

# Proving the Properties of Communicating Imperfectly-Clocked Synchronous Systems

**Julien Bertrane**, [bertrane@di.ens.fr](mailto:bertrane@di.ens.fr)

École Normale Supérieure, Paris, France

August 31st, 2006

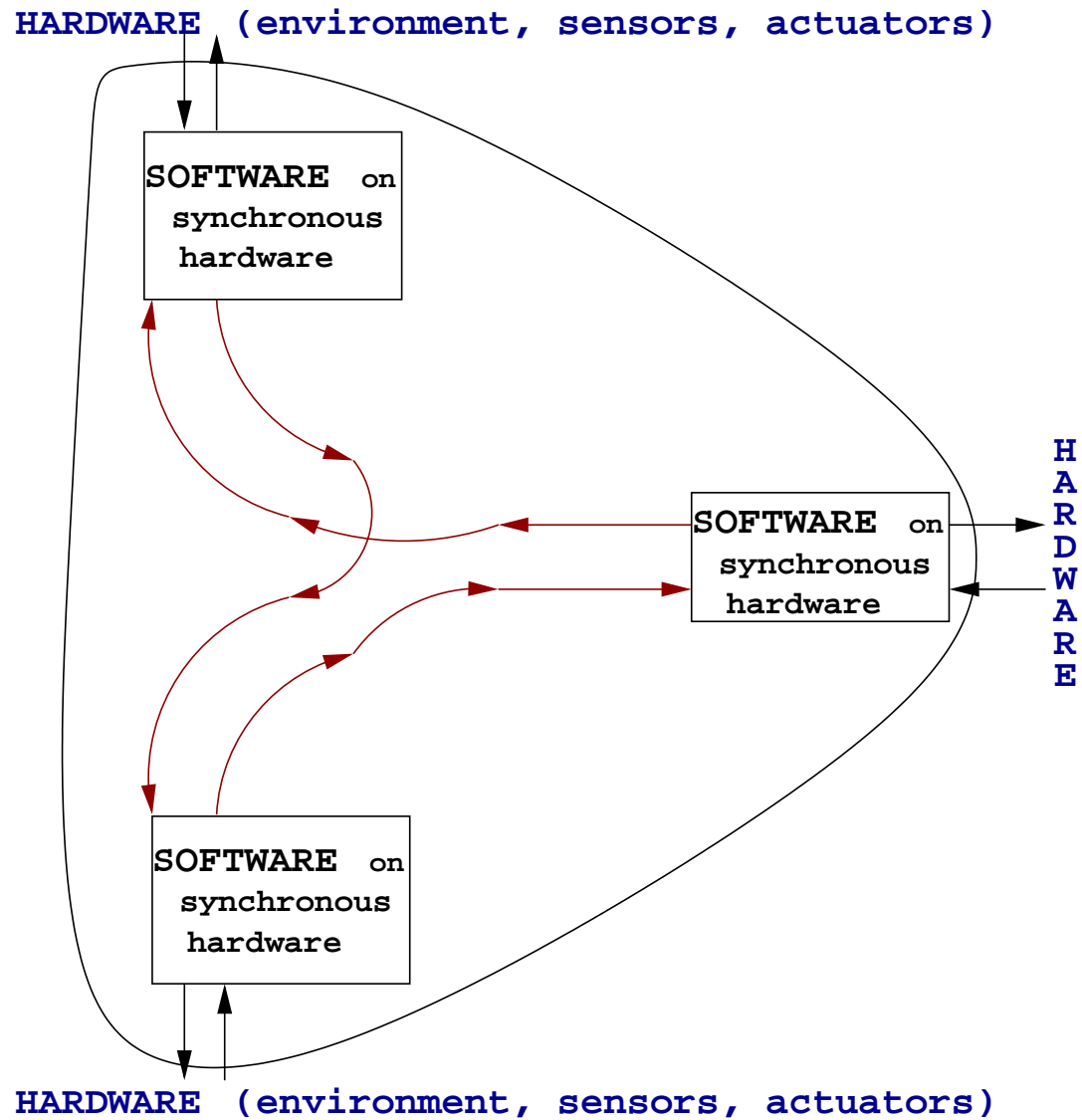
# Goals

---

- Proving the correctness of a system with respect to a formal specification.
- But “system”  $\triangleq$  set of several communicating synchronous programs (each one with its own clock)
