



HAL
open science

Validation of Real Time Applications

Françoise Simonot-Lion

► **To cite this version:**

Françoise Simonot-Lion. Validation of Real Time Applications. Séminaire Zhejiang University, 2002, Zhejiang/China, 29 p. inria-00107583

HAL Id: inria-00107583

<https://inria.hal.science/inria-00107583>

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Validation of Real Time Applications

Françoise Simonot - Lion

**Zhejiang University
China**

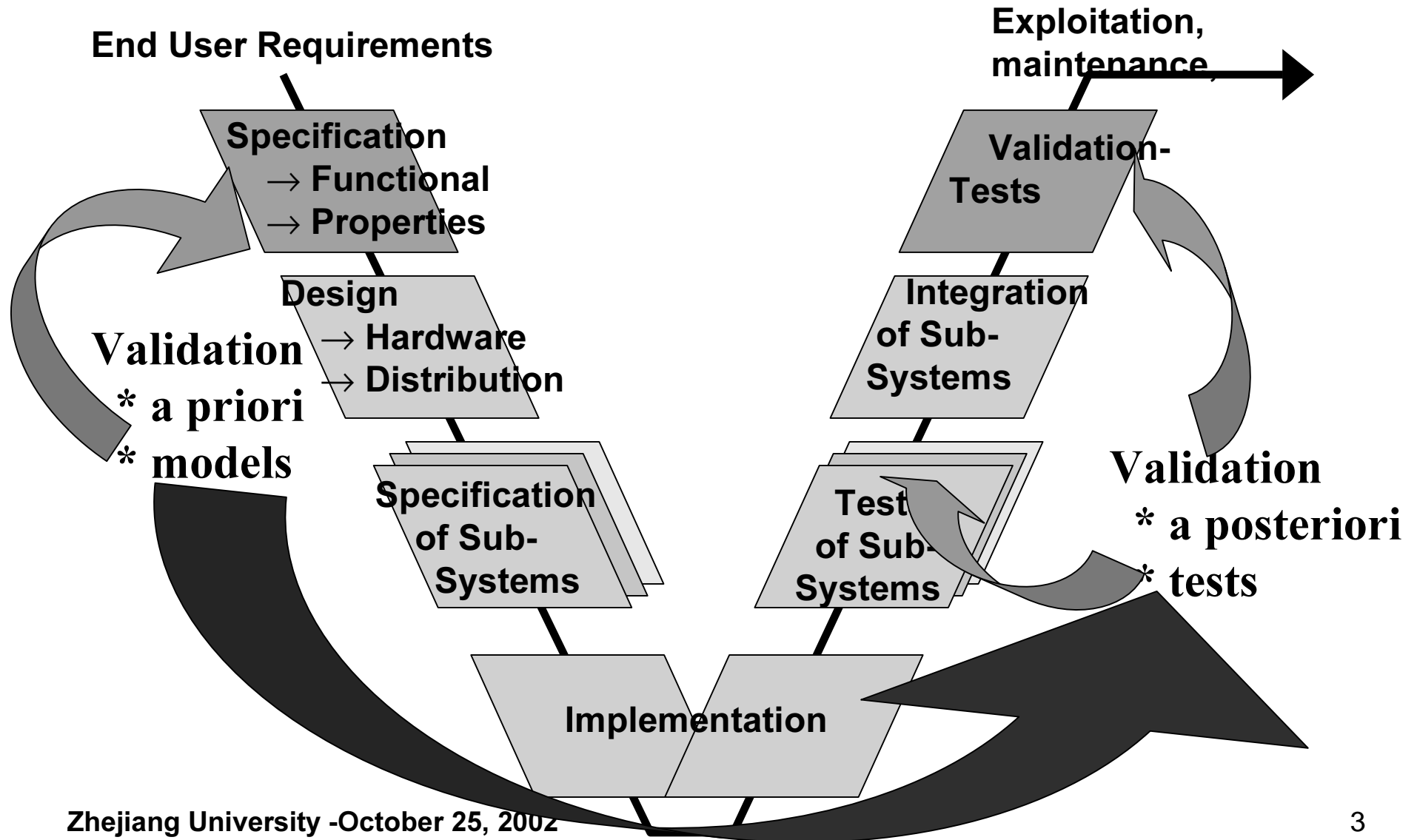
October 25, 2002



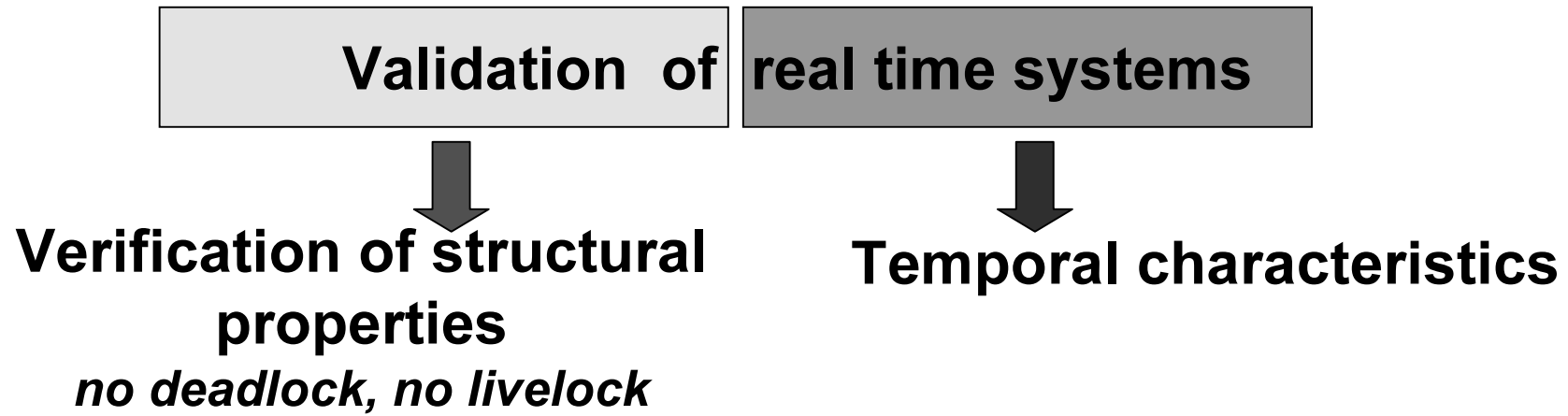
Contents

- 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



Model

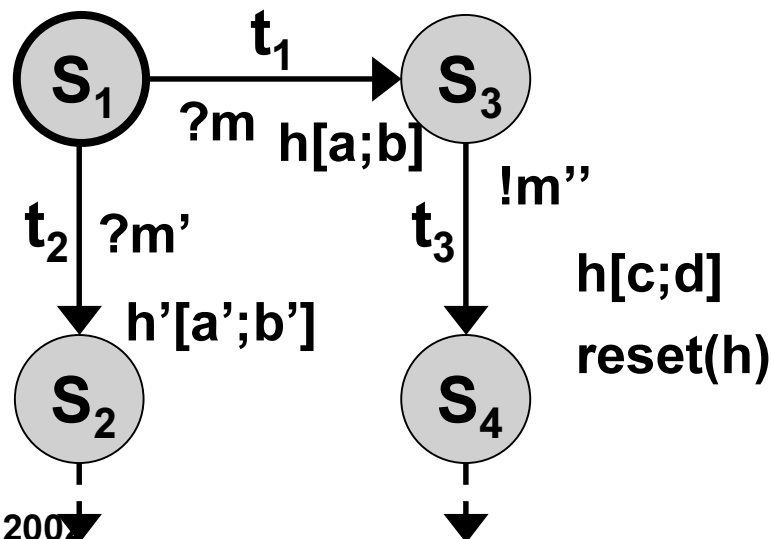
- state machines / temporal attributes
- deterministic approach

“Timed automata”

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

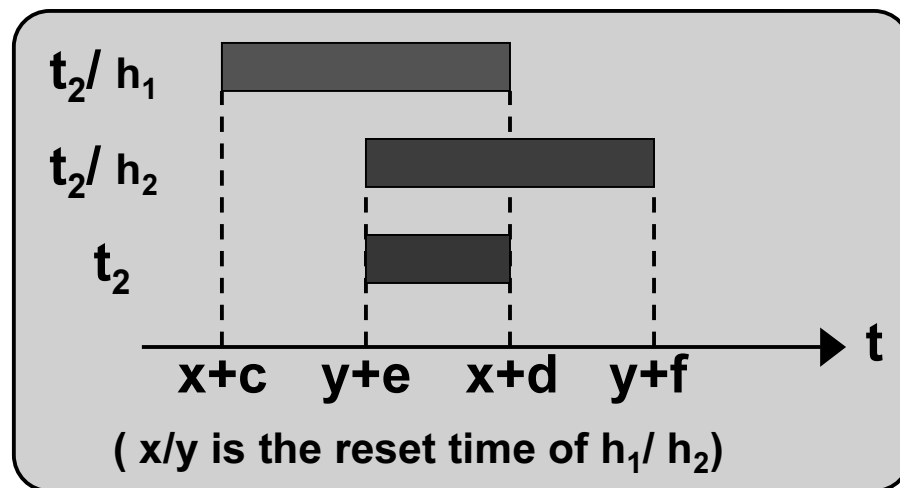
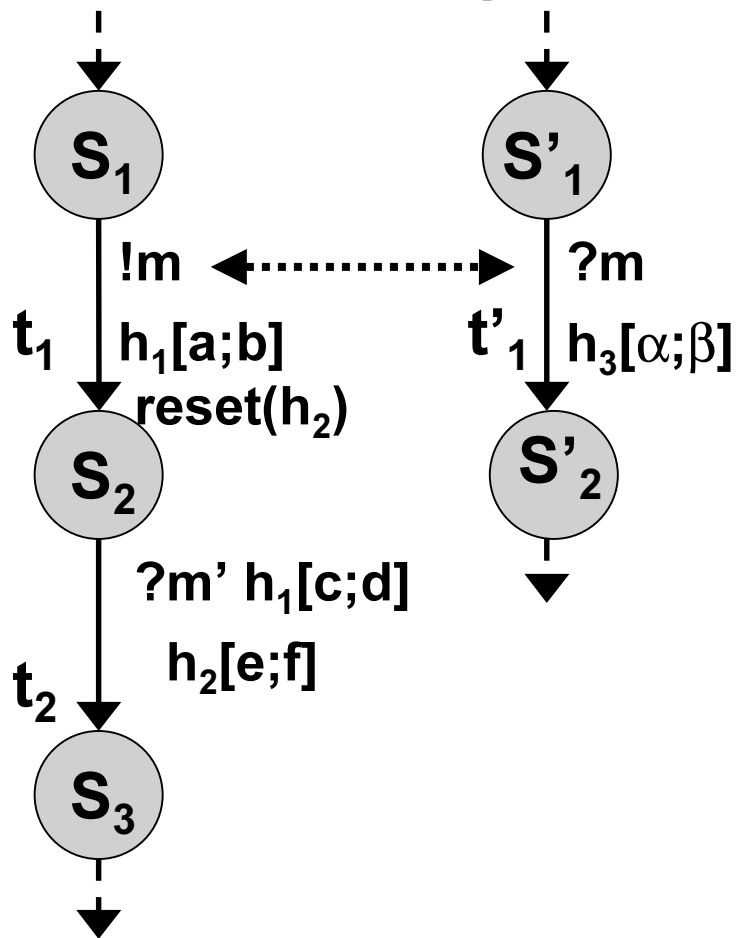
> TIOSM $T=(S, L, C, s_0, \varepsilon)$

