Pascal Racloz

Symbolic representation of markings and its application to Model Checking

Swiss Federal Institute
of Technology

Ecole Polytechnique Fédérale de Lausanne

© 1995 P. Racloz

CONTENT

Introduction

- Framework
- Temporal operator and predicate transformers
- Problems to solve

Petri nets

- Predicate structure
- Operations
- Results

Algrebraic Nets

- Motivations
- Syntax vs Semantic

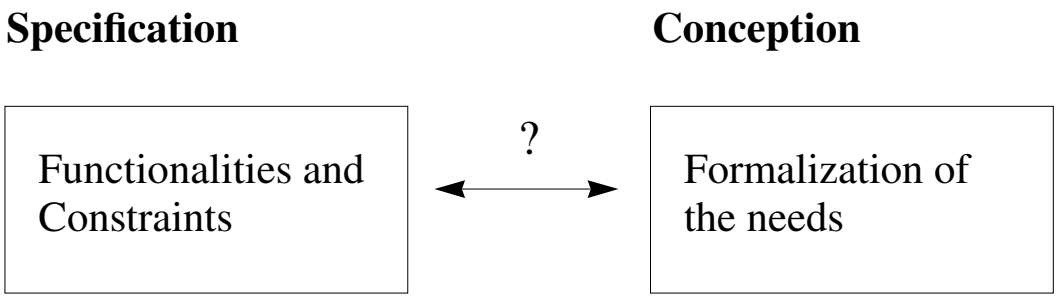
Prototype

- CO-OPN
- Environment
- Experiment

Conclusion

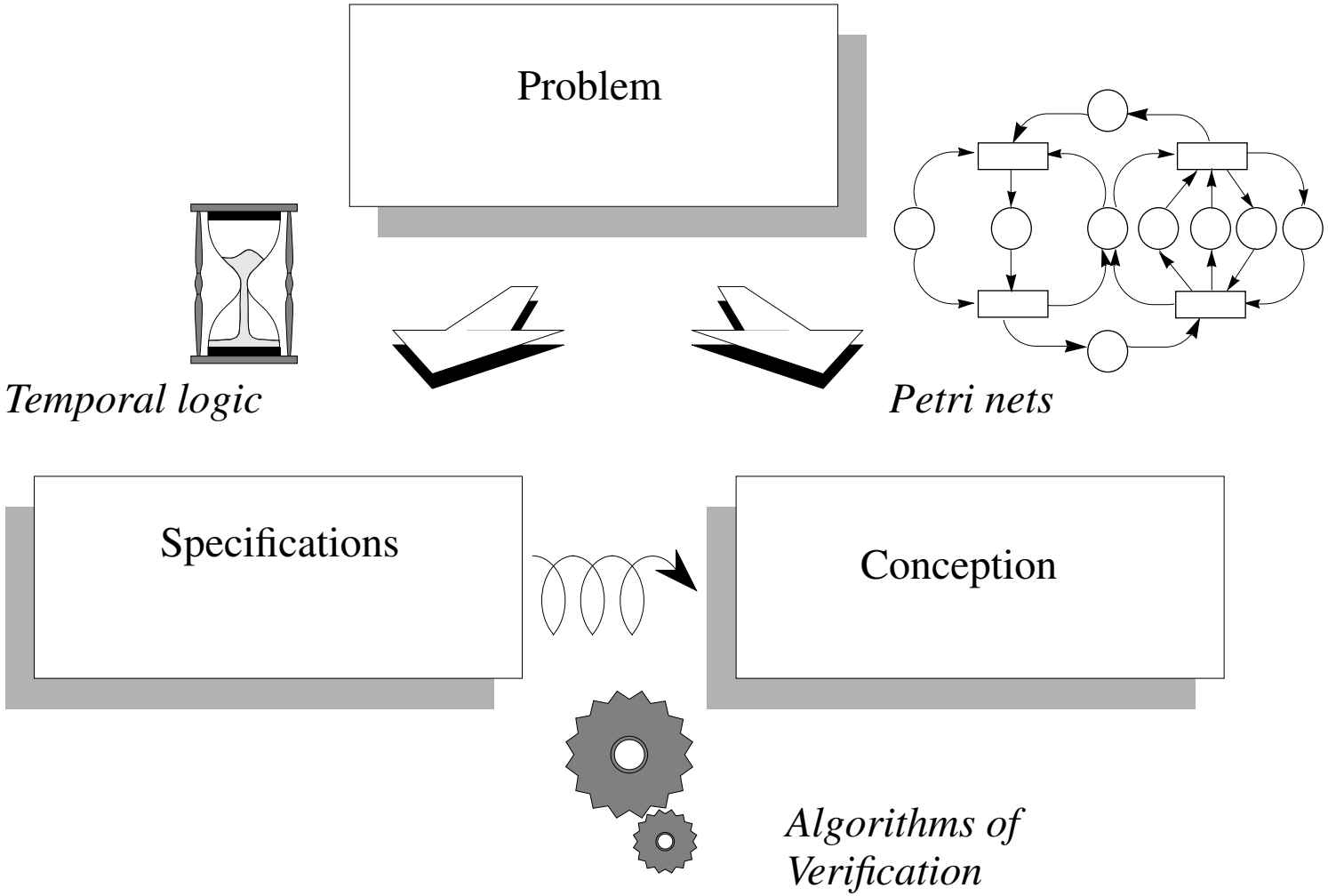
© 1995 P. Racloz

Is the conception appropriate?



