Validation of Real Time Applications

Françoise Simonot-Lion

Validation of Real Time Applications

Françoise Simonot - Lion

Zhejiang University
China

October 25, 2002

Contents

1 ⇨ **Validation : *When, What and How ?***

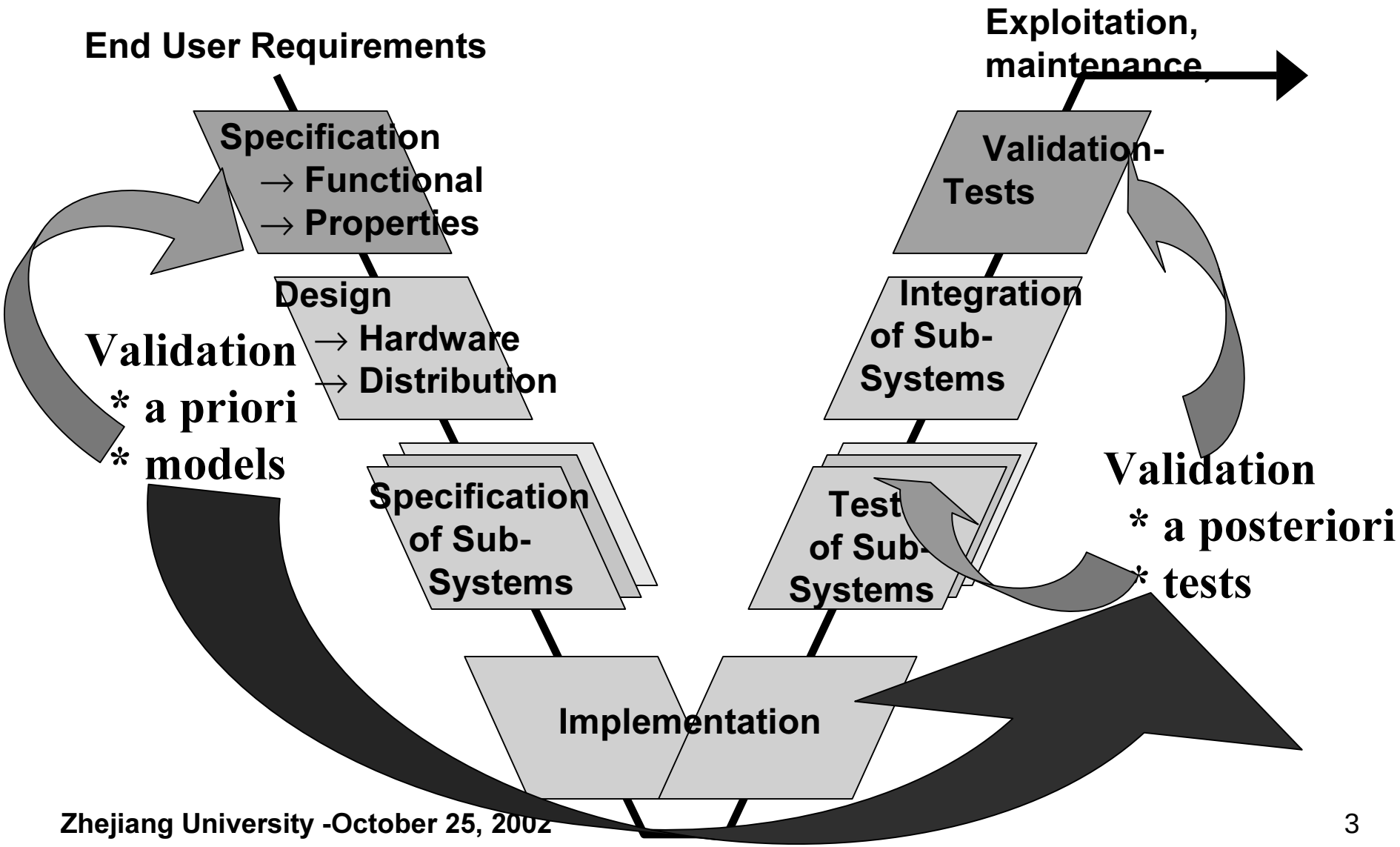
2 ⇨ **Timed Input Output State Machine (TIOSM)**
an adequate formalism

3 ⇨ **Validation a posteriori (Integration test)**
how to be efficient

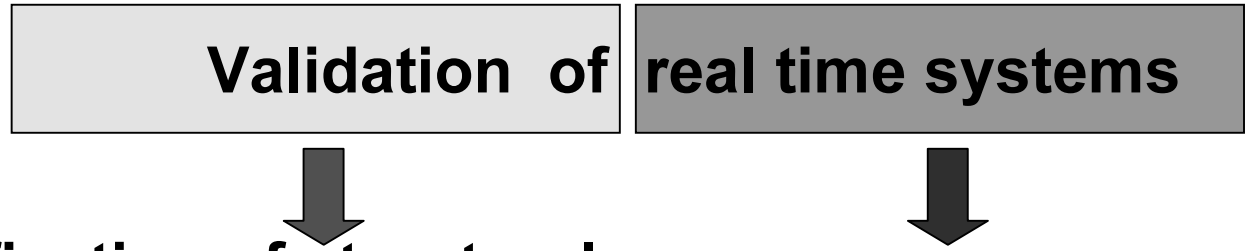
4 ⇨ **Validation a priori**
how to master the complexity

5 ⇨ **Conclusions**

Validation : when - what - how



Validation : when - what - how



Verification of structural properties

no deadlock, no livelock

Temporal characteristics

Model

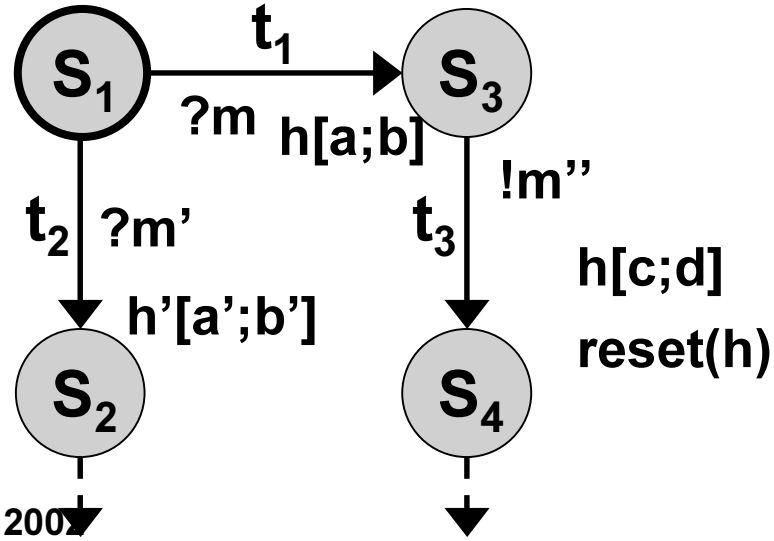
- **state machines / temporal attributes**
- **deterministic approach**

“Timed automata”

2 - Timed Input Output State Machine (TIOSM) *an adequate formalism*

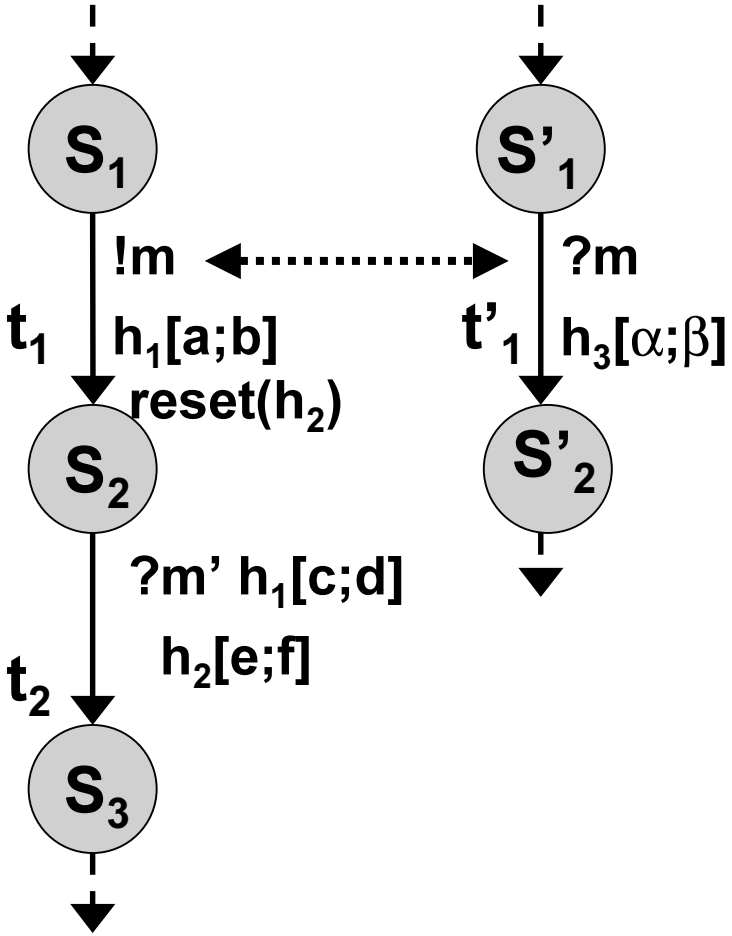
> TIOSM T=(S, L,C,s_0,ε)

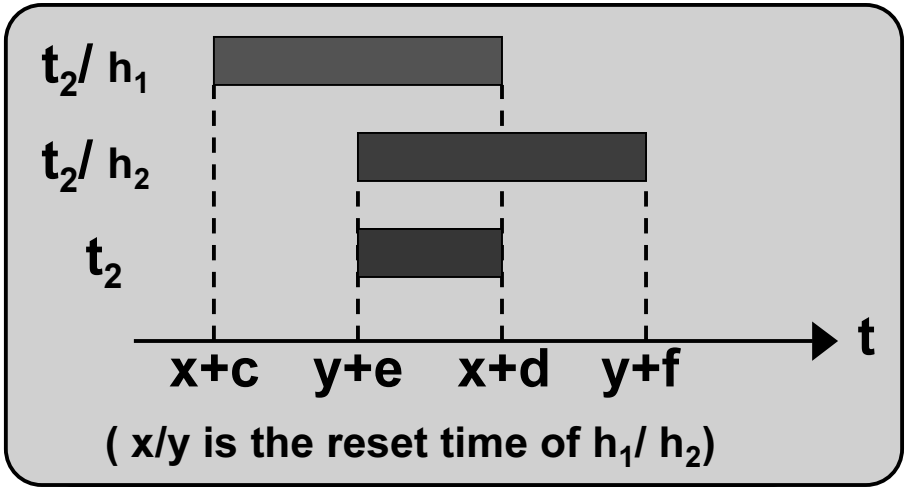
- S set of states / ε set of transitions
- s_0 is the initial state
- L set of messages
- C set of clocks