- Why consider multiclock systems ?
  - Some embedded systems are too big for only one clock : information flow between any two points is too slow
  - Critical systems often require redundancy (to prevent system physical failure of the unit  $\Rightarrow$  several clocks)

# Typical system

---



# Difficulties and subsequent hypotheses

---

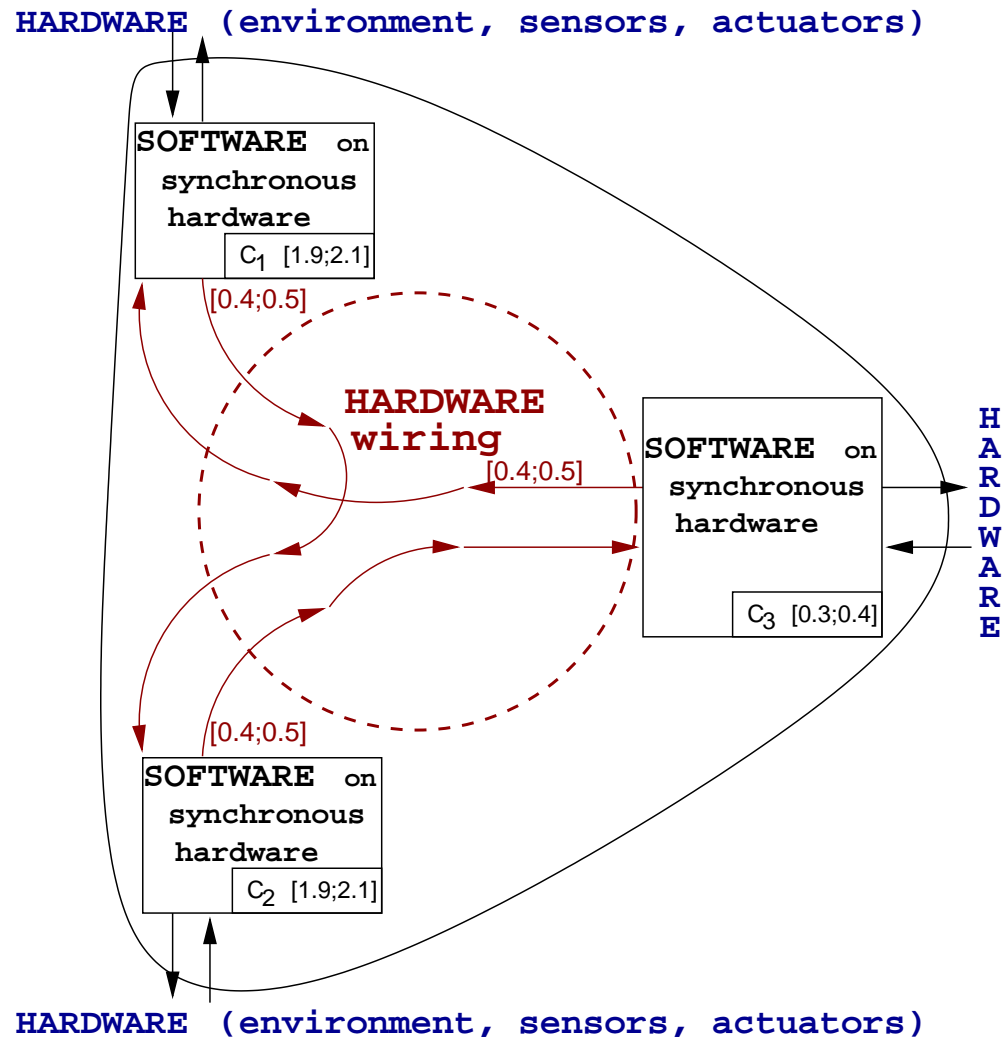
Framework includes realistic executions issues :