- Dynamic behavior
  - Simulations, tests
- Structure
  - Study of the modelling

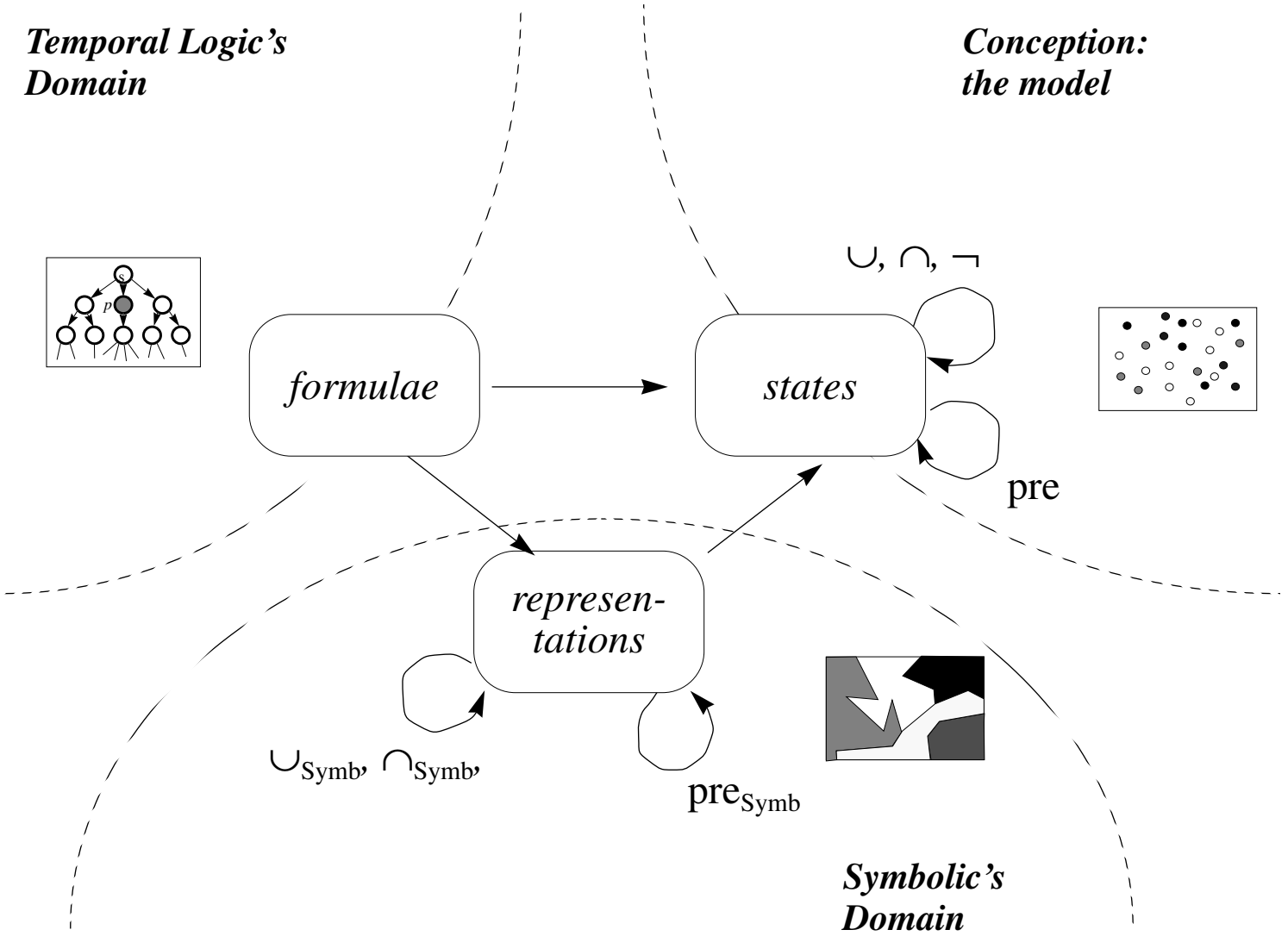
© 1995 P. Racloz



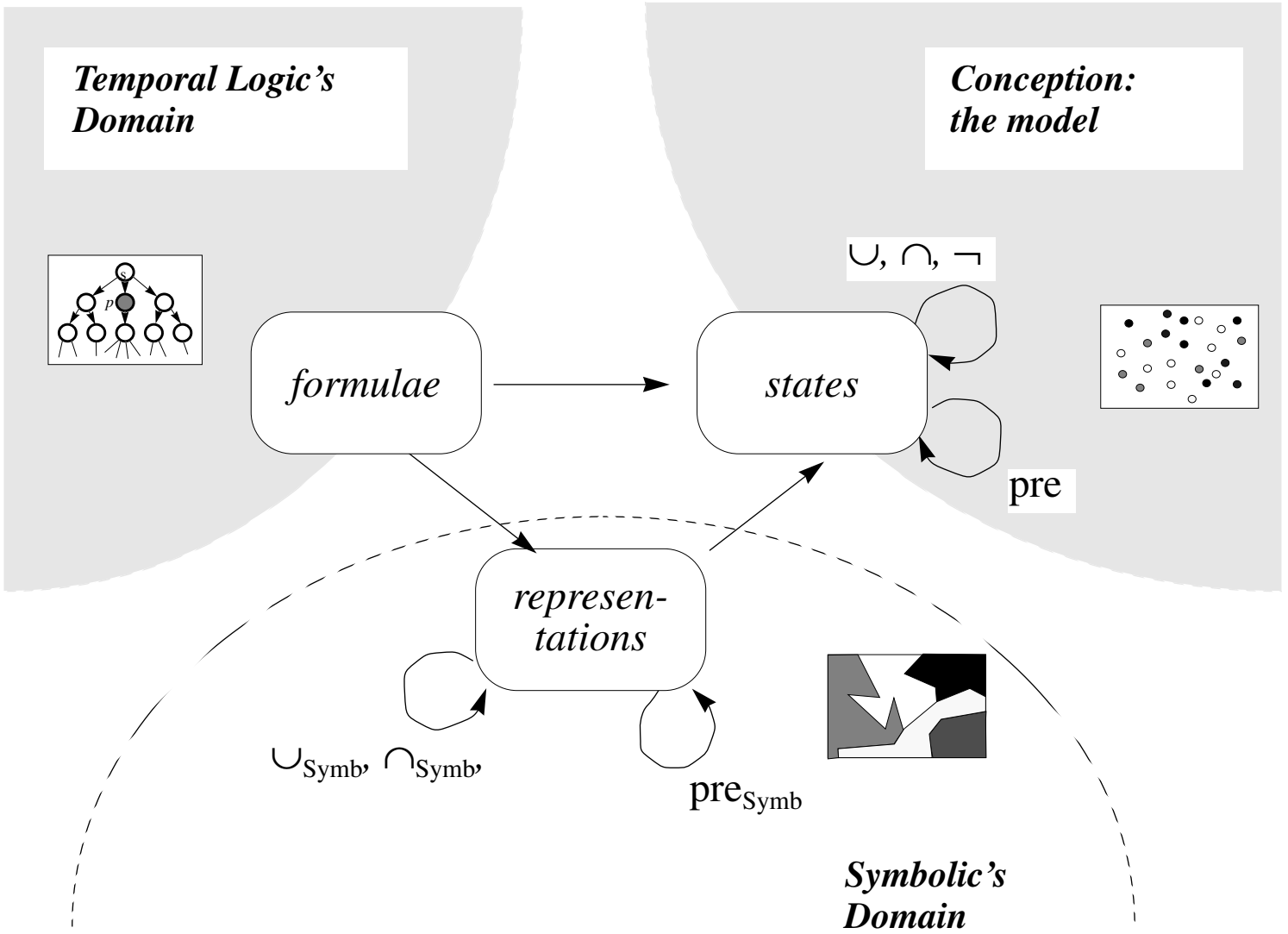
Goal:
To cope with the number of states

© 1995 P. Racloz

Motivations



© 1995 P. Racloz



Temporal logics

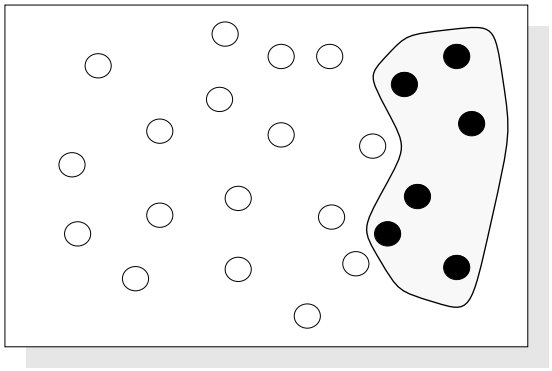
- Characteristics
  - Qualitative expression of the properties
  - Constraints on the events' arrangement making up the computations of a system
  - Intuitive meaning of the temporal operators
- Kind of properties
  - Potentiality: there exists a future...
  - Unavoidable: whatever the futures are...
  - Precedent: until... then...
  - ...
- Chosen temporal logic:

  CTL (Computation Tree Logic)

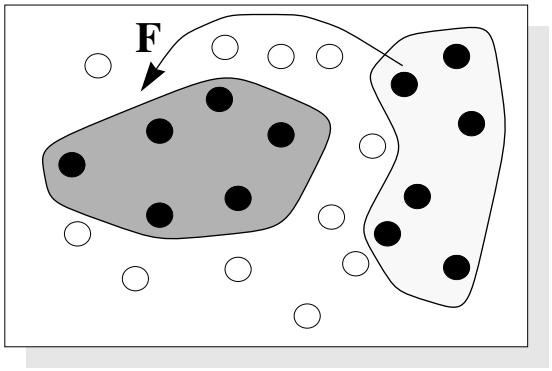
Temporal operator and functions on states

- Basic notions

Predicate



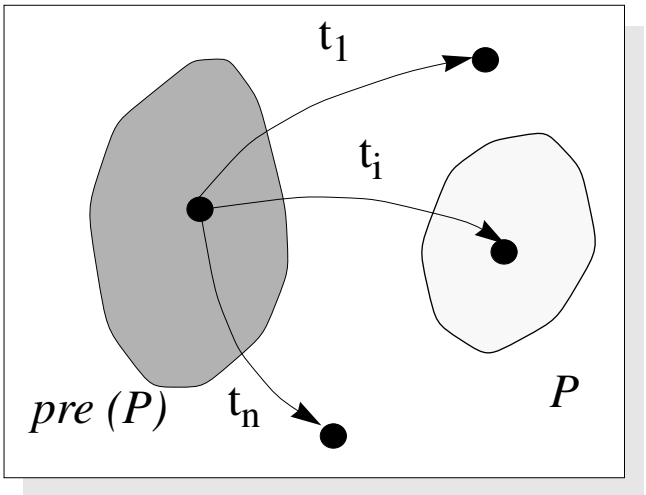
Predicate transformer



- Dynamic behavior

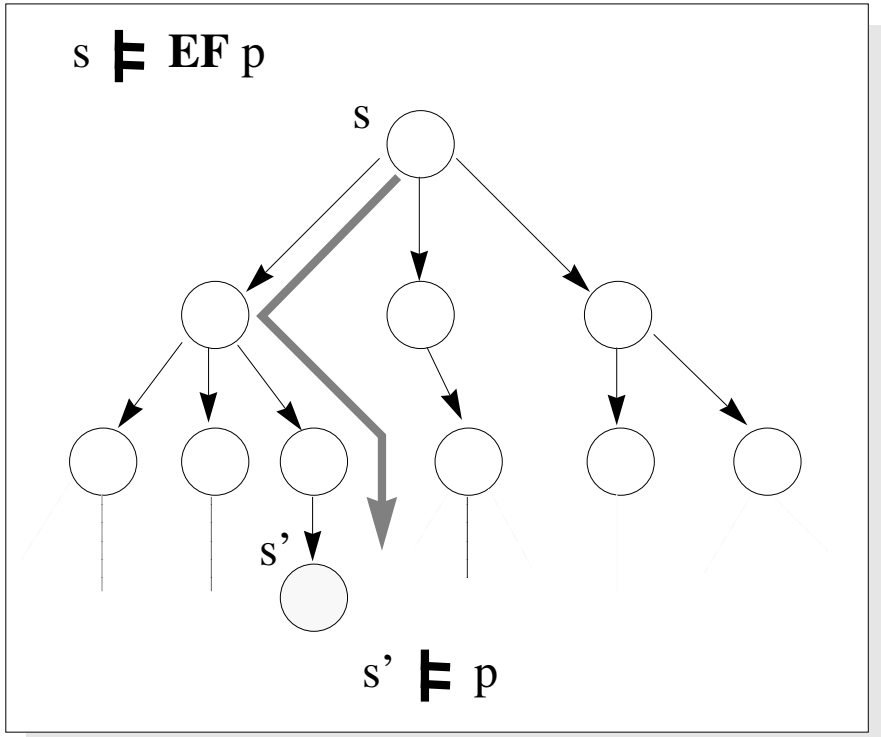
A particular predicate transformer:

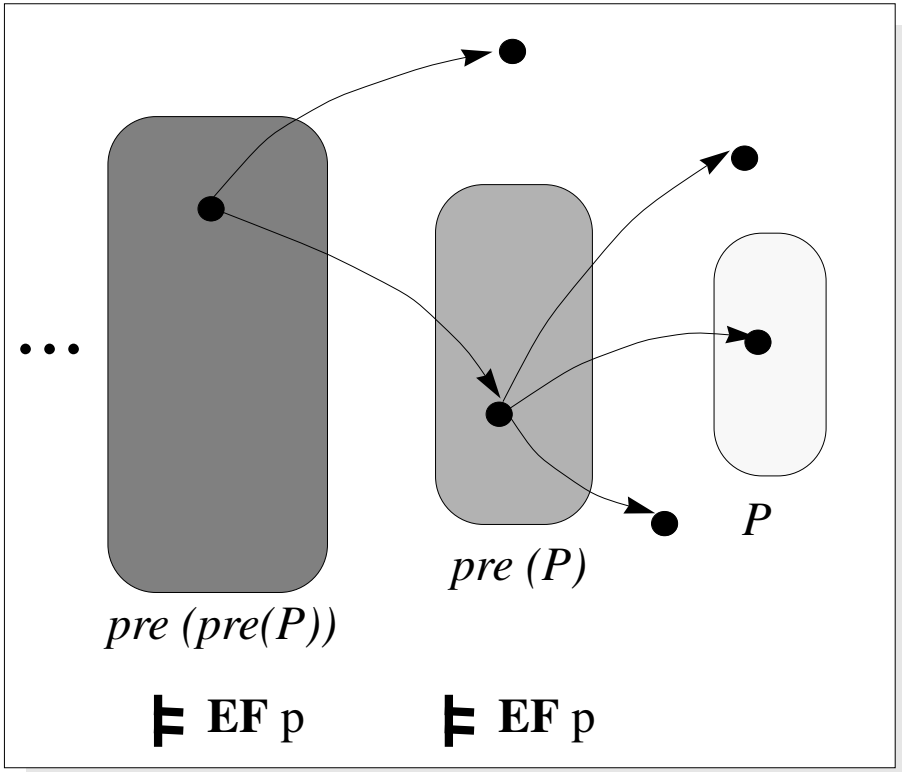
pre



© 1995 P. Racloz

- From the state *s*, the system is able to evolve towards a state *s'* where the property p est satisfied.
- From the state *s*, there exists a future where la property p est satisfied
- s ⊨ **EF** p





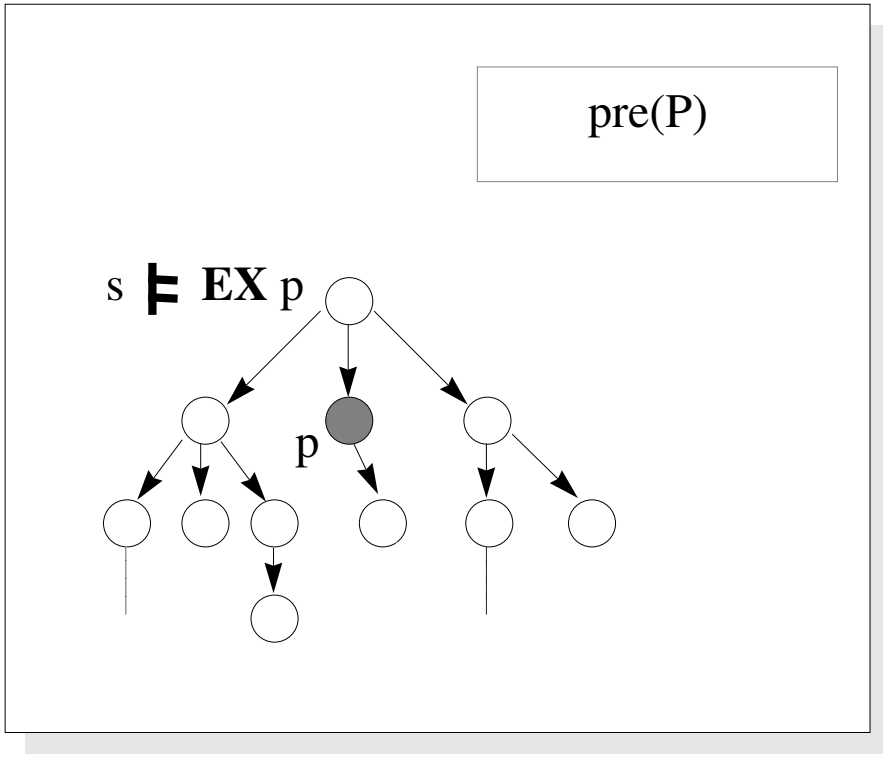
- Predicate transformer

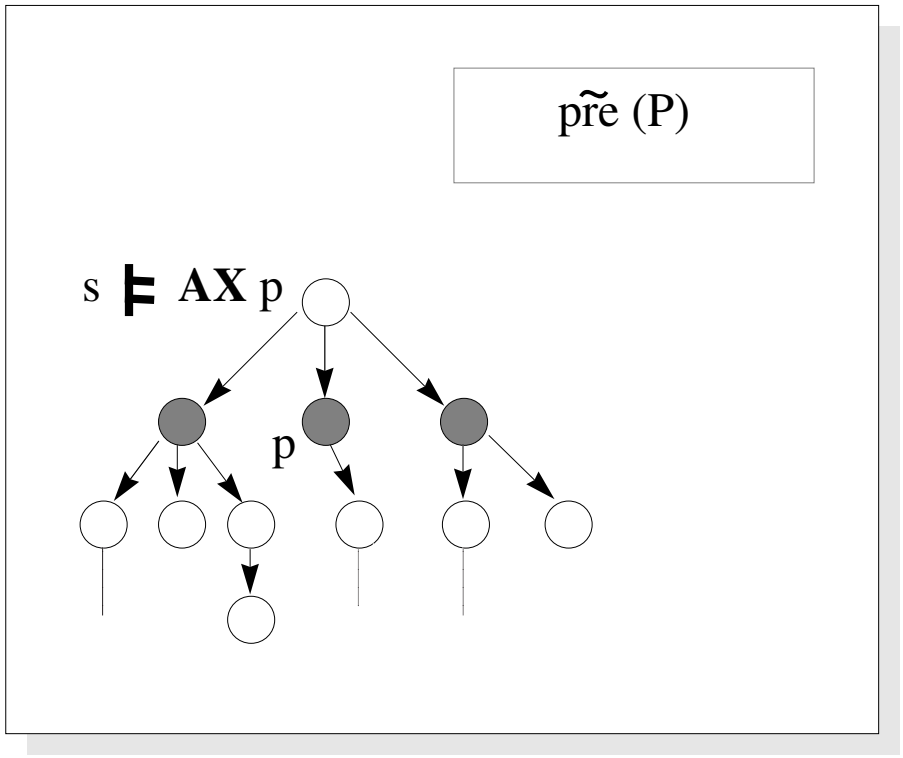
$$pre^*(P) = P \cup pre(P) \cup pre(pre(P)) \cup ...$$

© 1995 P. Racloz

EX

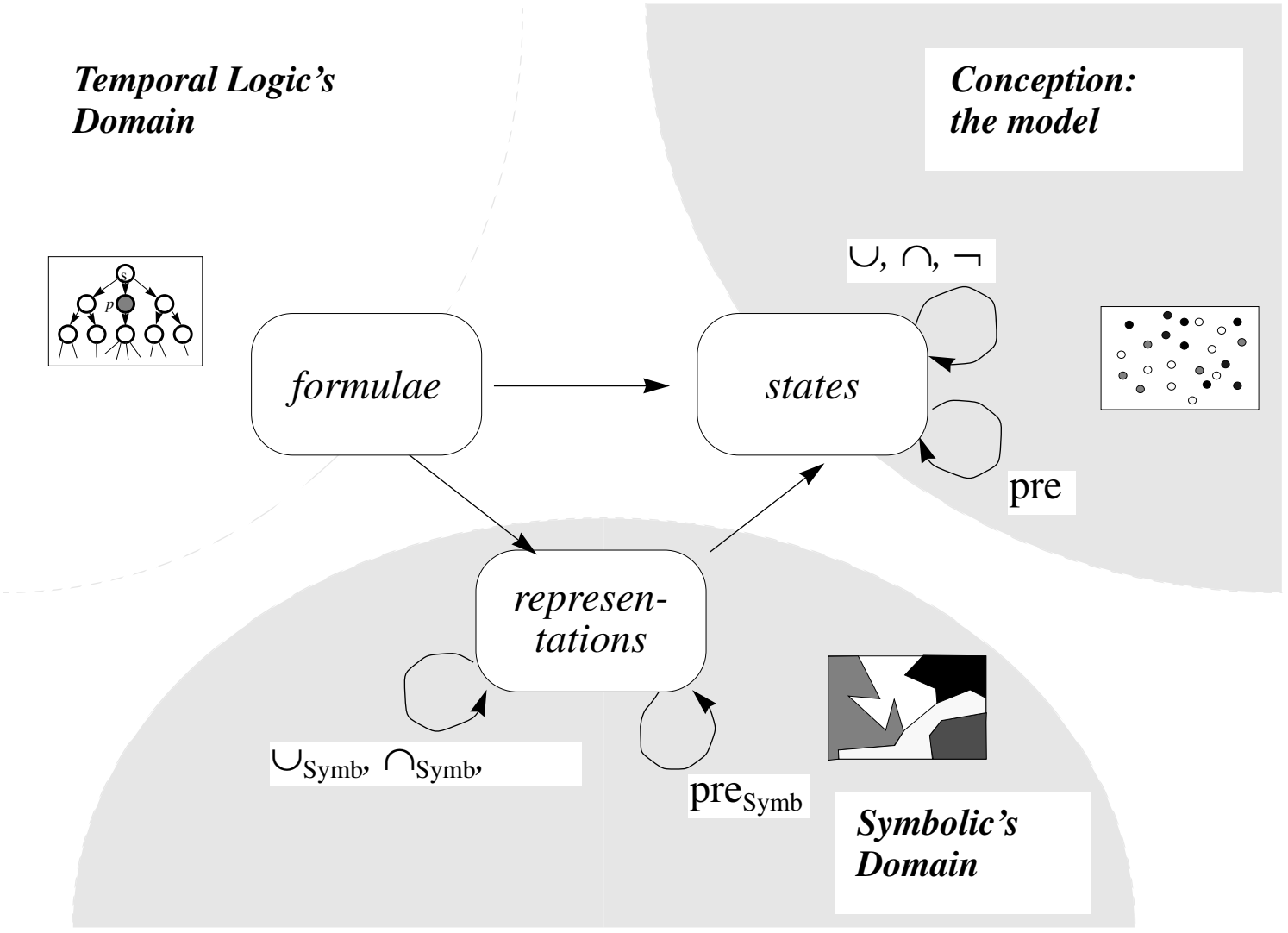
There exists at least one direct successor such that...