2 - Timed Input Output State Machine (TIOSM) *an adequate formalism*

> **Temporal behavior : an exemple**

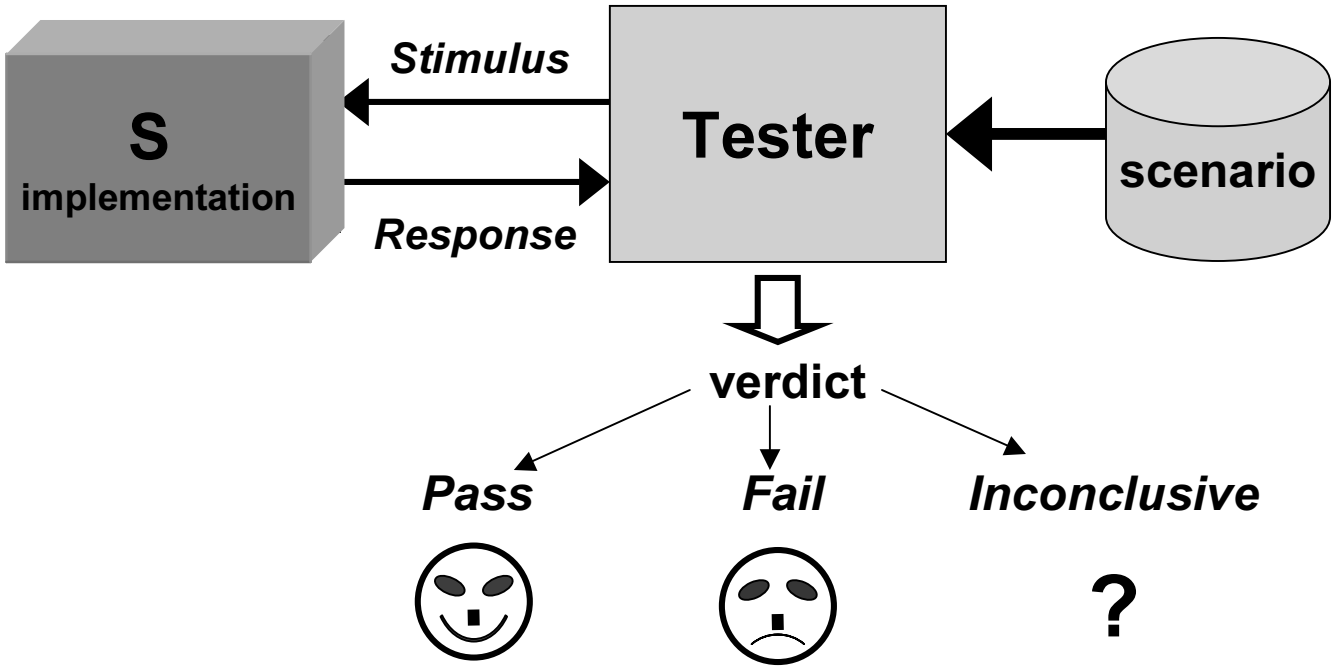




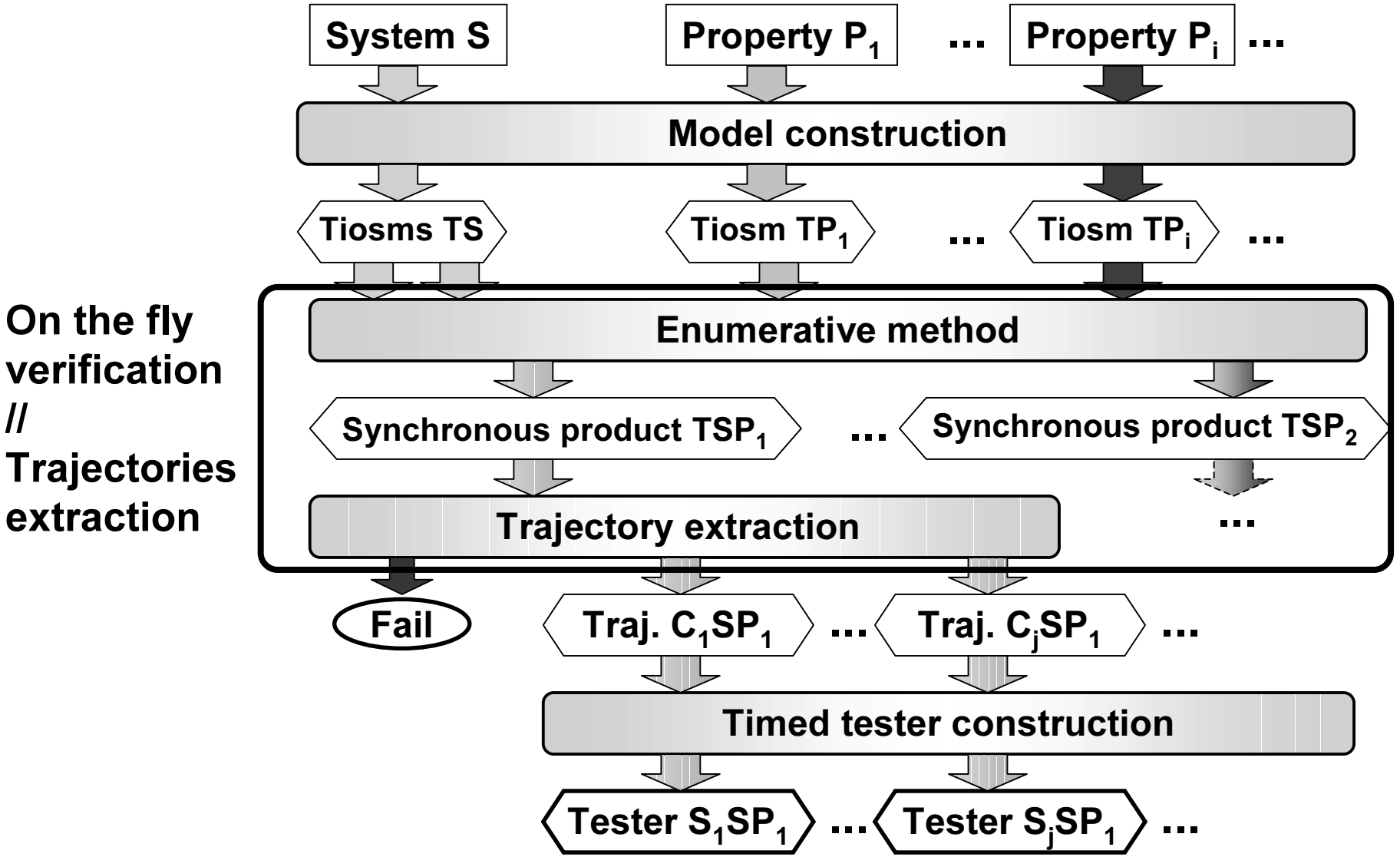
3 - Validation a posteriori (Integration test)

Implementation of S

Verification of Temporal Interoperability Properties Testing approach

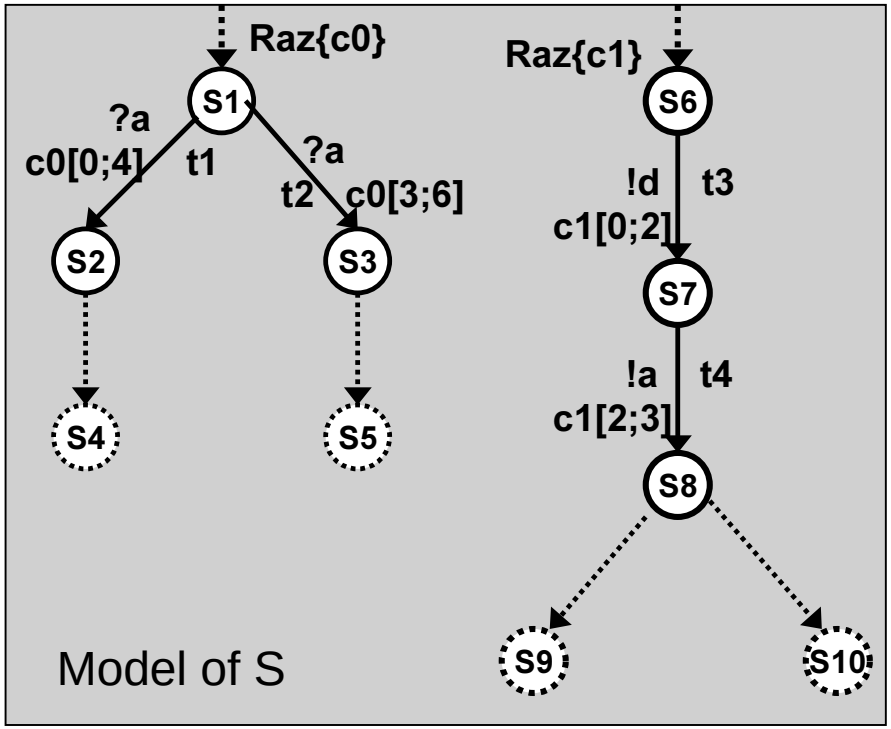


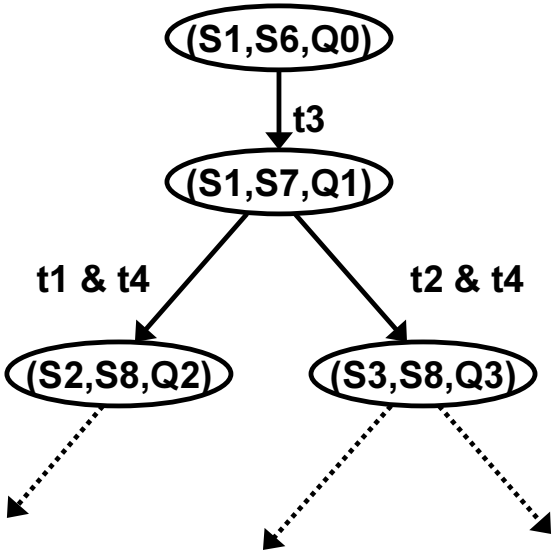
3 - Validation a posteriori (Integration test)



3 - Validation a posteriori (Integration test)

Construction of the state classes graph (cf. Berthomieu&Diaz) :





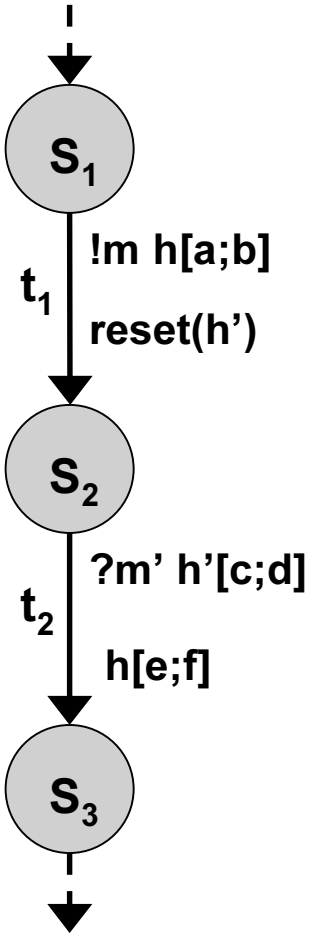
State classes graph of S

Si, Sj : current states of the TIOSM

Qk : inequations system

3 - Validation a posteriori (Integration test)

Construction of the inequations system / Pre-enabled transitions



Class 1

- **{S1}**
- **Q1**

$a \le \theta(t_1) \le b$

$e \le \theta(t_2) \le f$

$\theta(t_1) - \theta(t_2) \le b-e$

$\theta(t_2) - \theta(t_1) \le f-a$

...