- **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*

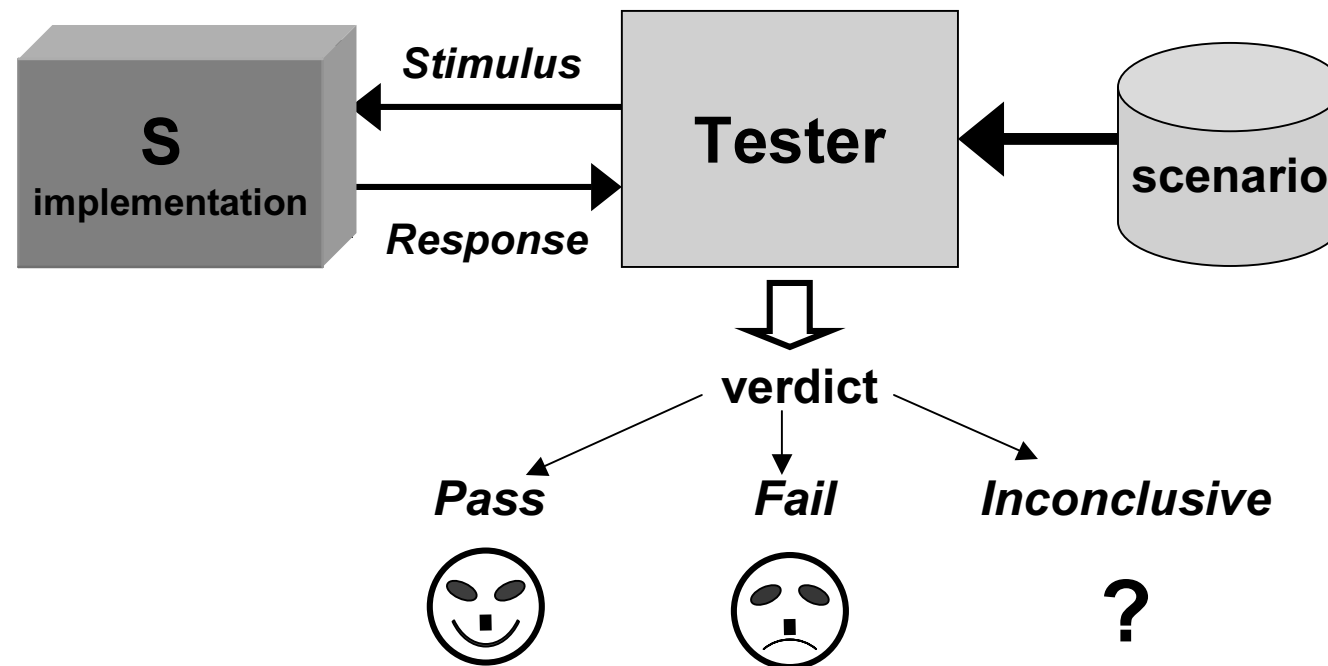
> Temporal behavior : an exemple



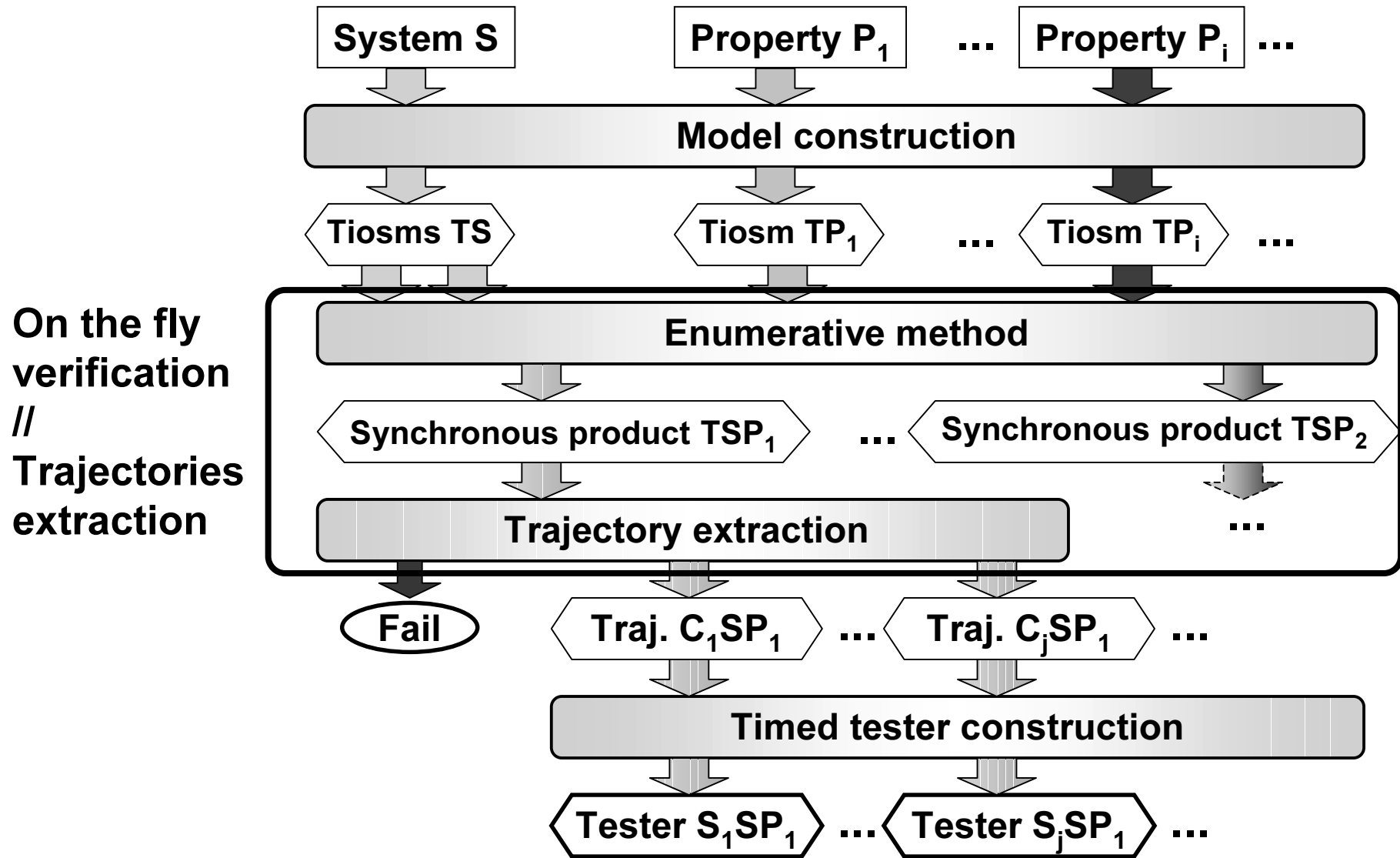
3 - Validation a posteriori (Integration test)

Implementation of S

Verification of Temporal Interoperability Properties
Testing approach

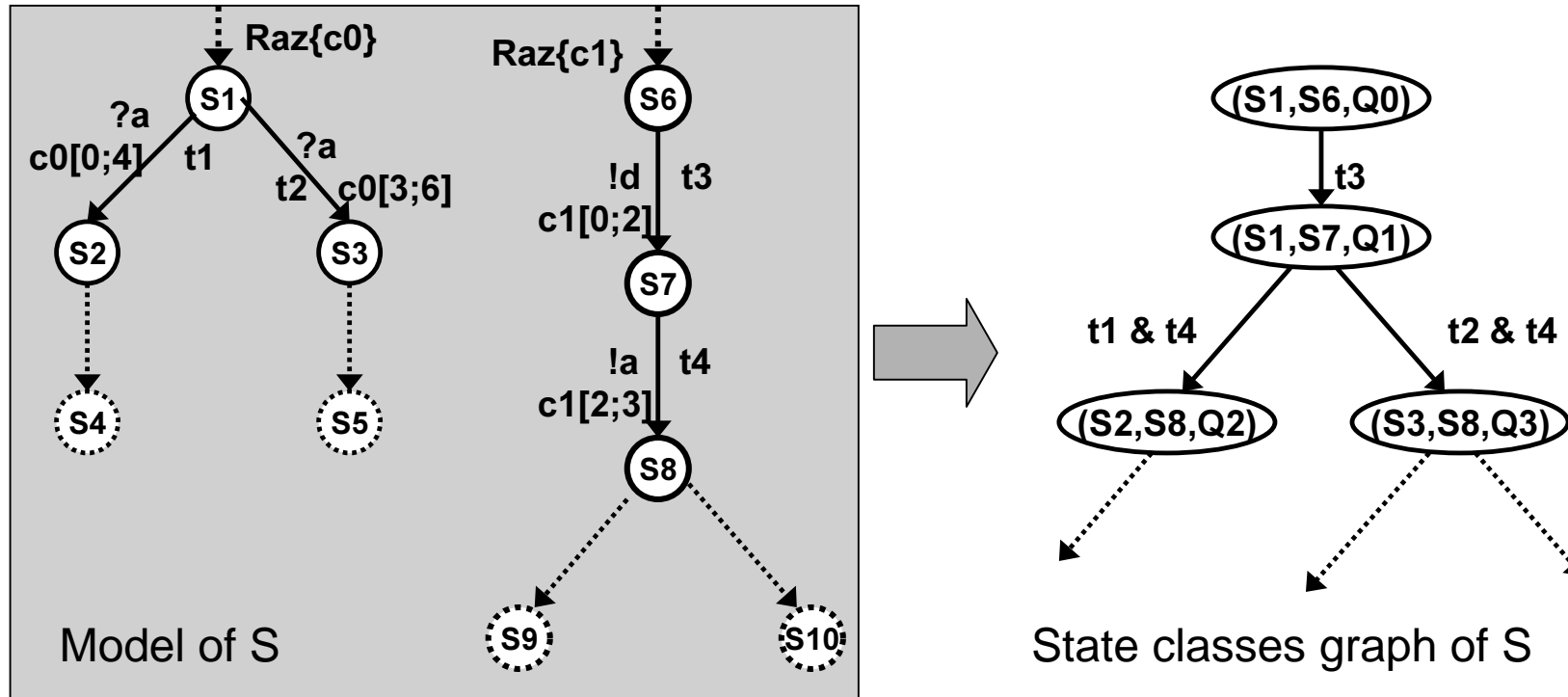


3 - Validation a posteriori (Integration test)