- Clock **desynchronization** allowed
- **Non-constant delays** during communications
- **Graphical** syntax : close to SCADE syntax

Simplifications :

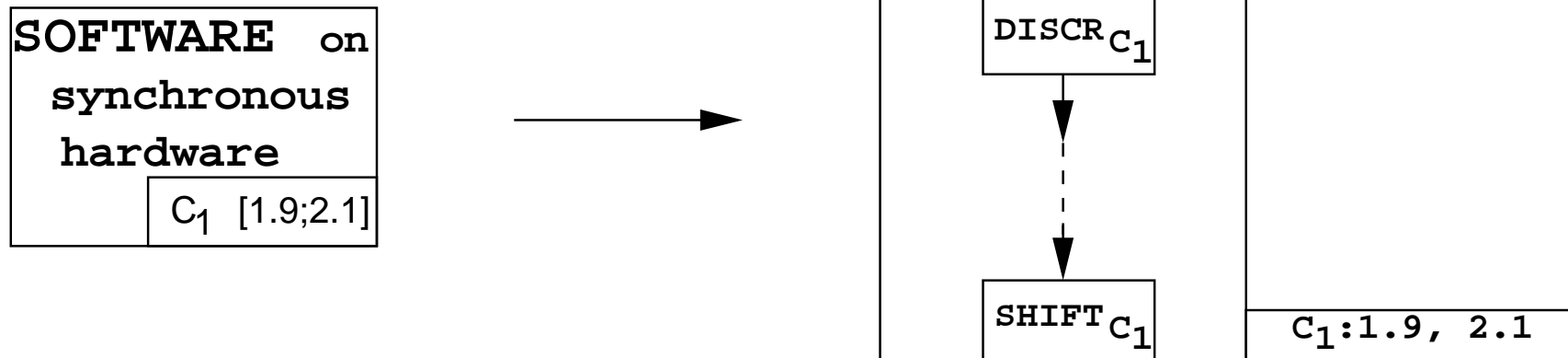
- Variables presently only considered **booleans**
- One-value only **buffers** for synchronous units input
- **Serial transmission** between synchronous systems

# Typical system :details



# Behavior of a synchronous system

---



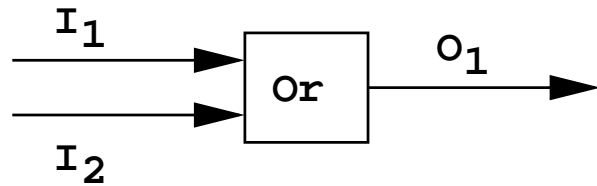
- a clock is a function  $c : \mathbb{N} \rightarrow \mathbb{R}^+$
- clock parameter  $[\alpha, \beta]$ , with  $\alpha, \beta \in \mathbb{R}^+$  and  $0 < \alpha \leq \beta$
- a clock  $c$  satisfies  $[\alpha, \beta]$  iff  $c_{n+1} - c_n \in [\alpha, \beta]$
- $\text{DISCR}_{C_1}$  models the periodic reading of the input buffer
- $\text{SHIFT}_{C_1}$  models the waiting for the next clock tick, and the emission of its result at this next clock tick