For each pre-enabled transition t_i :

$\mathbf{min_i - b \le \theta(t_i) \le max_i - a}$

$\mathbf{0 \le \theta(t_i)}$

...

$\mathbf{e-b \le \theta(t_2) \le f-a}$

$\mathbf{0 \le \theta(t_2)}$

...

Class 2

- **{S2}**
- **Q2**

$e-b \le \theta(t_2) \le f-a$

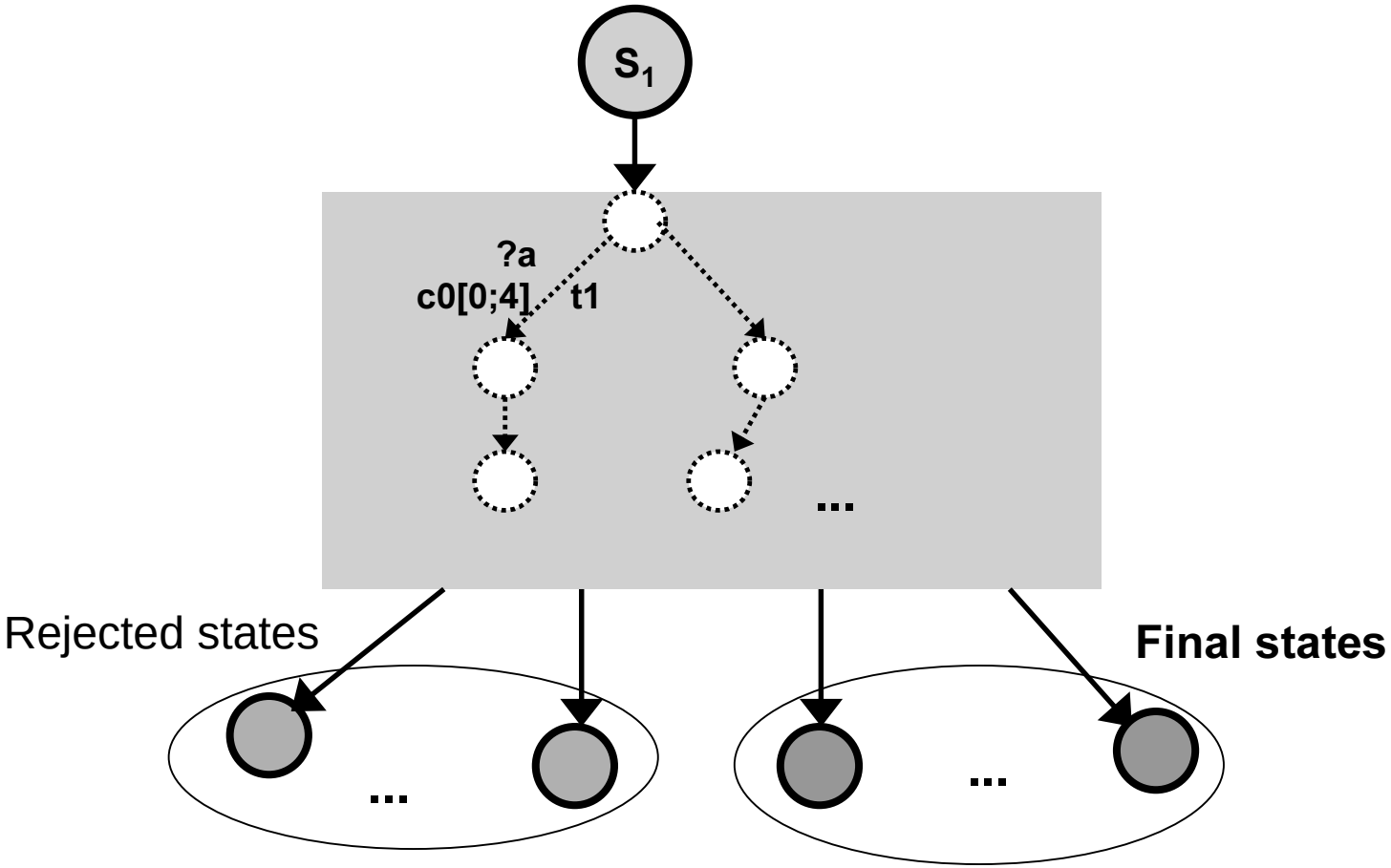
$c \le \theta(t_2) \le d$

$0 \le \theta(t_2)$

...

3 - Validation a posteriori (Integration test)

