and Miller tree

The reachability tree revisited

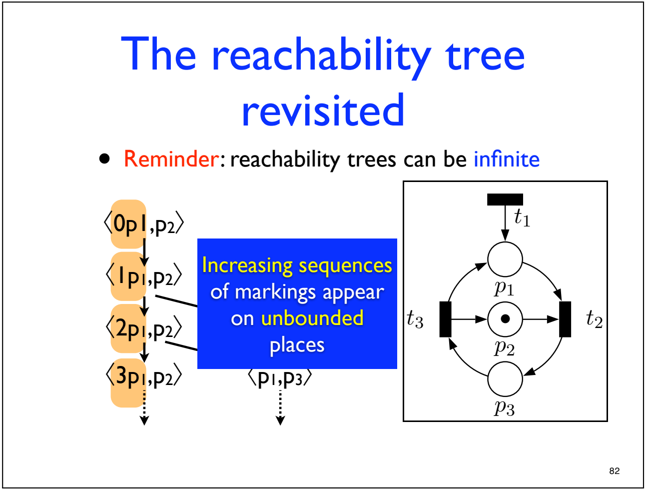
- Reminder: reachability trees can be infinite



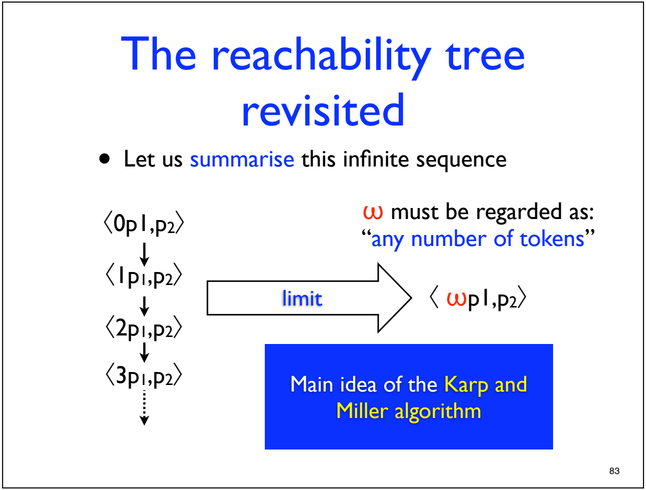


82

Karp and Miller tree



Karp and Miller tree



©G. Geeraerts

Karp and Miller tree

Karp & Miller

- Propose in 1969 a solution to detect unbounded places of a Petri net

84

Karp and Miller tree

Monotonicity

- Petri nets induce (strongly) monotonic transition systems:

$$\begin{array}{ccc} m_3 & \xrightarrow{t} & m_4 \\ \succ & & \succ \\ m_1 & \xrightarrow{t} & m_2 \end{array}$$

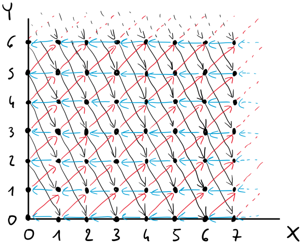
- In particular:

if $\langle i_1, i_2, i_3 \rangle \xrightarrow{\quad} \langle i_1', i_2', i_3' \rangle$, with $i_2 < i_2'$, then p_2 is unbounded

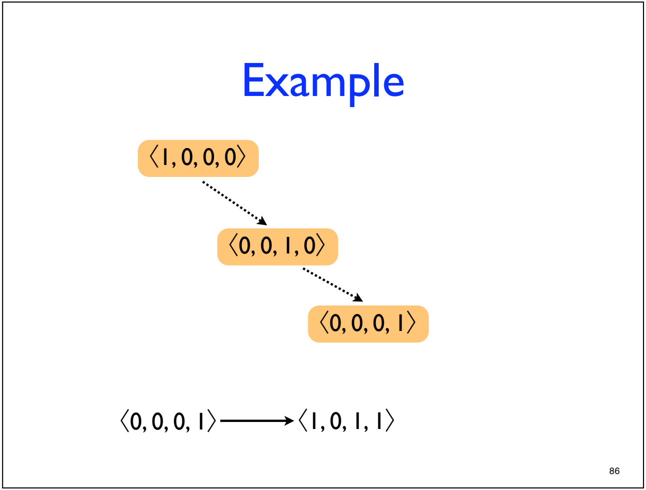
85

©G. Geeraerts

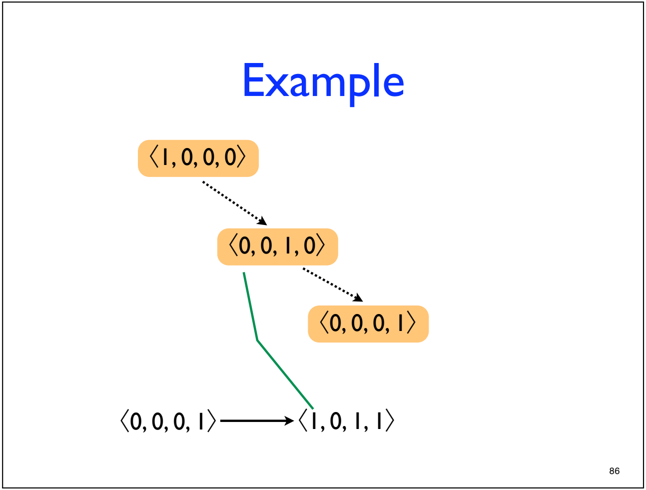
This is clearer on the cartesian plane representation:



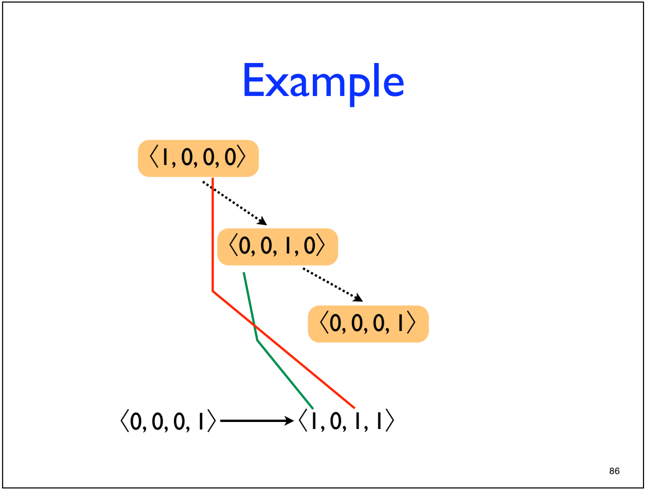
Karp and Miller tree



Karp and Miller tree

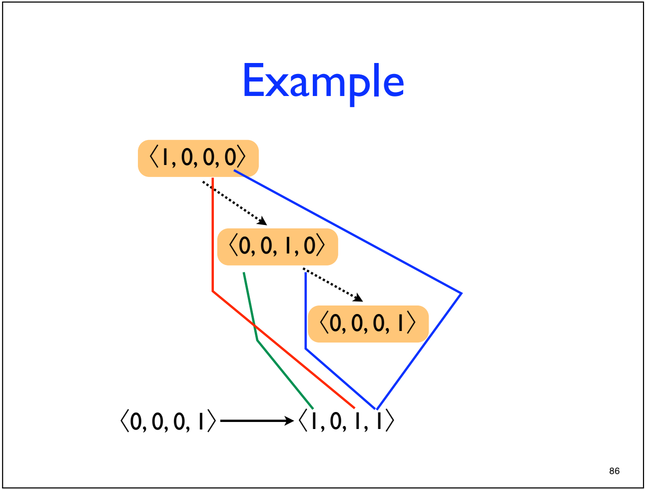


Karp and Miller tree



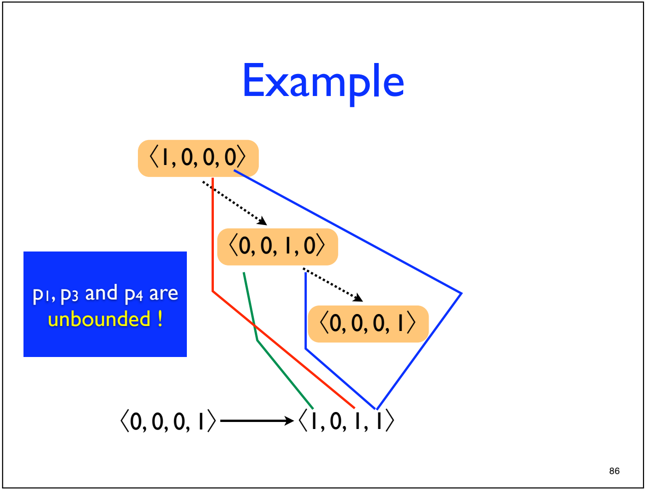
©G. Geeraerts

Karp and Miller tree

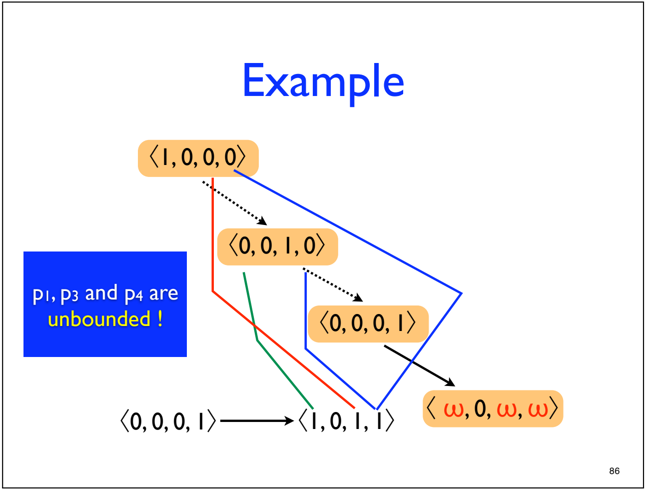


©G. Geeraerts

Karp and Miller tree

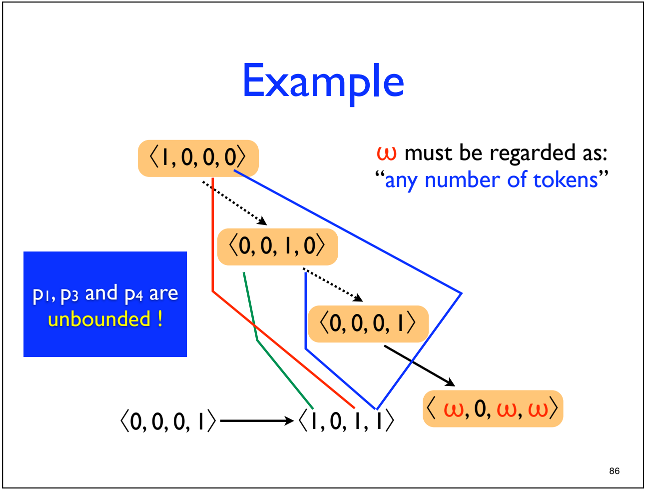


Karp and Miller tree



©G. Geeraerts

Karp and Miller tree



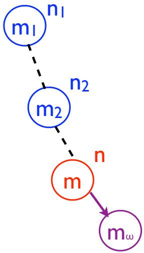
©G. Geeraerts

Karp and Miller tree

Karp & Miller Acceleration

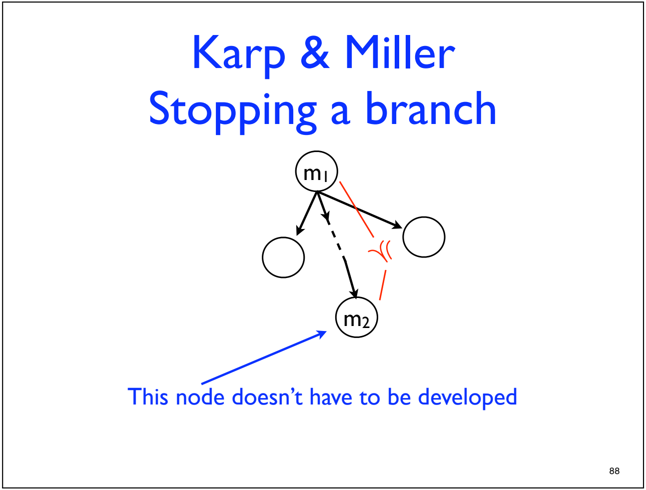
This is how we compute the successors of a node n:

foreach *Successor* m' *of* m **do**
  $m_\omega \leftarrow m'$;
  foreach *ancestor* n_i *s.t.* $m_i \prec m'$ **do**
    foreach *place* p *s.t.* $m_i(p) < m'(p)$ **do**
      $m_\omega(p) \leftarrow \omega$;
