Petri Nets and Model Checking

Natasa Gkolfi

University of Oslo

March 31, 2017

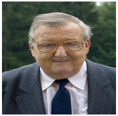
Petri Nets

Petri Nets :

- mathematically founded formalism
- concurrency
- synchronization
- modeling distributed systems

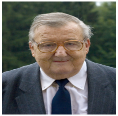
Petri Nets

Petri Nets :

- mathematically founded formalism
- concurrency
- synchronization
- modeling distributed systems
- Invented by C.A.Petri 

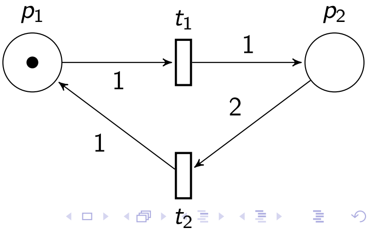
Petri Nets

Petri Nets :

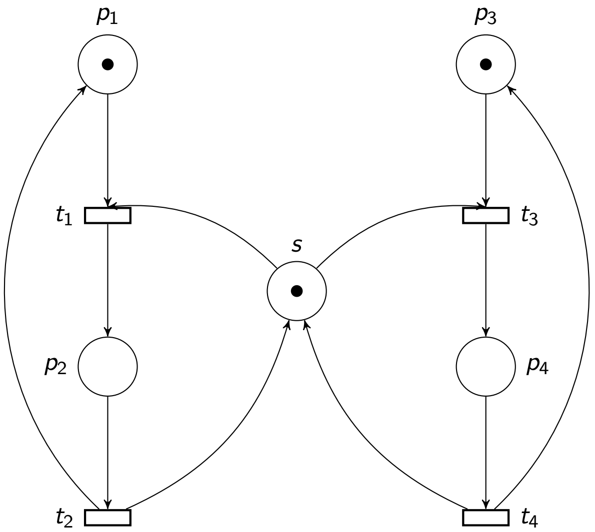
- mathematically founded formalism
- concurrency
- synchronization
- modeling distributed systems
- Invented by C.A.Petri 

They are consisting of:

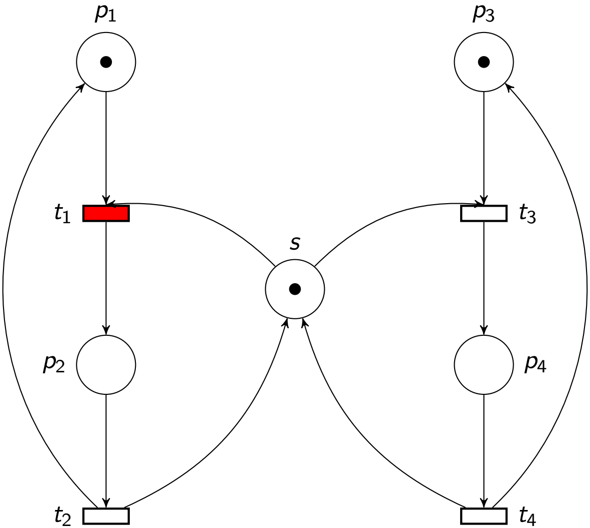
- places
- transitions
- arcs
- tokens
- initial marking