3 - Validation a posteriori (Integration test)

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

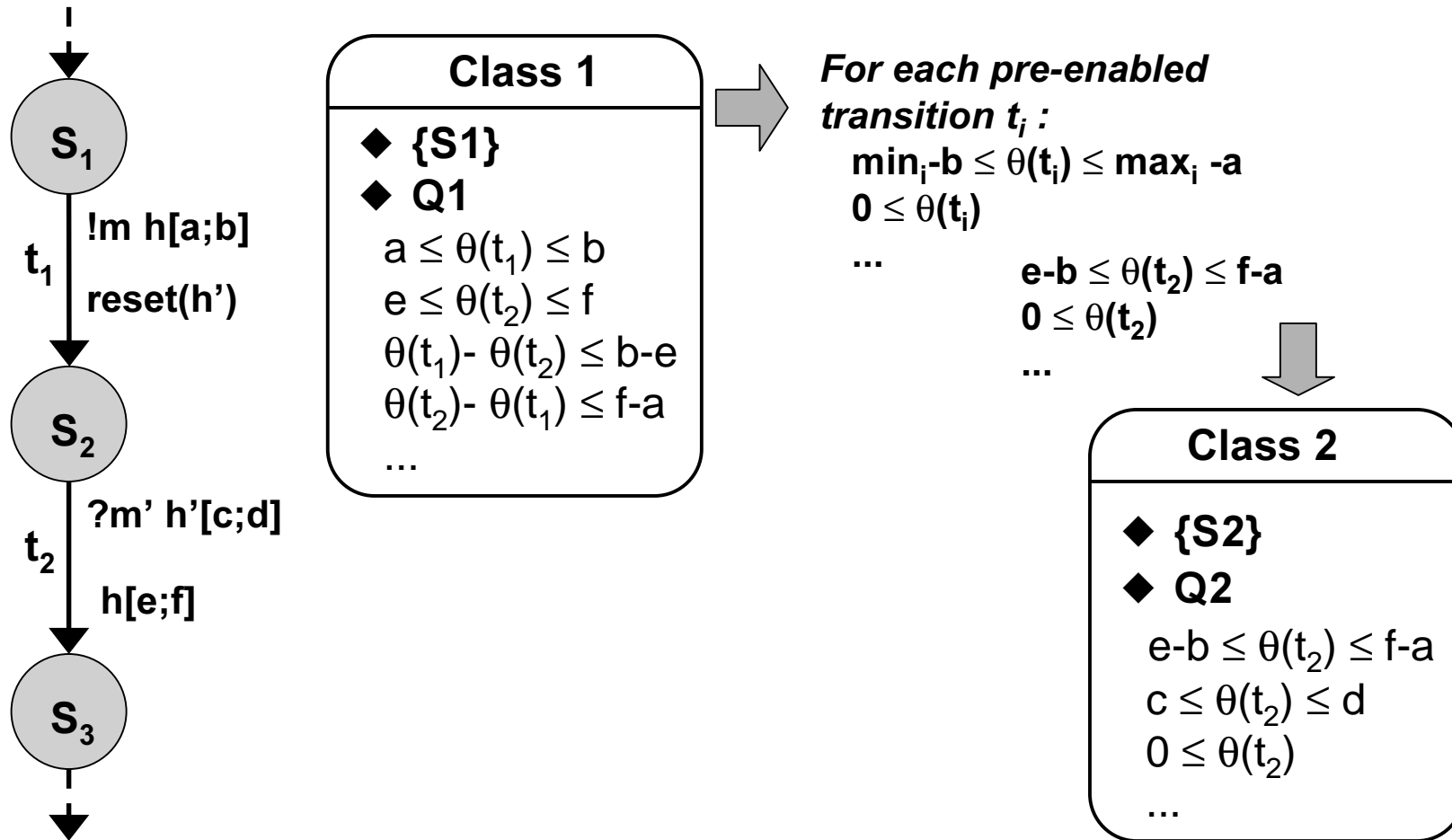


S_i, S_j : current states of the TIOSM

Q_k : inequations system

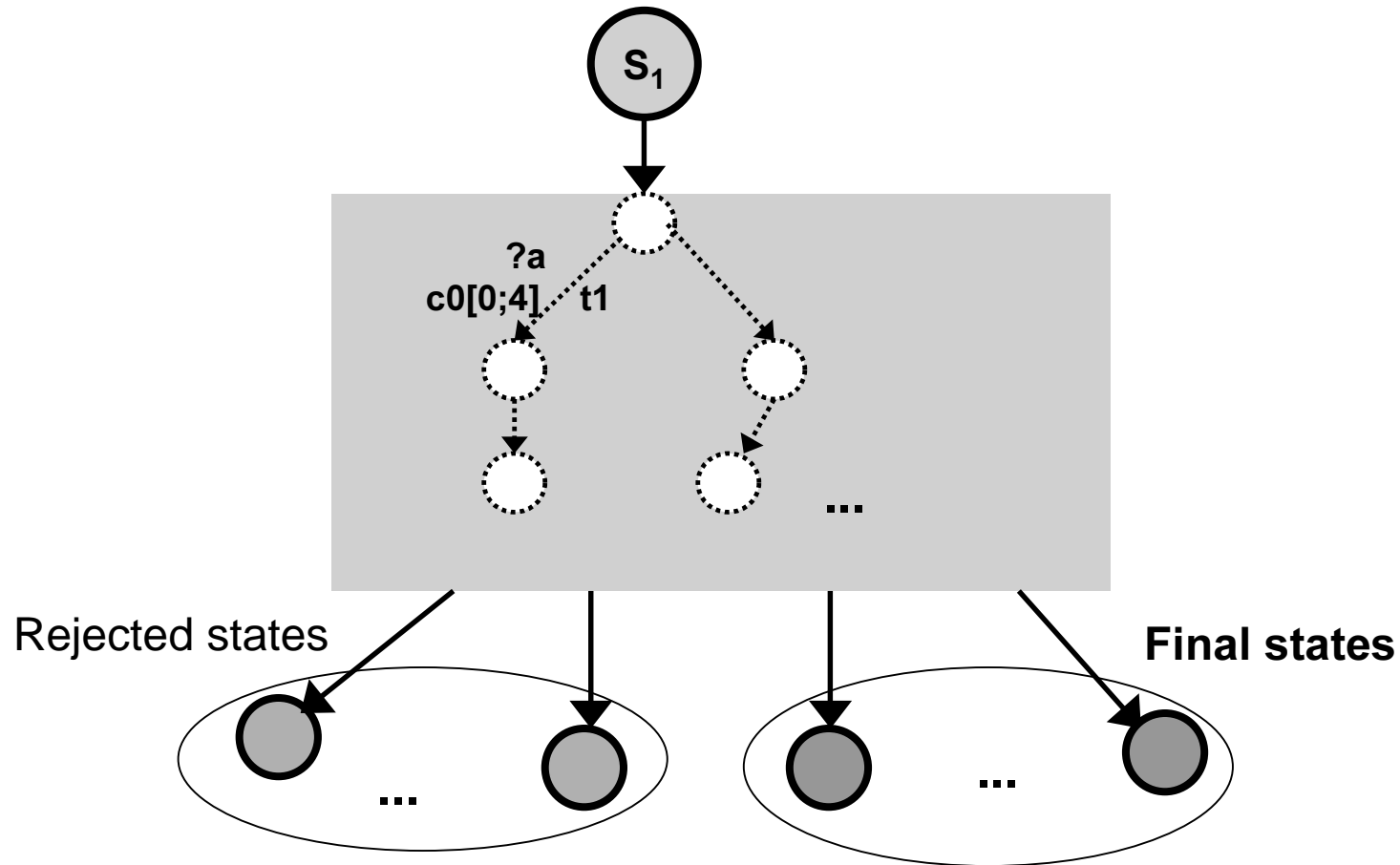
3 - Validation a posteriori (Integration test)

Construction of the inequations system / Pre-enabled transitions



3 - Validation a posteriori (Integration test)

> Model of a Property T_p (TIOSM)

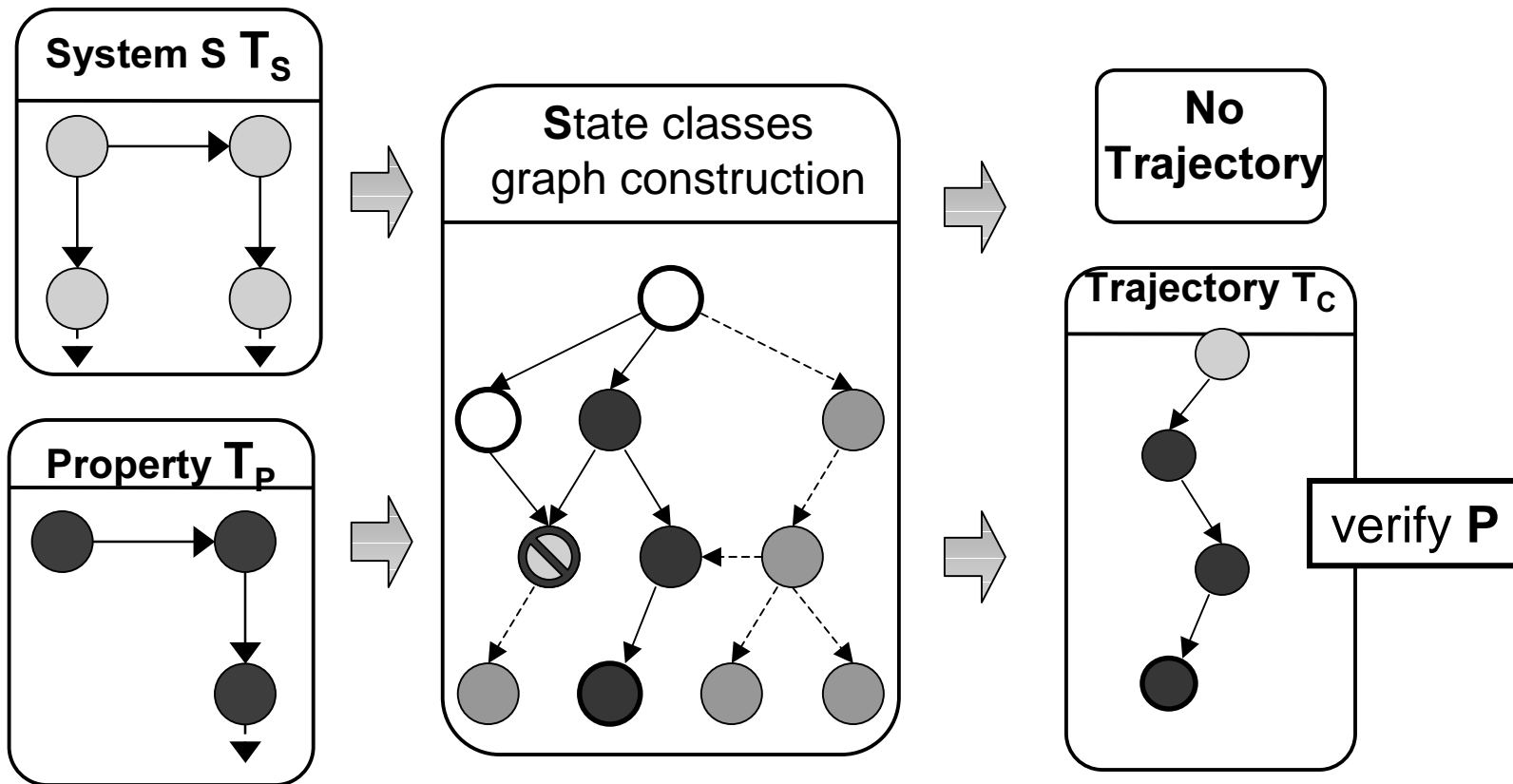


3 - Validation a posteriori (Integration test)