  Add m_ω as child of n;



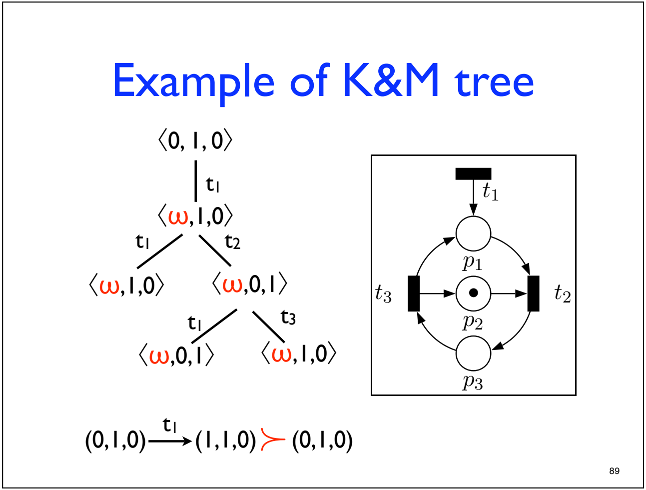
87

Karp and Miller tree



©G. Geeraerts

Karp and Miller tree



Karp and Miller tree

Properties

- Theorem: the K&M tree is always finite.
  - Idea of the proof:
    - if the net is not bounded, it is because of some infinite increasing sequence of markings.
    - such sequences are detected in a finite amount of time by adding ω in the unbounded places.

90

Karp and Miller tree

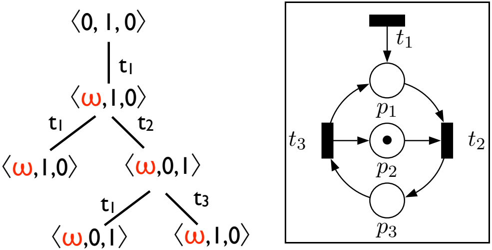
Properties

- Theorem: a net is bounded iff there is no node containing an ω in its K&M tree.
- Theorem: place p is unbounded iff there exists a node labeled by m in the K&M tree s.t. m(p) = ω.
- Theorem: transition t is semi-live iff there exists a node labeled by m in the K&M tree s.t. t can fire in m.