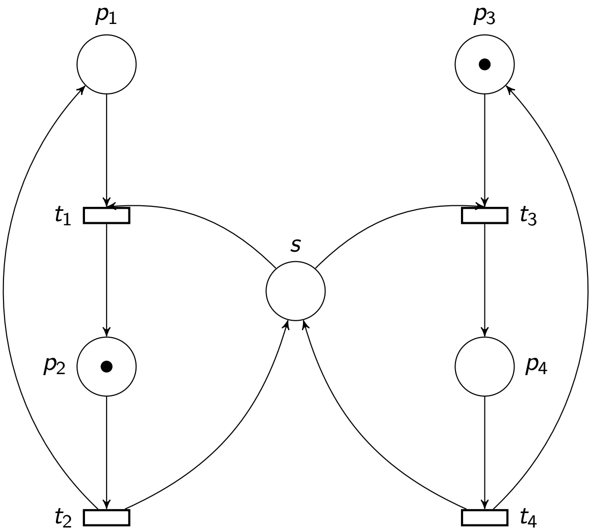
Petri Nets - Mutual Exclusion



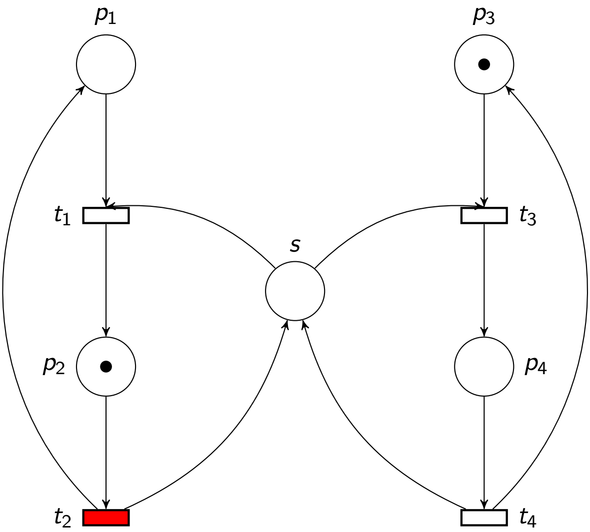
Petri Nets - Mutual Exclusion



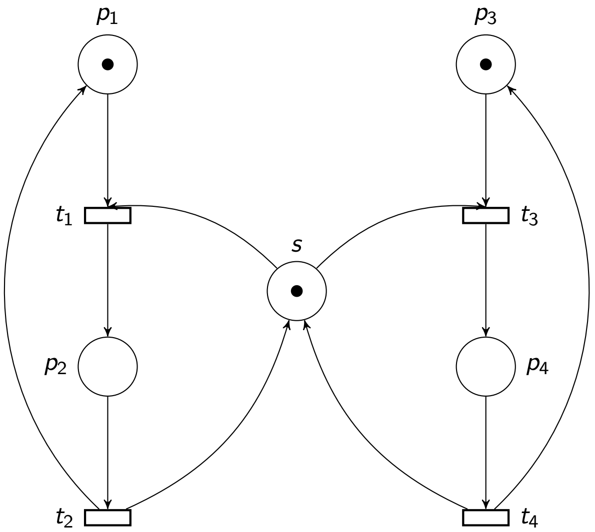
Petri Nets - Mutual Exclusion



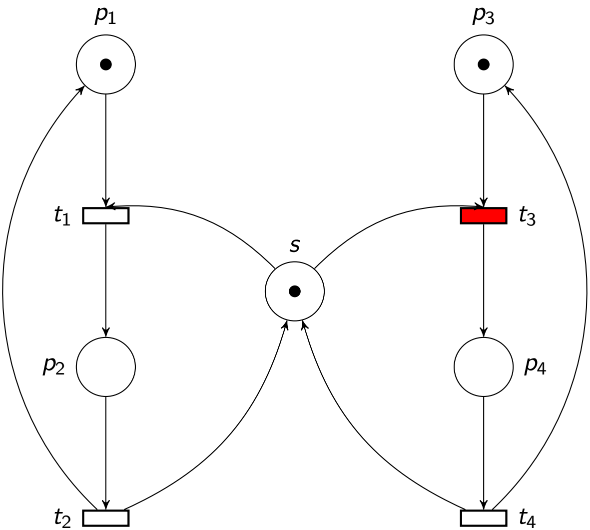
Petri Nets - Mutual Exclusion



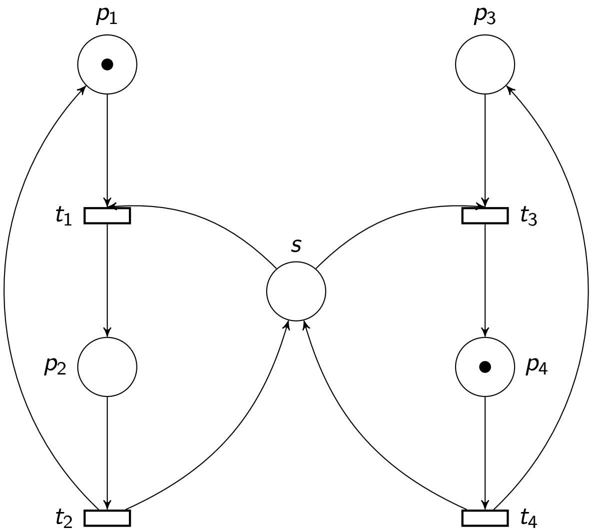
Petri Nets - Mutual Exclusion



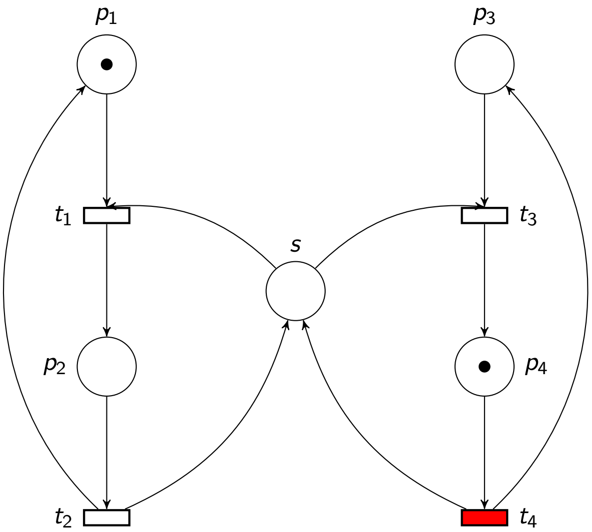
Petri Nets - Mutual Exclusion



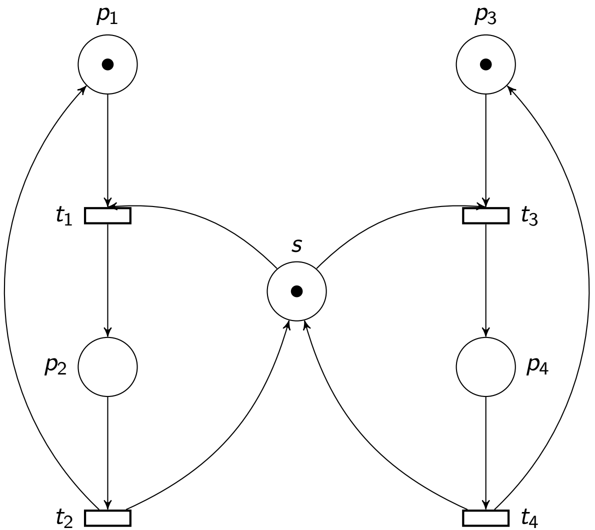
Petri Nets - Mutual Exclusion



Petri Nets - Mutual Exclusion



Petri Nets - Mutual Exclusion

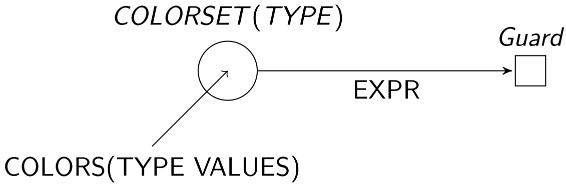


