Testing of software and of communication systems

Richard Castanet,

LaBRI

Bordeaux

Overview

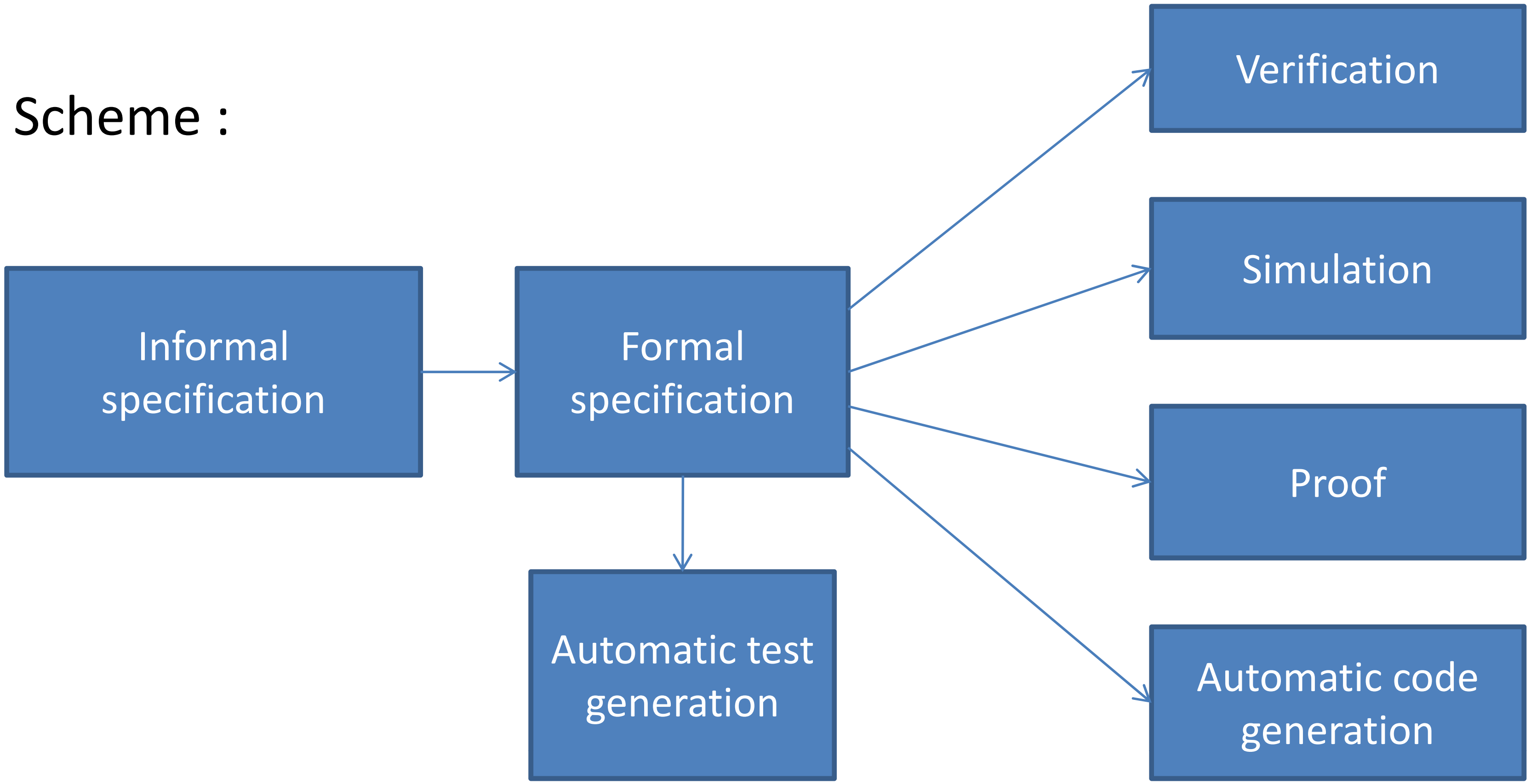
- Test positioning, definitions and norms
- Automatization and formal methods
- Test of reactive systems
- Optimizations (test on the fly, symbolic testing)
- Test of critical systems with temporal constraints
- Robustness testing
- Perspectives

Definition et positioning

- Test definitions : dynamic validation method , non exhaustive
- Need of the test : cost of systems and human life
- Live cycle : Position of the test : test of an implementation
- Types of test
  - Conformance testing, interoperability testing, robustness testing, performance testing
  - Test of unities, integration testing, test of the global system, test of acceptance
- Structural testing (white box) and functional testing (black box)
- Norms : ISO9646, DO178B, CEI 60880, CENELEC EN 50128 …
- Need of automatization: test conception and test campaign
- Main questions about test : selection, coverage, testability, controllability

Use of formal methods

- Formal description language (well defined semantic)
- Scheme :

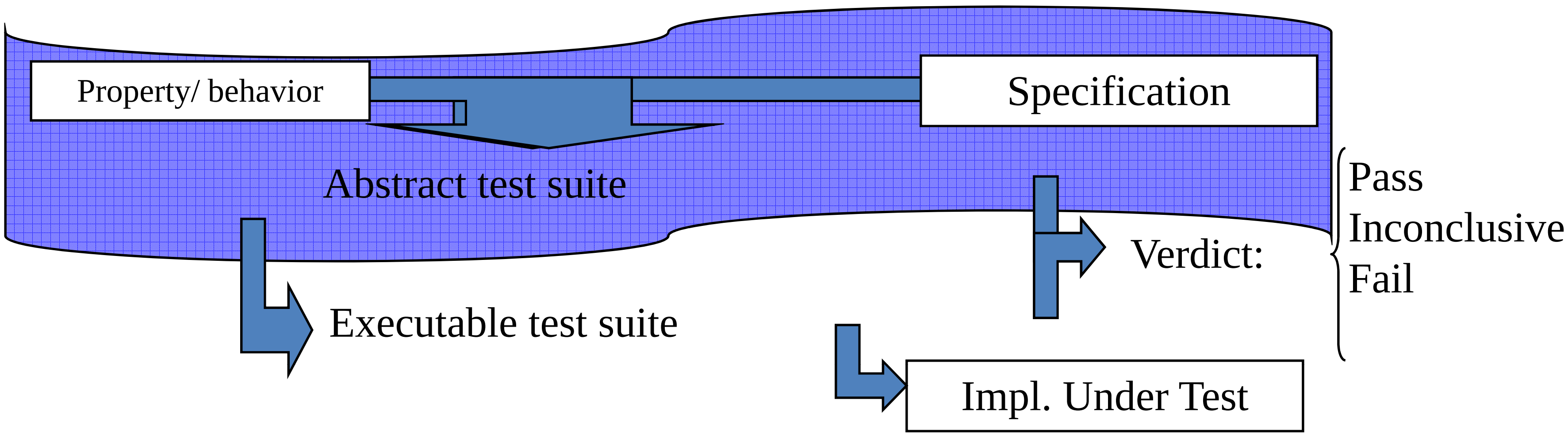


Test of reactive systems

- Formal languages: SDL, LOTOS, LUSTRE, ESTEREL …
- Based model: transition systems: Labeled Transition Systems (LTS), IOLTS (semantic for non deterministic reactive systems) (Tretmans, Jéron), automata …
- Formal approach for the test: conformance relation : conf, ioco (traces and suspension inclusion)