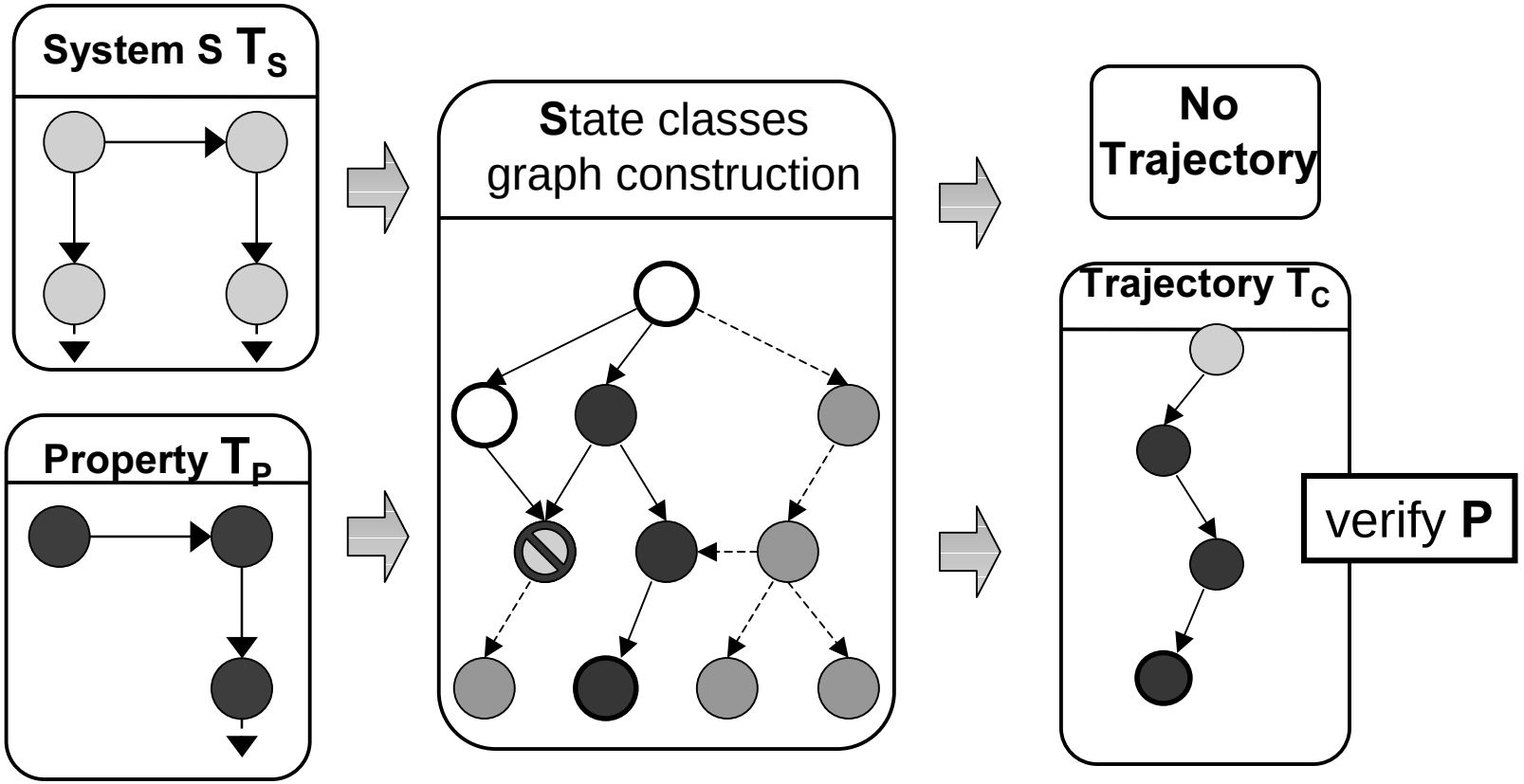
> **Model of a Property T_p (TIOSM)**



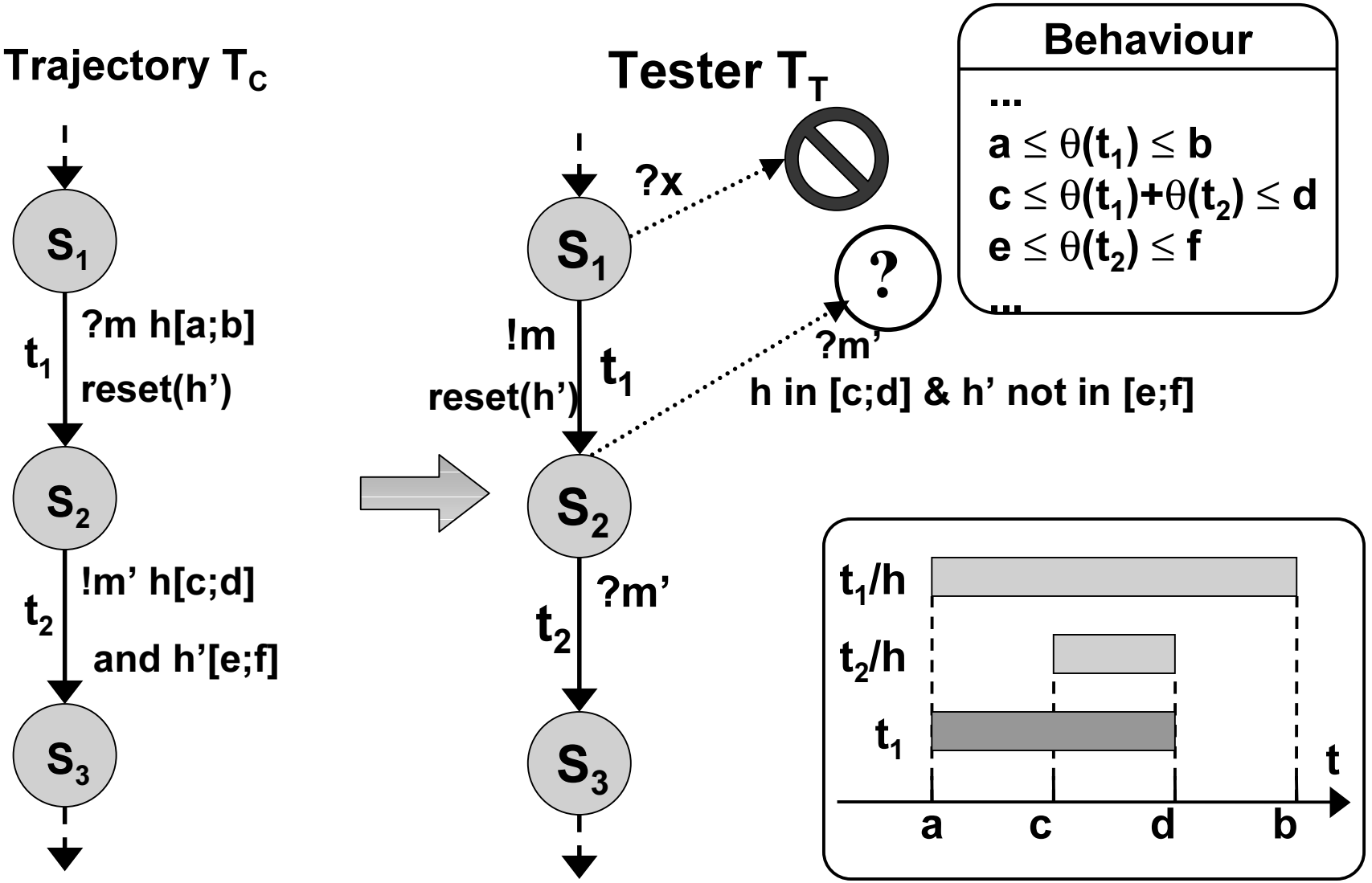
3 - Validation a posteriori (Integration test)

On the fly trajectories extraction

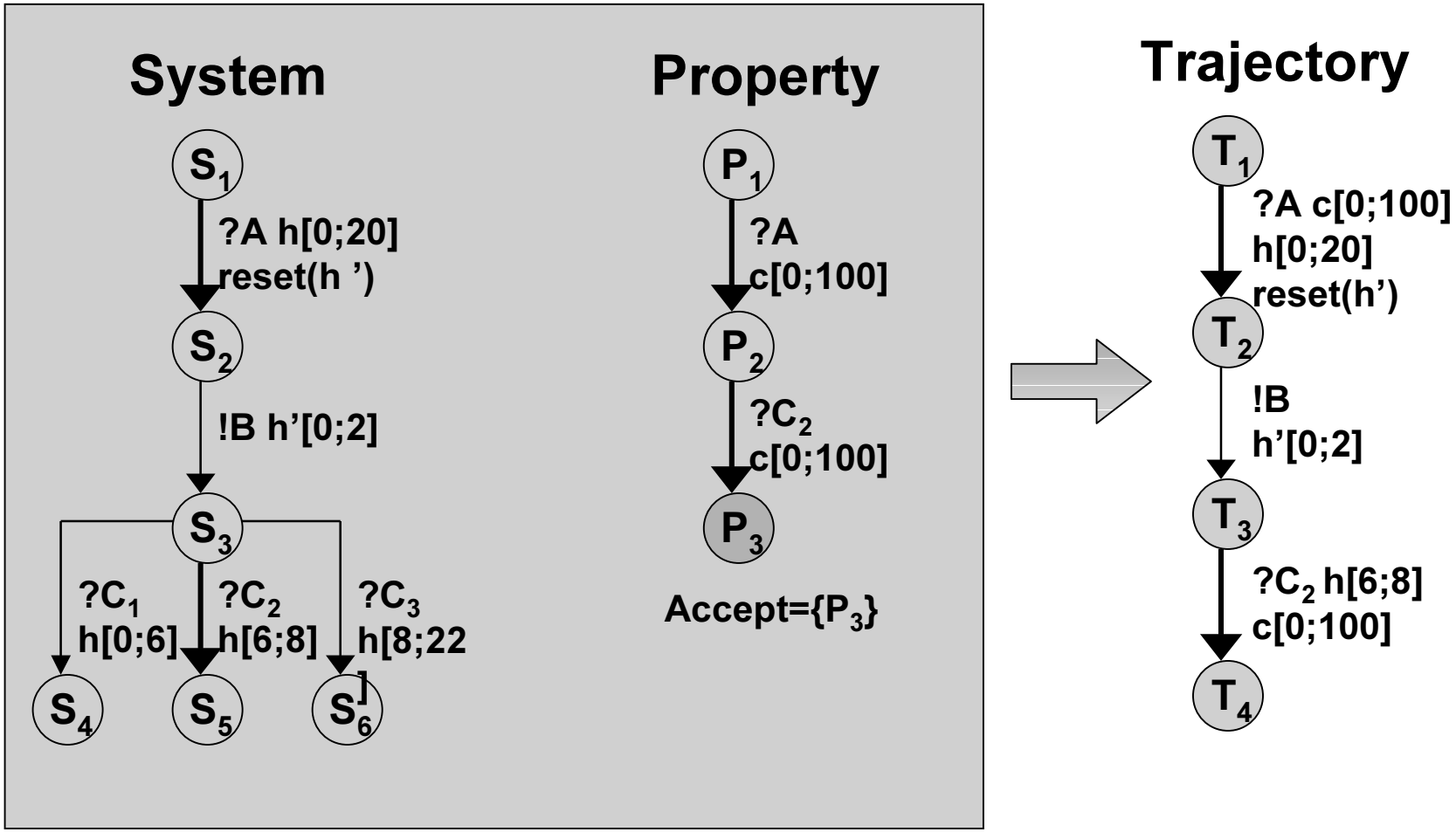
- **The property is searched during the state classes graph construction**
- **Stop as soon as the property is founded or ...**



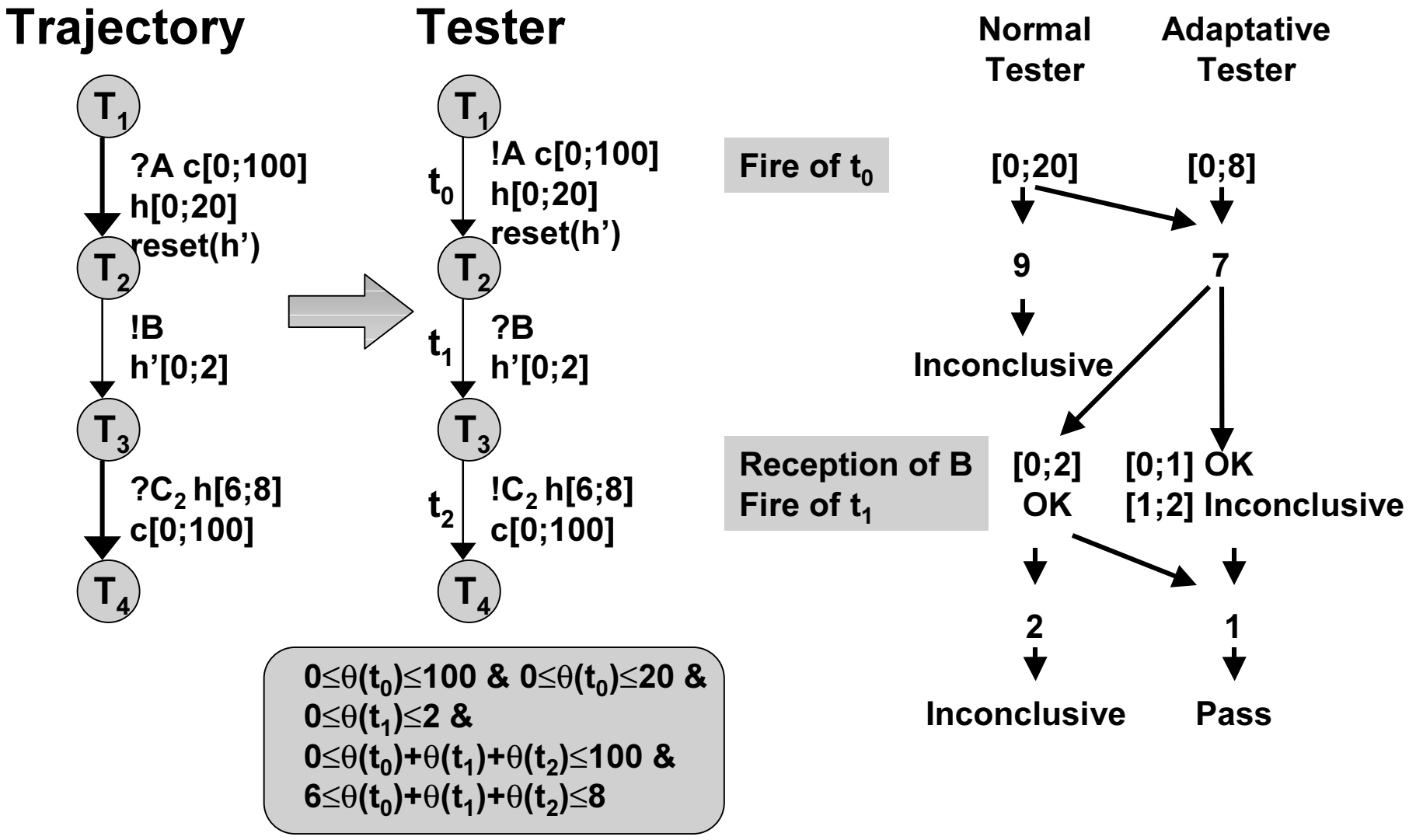
3 - Validation a posteriori (Integration test)