91

Karp and Miller tree

Example



t_2 is semi-live

p_1 is unbounded

p_2 and p_3 are bounded

The net is unbounded

92

Coverability set

Coverability set

- Question: what is the relationship between:
  - the set of reachable markings and
  - the set of labels of the nodes of the K&M tree ?

93

Coverability set

Coverability set

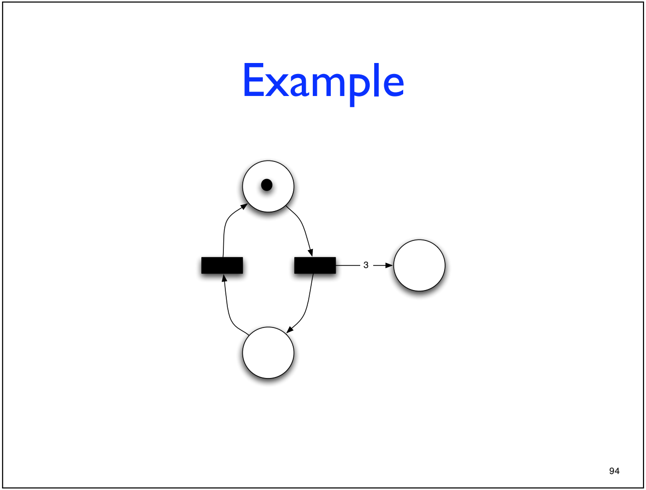
- Question: what is the relationship between:
  - the set of reachable markings and
  - the set of labels of the nodes of the K&M tree ?

might be infinite

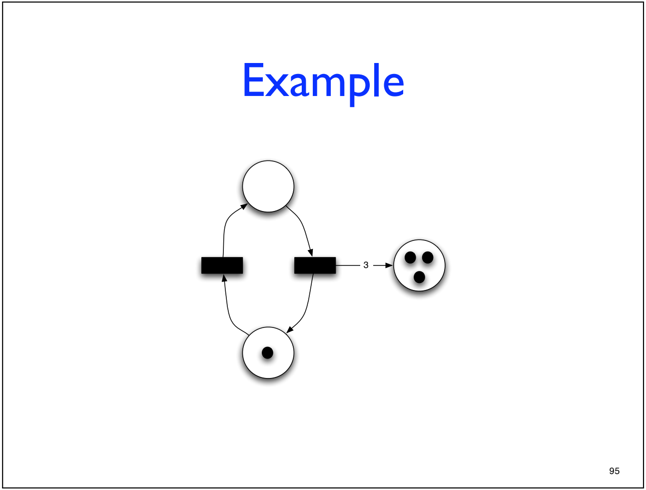
always finite

93

Coverability set



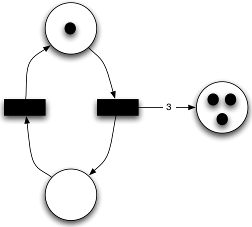
Coverability set



©G. Geeraerts

Coverability set

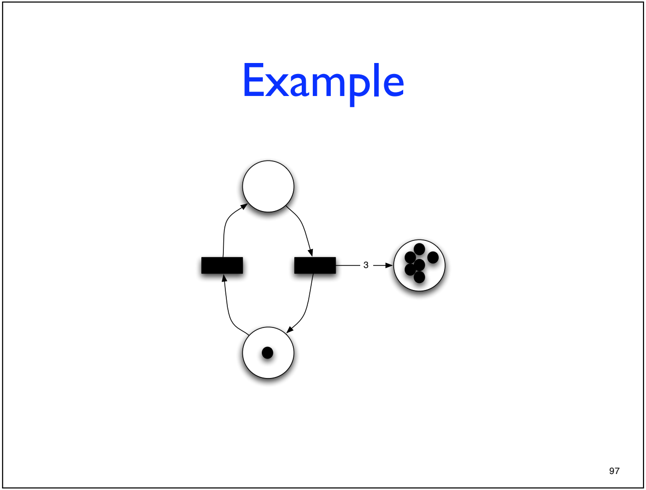
Example



96

©G. Geeraerts

Coverability set



Coverability set

Example

- Set of reachable markings:

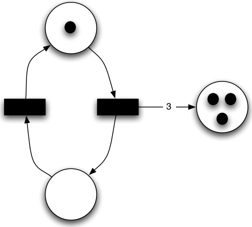
{ ⟨1, 0, 3.i⟩ , ⟨0, 1, 3.i⟩ | i⩾0 }

- Set of nodes of the K&M tree:

{ ⟨1, 0, 0⟩ ⟨1, 0, ω⟩ , ⟨0, 1, ω⟩ }

- This set "represents":