On the fly trajectories extraction

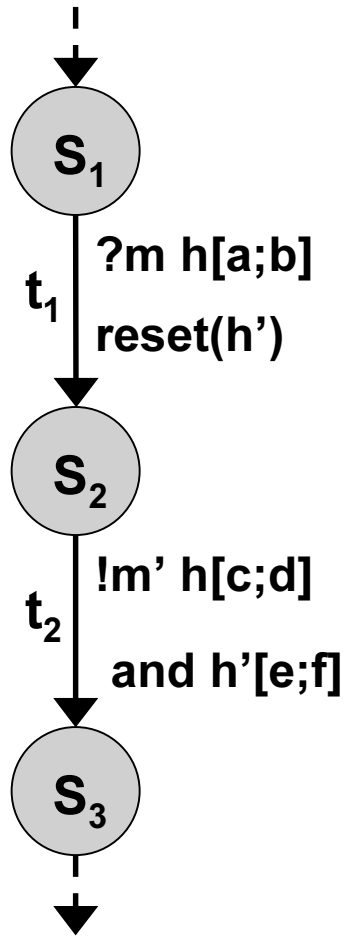
- The property is searched during the state classes graph construction

- Stop as soon as the property is founded or ...

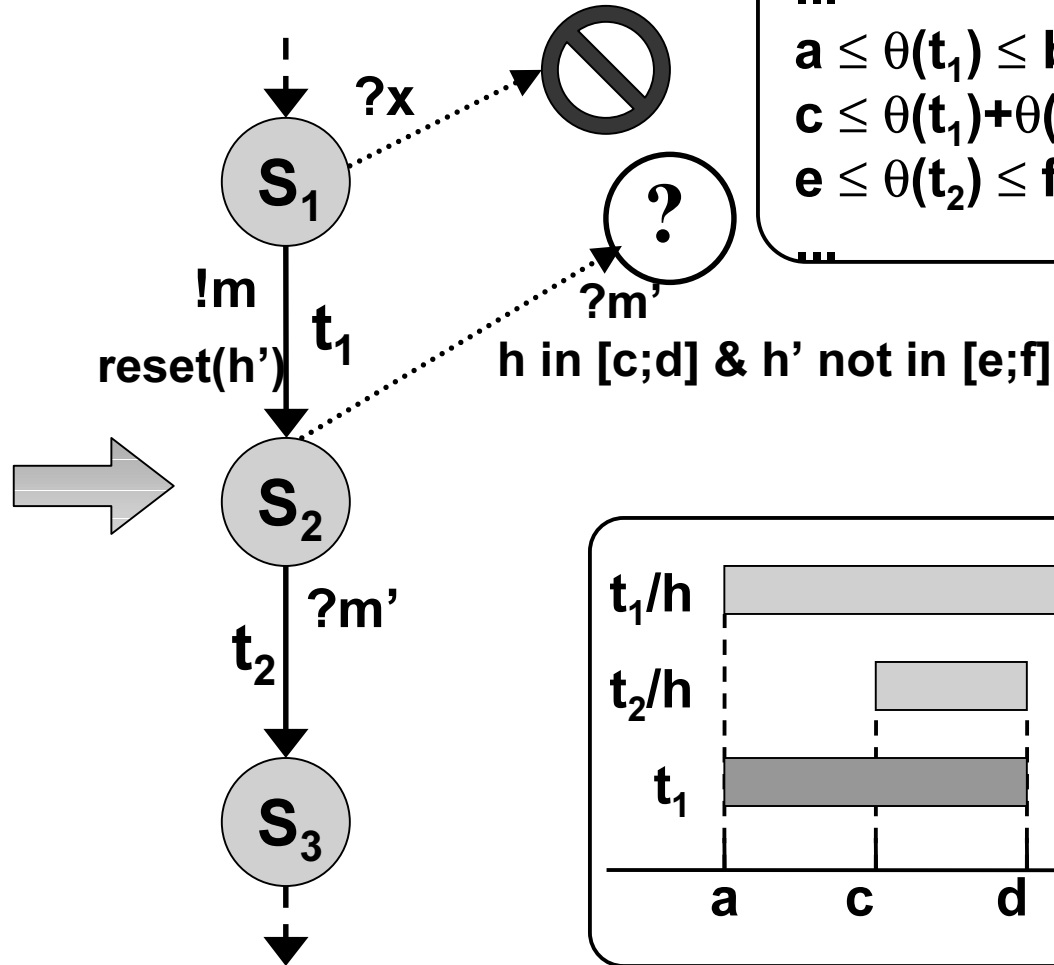


3 - Validation a posteriori (Integration test)

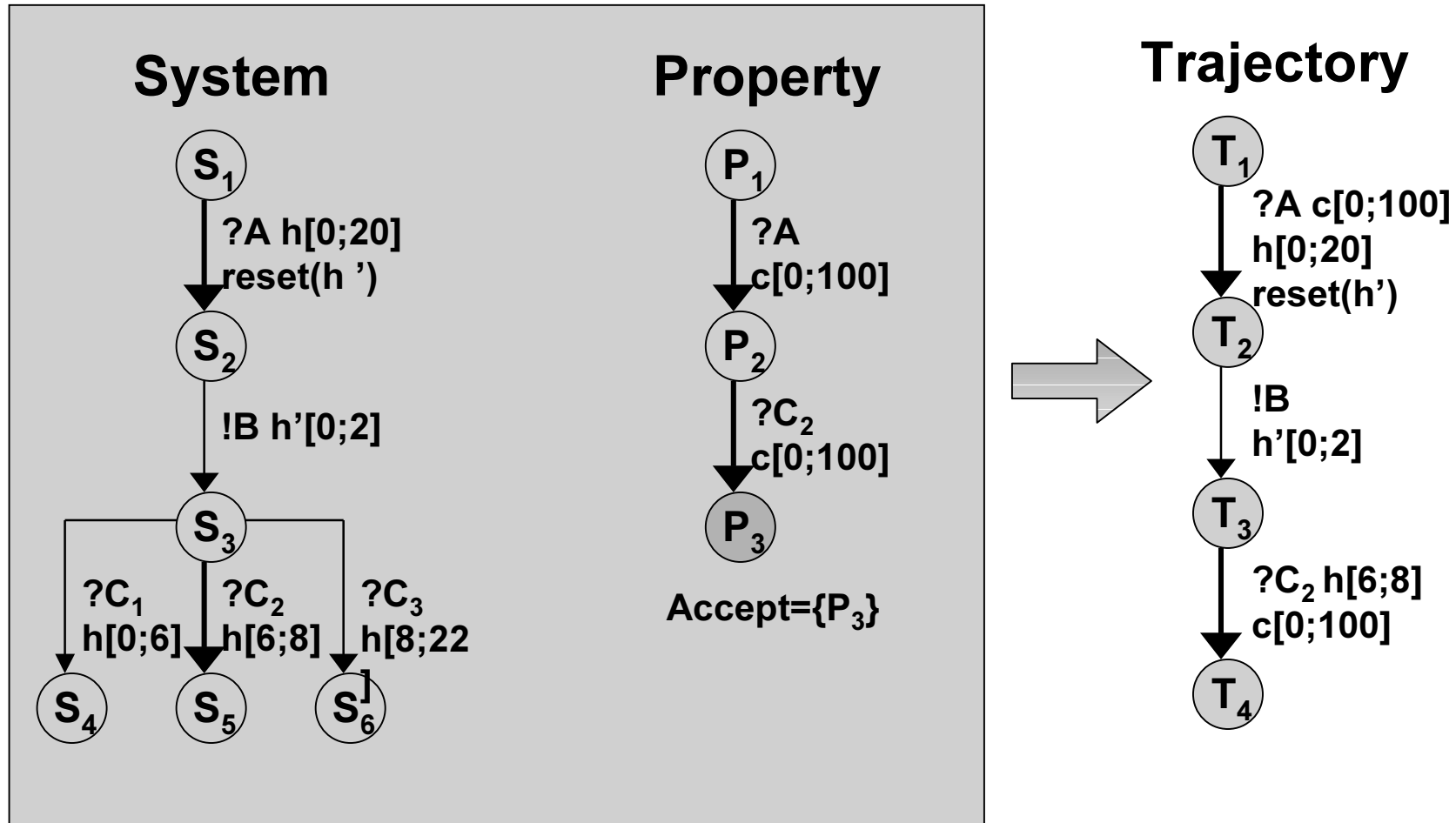
Trajectory T_C



Tester T_T

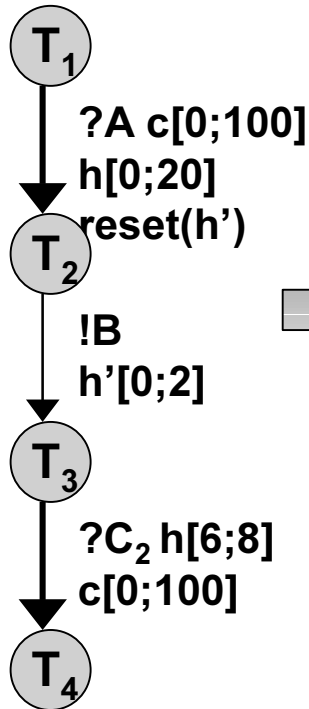


3 - Validation a posteriori (Integration test)

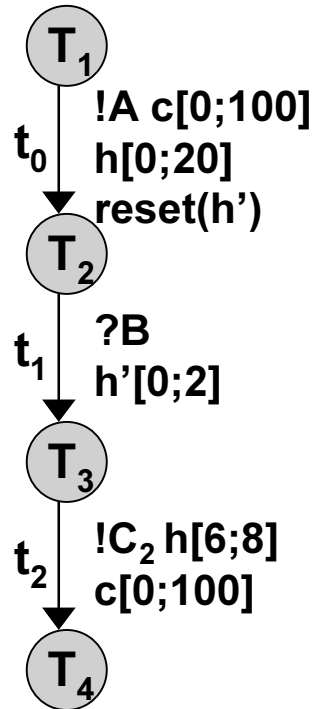


3 - Validation a posteriori (Integration test)