3 - Validation a posteriori (Integration test)



3 - Validation a posteriori (Integration test)

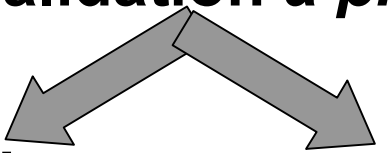


3 - Validation a posteriori (Integration test)

- **Conclusions**
  - **Formal generation of timed test sequences**
  - **Reduction of *inconclusive* cases during the test process**

4 - Validation a priori
how to master the complexity

Validation *a priori*



model analysis

+ exhaustivity
- number of states

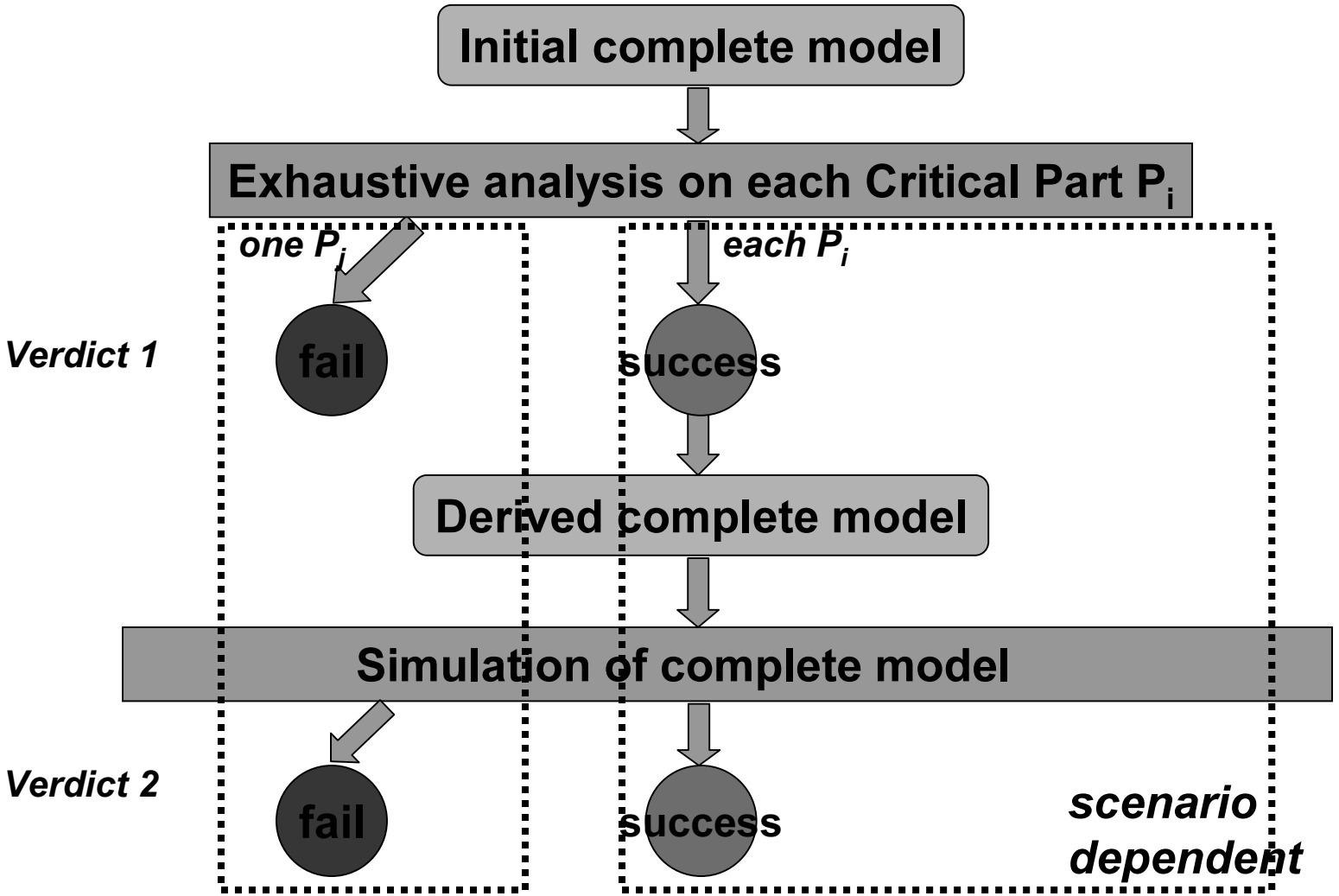
model simulation

+ easy to use
- non exhaustivity
scenario dependent

An hybrid validation method

exhaustive analysis on critical parts
+
simulation on the complete system

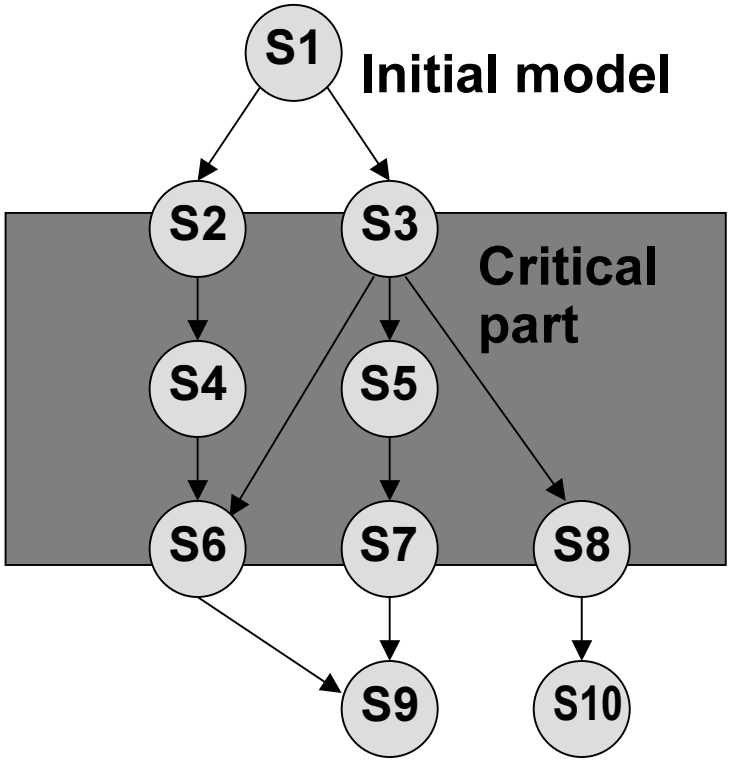
4 - Validation a priori - Hybrid Method



4 - Validation a priori - Hybrid Method

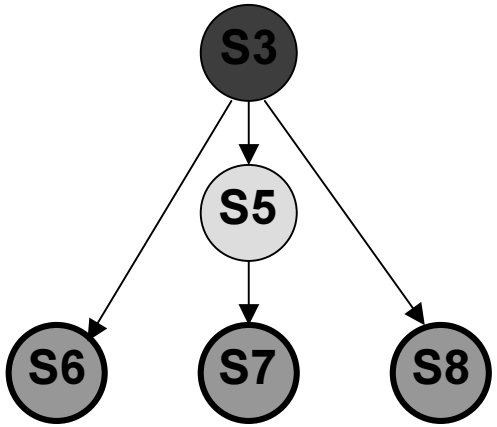
Critical parts - partial models definition

- 1 and only 1 input state
- at least one output states
- 2 partial models cannot be overlapped