{ ⟨1, 0, i⟩ , ⟨0, 1, i⟩ | i⩾0 }

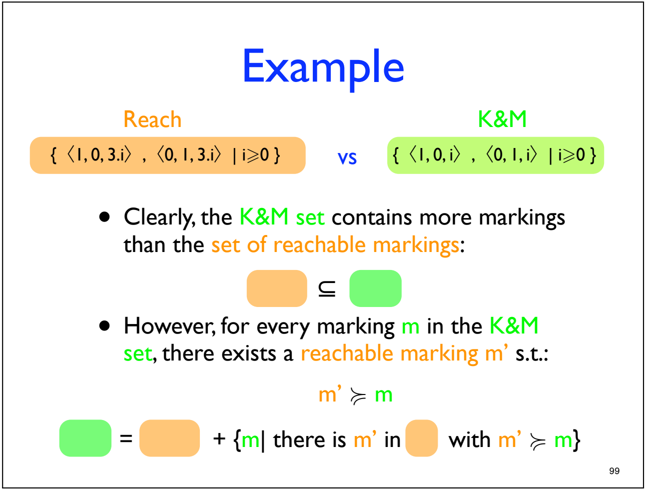


Clearly: [orange] ≠ [green]

98

©G. Geeraerts

Coverability set



Coverability set

Downward-closure

- Let us assume that any natural number i is s.t.

$$i < \omega$$

- Let m be a marking (possibly with ω), then its downward-closure is the set:

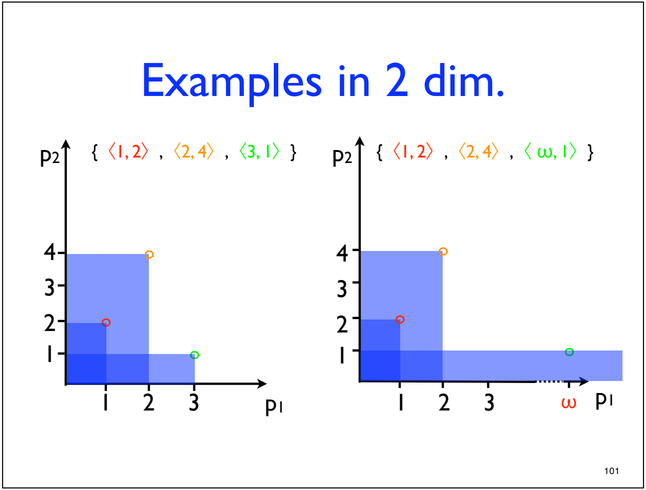
$$\downarrow m = \{m' \mid m' \preccurlyeq m\}$$

- Let $S=\{m_1, m_2, \ldots m_k\}$ be a set of markings, then:

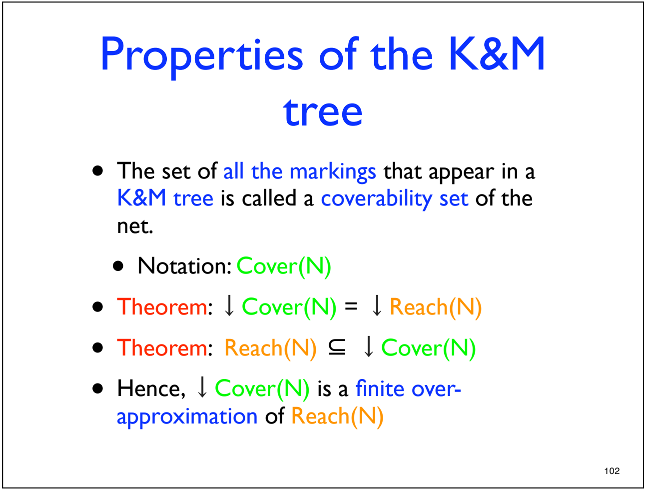
$$\downarrow S = \downarrow m_1 \cup \downarrow m_2 \cup \ldots \cup \downarrow m_k$$

100

Coverability set



Coverability set



$\downarrow Cover(N)$ is another overapproximation of the set of reachable markings

If a marking is not in $\downarrow Cover(N)$, it is not reachable!

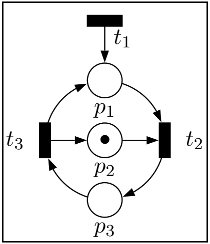
- again: sufficient condition!

Coverability set

Example

Reach(N)
=
{ ⟨i, 1, 0⟩ , ⟨i, 0, 1⟩ | i ≥ 0 }

Cover(N)
=
↓{ ⟨ω, 1, 0⟩ , ⟨ω, 0, 1⟩ }
=
Reach(N) ∪ { ⟨0, 0, 0⟩ }

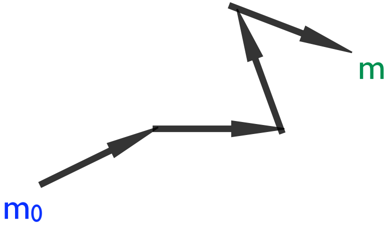


103

An alternative to reachability

Reachability: a natural question

- The reachability problem: given a marking m is it reachable from m_0 ?