Trajectory

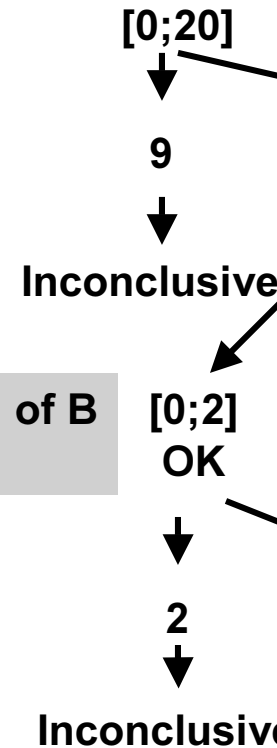


Tester

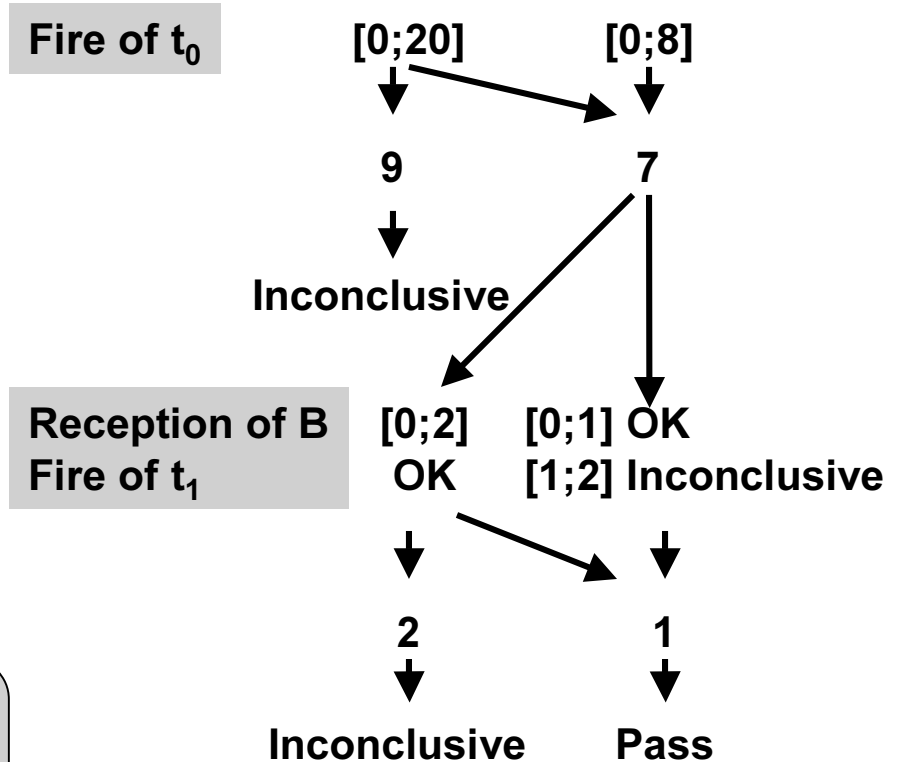


$$\begin{aligned}
 &0 \leq \theta(t_0) \leq 100 \ \& \ 0 \leq \theta(t_0) \leq 20 \ \& \\
 &0 \leq \theta(t_1) \leq 2 \ \& \\
 &0 \leq \theta(t_0) + \theta(t_1) + \theta(t_2) \leq 100 \ \& \\
 &6 \leq \theta(t_0) + \theta(t_1) + \theta(t_2) \leq 8
 \end{aligned}$$

Normal Tester



Adaptative Tester



Fire of t_0

Reception of B
Fire of t_1

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

model simulation

+ exhaustivity
- number of states

+ easy to use
- non exhaustivity
scenario dependent

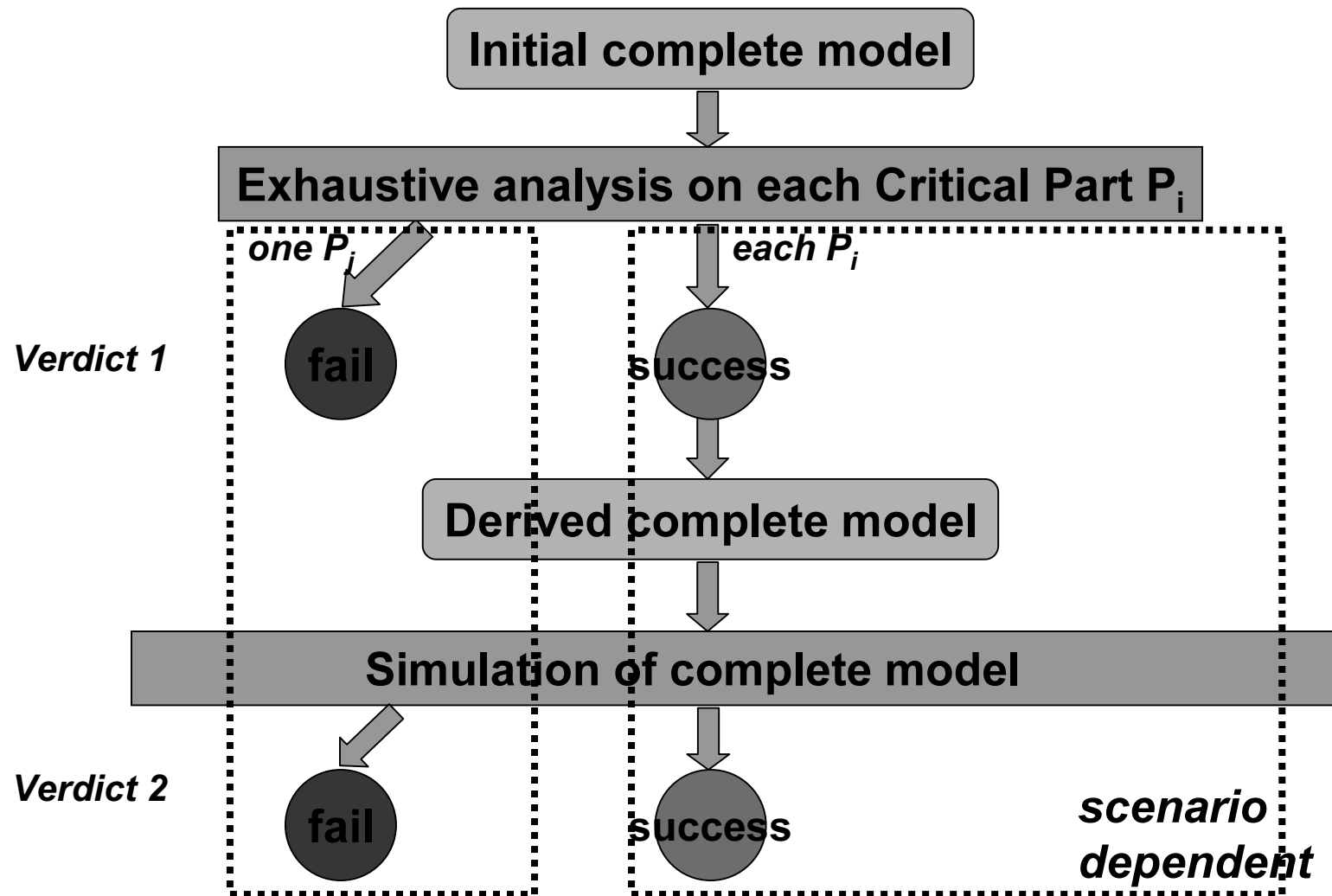
An hybrid validation method

{ exhaustive analysis on critical parts }

+

{ simulation on the complete system }

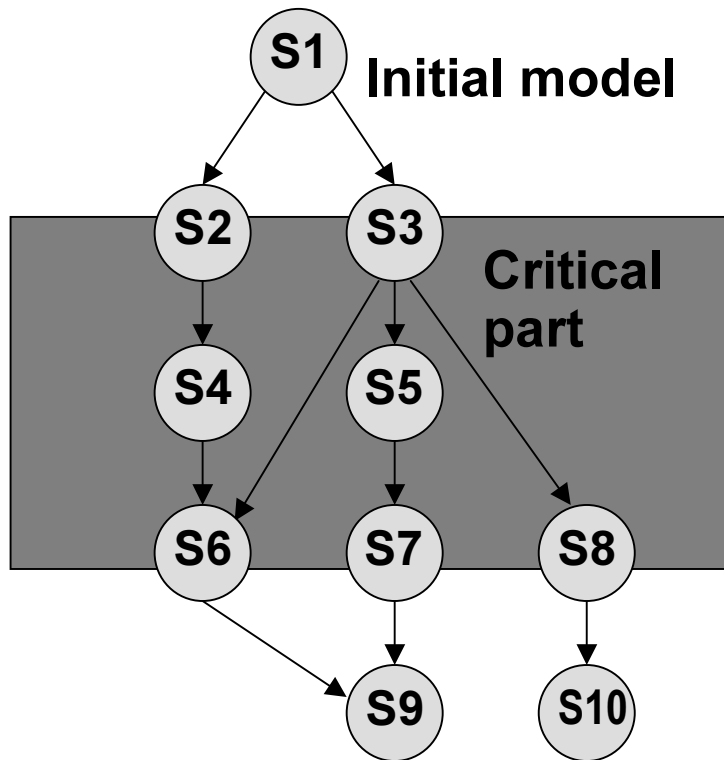
4 - Validation a priori - Hybrid Method



4 - Validation a priori - Hybrid Method

Critical parts - partial models definition

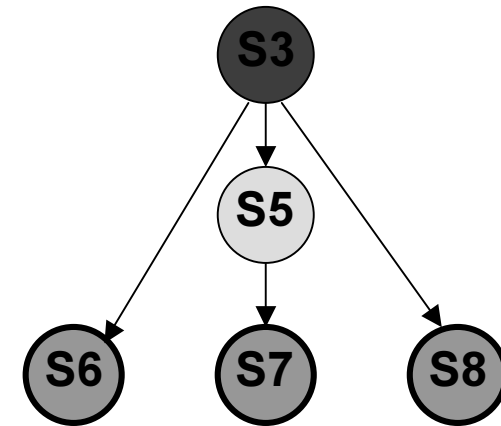
- 1 and only 1 input state
- at least one output states
- 2 partial models cannot be overlapped