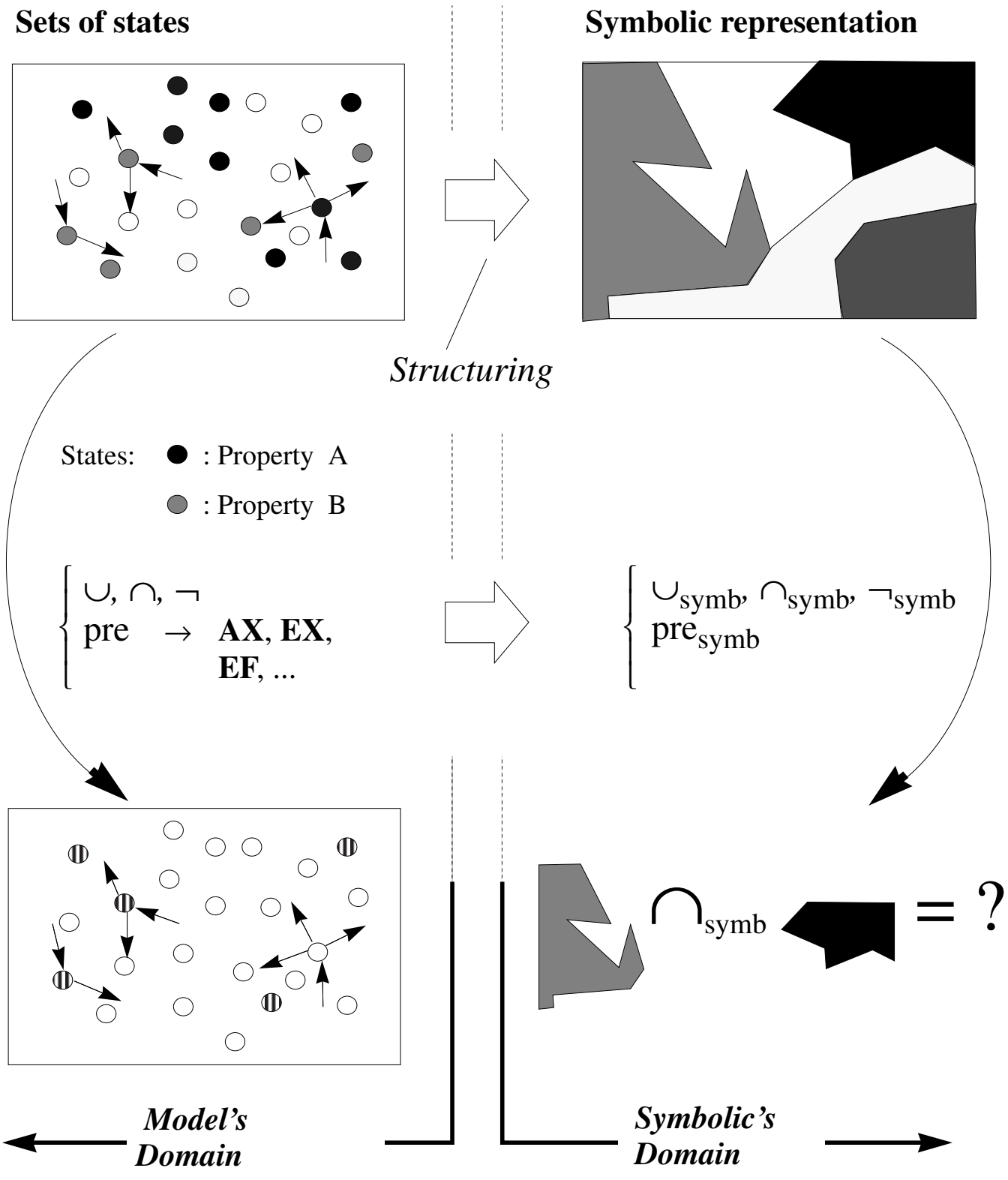
AX

For all direct successors...

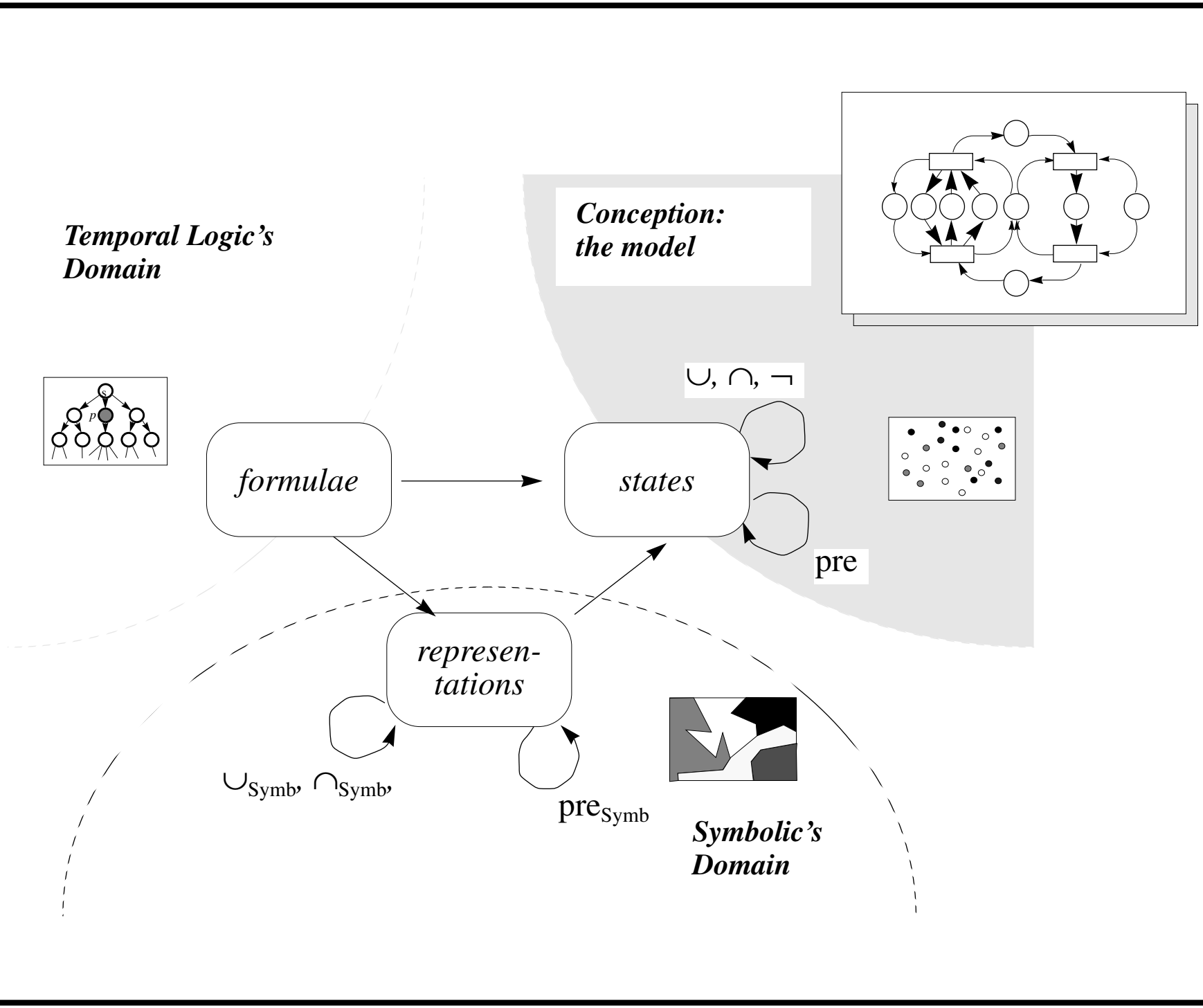


© 1995 P. Racloz

Cope with the number of states: characteristics of a symbolic representation



© 1995 P. Racloz



The formalism of Petri nets

- Modelling and analysis of discrete systems
- Distribution of the state and the evolution
- Different kind of synchronizations
- True concurrency

- Properties of modellings done with Petri nets
  - Static and structural analysis
  - Dynamic analysis

- Atomic proposition of the temporal logic:

  Can a given event happen?