# Semantics : choices

---

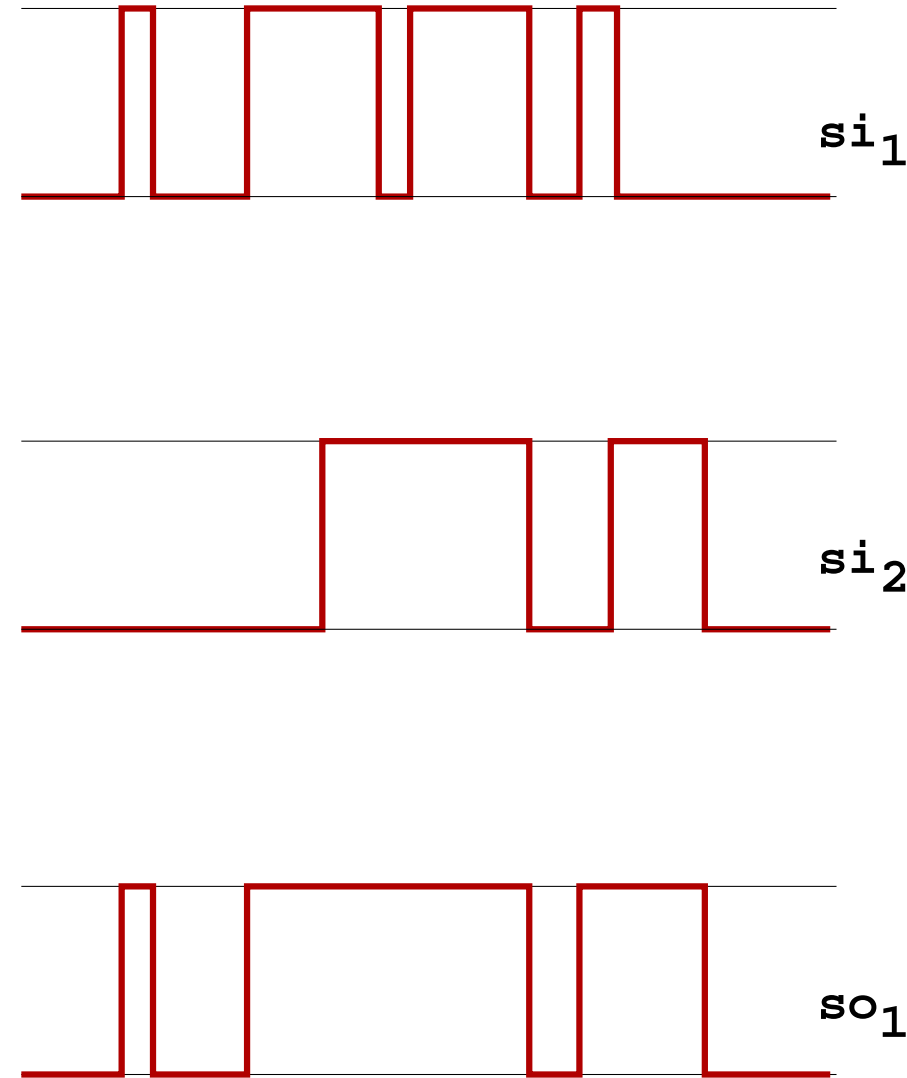
- the semantics connects each point of control to a set of **signals** (i.e. element of  $f : \mathbb{R}^+ \rightarrow \mathbb{B}$ )
- a signal belongs to the semantics at point  $p$  if there is a vector connecting each any point but  $p$  to a signal **compatible** with  $p$ .
- if no-empty, the semantics often contains a non-countable infinity of signals

# Semantics of time-independent operators

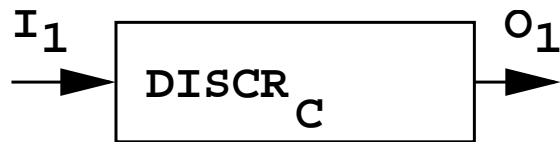


$$so_1(t) = \begin{cases} \bullet \text{ true} \\ \text{if } si_1(t) = \text{true} \\ \text{or } si_2(t) = \text{true} \\ \bullet \text{ false else} \end{cases}$$