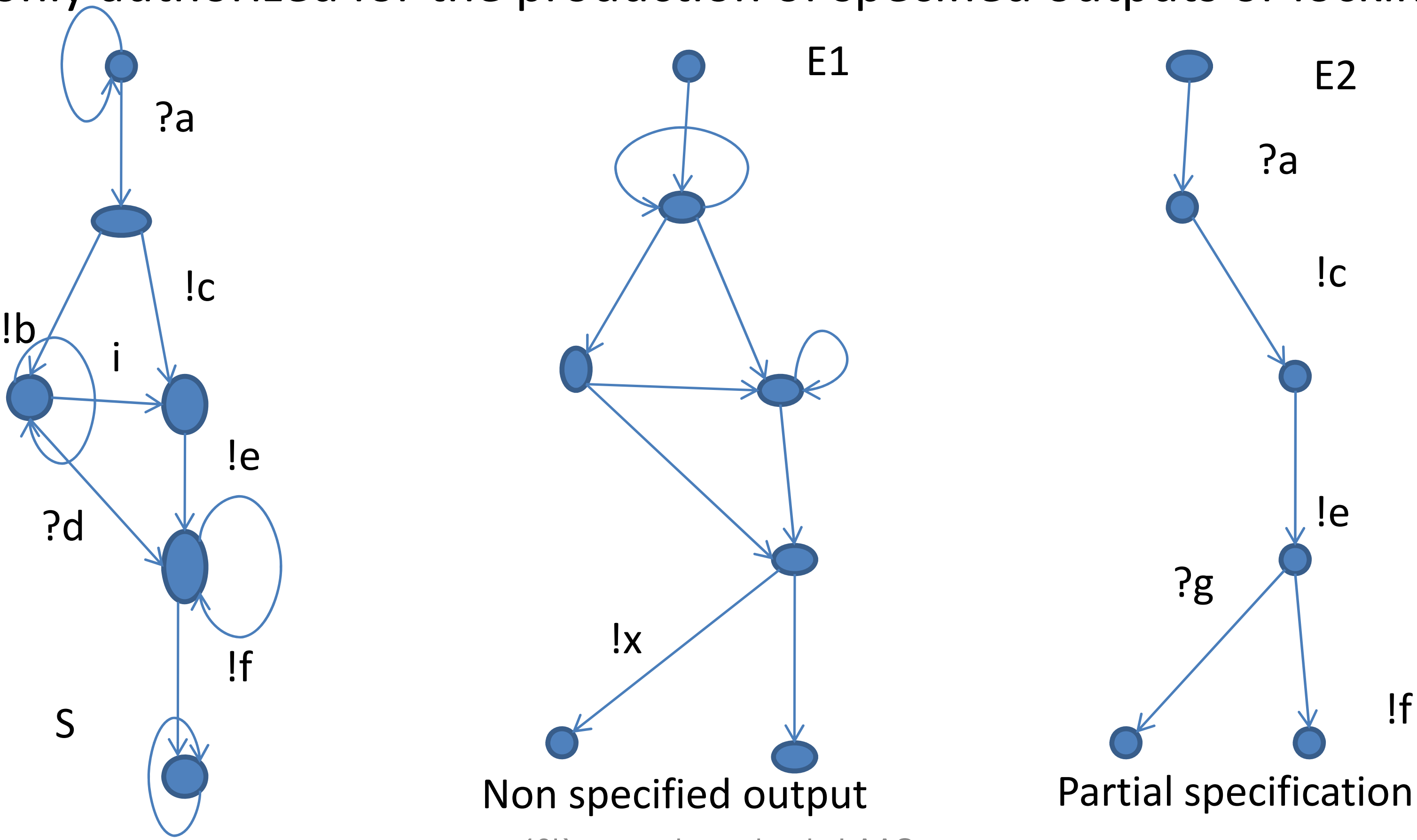
Automatic test sequence generation

- Types of test:
  Conformance/Interoperability/Performance/Robustness/
- Conformance testing :
  Implantation conform to a specification
- Interoperability:
  Capacity several communicating system to interoperate
- Test Process:



Conformance relation : example

- After a visible behavior of the specification, the implementation is only authorized for the production of specified outputs or locking

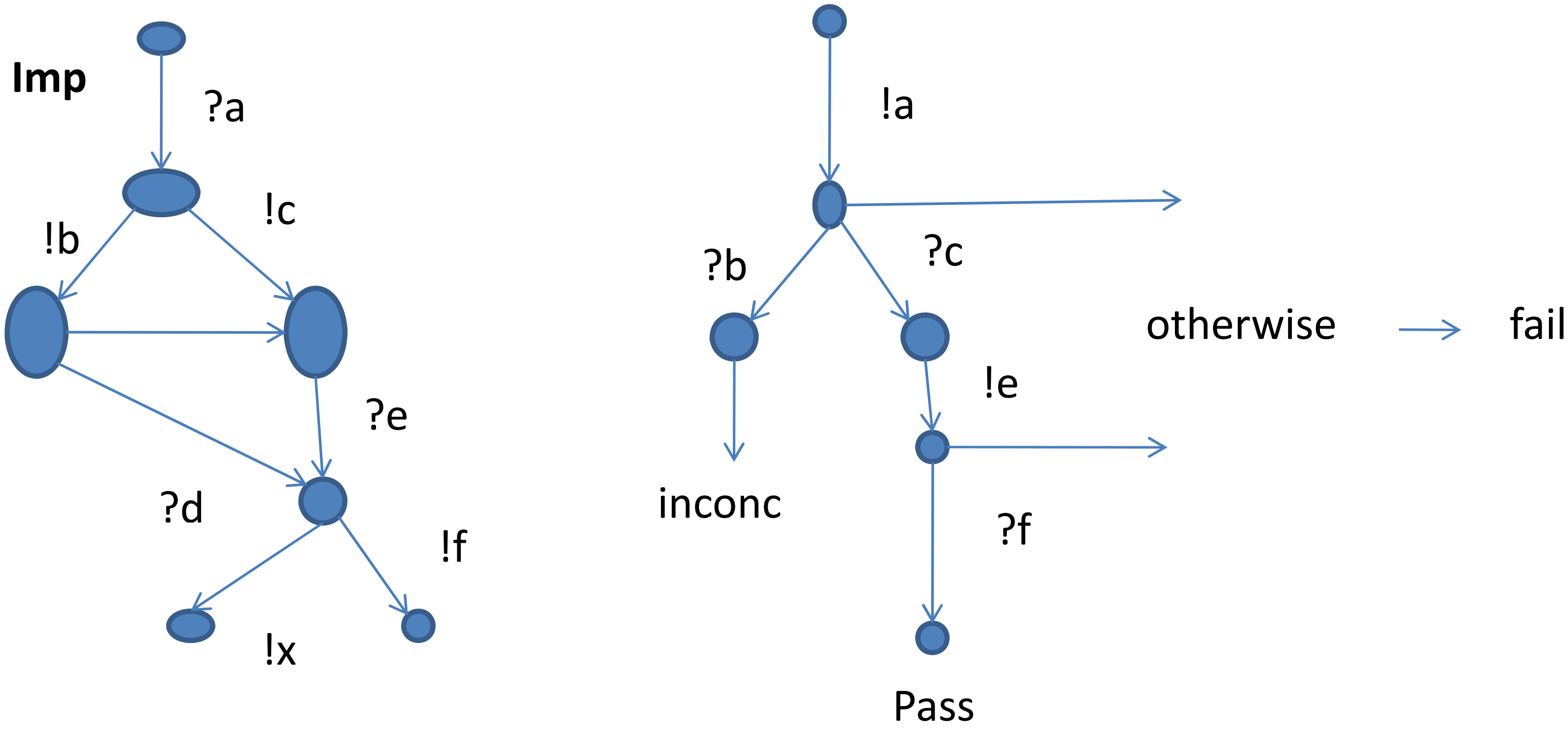


Non specified output

Partial specification

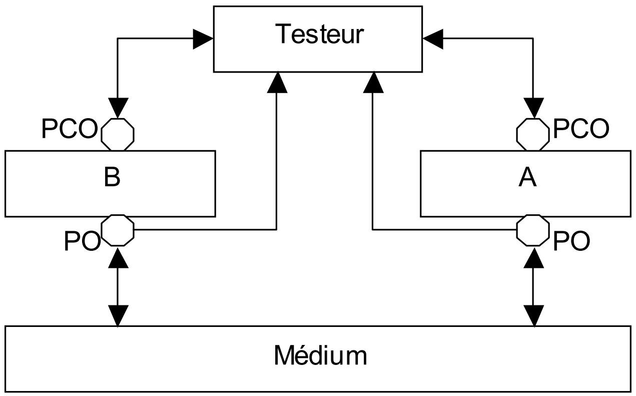
Test case and tester

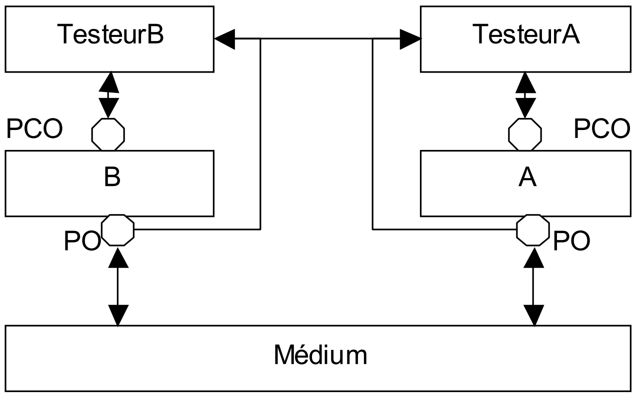
- Principles of a tester for communication with the implementation: inversion of inputs - outputs



Test Architectures : interoperability testing

- **Observation Point (OP) and Control (COP)**
- **2 types of test : black box/ grey box**
- **Architectures for interoperability testing:**





Automatic Test Generation

- Methods based on automata
  - Paths in graphs
  - Eulerian circuits
  - Fault model, mutant method
  - TT, W, UIO methods
  - Optimization by chinese postman algorithm
  - Executability problem

Generation Methods based on verification and simulation

- Construction of the reachability graph (all behaviors)
- Test sequence : path in the reachability graph

  → combinatorial explosion

  → enumerative methods, interlacing ….

  → need of reduction methods (test number, test sequences length, …)