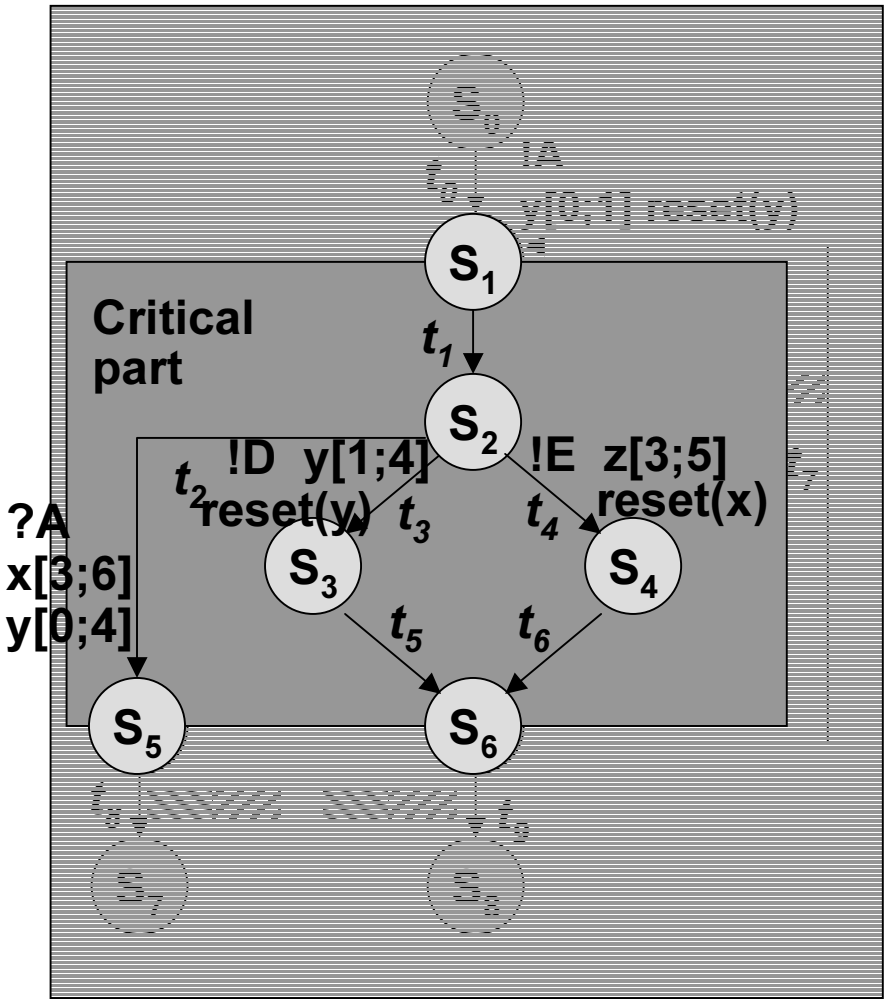


Partial model PM_1

Partial model PM_2

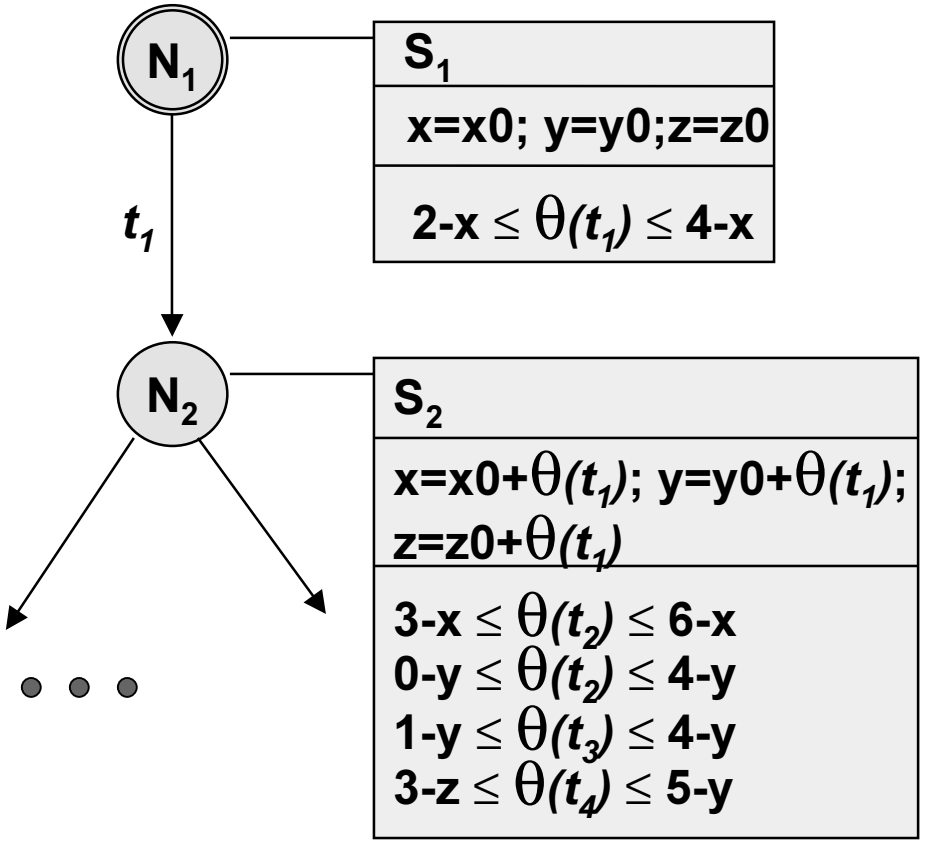


4 - Validation a priori - Hybrid Method



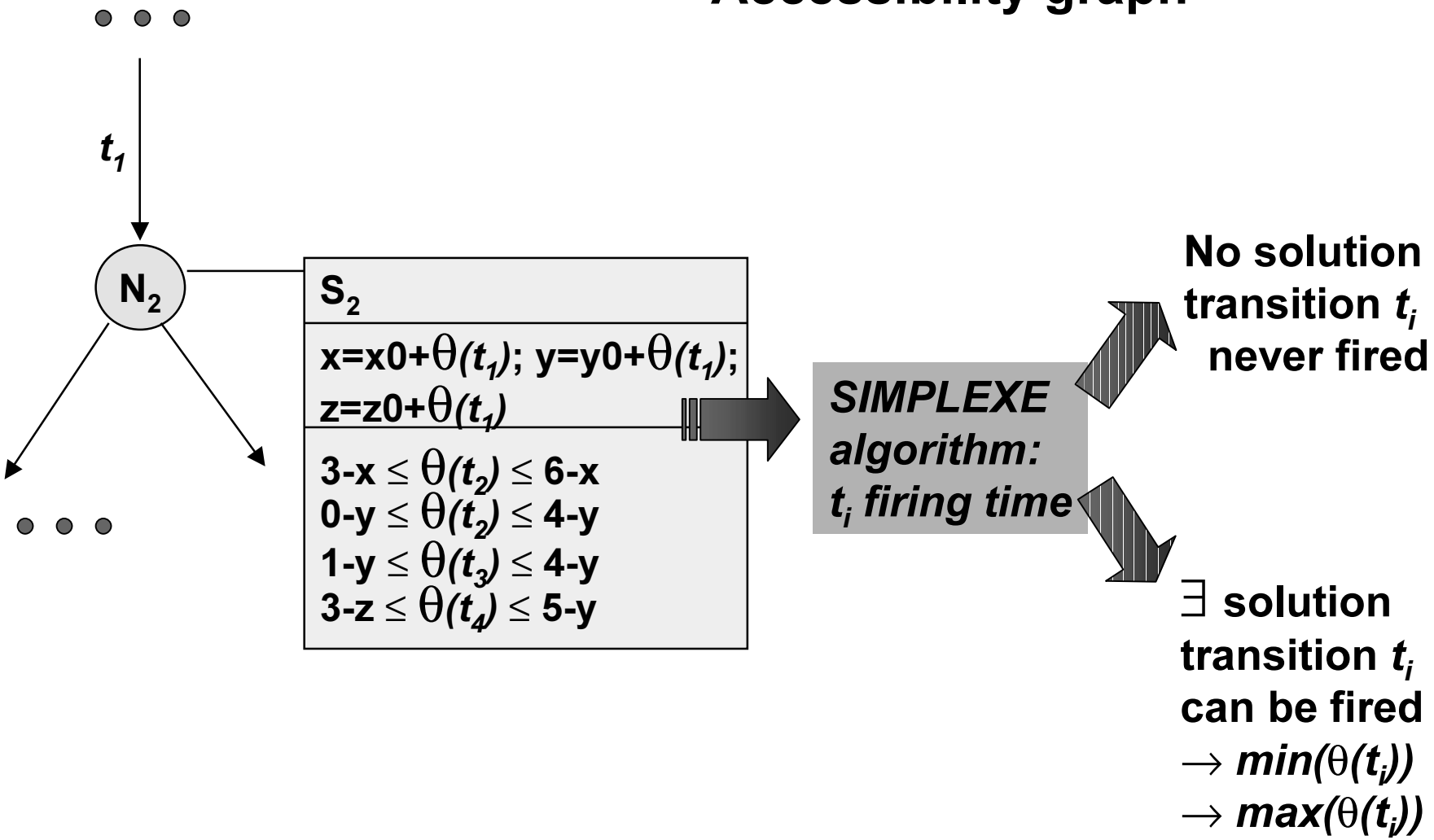
Accessibility graph

Set of clocks : C = {x,y,z}



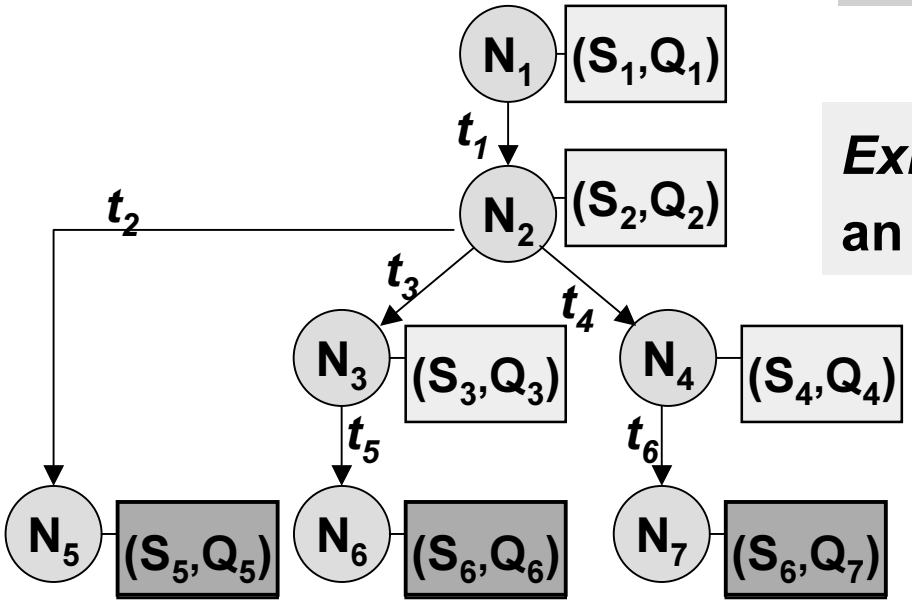
4 - Validation a priori - Hybrid Method

Accessibility graph



4 - Validation a priori - Hybrid Method

Accessibility graph



Exit states **of partial model : {S5, S6}**

Exit Nodes **: node characterized with an exit state**

Every exit node can be reached

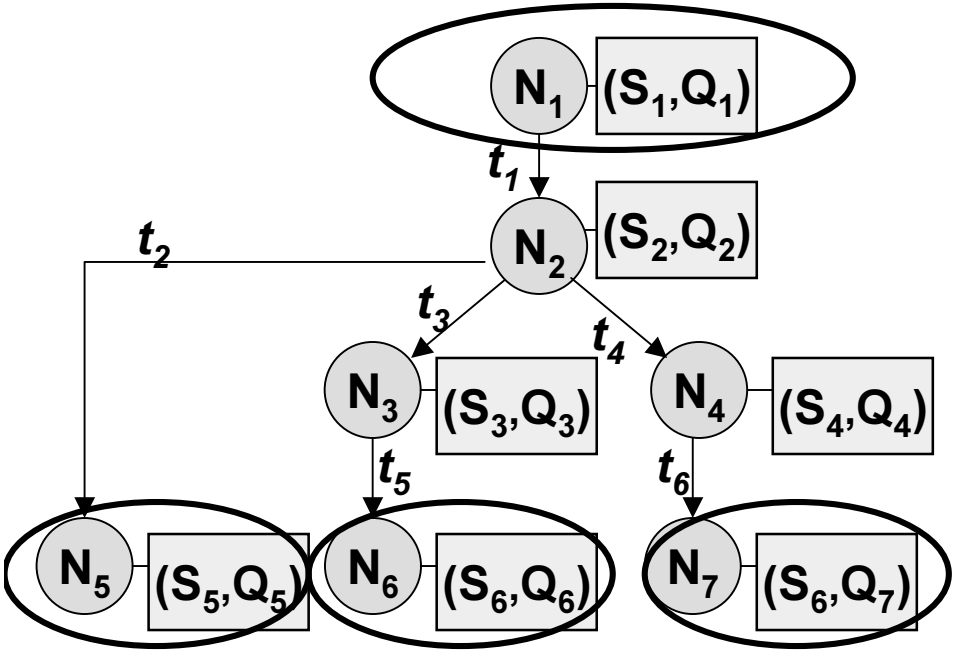
⇒ **Success**

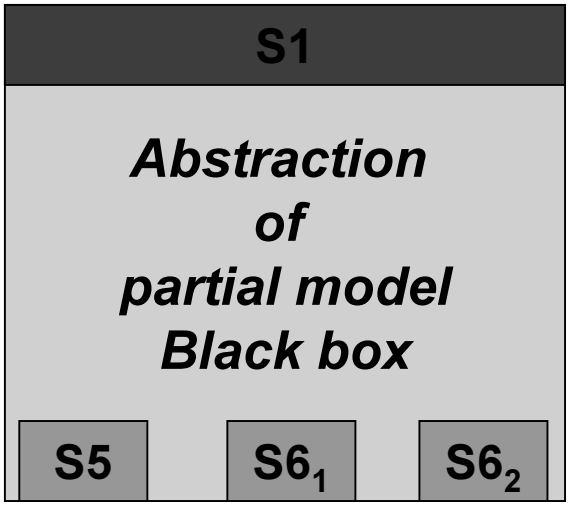
One exit node cannot be reached

⇒ **Fail**

4 - Validation a priori - Hybrid Method

Partial model	Accessibility graph	Abstraction of partial model
Initial state : S_1	Initial node : $N_1 = (S_1, Q_1)$	Initial state : S_1
Exit state : S_i	Node : $N_k = (S_i, Q_i)$	Exit state : $S_{i,l}$