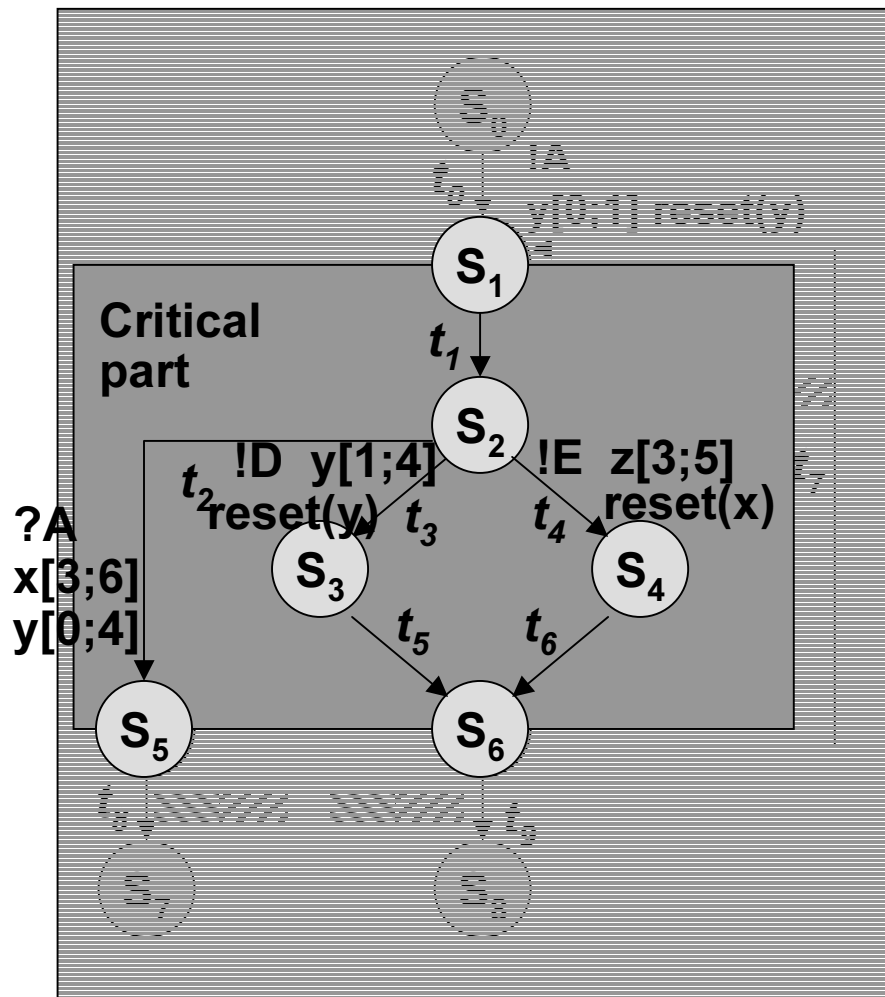
Partial model
 PM_1



Partial model
 PM_2

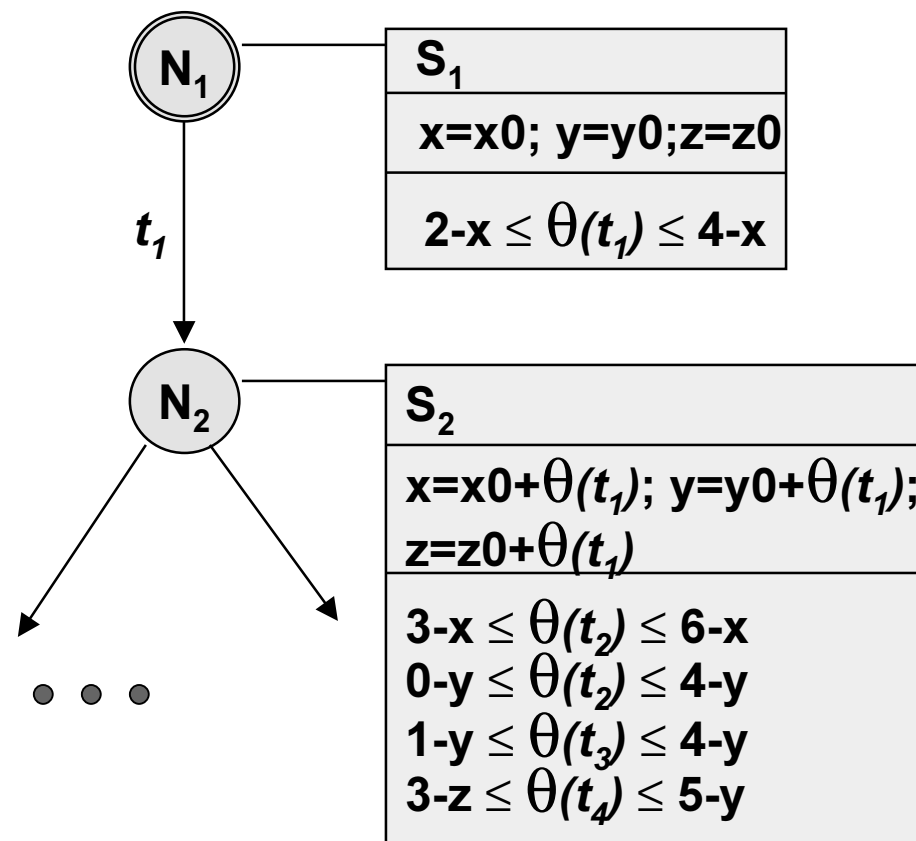


4 - Validation a priori - Hybrid Method



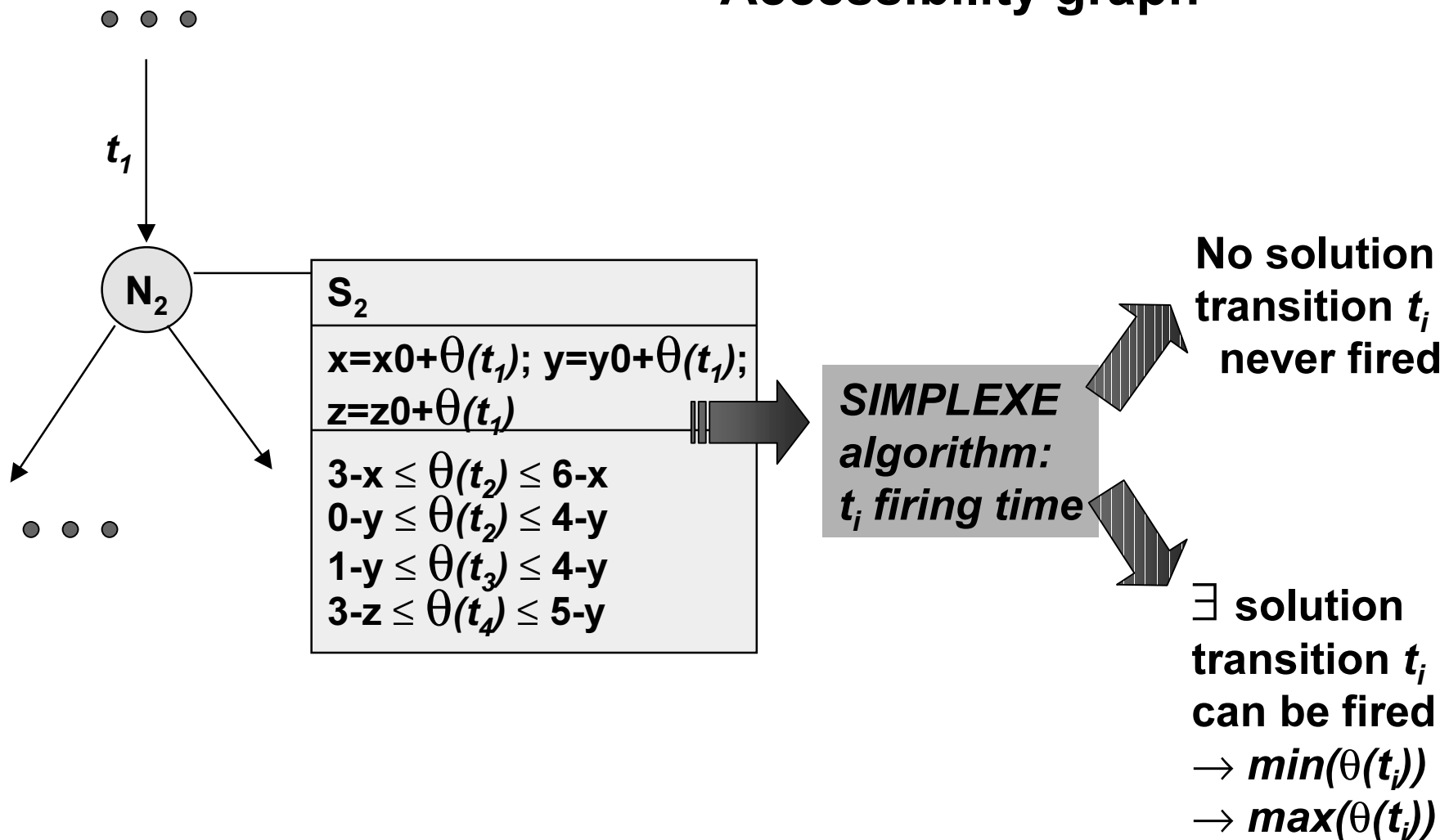
Accessibility graph

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



4 - Validation a priori - Hybrid Method

Accessibility graph

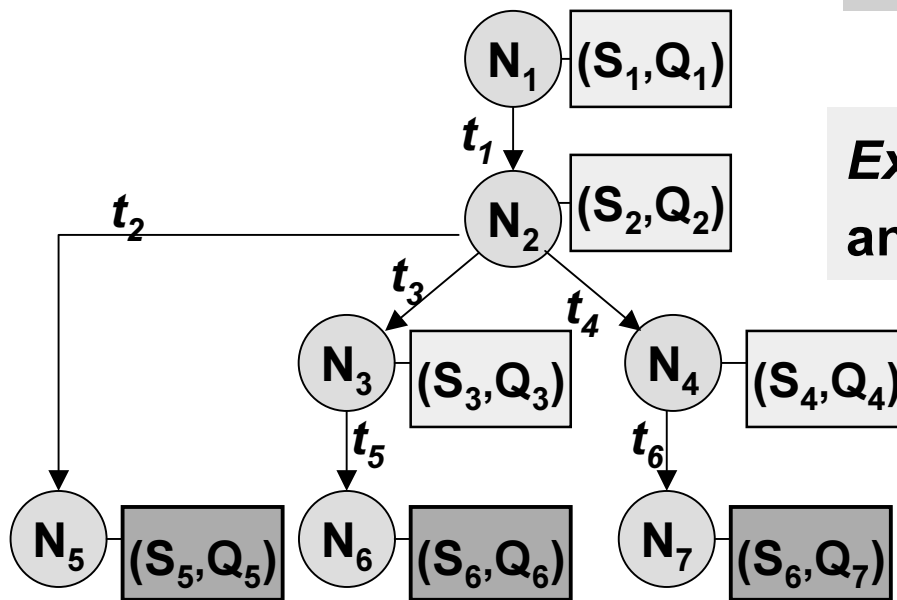


4 - Validation a priori - Hybrid Method

Accessibility graph

Exit states of partial model : {S5, S6}

Exit Nodes : node characterized with an exit state



Every exit node can be reached

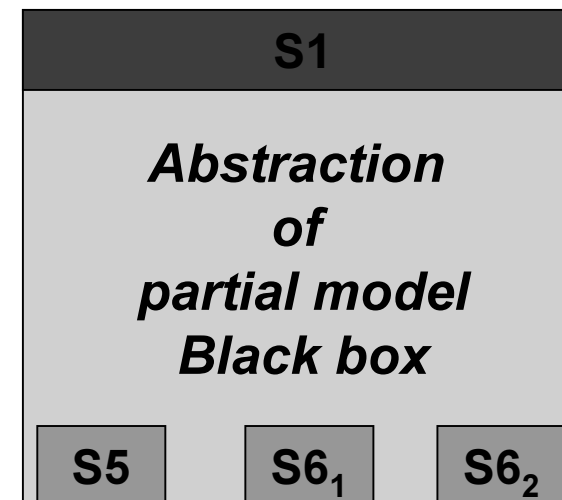
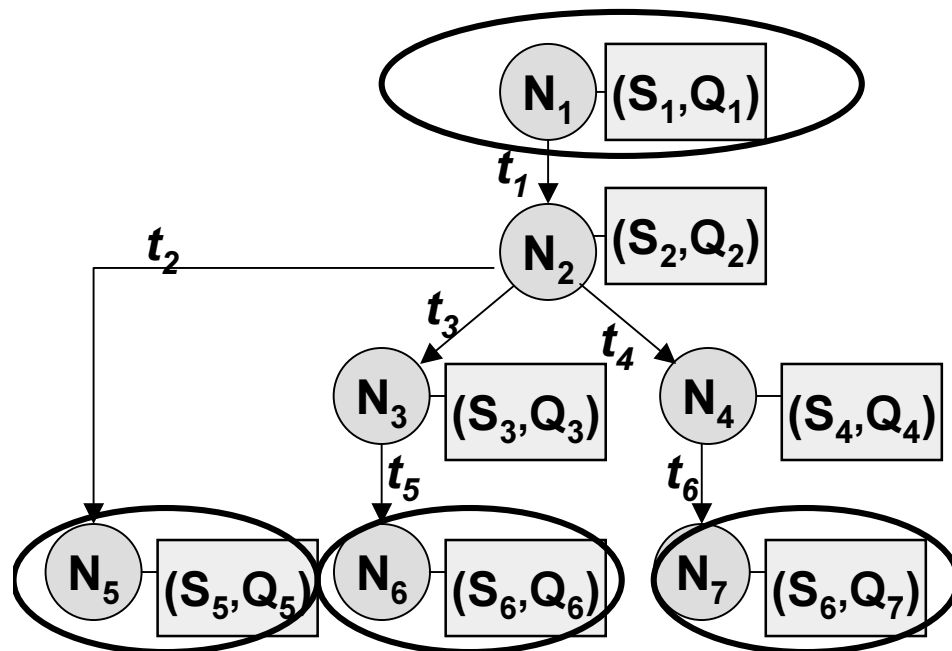
➔ **Success**