Colored Petri nets

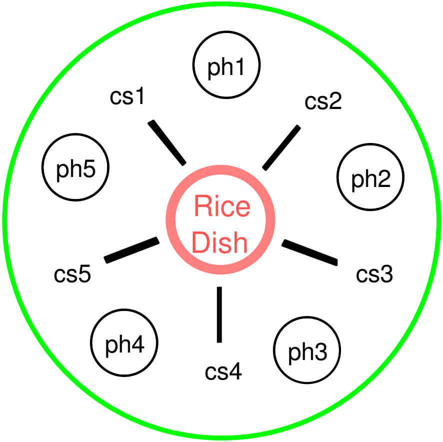
High-level Petri nets

The extension of Petri nets (called *place/transition nets*) with abstract data types.

Colored Petri nets

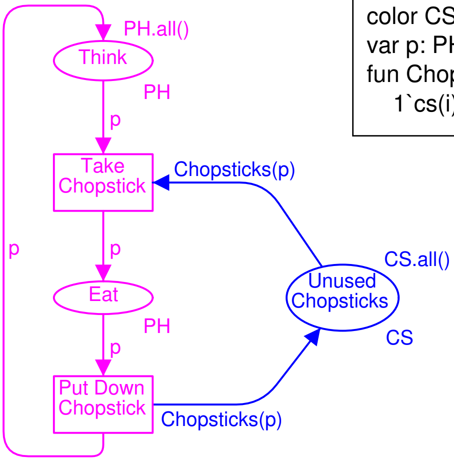


Example: Dining Philosophers



Example: Dining Philosophers

```
val n = 5;
color PH = index ph with 1..n;
color CS = index cs with 1..n;
var p: PH;
fun Chopsticks(ph(i)) =
   1`cs(i)++1`cs(if i=n then 1 else i+1);
```



State Space

State Space

A directed graph having a node for each reachable marking and an arc for each occurring binding element.