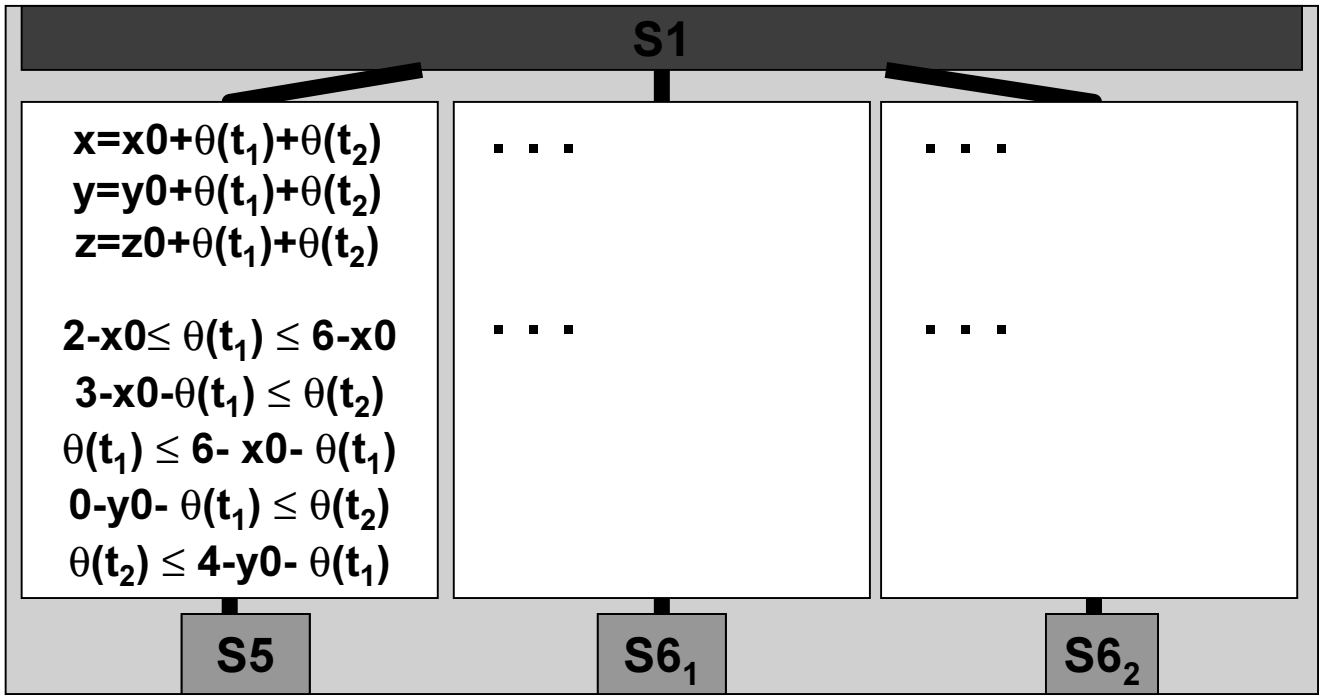


4 - Validation a priori - Hybrid Method

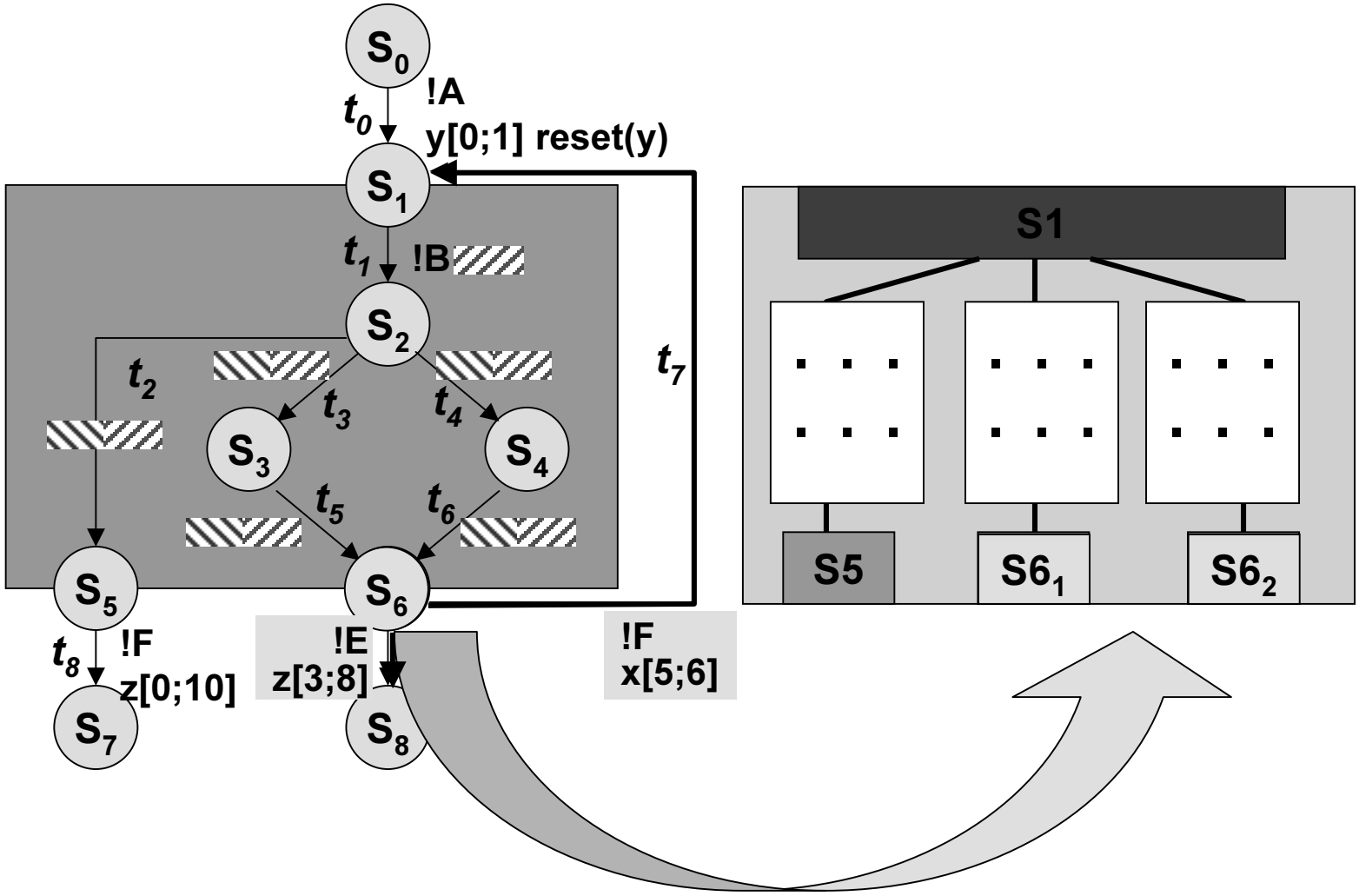
For each exit state S

Value of each clock when reaching this state

Constraints on clocks along the path from init state to S

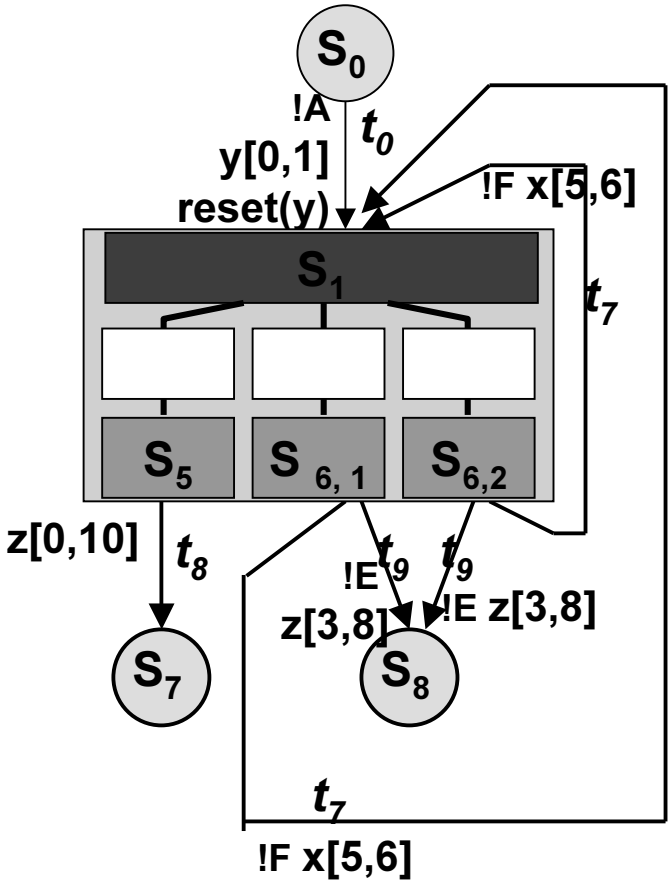


4 - Validation a priori - Hybrid Method

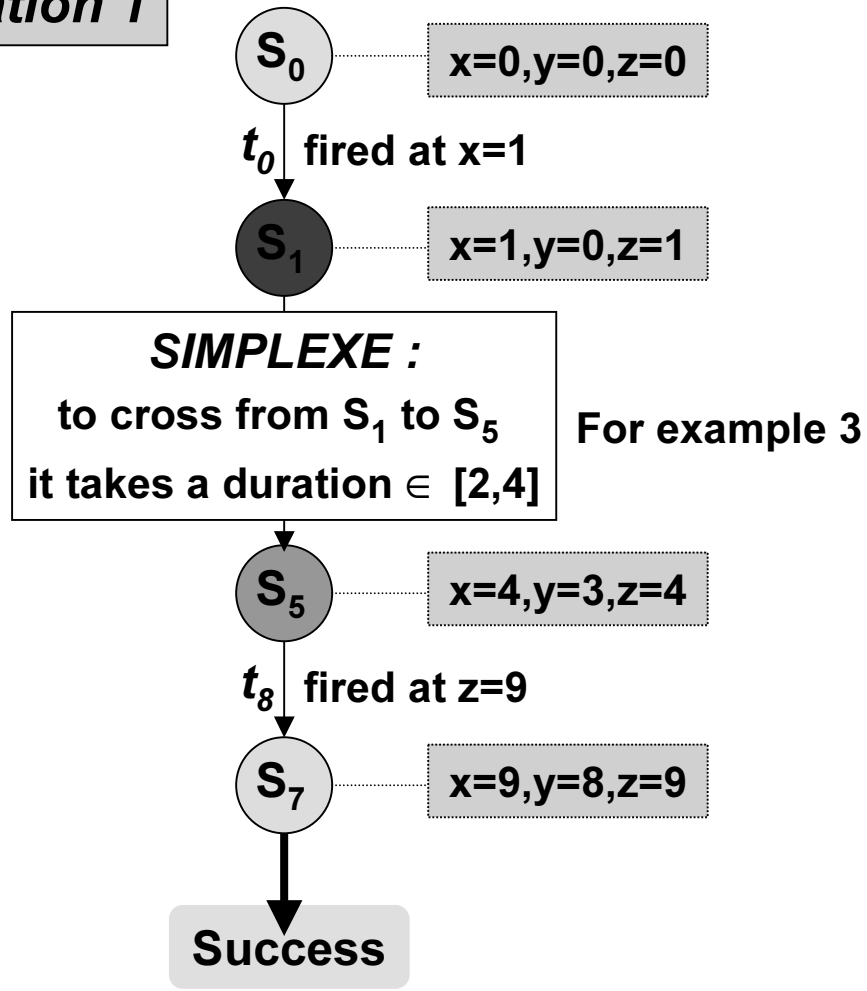


4 - Validation a priori - Hybrid Method

Derived model

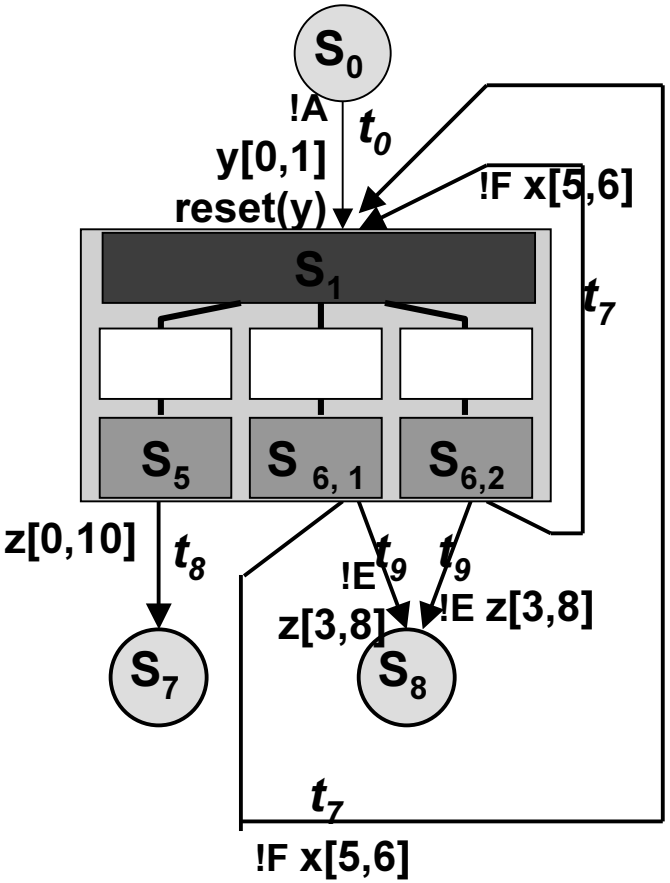


Simulation 1