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 : $S_{i,l}$

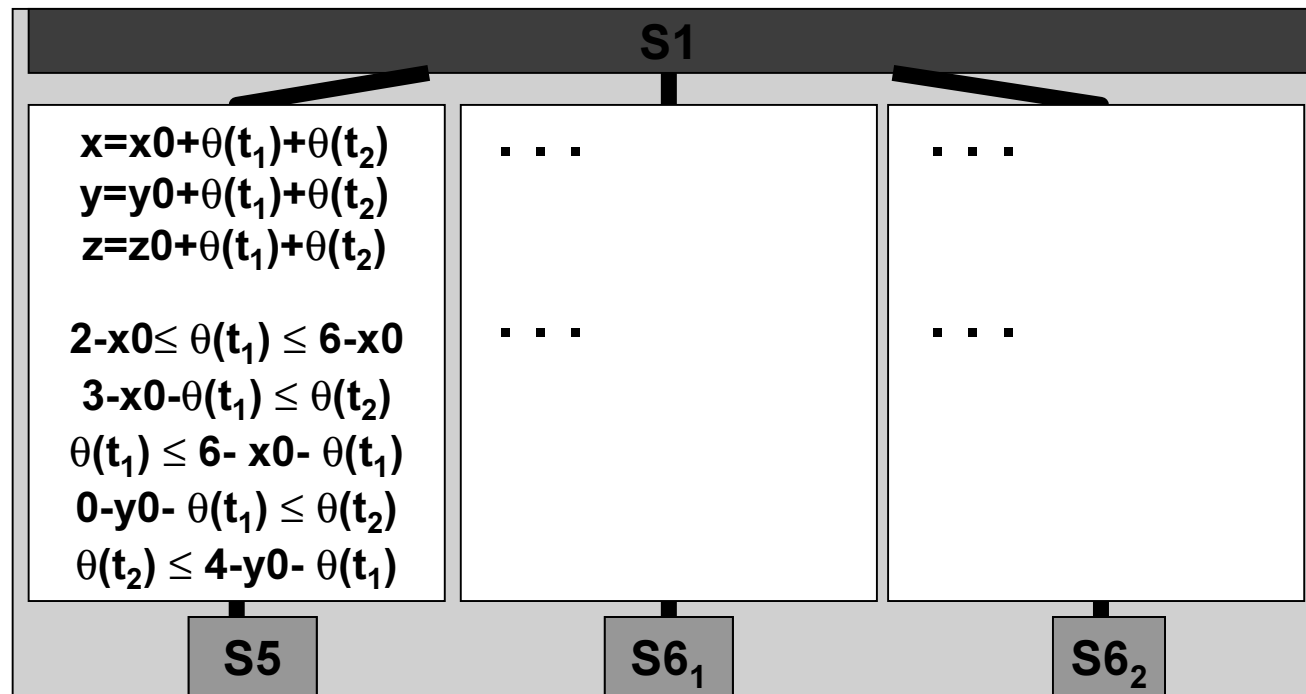


4 - Validation a priori - Hybrid Method

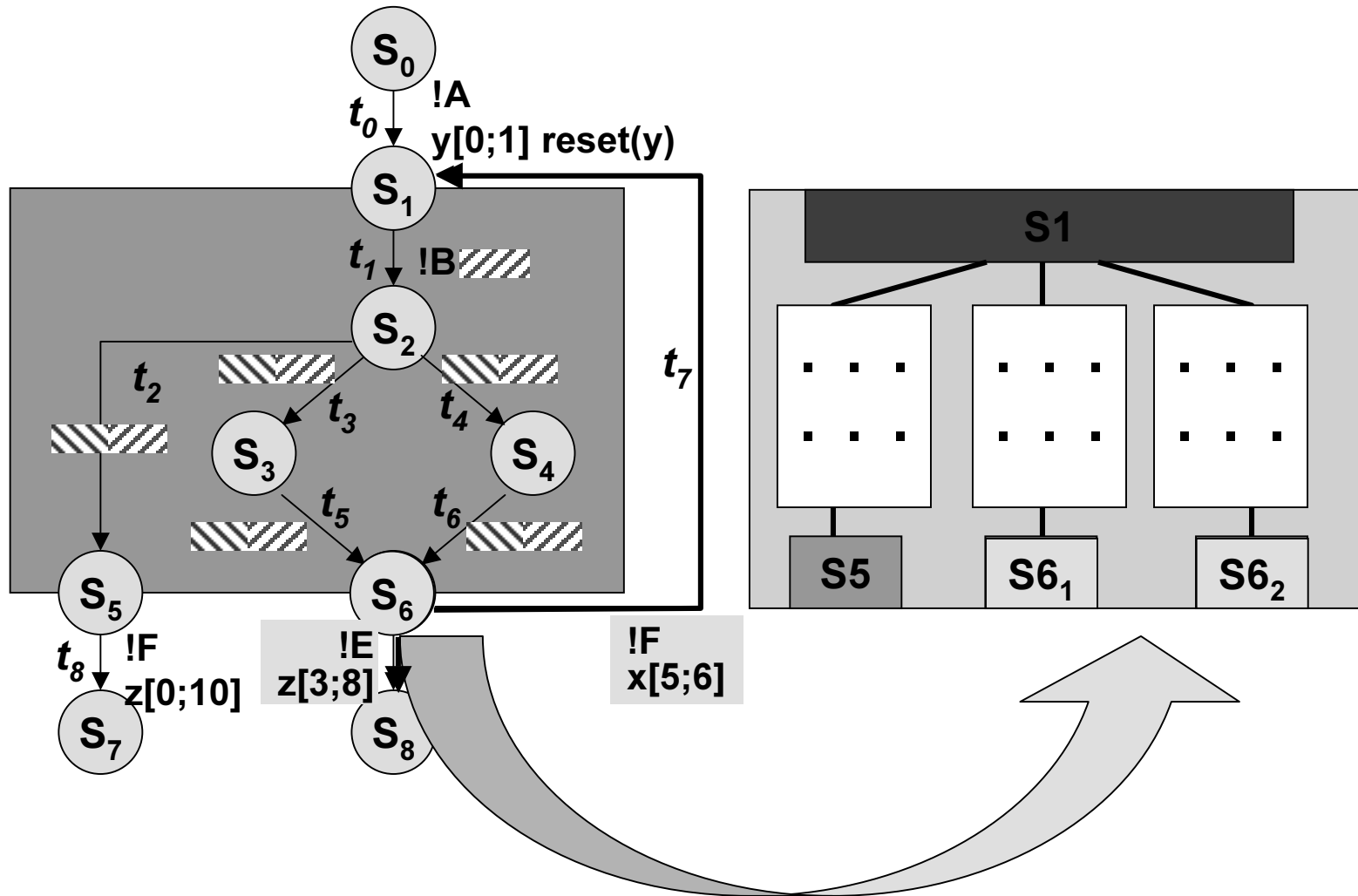
For each exit state S

Value of each clock when reaching this state

Constraints on clocks along the path from init state to S

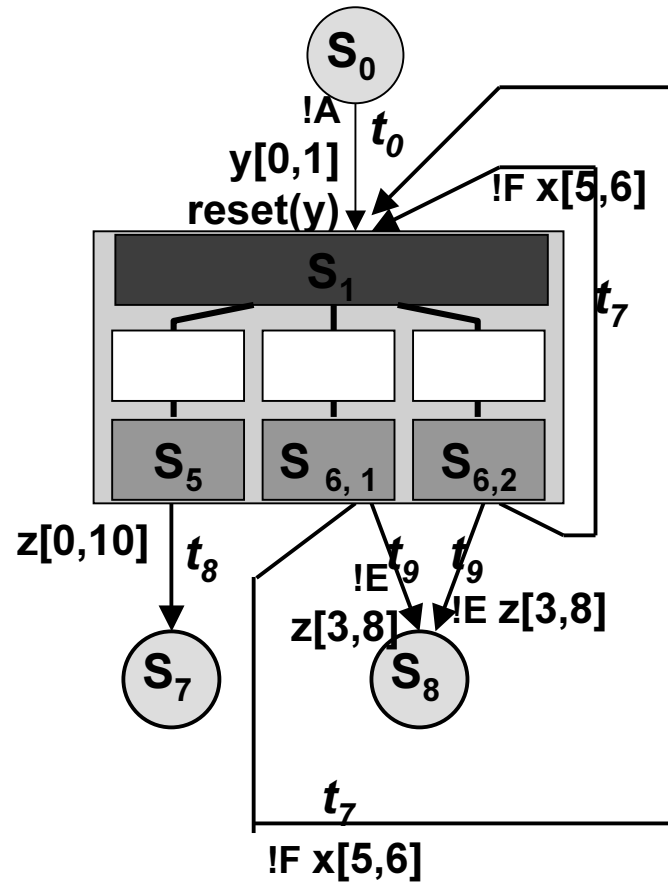


4 - Validation a priori - Hybrid Method

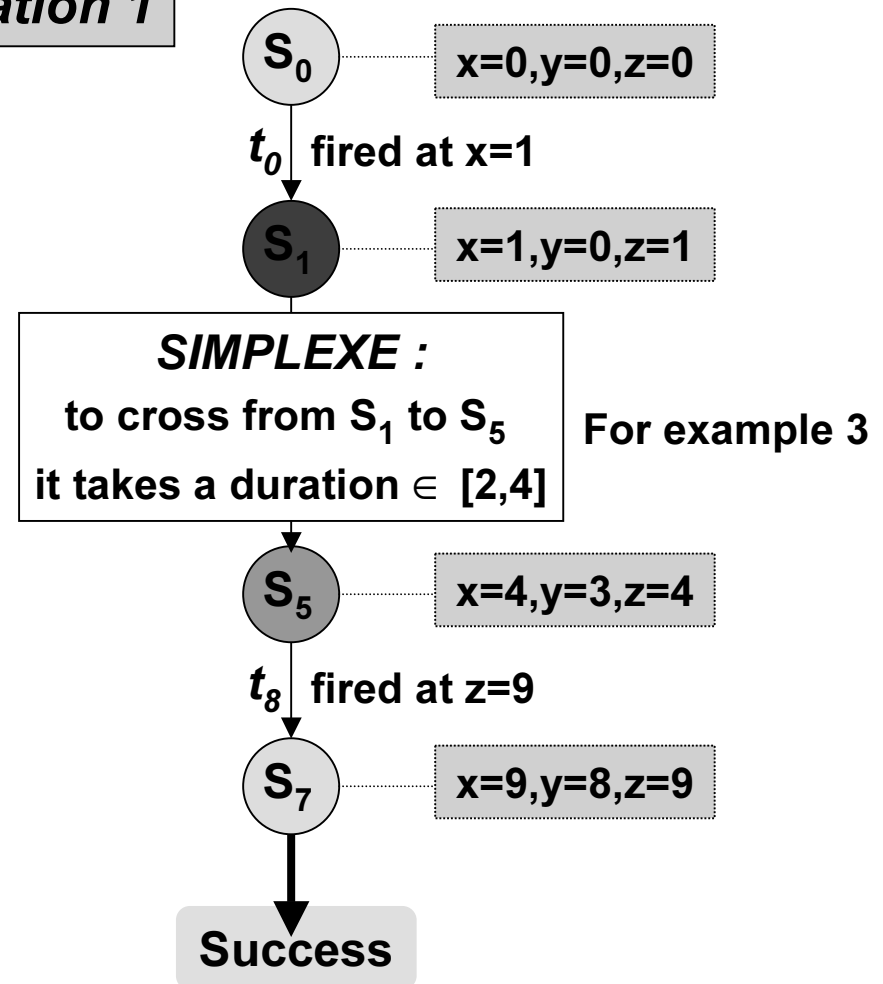


4 - Validation a priori - Hybrid Method