Symbolic Representation

Example

Firing of t from M

$$\lambda^{in}(t) = \begin{pmatrix} in_1 \\ in_2 \\ in_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix} \subseteq M$$

Transition t not firable

$$\lambda^{in}(t) = \begin{pmatrix} in_1 \\ in_2 \\ in_3 \end{pmatrix} \not\subset M$$

- Generalization

From M...

$t_1, t_2, ..., t_n$ *firable*
and
$t'_1, t'_2, ..., t'_m$ *not firable*

$\Leftrightarrow$

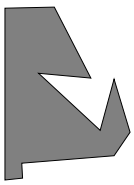
$\forall\ 1 \le i \le n\ \lambda^{in}(t_i) \subseteq M$
and
$\forall\ 1 \le j \le m\ \lambda^{in}(t'_j) \not\subset M$

- Definition

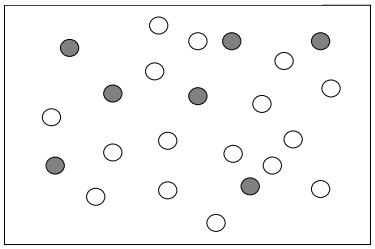
Predicate structure:

(A, B) where A,B two sets of markings

$A = \{ M_{a1}, ..., M_{an} \}$
$B = \{ M_{b1}, ..., M_{bm} \}$

(A, B) = 

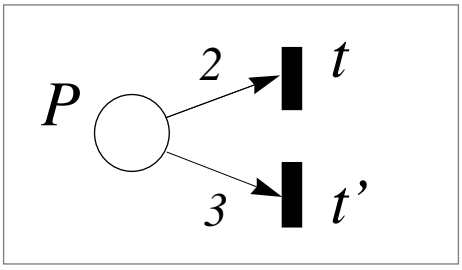
UM(A, B) = ' ● '



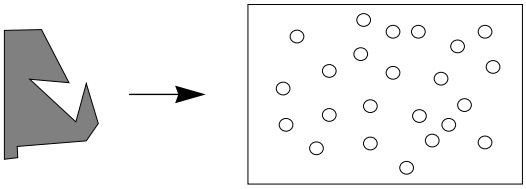
Set of states represented:
UM(A, B)

$M \in UM(A,B) \Leftrightarrow$
$\forall\ M_a \in A,\ M_a \subseteq M$
$\forall\ M_b \in B,\ M_b \not\subset M$

Predicate structure: its properties



- Incompatibility



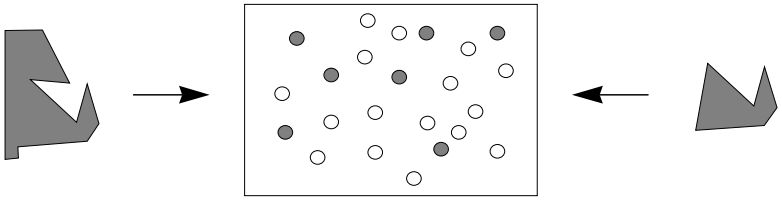
t' firable
and
t not firable

$\Rightarrow$

$(A, B) = (\{\lambda^{in}(t')\} , \{\lambda^{in}(t)\})$

$UM(A, B) = \varnothing$

- Simplifications



t' et t not firable
$\Leftrightarrow$
t not firable

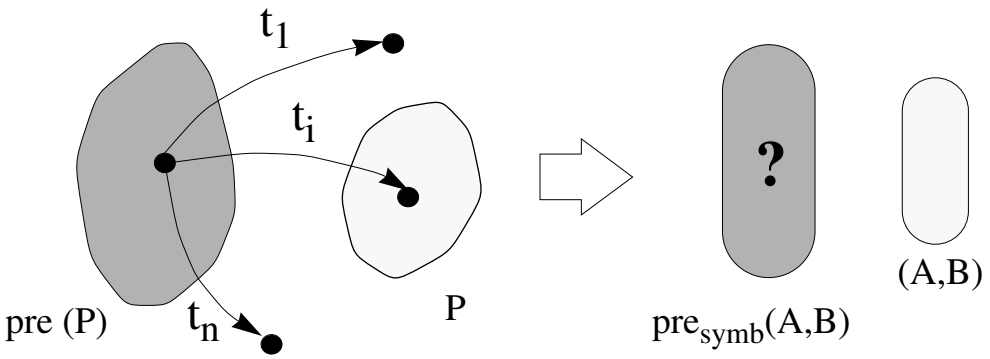
$\Rightarrow \approx$

$(A, B) = (\varnothing, \{\lambda^{in}(t'), \lambda^{in}(t)\})$

$(A', B') = (\varnothing, \{\lambda^{in}(t)\})$

Predicate structure: the associated operations

- 'Basic': union, intersection, negation
- Calculation of pre



- **preT:** M **x T** $\rightarrow M$

$\forall$ (M,t) $\in$ M x T, $\forall$ p $\in$ P

preT(M,t)(p) = λ(p,t) if M(p) $\leq$ λ(t,p)

= M(p) + λ(p,t) - λ(t,p) if λ(t,p) $\leq$ M(p)

- $\forall$ **t** $\in$ **T, preP$_t$, preN$_t$:** $P(M) \rightarrow P(M)$

(i) $preP_t(\emptyset) = \{ \lambda^{in}(t) \}$

(ii) $preP_t(F) = \bigcup_{M \in F}$ preT(M,t) where F $\subseteq$ M and F $\neq \emptyset$

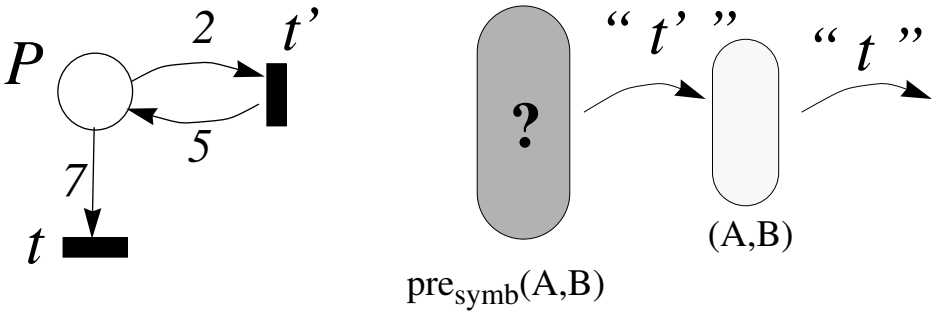
(iii) $preN_t(\emptyset) = \emptyset$

(iv) $preN_t(F) = \bigcup_{M \in F}$ preT(M,t) where F $\subseteq$ M and F $\neq \emptyset$.

- **pre$_{PS}$: Str$_M$** $\rightarrow P$**(Str$_M$)**

$\forall$ (A,B) $\in$ Str_M $pre_{PS}(A,B) = \bigcup_{t \in T}$ $(preP_t(A),preN_t(B))$

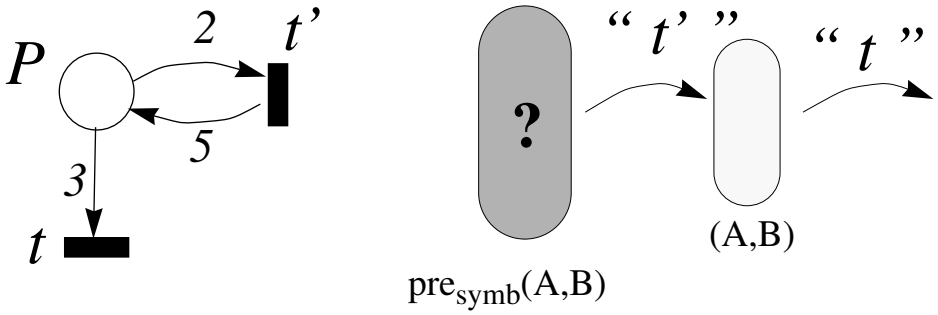
Particular cases:



(A,B) = ({(7)}, ∅)

Relatively to t' :

$pre_{symb}(A,B) = (\{(4)\}, \emptyset)$

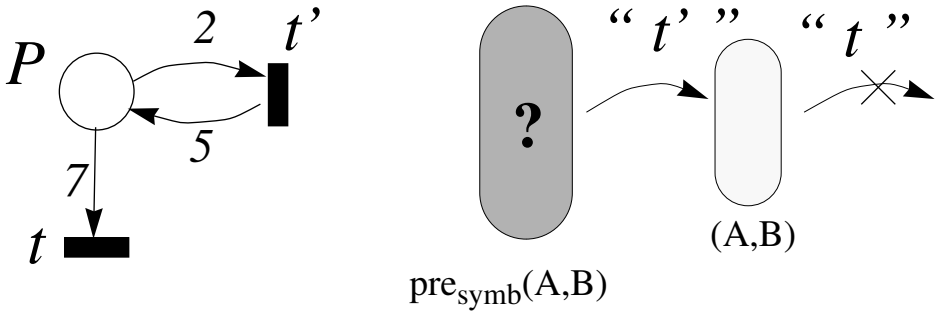


(A,B) = ({(3)}, ∅)

Relatively to t' :