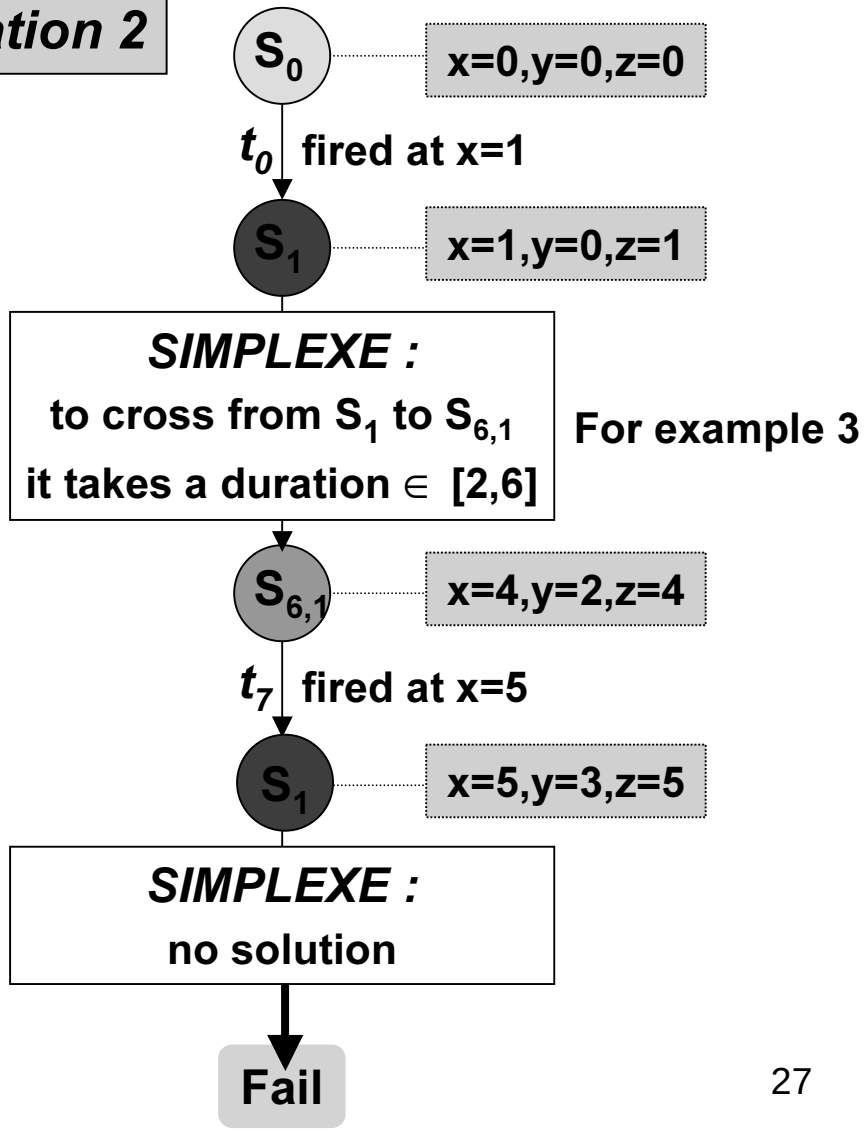


4 - Validation a priori - Hybrid Method

Derived model



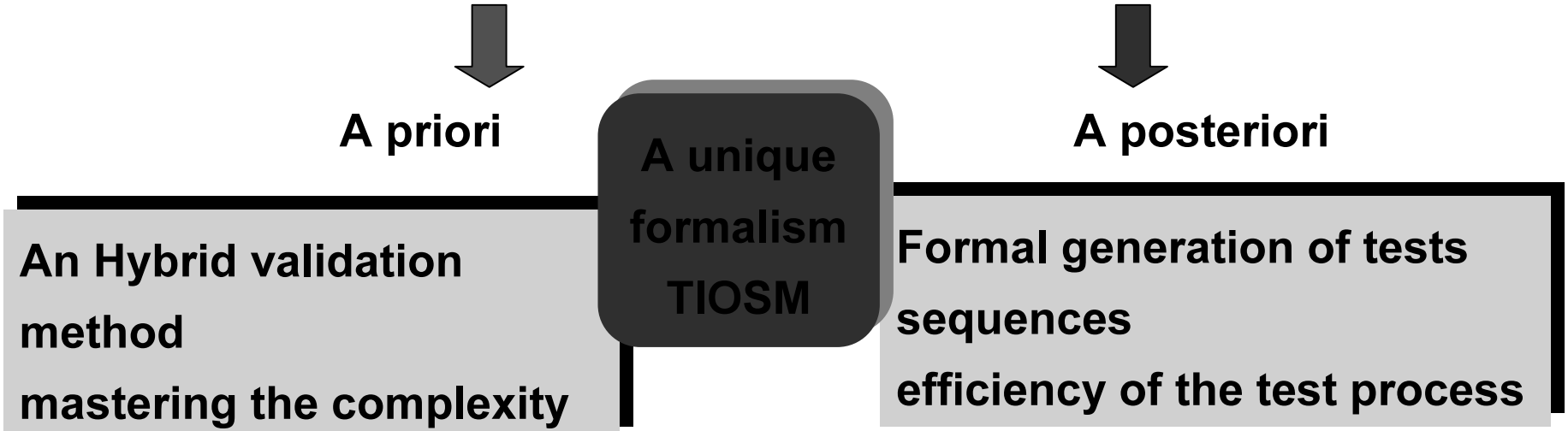
Simulation 2



4 - Validation a priori - Hybrid Method

- **Conclusions**
  - **A validation method decreasing the complexity**
  - **Simulation and Exhaustive analysis**

5 - Validationof Real Time Systems



Future trends

- ***Composition / communication between critical parts***
- ***COTS (Components off-the-shelf)***
- ***Industrial Tester - CANoe (TTCN)***