Test generation on the fly

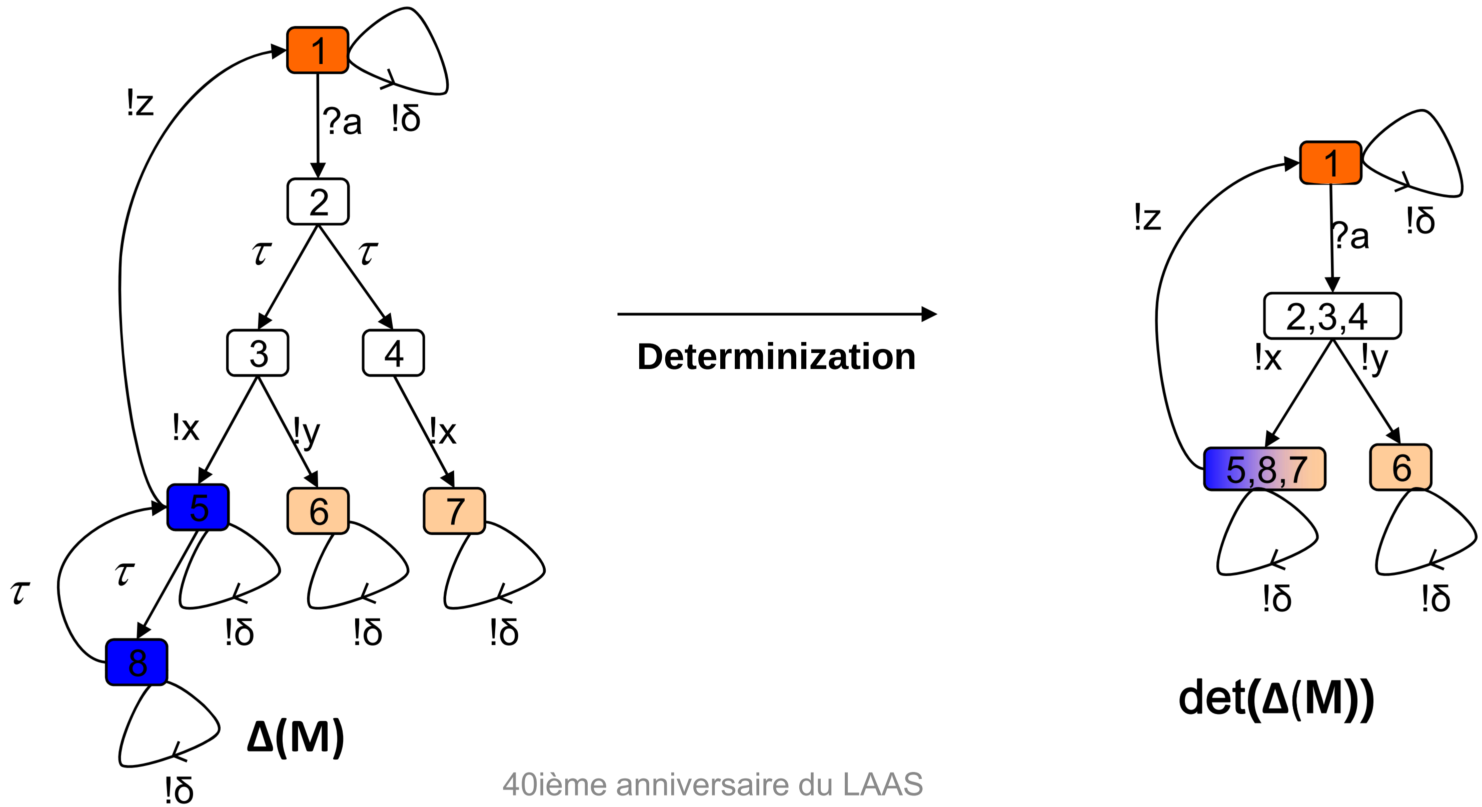
- A method to reduce the combinatorial explosion
- Synchronous product between a test purpose and the specification
- Notion of observer
- Production of a part of all behaviors
- Optimizations with determinization of the graph

Suspension automaton

The absence of visible behaviors is modeled by an output event !δ

Δ(M) = M + loops of !δ on each quiescent states

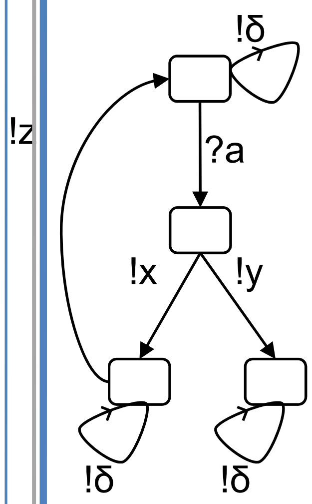
- Suspended traces of M: STraces(M) = Traces(Δ(M))
- det(Δ(M)) characterizes the visible behaviors of M.



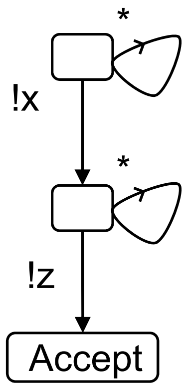
An example of a specific tester related to a test purpose

- Specification: S
- Conformance Relation: *ioco*
- *Property:* Test purpose TP

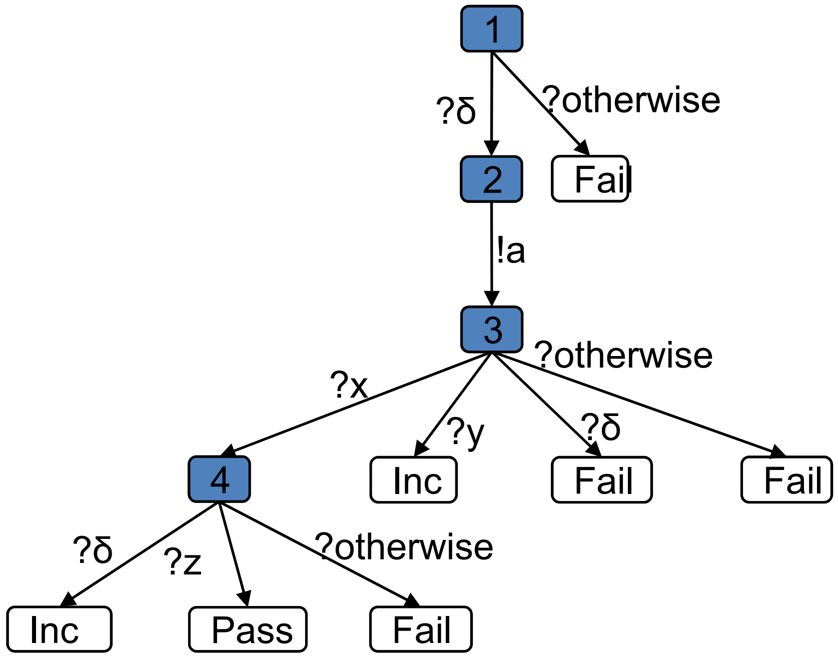
Reachability of Det(Δ(S)): suspended trace of S accepted by TP



Det(Δ(S))



TP

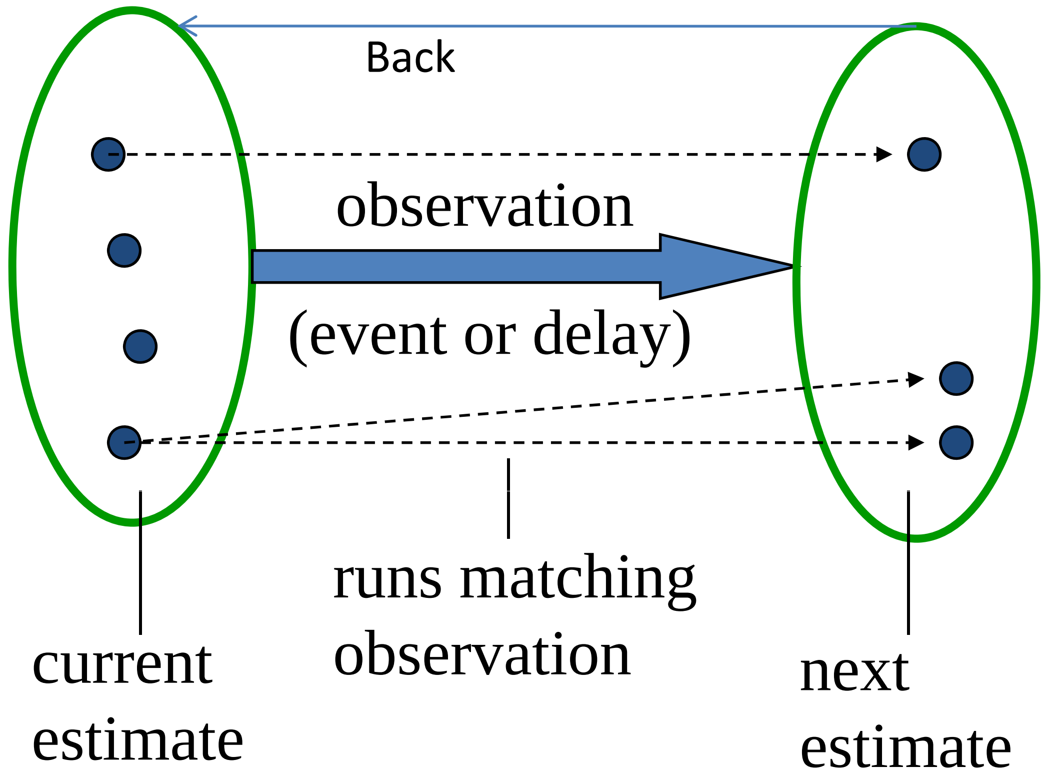


TC

Symbolic testing

- Method used with test purpose
- Each state is linked a descriptor giving all the constraints on the variables
- From a state to the next one, the constraints are modified by the actions of the state
- Use of a constraint solver