106

An alternative to reachability

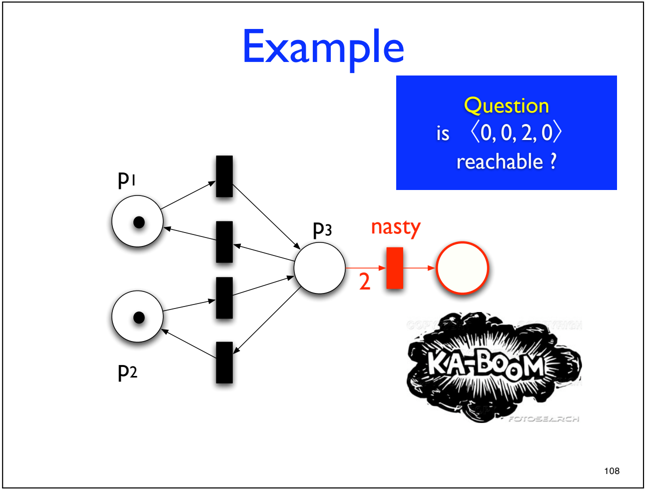
Reachability: a natural question ??

- In the case of Petri nets, asking whether a given marking is reachable does not always make sense...
- ... because Petri nets are monotonic

107

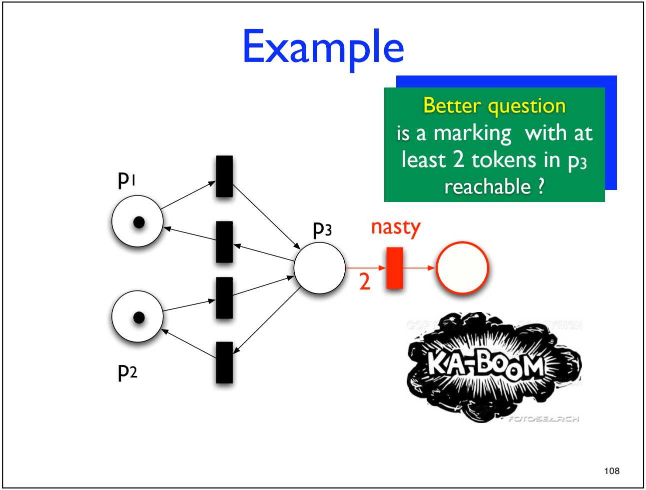
©G. Geeraerts

An alternative to reachability



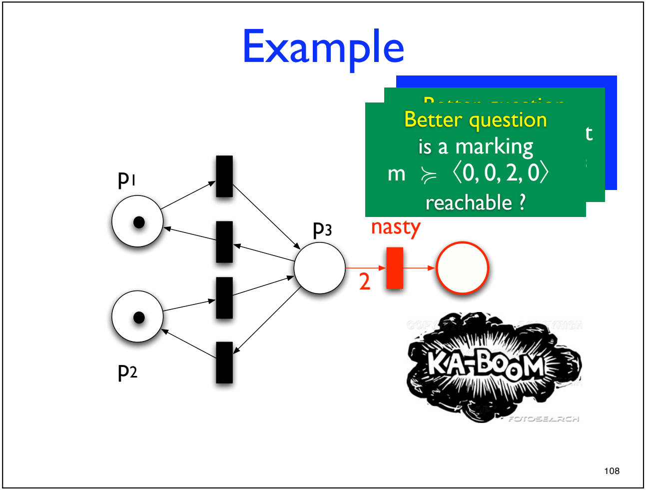
©G. Geeraerts

An alternative to reachability



©G. Geeraerts

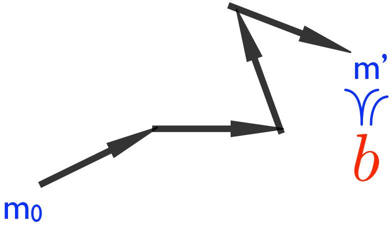
An alternative to reachability



The coverability problem

The coverability problem

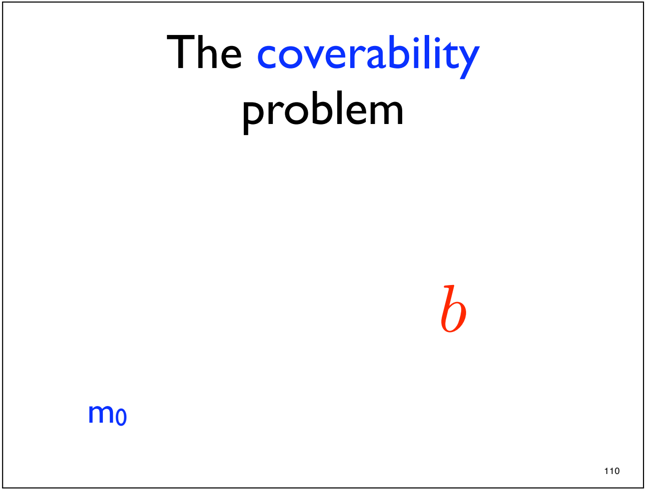
Does there exist a reachable marking which is larger than some marking b ?