$pre_{symb}(A,B) = (\{(2)\}, \emptyset)$

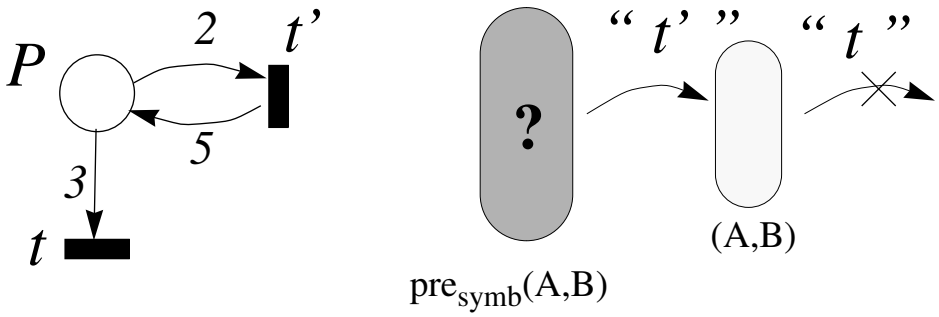
Particular cases:



(A,B) = (∅, {(7)})

Relatively to t' :

$pre_{symb}(A,B) = (\{(2)\}, \{(4)\})$

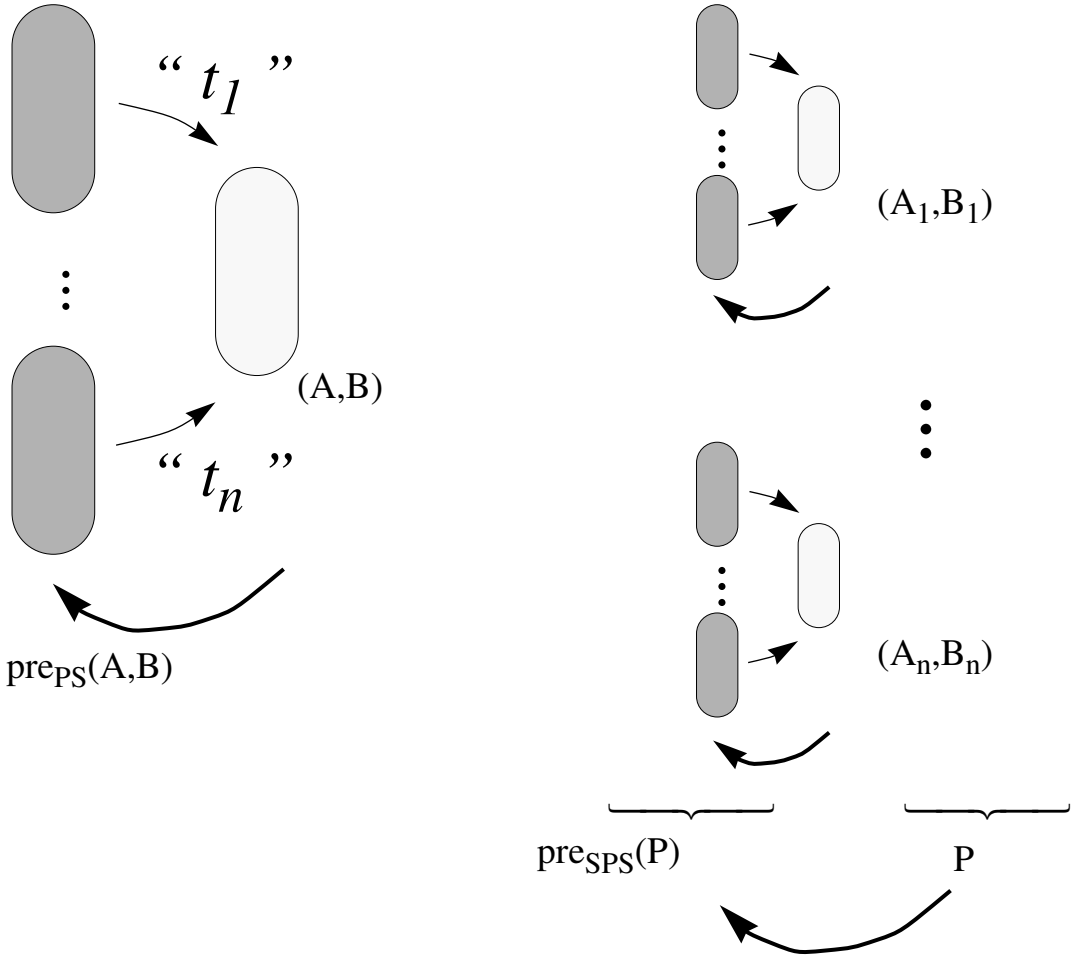


(A,B) = (∅, {(3)})

Relatively to t' :

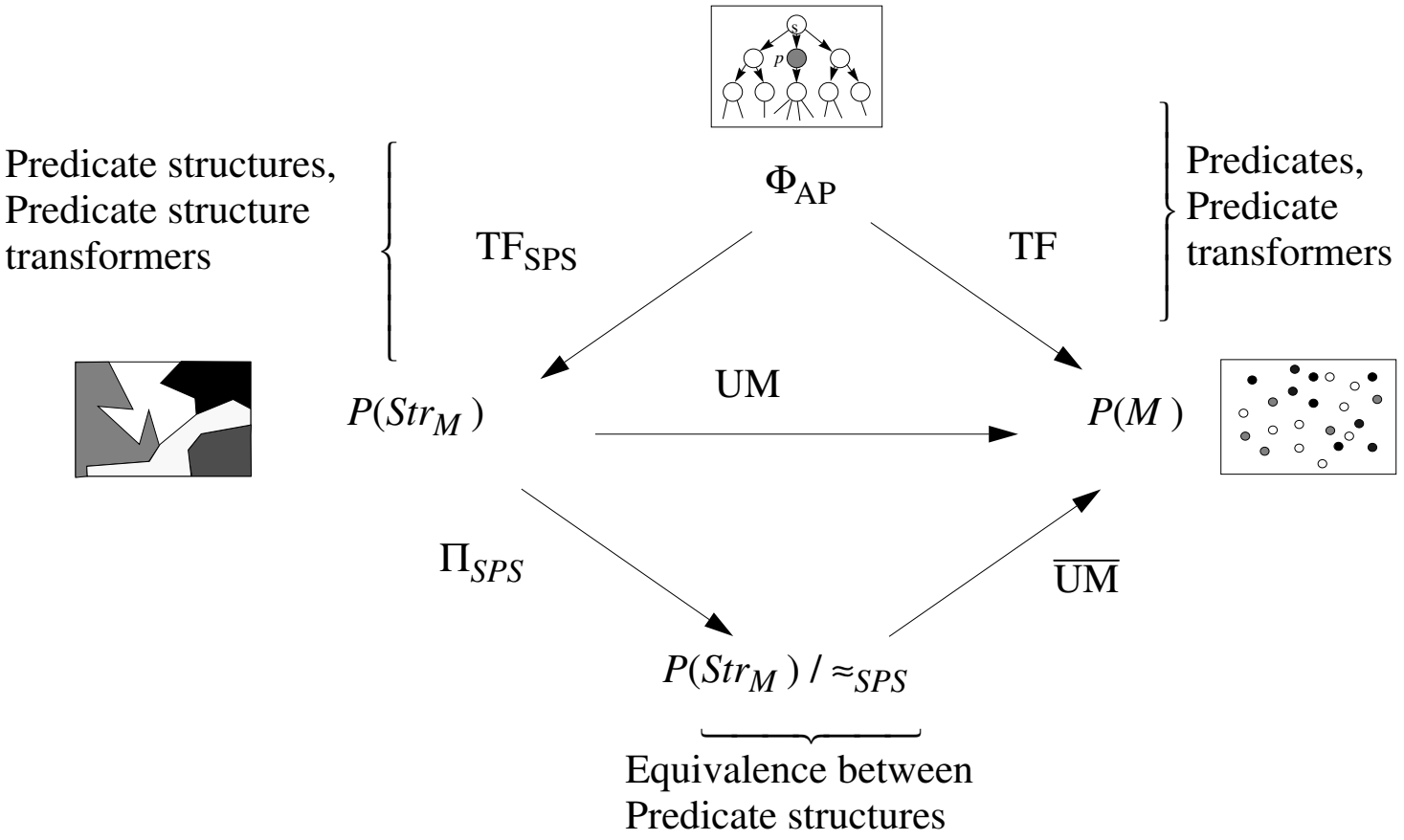
$pre_{symb}(A,B) = (\{(2)\}, \{(2)\})$

Importance of the simplification and elimination processes



© 1995 P. Racloz

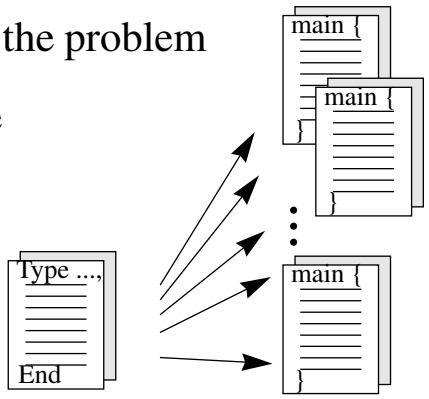
Petri nets: the results



Abstract data type

Formal specification, specification language

- Abstraction from the reality
- Expression of the relevant aspects of the problem
- Complete expression of the language
- Well defined semantics
- Genericity
- Modularity



Components of a ADT. Syntactical Domain (based on a sorted equational logic)