Symbolic Reachability



Test of critical temporal systems

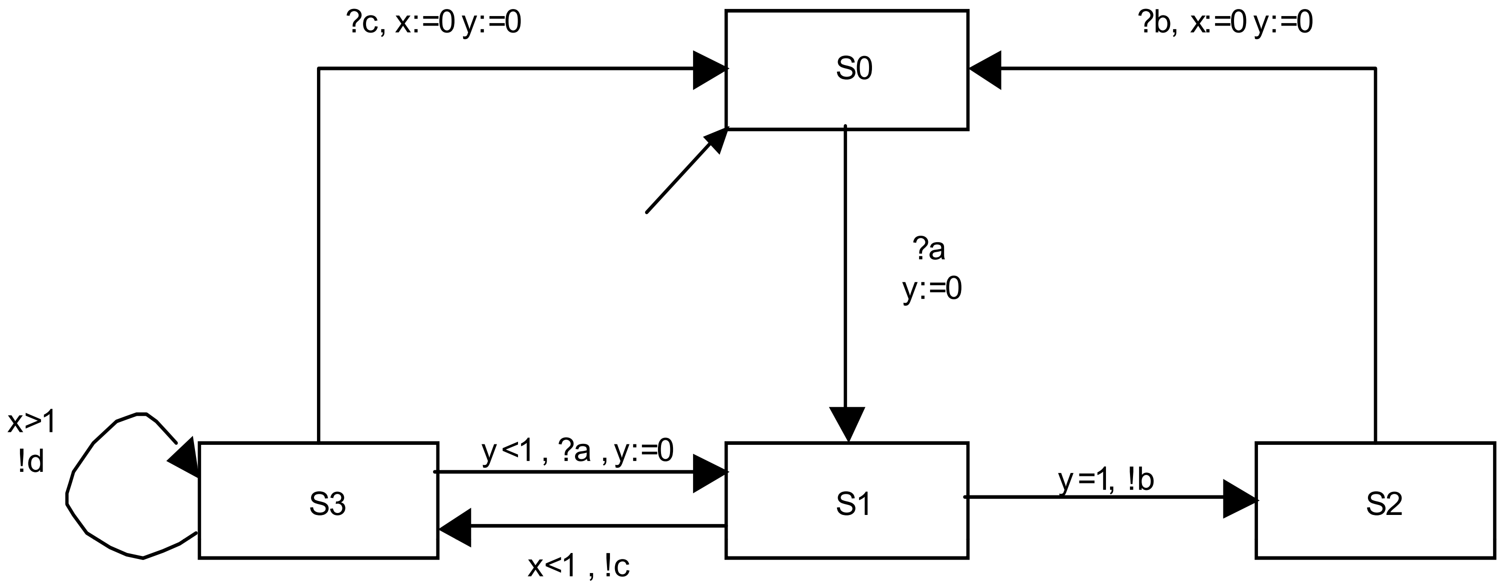
- Need of test for critical system
- Model for critical temporal system : temporized automata

Temporized automaton

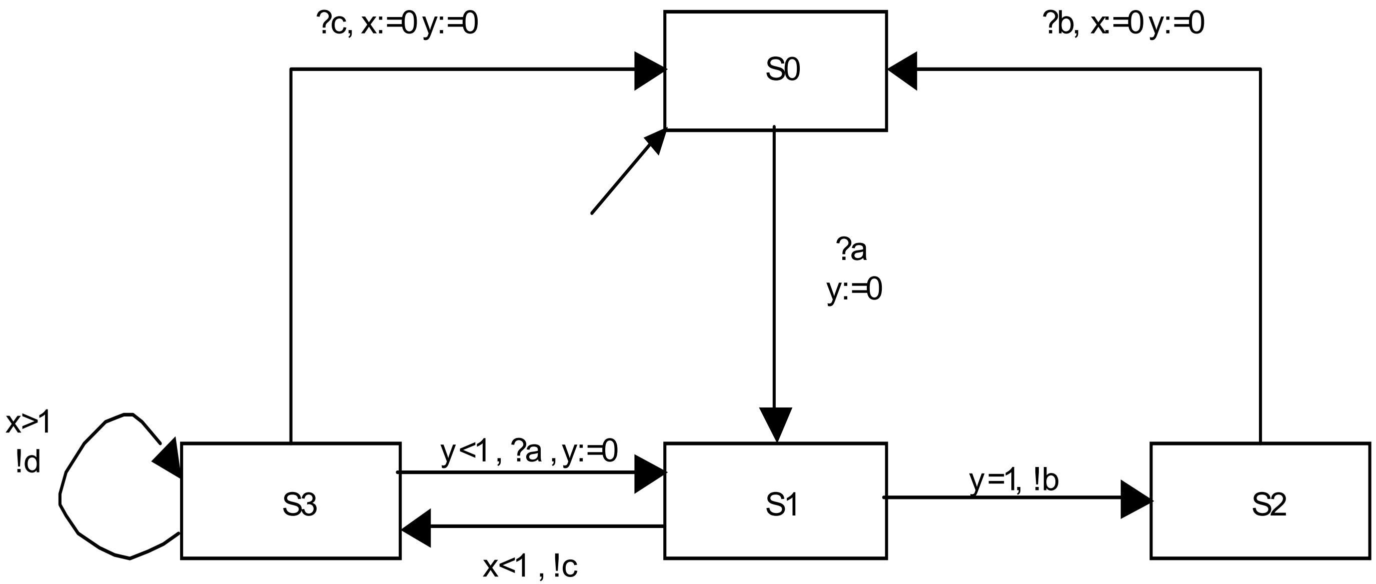
<u>Temporized automaton:</u>

(S,L, C ,S0,T) with :

- S_0 □S : initial states,
- T □ S x S x L x 2^{C} x $\Phi(C)$.



Execution of a temporized automaton



(S0, x=y=0) —? a, t=0.5→ (S1, x=0.5 y=0) —! c, t=0.75→ (S3, x=0.75 y=0.25) —? a, t=0.85→ (S1, x=0.85 y=0)

—! b, t=1.85→ (S2, x=1.85 y=1) —? b, t=6.85→ (S0, 0=x=y) —? a, t=7.85→ (S1, x=1 y=0)

—! b, t=8.85→ (S2, x=2 y=1)

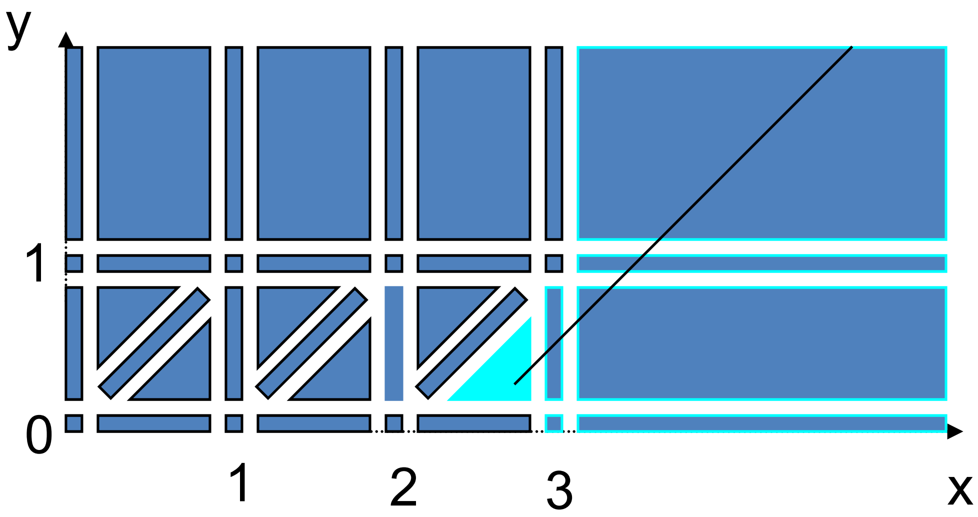
Automatic test generation

- Discrete time or continous time
- Infinite states
- Method based on the region graph (finite graph and exponential complexity)
- Test on the fly
- Timed constraints resolution

Clock regions and region graph

Let $C = \{x,y\}$ and $\Phi(C)$ a set of constraints on C with $C_x=3$ and $C_y=1$.

An example for a region graph:



r1 [(2<x<3), (0<y<x-1)]

r2 [(x=3), (0<y<1)]

r3 [(x>3), (0<y<1)]

r4 [(x>3), (y=1)]

r5 [(x>3), (y>1)]

r6 [2<x<3, y=0]

Temporal successor of a region:

The temporal successors of r1: r2, r6, r7.

Region automaton

Let *(S,L, C ,S0,T)* a temporized automaton)

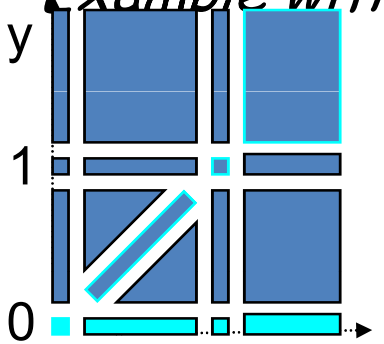
State representation : <state, region>.

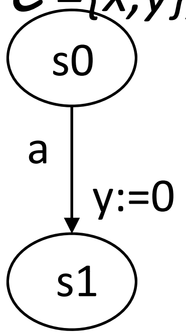
Transition representation : *<<e1,r1>, a ,<e2,r2>>* with:

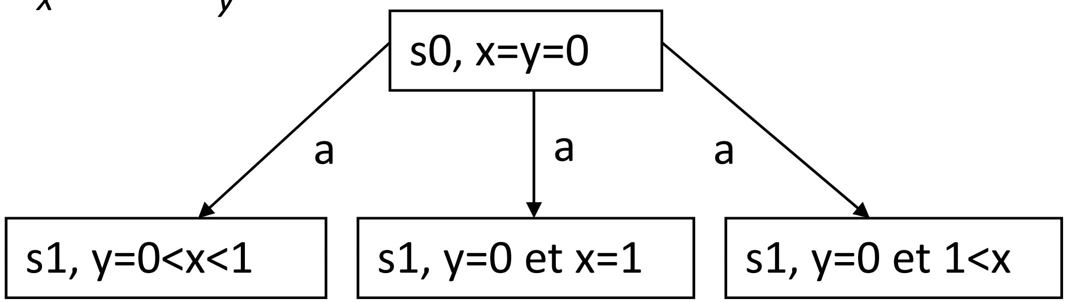
- *<e1, a, c, r, e2>* $\in$ *T*,
- *r2* is a temporal successor of *r1*,
- *r2* satisfies *c*.