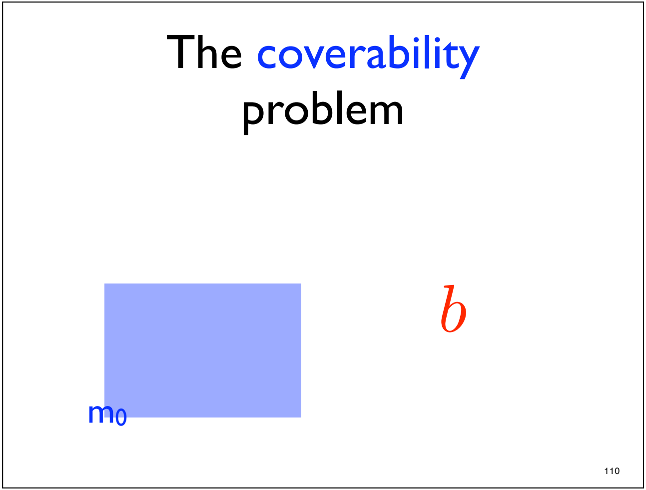
109

©G. Geeraerts

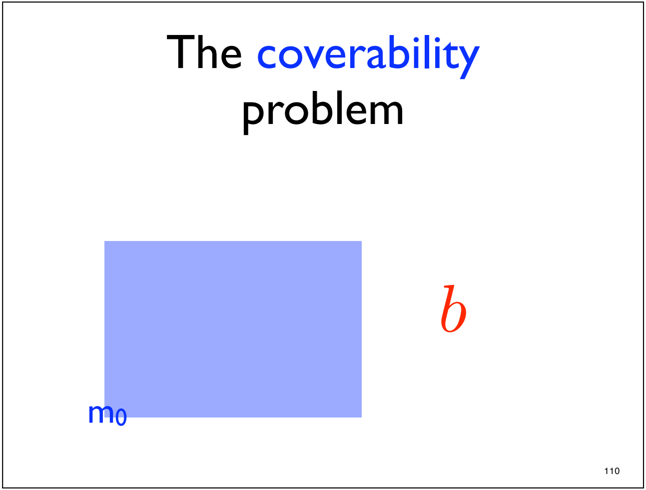
The coverability problem



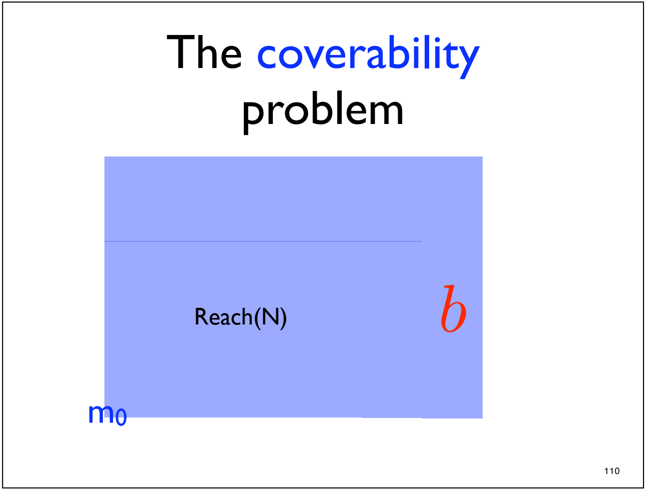
The coverability problem



The coverability problem

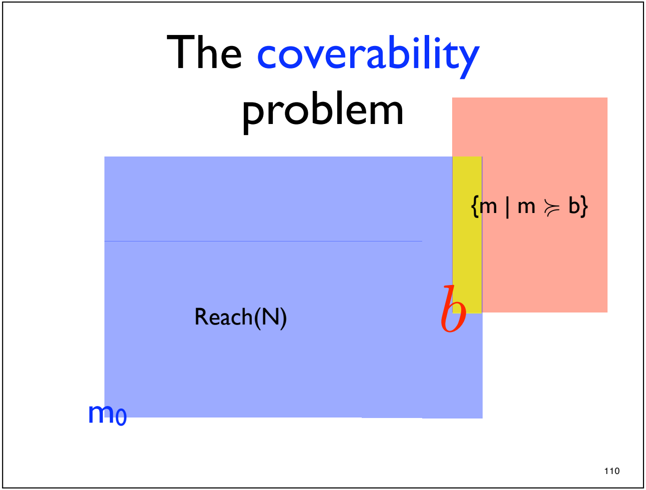


The coverability problem



©G. Geeraerts

The coverability problem



The coverability problem

The coverability problem

- Two alternative definitions:
  - Is there a reachable marking m s.t. $m \succcurlyeq b$?
  - Does $Reach(N) \cap \{m \mid m \succcurlyeq b\} \neq \phi$?