- Sortes
- Constructors
- Operations
- Axioms

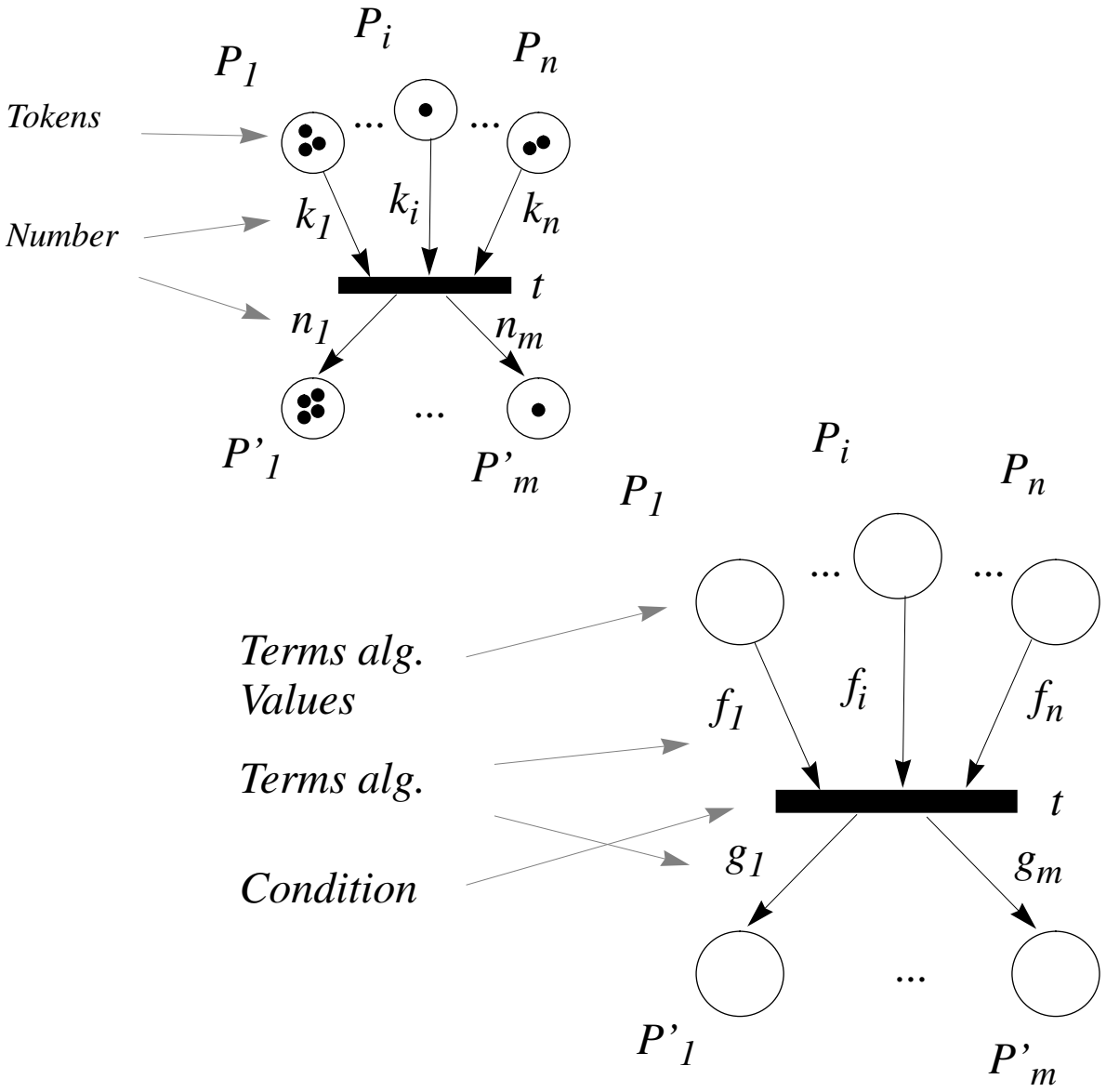
Fonction of interpretation

Semantical Domain

- Algebra
- Models

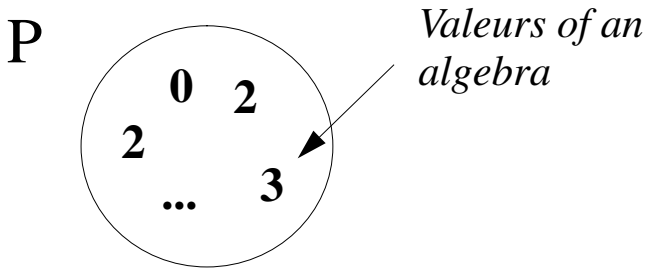
Algebraic Nets

- Petri net: structural aspect
- Abstract data type: functionnal aspects

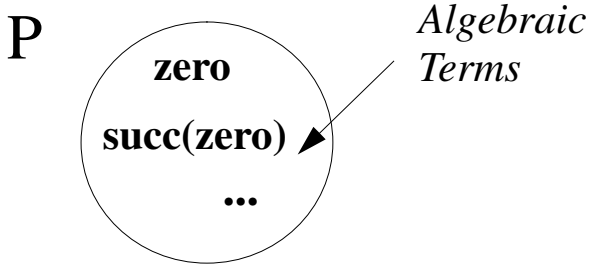


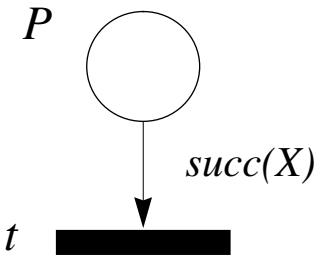
Algebraic Nets:
syntactical vs semantical approach

semantical approach



syntactical approach

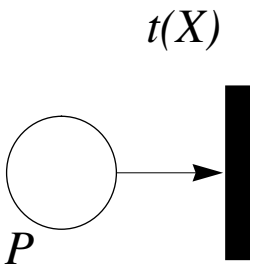




Results with the *semantical approach*

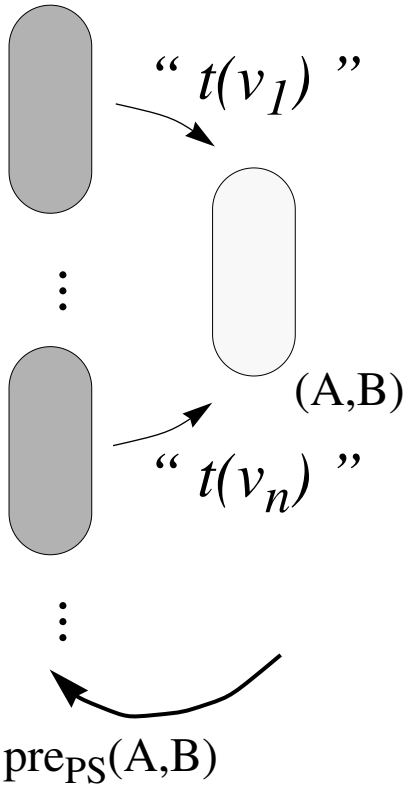
- Predicate structure similar (i.e. usual Petri net)
- Markings with values in an algebra
- All the operations have been defined

Problem of the parametrized transitions



Domain of interpretation of X:

$\{v_1, ..., v_n, ...\}$



Result of the *syntactical approach*

- Cope with the difficulties of the variables
- Markings with terms
- Keep the ‘abstraction’ of the variables

Predicat structure

- One part similar: (A,B)
- Constraints on the variables in (A,B)
- Generic predicate structure

Problems

- ☒ transformer of the predicate structure *pre* (inference rules)
- ☒ union, intersection
- ☐ negation
- ☐ quantification of the variables

© 1995 P. Racloz

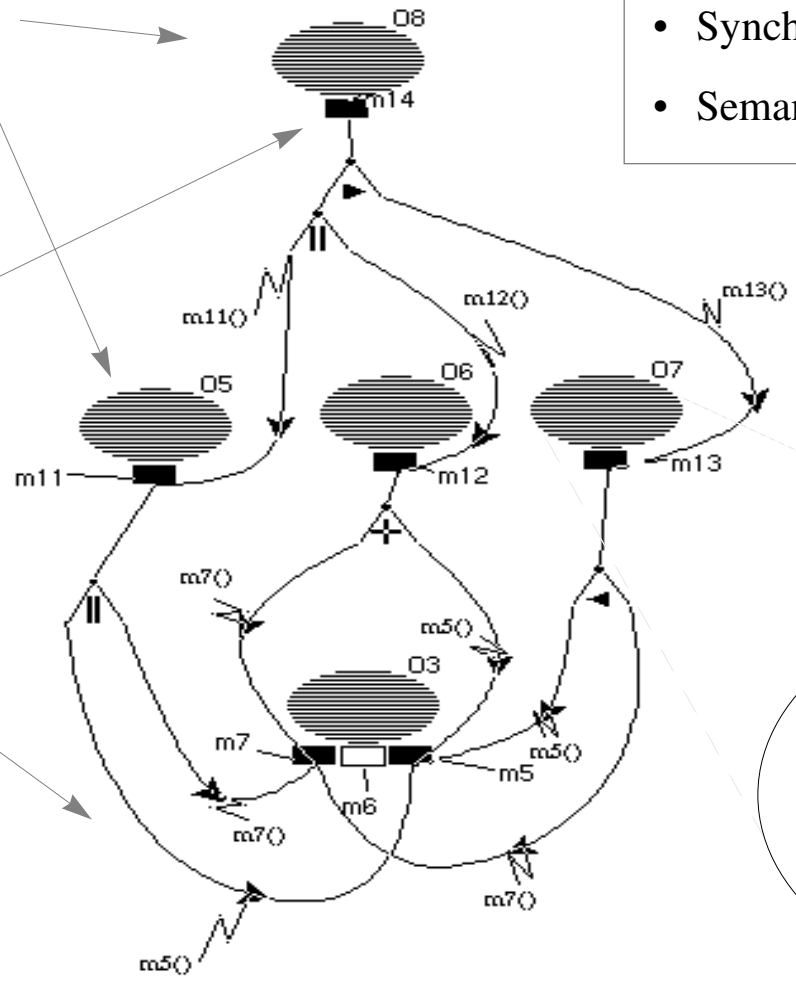
Concurrent Object Oriented Petri Net (CO-OPN)