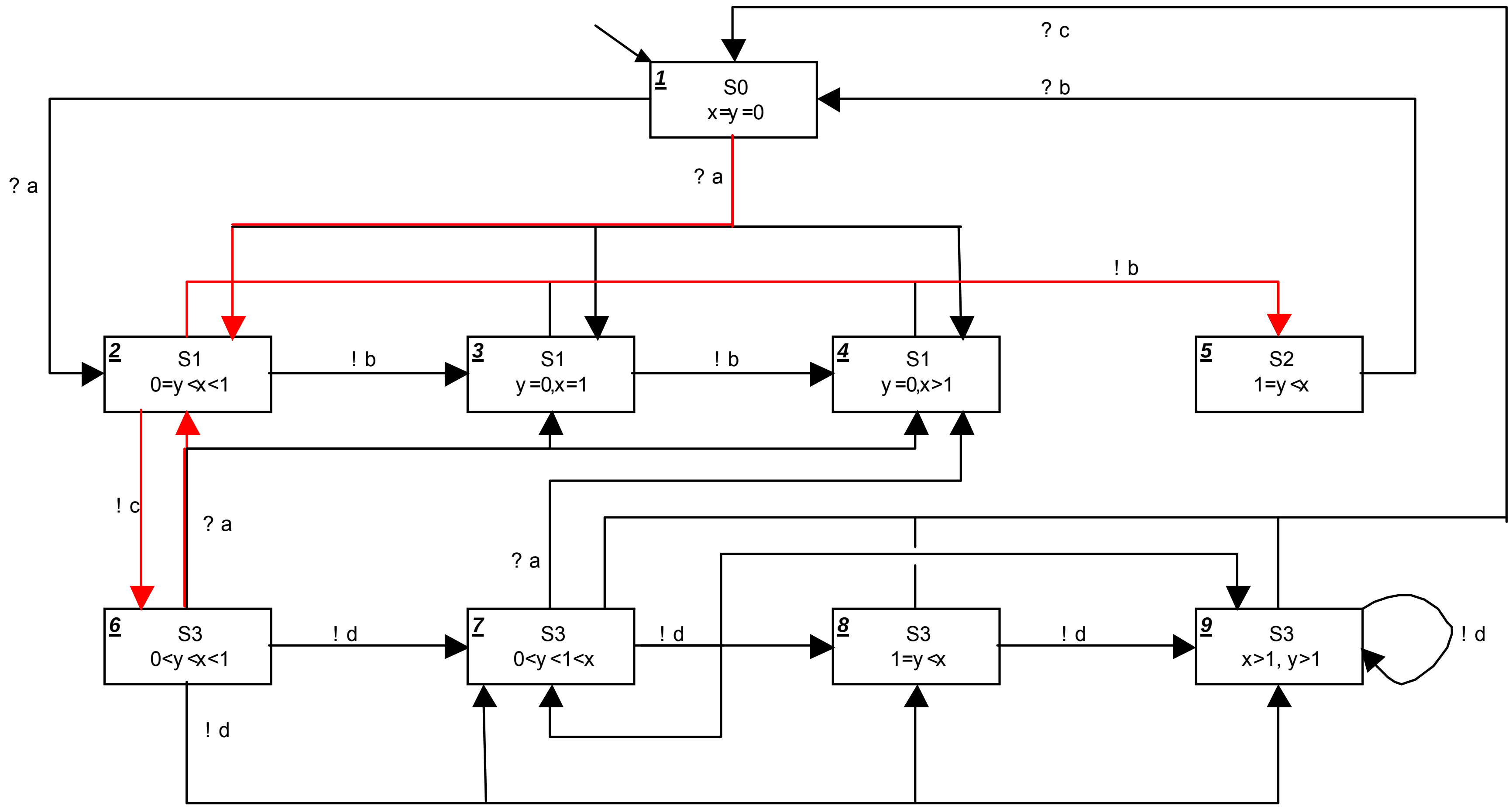
Ex

Example with C={x,y}, C_x=1 et C_y=1.





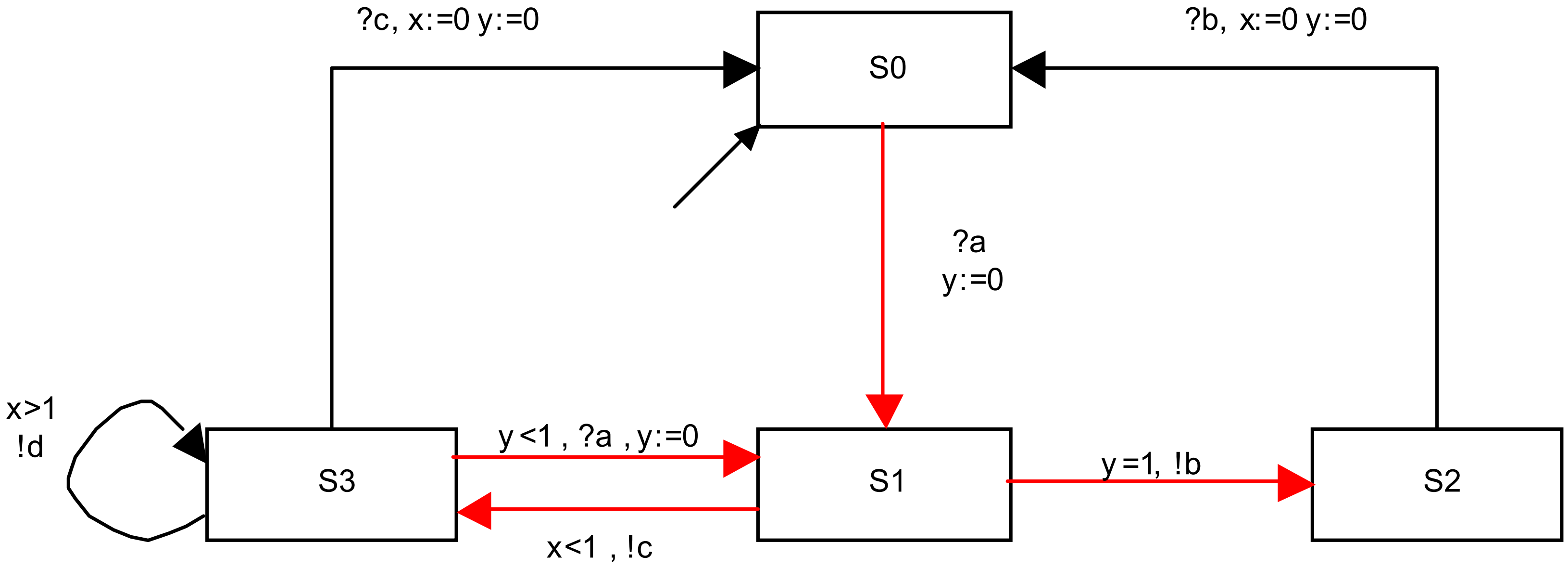




(S0, x=y=0) —? a, δ=0.5→ (S1, 0=y<x<1) —! c, δ=0.25→ (S3, 0<y<x<1) —? a, δ=0.1→ (S1, 0=y<x<1)

—! b, δ=1→ (S2, 1=y<x) —? b, δ=5→ (S0, 0=x=y) —? a, δ=1→ (S1, x=1 0=y)

—! b, δ=1→ (S2, 1=y<x)



(S0, x=y=0) —? a, t=0.5→ (S1, x=0.5 y=0) —! c, t=0.75→ (S3, x=0.75 y=0.25) —? a, t=0.85→ (S1, x=0.85 y=0)

—! b, t=1.85→ (S2, x=1.85 y=1) —? b, t=6.85→ (S0, 0=x=y) —? a, t=7.85→ (S1, x=1 y=0)

—! b, t=8.85→ (S2, x=2 y=1)

Test on the fly

Approaches with test purpose

- Main idea for test with test purpose
  - Let :
    - S specification of the system to be tested,
    - O test purpose
  - Problem:
    - Find an execution of S drived by the test purpose.
- Model : temporized automaton
- Several approaches (conformance testing)
  - Region graph
  - Test purpose
  - Proof assistant

Region graph approach

- Extraction of a test sequence
  - Producing a trace drived by a test purpose on the region automaton
    1. Sequence of transitions of the region automaton (non temporized)
    2. Choice of δ : fire instants
- disadvantage: combinatory explosion
  - Abstraction of the specification (temporized automaton)
  - Equivalence of states
  - Minimization of the region graph (partition of the state space)

Method with test purpose

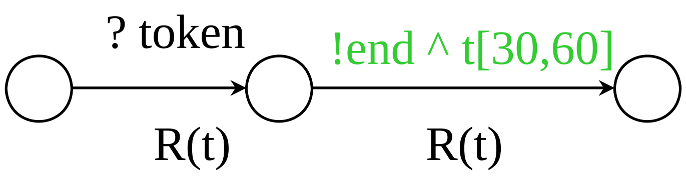
Definition: A Test purpose is an deterministic atomaton without cycle automaton and with an non empty set of special states : Accept(TP).

Goal: Find a sequence of transitions of the specfication according to the test purpose.