Derived model

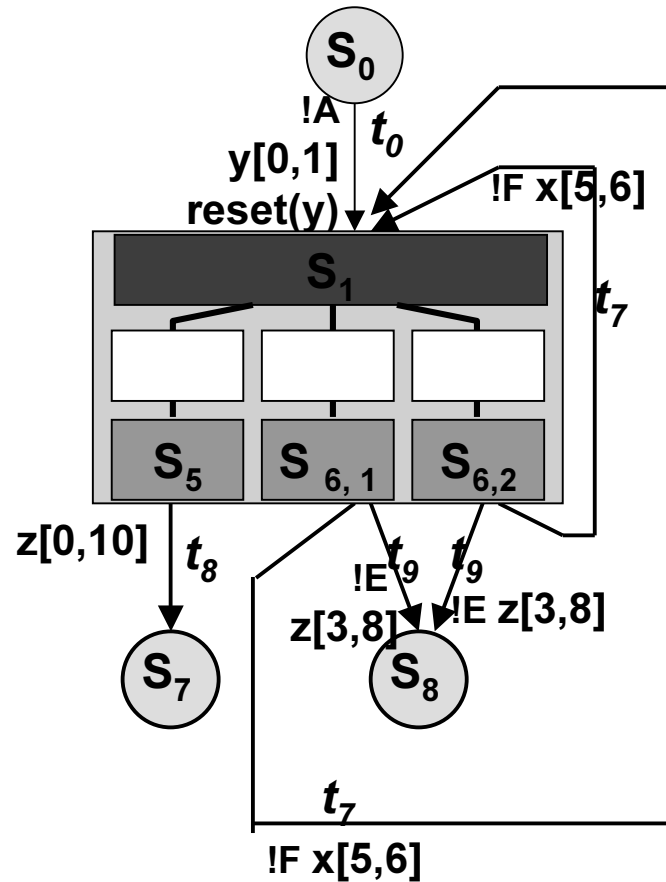


Simulation 1

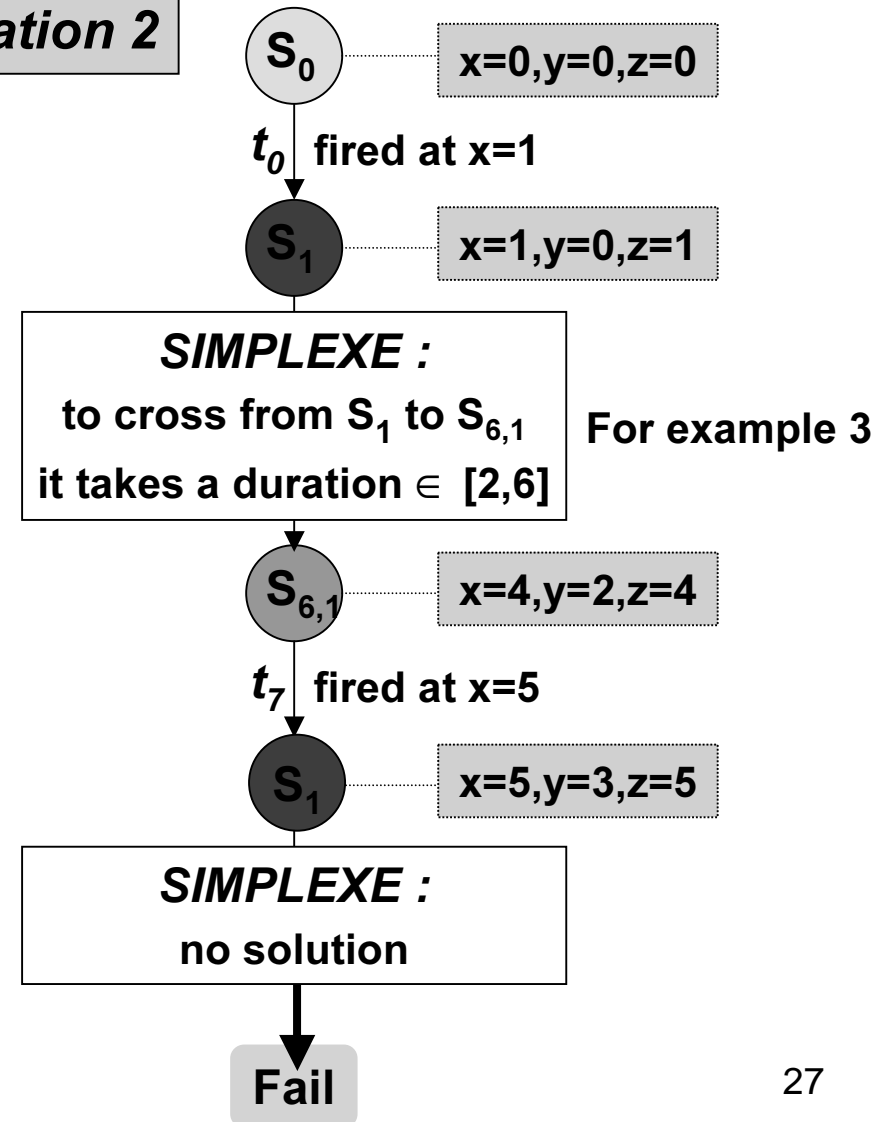


4 - Validation a priori - Hybrid Method

Derived model



Simulation 2

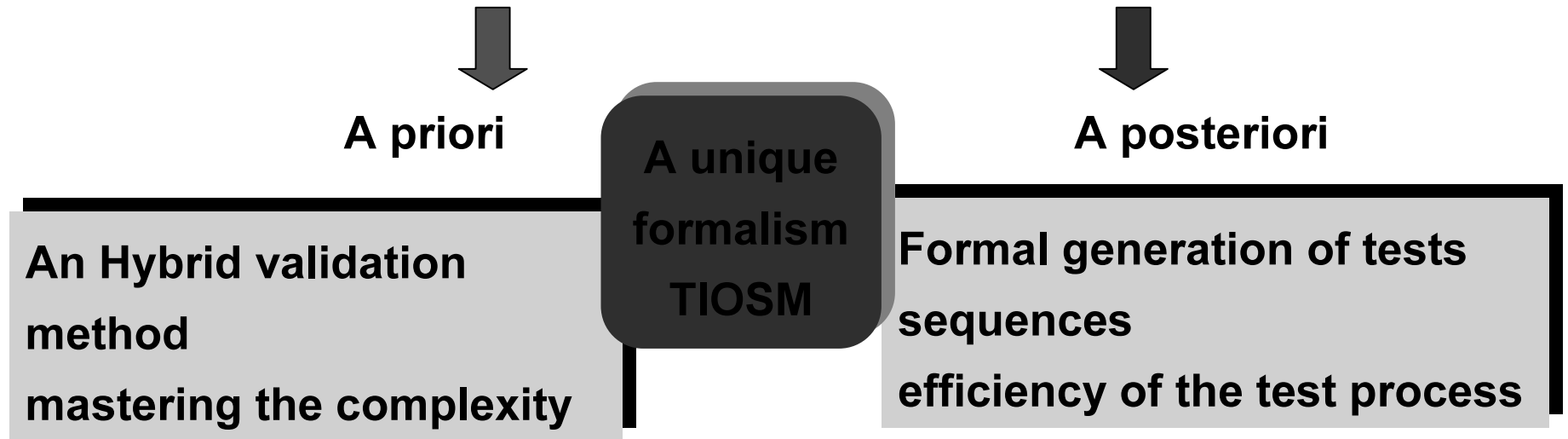


4 - Validation a priori - Hybrid Method

- **Conclusions**

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

5 - Validation of Real Time Systems



Future trends

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