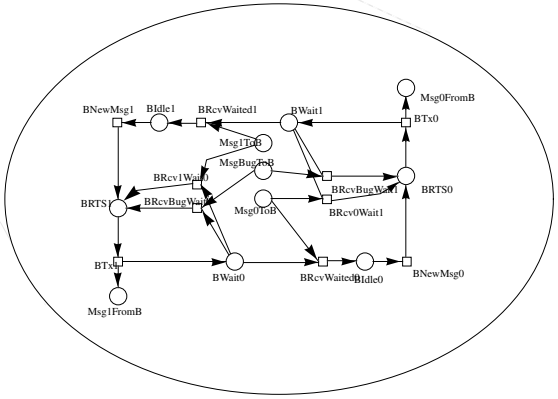
- Systems of objects
  (algebraic nets + modularity)
- Synchronization mechanisms
- Semantic

Algebraic Nets

External methods

Synchronizations between objects





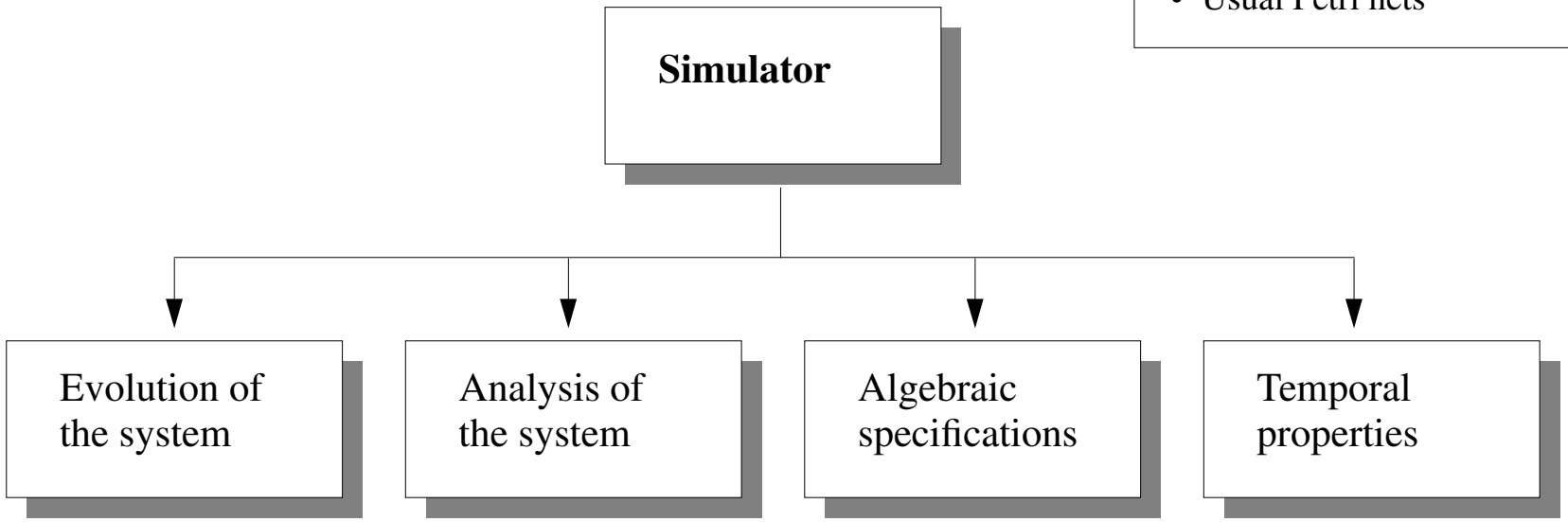
The environment:

SANDS
(Structured Algebraic Nets Development System)

- Compiler (generation of Prolog), Execution monitor
- Graphical editor
- Simulator
- Temporal properties
- Transformations
- Parallel implementation

© 1995 P. Racloz

- Algebraic nets
  - modular
  - simple
- Usual Petri nets



- Step by step
- Complex synchronization expressions

- Markings
- Reachability graph

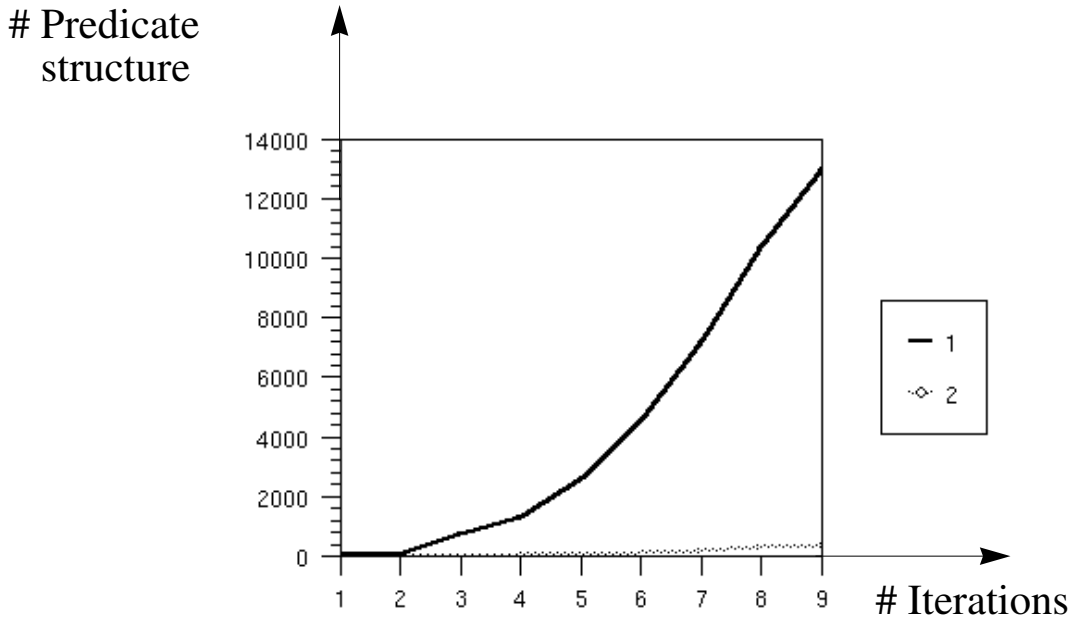
- Terms' evaluation
- Equality between terms

- Predicate structure
- Temporal operators

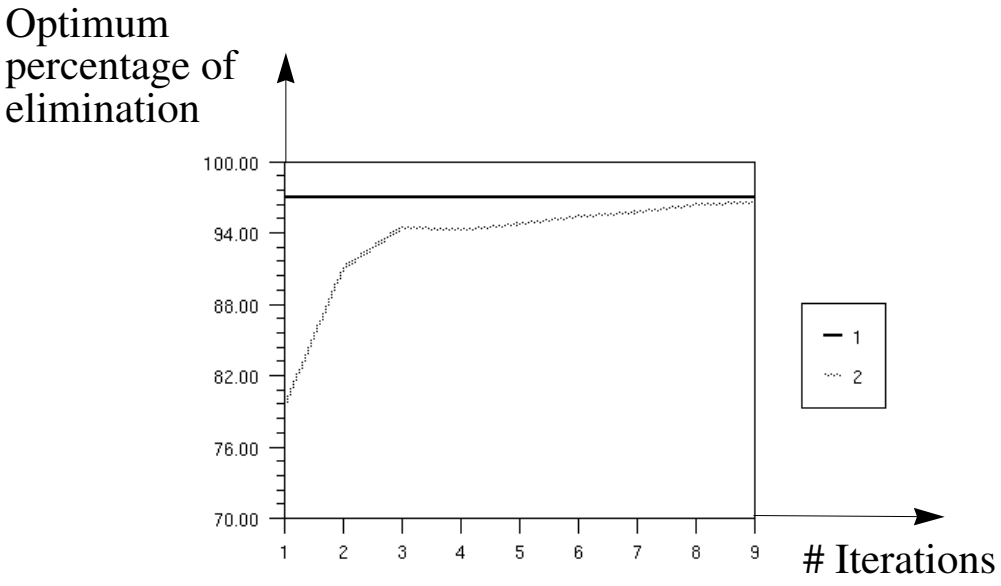
Experiment: alternating bit protocol.

- Modelling with a Petri net:

Elimination and simplification processes



Estimation of the convergence's speed



Conclusion

Contribution

- Symbolic representation
  - Simple Petri nets
  - Algebraic nets (semantic level)

Points to improve

- Study of the possible extensions to algebraic nets (syntactic level)
- Temporal tool
  - Efficient implementation
  - New functionalities

In the future

- Large scale experiments
- Study of the impact of the use of modular modellings

© 1995 P. Racloz