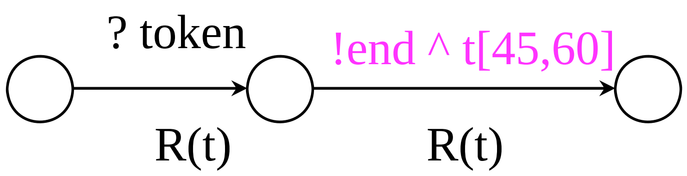
Verdicts

- (Pass) : the event satisfies the Spec and the TP
- Pass : The event satisfies the Spec and the TP and TP is in an acceptance state
- Fail : the event does not verify the Spec
- Inc : the event satisfies the Spec, but not the TP.

Example of a coffee machine:



Specification (Spec)



Test Purpose(TP)

Synchronized product

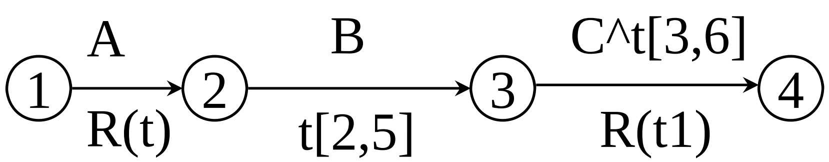
Synchronized product on the events:

Sync : product automaton :

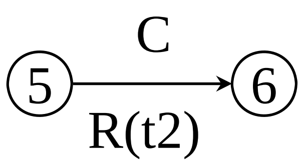
<u>Rule 1</u>:

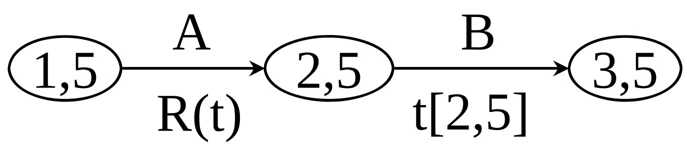
$$\frac{(s1,s3) \in S(Sync) \wedge \\ (s1,\mu,Ct1,Cv1,\rho1,\beta1,s2) \in T(spec) \wedge (s3,\mu,Ct2,Cv2,\rho2,\beta2,s4) \notin T(Ot)}{(s2,s3) \in S(Sync) \wedge ((s1,s3),\mu,Ct1,Cv1,\rho1,\beta1,(s2,s3)) \in T(Sync)}$$

Spec:



TP:





Product automaton

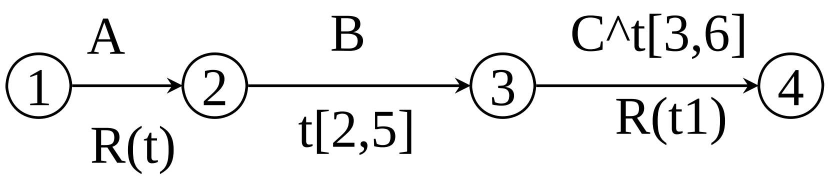
Synchronized product

Synchronized product:

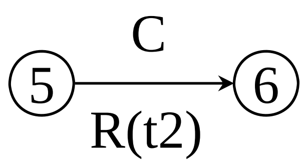
<u>Rule 2</u>:

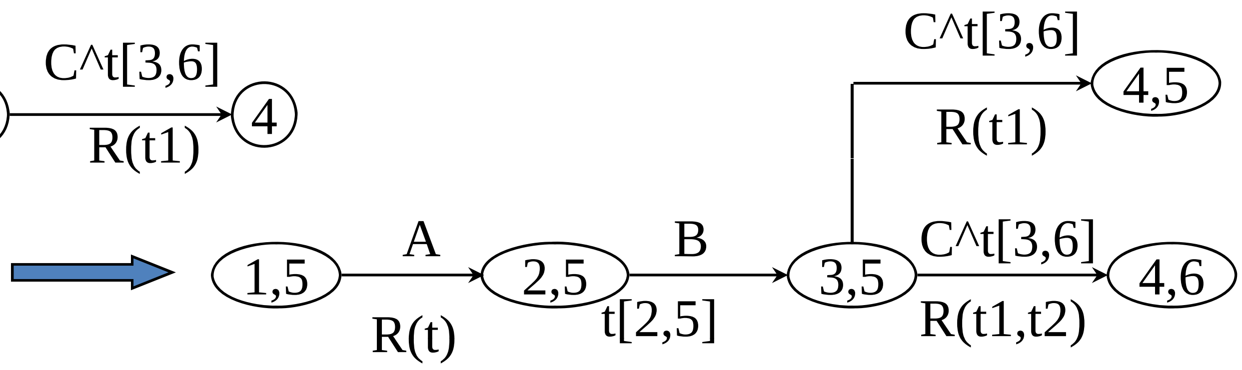
$$\frac{(s1,s3) \in S(Sync) \wedge \\ (s1,\mu,Ct1,Cv1,\rho1,\beta1,s2) \in T(spec) \wedge (s3,\mu,Ct2,Cv2,\rho2,\beta2,s4) \in T(Ot)}{(s2,s4) \in S(Sync) \wedge ((s1,s3),\mu,Ct1 \cup Ct2,Cv1 \cup Cv2,\rho1 \cup \rho2,\beta1 \cup \beta2,(s2,s4)) \in T(Sync) \\ \wedge (s2,s3) \in S(Sync) \wedge ((s1,s3),\mu1,Ct1,Cv1,\rho1,\beta1,(s2,s3)) \in T(Sync)}$$

Spec:



TP:





Automate Produit

Synchronized product

Temporal Synchronisation :

(method similar than the computtion of the state classes of the temporized Petri nets)

Main idea: inequations system keeping the temporal relationship between the different clocks.

a transition is aware once all the clocks in his temporal constraint have been reset.

Synchronized product

When a transition is fired, the system of inequalities is updated in 3 stages:

1) Calculating the time remaining to make transitions sensitized

2) Remove unnecessary relations.

3) Taking into account the new transitions sensitized by resetting clocks.

Robustness testing

Robustness definitions :

IEEE : degree from one system to function properly in the presence of invalid entries or stressful environment

⟹ ability to exhibit acceptable behavior in the presence of hazards

hazard

- Fault
  - extern/intern
  - Accidental / intentional

Change use profile and charge

correct or acceptable

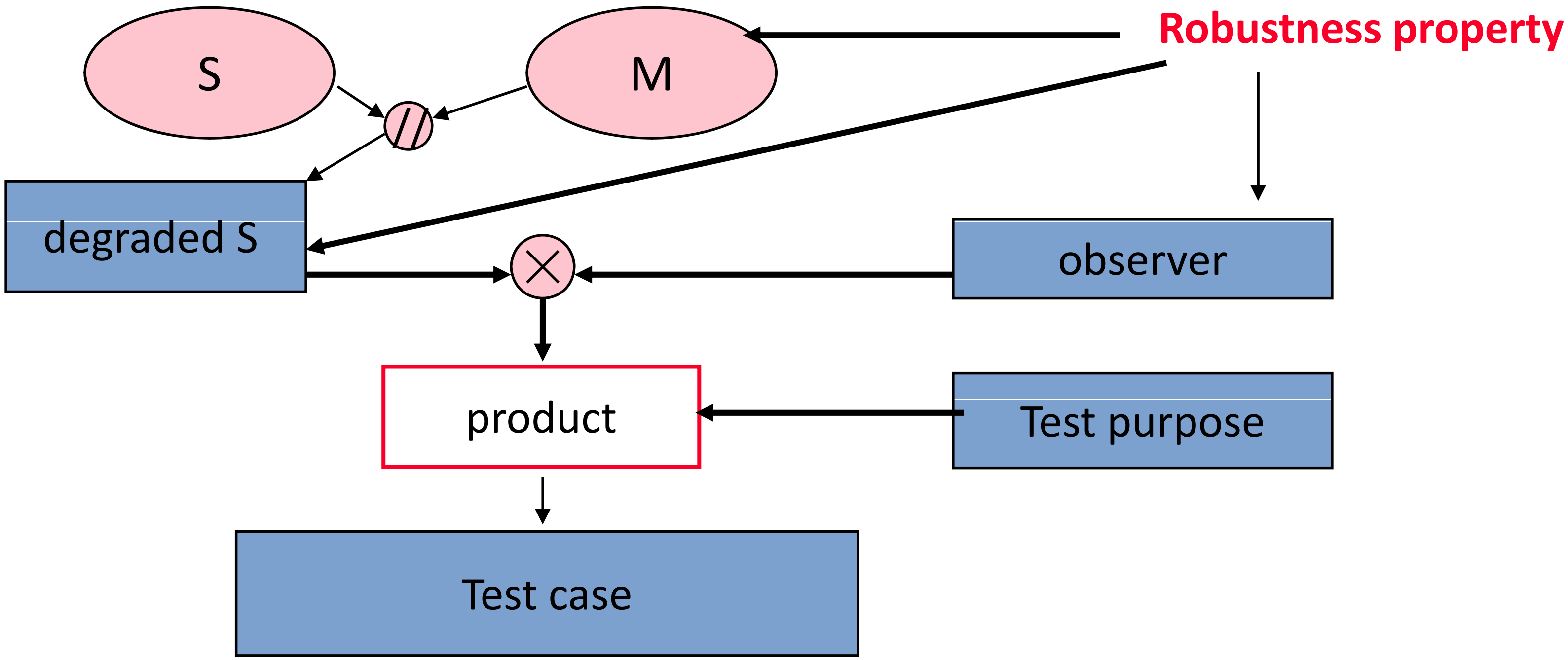
- robustness stronger than conformity or orthogonal?
- sometimes, no specification of expected behavior over hazard

Classification of the hazards

- Internal, external, out of the system
- Representable or no representable

Methods based on behavior models

P : S specification, M fault model (set of potential faults and unanticipated events planned), P robustness property : an implantation I is robust iff I met P even in the presence of faults