State Space

State Space

A directed graph having a node for each reachable marking and an arc for each occurring binding element.

There is one to one correspondence between the paths in the state space and the occurrence sequences (where all steps consisting of a single binding element)

State Space

State Space

A directed graph having a node for each reachable marking and an arc for each occurring binding element.

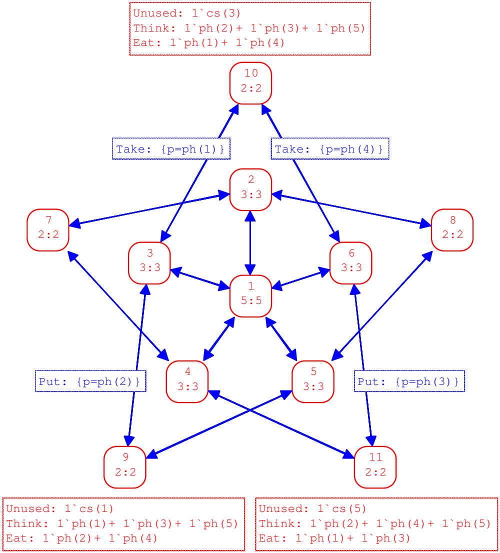
There is one to one correspondence between the paths in the state space and the occurrence sequences (where all steps consisting of a single binding element)

The *strongly-connected-component graph (SCC graph)* is the graph derived from the state space where each node is a SCC of the state space.

SCC graph

- is an acyclic graph
- fewer nodes than the ss mean that there exist infinite occurrence sequences
- more efficient since often much smaller than the ss

Example: Dining Philosophers State Space



Behavioral Properties

Boundedness properties

How many and which tokes a place may hold when all reachable markings are considered.

Home Properties

A *home marking* is a marking that can be reached from any reachable marking

- *All the markings in a (single) terminal SCC are home markings*

Behavioral Properties

Liveness Properties

A *dead marking* is a marking in which no binding elements are enabled.
Similarly *dead transition*.
A transition is *live* if, starting from any reachable marking, we can always find an occurrence sequence containing it.

Behavioral Properties

Liveness Properties

A *dead marking* is a marking in which no binding elements are enabled.
Similarly *dead transition*.
A transition is *live* if, starting from any reachable marking, we can always find an occurrence sequence containing it.

Fairness Properties

How often transitions occur in infinite occurrence sequences.
A transition is *impartial* if it occurs infinitely often in all infinite occurrence sequences.

- *Removal of this transition implies no infinite occurrence sequences!*

Example: Dining Philosophers

|PH|	Nodes	Arcs
2	3	4
3	4	6
4	7	16
5	11	30
6	18	60
7	29	112
8	47	208
9	76	378
10	123	680
15	1,364	11,310

State Space Reduction Methods

- Sweep-Line method

A *progress measure* is a function that maps each marking into a *progress value*.
For a given marking, the progress value of any successor marking must be greater or equal to its progress value.

State Space Reduction Methods

- Sweep-Line method

A *progress measure* is a function that maps each marking into a *progress value*.
For a given marking, the progress value of any successor marking must be greater or equal to its progress value.

- Symmetry method

Equivalence classes used for symmetric markings and symmetric binding elements.

  - the ss can be significantly reduced
  - can check all behavioral properties that are invariant under symmetry
  - computing canonical representations of markings and binding elements is computationally expensive

State Space Reduction Methods

- Sweep-Line method

A *progress measure* is a function that maps each marking into a *progress value*.
For a given marking, the progress value of any successor marking must be greater or equal to its progress value.

- Symmetry method

Equivalence classes used for symmetric markings and symmetric binding elements.

  - the ss can be significantly reduced
  - can check all behavioral properties that are invariant under symmetry
  - computing canonical representations of markings and binding elements is computationally expensive

- Equivalence method

A generalization of the symmetry method. Here, no requirement that the equivalence relations should be induced by symmetries.

Thank you!