111

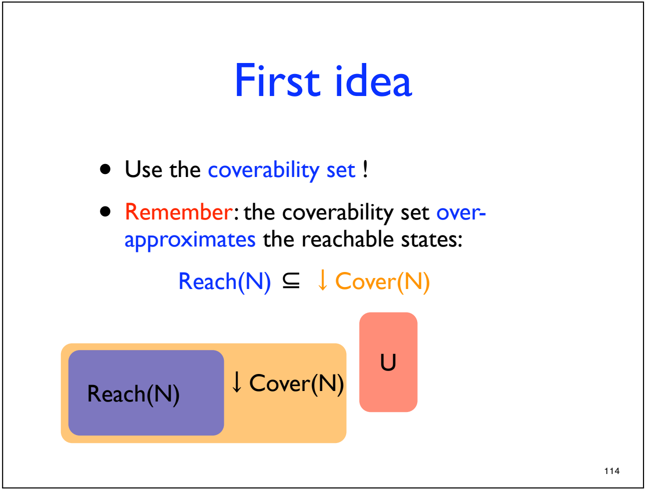
The coverability problem

Coverability: a natural question (indeed)

- Coverability might be regarded as the most natural reachability question in the framework of Petri nets
- Besides, coverability is much more easily solved than reachability

112

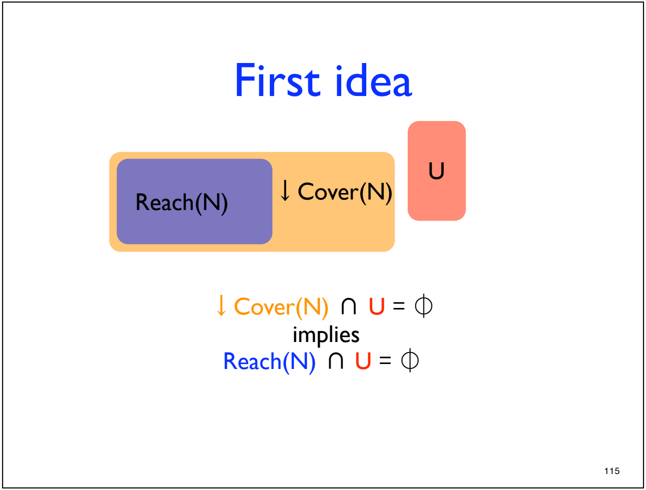
The coverability problem



©G. Geeraerts

The coverability set can be computed (easily) by using the Karp and Miller algorithm

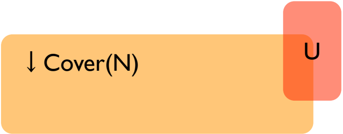
The coverability problem



©G. Geeraerts

The coverability problem

What if ?

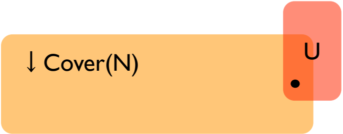


- There is m in ↓ Cover(N) ∩ U
- Hence, there is m' $\succcurlyeq$ m which is in Reach(N)
- However, any m' $\succcurlyeq$ m is also in U
- Thus, there is m' both in Reach(N) and U

116

The coverability problem

What if ?

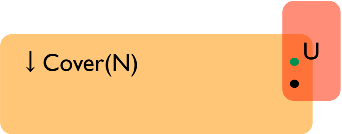


- There is m in ↓ Cover(N) ∩ U
- Hence, there is m’ $\succcurlyeq$ m which is in Reach(N)
- However, any m’ $\succcurlyeq$ m is also in U
- Thus, there is m’ both in Reach(N) and U

116

The coverability problem

What if ?

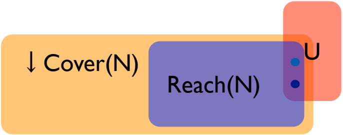


- There is m in ↓ Cover(N) ∩ U
- Hence, there is m' $\succcurlyeq$ m which is in Reach(N)
- However, any m' $\succcurlyeq$ m is also in U
- Thus, there is m' both in Reach(N) and U

116

The coverability problem

What if ?



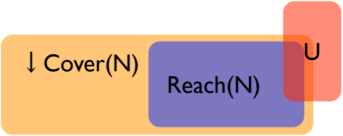
- There is m in ↓ Cover(N) ∩ U
- Hence, there is m' $\succcurlyeq$ m which is in Reach(N)
- However, any m' $\succcurlyeq$ m is also in U
- Thus, there is m' both in Reach(N) and U

116

NOTE: $m' > m$ since $\downarrow Cover = \downarrow Reach$

The coverability problem

What if ?



Reach(N) $\cap$ U = ϕ

implies

↓ Cover(N) $\cap$ U = ϕ

117

The coverability problem

Coverability set and coverability problem

- Theorem:

  Reach(N) ∩ U = ϕ iff ↓ Cover(N) ∩ U = ϕ

118

Summing up:
In order to check whether a marking in U is reachable, we can use the Karp and Miller approach to compute the $\downarrow Cover(N)$ set, and check whether it has a non-empty interesection with U!