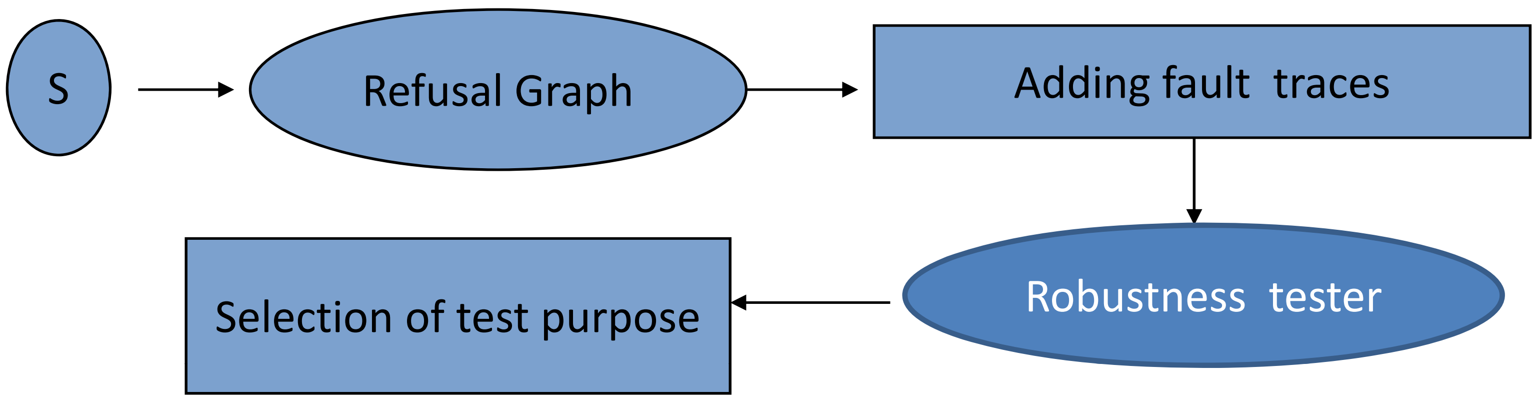


Approach « increasing the specification »and refusal graph

- specification S : LTS, extension with unknown or incorrect event (increase the specification)
- Construction of the refusal graph (set of refuse in each state)
- Adding fault traces (sequence of actions alternating with set of refusal)
- Construction of the robustness tester: refusal graph + fault traces,

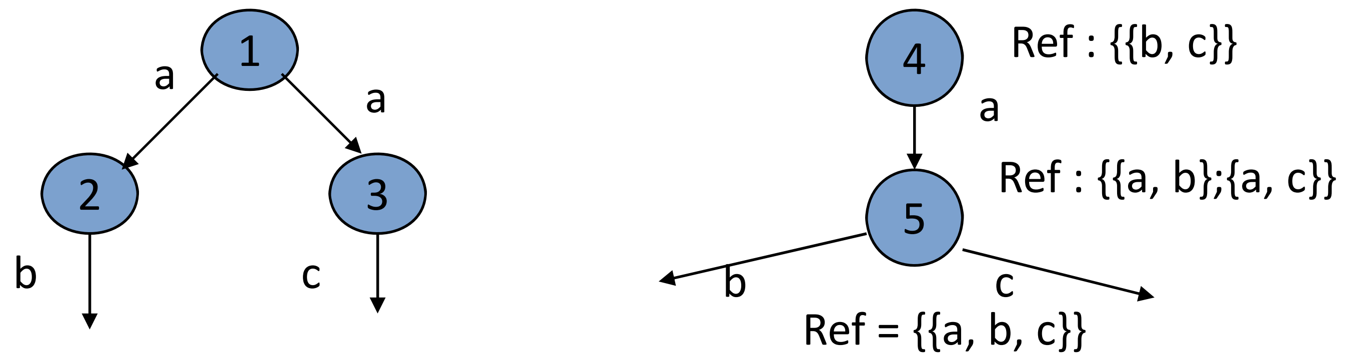
Adding of unobservable action to go back to the initial state

-Test selection and coverage computation

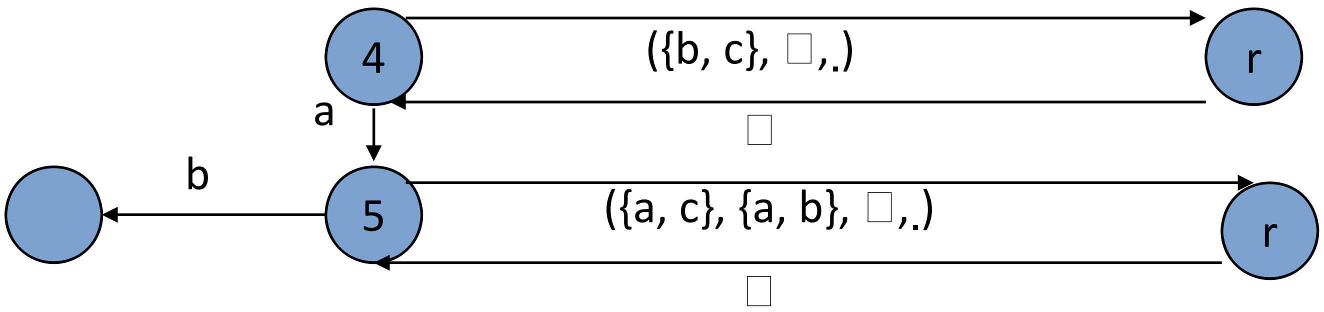


Example

Construction of the refusal graph



Fault traces: alternating sequences of actions and set of refusal



Construction of the robustness tester

Approach based on a model on the entries

- Operational profile
- Equivalence classes
- …

Model for the nominal behavior +
model of the hazards

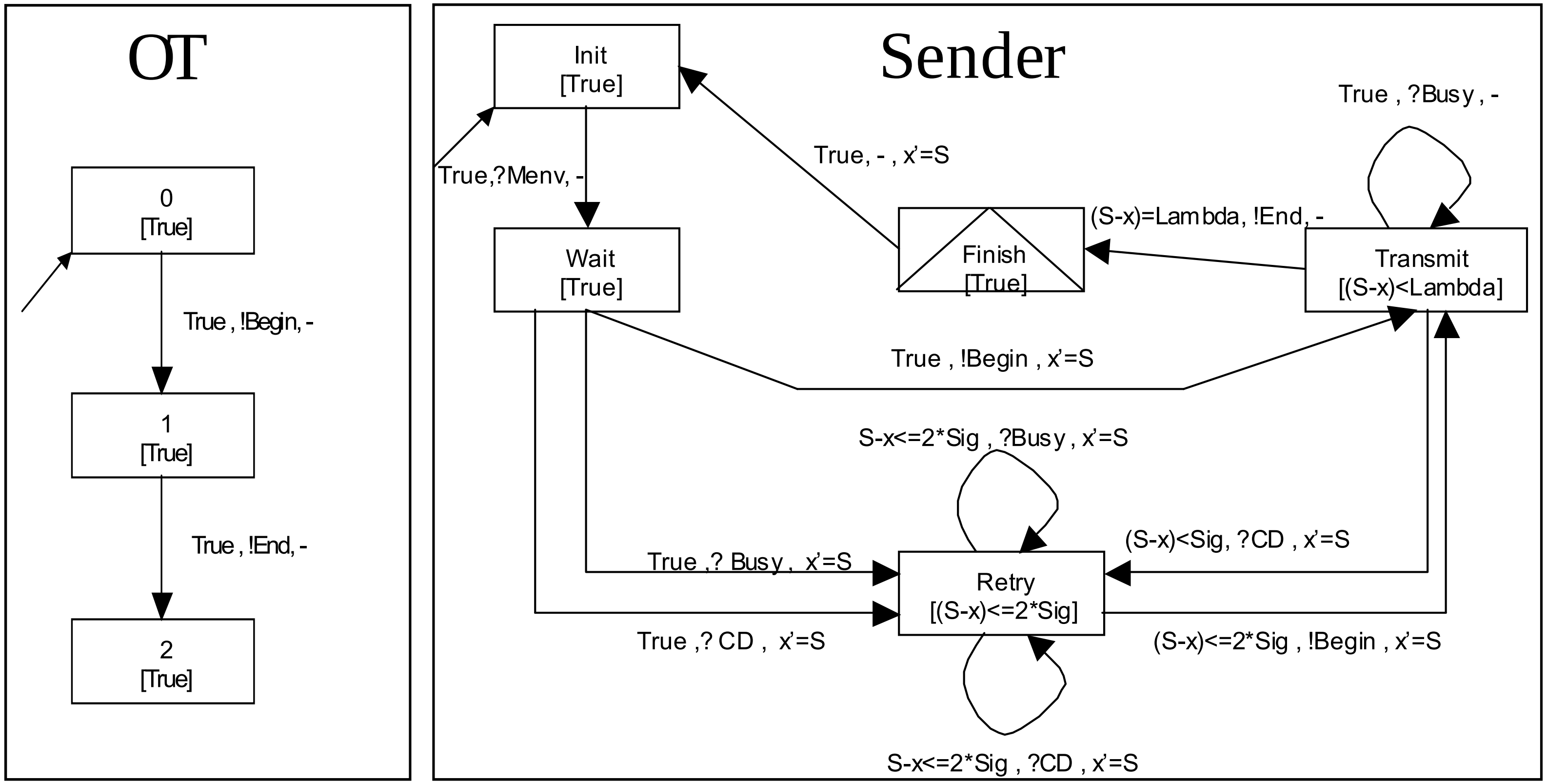
Overlay nominal activity nominal x hazards ?

- Problem of insignificant experiences
  Ex: injection of a fault that will not be activated in the system
  analysis : Online system activity, gray box analysis
- Selecting relevant case in an objective verification (≠ evaluation)
  Ex: heuristic optimization to guide the research of test the most "dangerous" case (≠ the most representative)

Perspectives

- Improvement of the use of formal methods in industries
- Treatment of real, complex systems
- Best selection of the tests
- Best coverage of a test suite
- Combining several methods : verification, proof, …

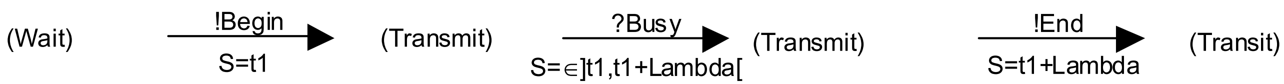
An example for conformance testing



Paramètres:

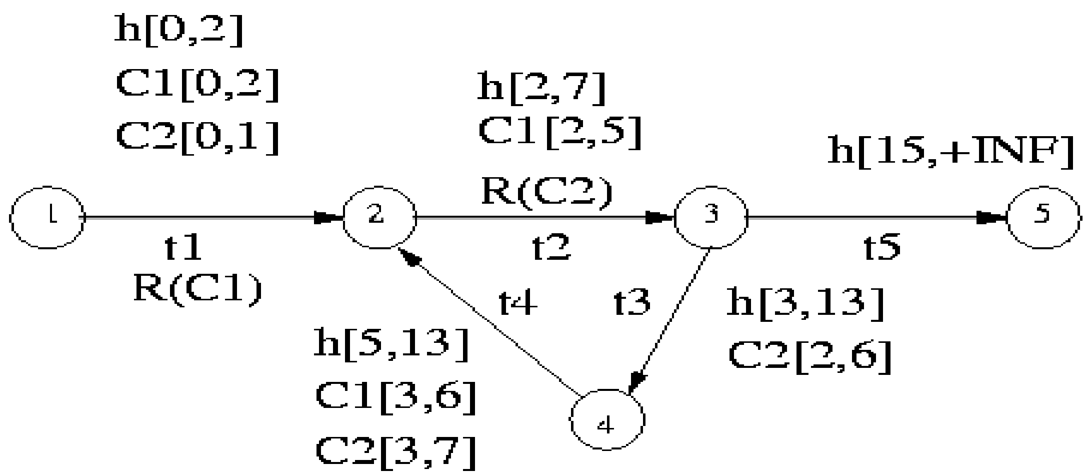
Lambda=808

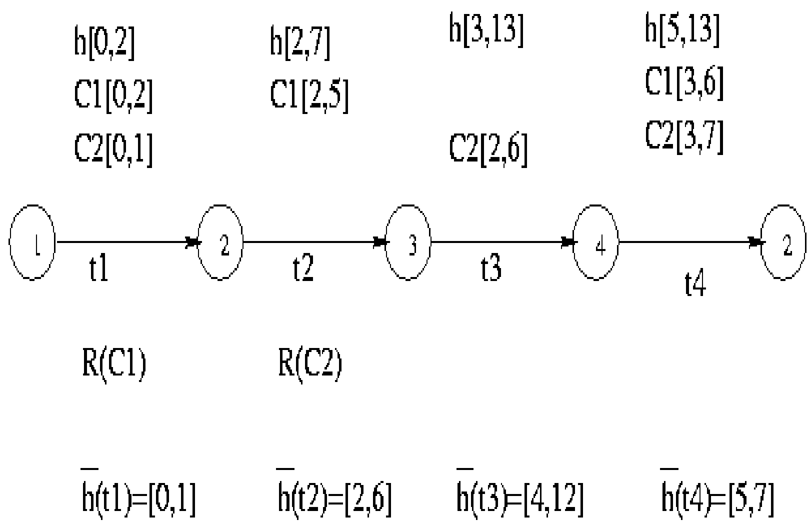
Sig=26



Calcul des intervalles de l'horloge h(2)

Exemple:





$\overline{h(t_0)}= [0,0]$

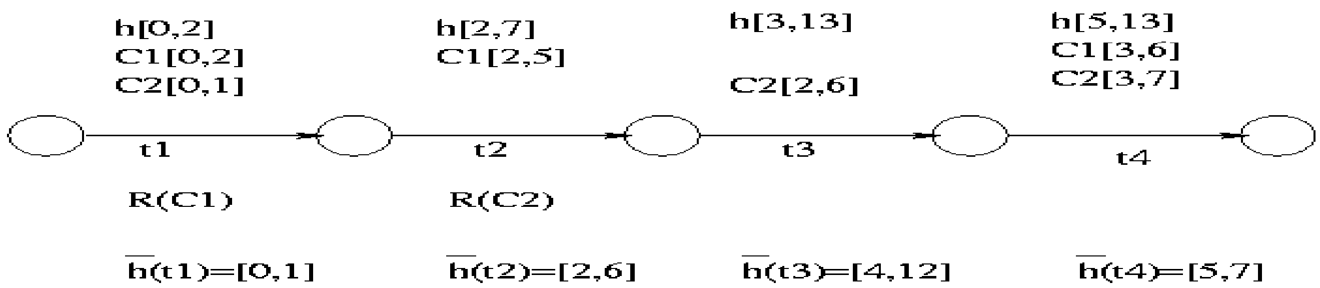
$\overline{h(t_1)}= [0,2] \cap [0,2] \cap [0,1]=[0,1]$

$\overline{h(t_2)}= [0+2,1+5] \cap [2,7]=[2,6]$

$\overline{h(t_3)} =[2+2,6+6] \cap [3,13]=[4,12]$

$\overline{h(t_4)} =[0+3,1+6] \cap [2+3,6+7] \cap [5,13] =[5,7]$

Exemple



Pour atteindre t_1

Horloge C1: $0 \leq \tau_1 \leq 2$

Horloge C2: $0 \leq \tau_1 \leq 1$

Horloge h : $0 \leq \tau_1 \leq 1$

$RC(t_1,h) = [0,1]$

$$\left\langle s1, \begin{pmatrix} h[0,1] \\ C1[0,1] \\ C2[0,1] \end{pmatrix} \right\rangle$$

Pour atteindre t_2

Transition t_2:

Horloge C1: $0 \leq \tau_1 \leq 1$

$2 \leq \tau_2 - \tau_1 \leq 5$

$2 \leq \tau_2 \leq 6$

Transition t_1

Horloge C1: $0 \leq \tau_1 \leq 2$

Horloge C2: $0 \leq \tau_1 \leq 1$

Horloge h : $0 \leq \tau_1 \leq 1$

$\tau_1 \leq \tau_2$

$RC(t_1,h) = [0,1]$

$RC(t_2,h) = [2,6]$

$$\left\langle s1, \begin{pmatrix} h[0,1] \\ C1[0,1] \\ C2[0,1] \end{pmatrix} \right\rangle$$

$$\left\langle s2, \begin{pmatrix} h[2,6] \\ C1[2,5] \\ \end{pmatrix} \right\rangle$$

Exemple

Pour atteindre t_4

Transition t_4:

Horloge h: $5 \leq \tau_4 \leq 7$

Horloge C1: $3 \leq \tau_4 - \tau_1 \leq 6$

Horloge C2: $3 \leq \tau_4 - \tau_2 \leq 7$

Transition t_3:

Horloge h: $4 \leq \tau_3 \leq 12$

Horloge C2: $2 \leq \tau_3 - \tau_2 \leq 6$

Transition t_2:

Horloge h: $2 \leq \tau_2 \leq 6$

Horloge C1: $2 \leq \tau_2 - \tau_1 \leq 5$

Transition t_1:

Horloge h : $0 \leq \tau_1 \leq 1$

Horloge C1: $0 \leq \tau_1 \leq 2$

Horloge C2: $0 \leq \tau_1 \leq 1$

$\tau_1 \leq \tau_2 \leq \tau_3 \leq \tau_4$

$RC(t_1,h) = [0,1]$

$RC(t_2,h) = [2,4]$

$RC(t_3,h) = [4,7]$

$RC(t_4,h) = [5,7]$

__Domaine de tir potentiel__

$$\left\langle s1, \begin{pmatrix} h[0,1] \\ C1[0,1] \\ C2[0,1] \end{pmatrix} \right\rangle$$

$$\left\langle s2, \begin{pmatrix} h[2,4] \\ C1[2,4] \\ \end{pmatrix} \right\rangle$$

$$\left\langle s3, \begin{pmatrix} h[4,7] \\ \\ C1[2,5] \end{pmatrix} \right\rangle$$

$$\left\langle s4, \begin{pmatrix} h[5,7] \\ C1[4,6] \\ C2[3,5] \end{pmatrix} \right\rangle$$

__Domaine de tir__

$$\left\langle s1, \begin{pmatrix} h[0,1] \\ C1[0,1] \\ C2[0,1] \end{pmatrix} \right\rangle$$

$$\left\langle s2, \begin{pmatrix} h[2,4] \\ C1[2,3] \\ \end{pmatrix} \right\rangle$$

$$\left\langle s3, \begin{pmatrix} h[4,7] \\ \\ C1[2,4] \end{pmatrix} \right\rangle$$

$$\left\langle s4, \begin{pmatrix} h[5,7] \\ C1[5,6] \\ C2[3,4] \end{pmatrix} \right\rangle$$