$$so_1 \triangleq \Psi_{\text{OR}}(si_1, si_2)$$



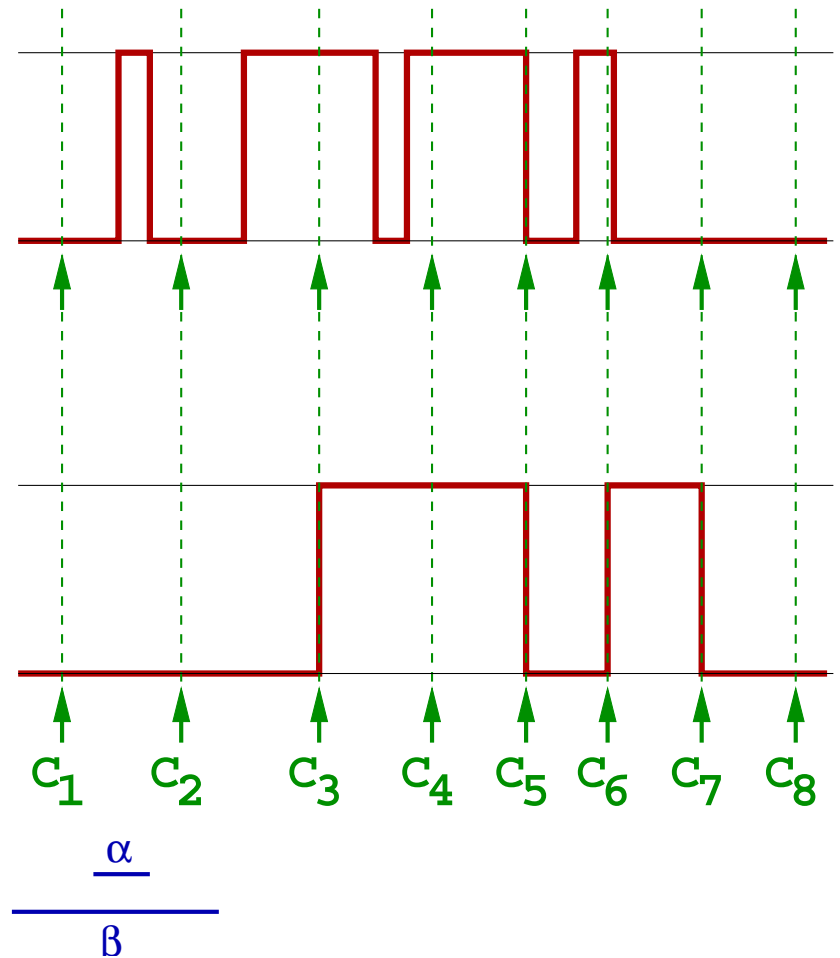
# Semantics of time-dependent operators



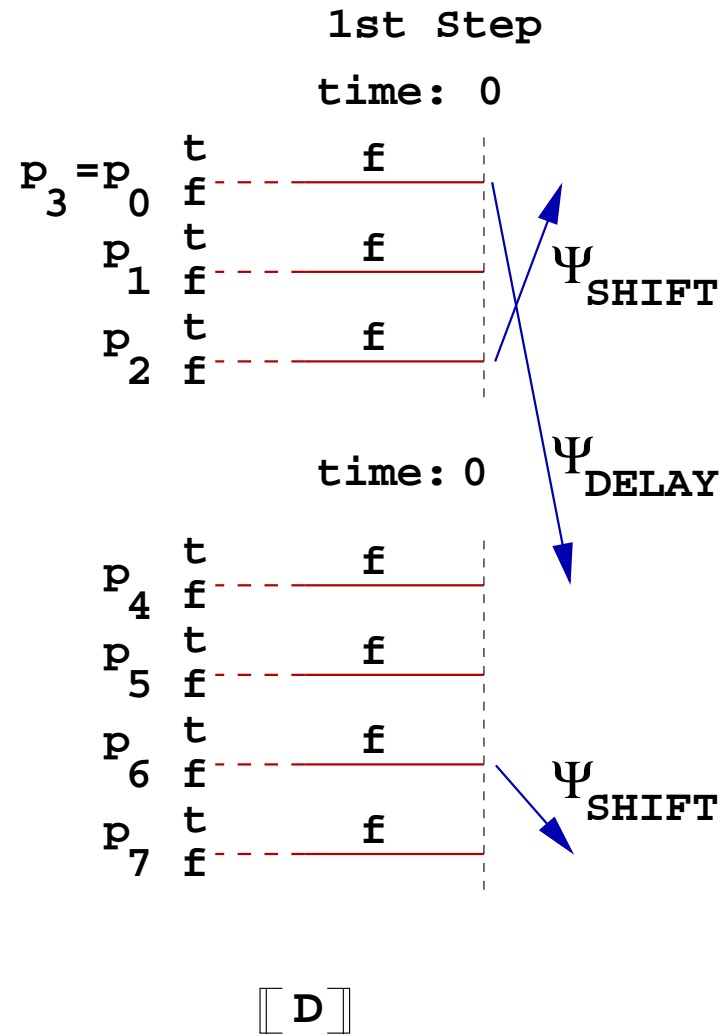
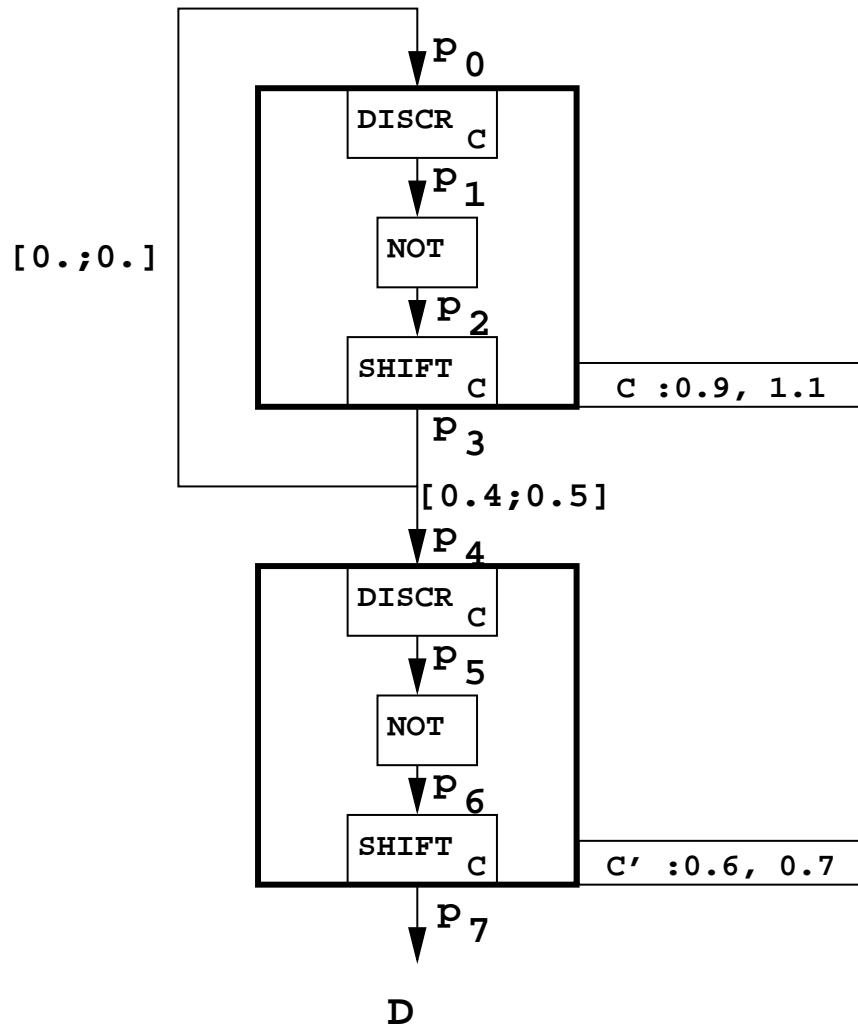
$[\alpha, \beta]$  parameter of clock  $C$

$$so_1(t) = \begin{cases} \bullet \text{ false} & \text{if } t < c(0) \\ \bullet si_1(c_n) & \text{if } t \in [c_n, c_{n+1}) \end{cases}$$

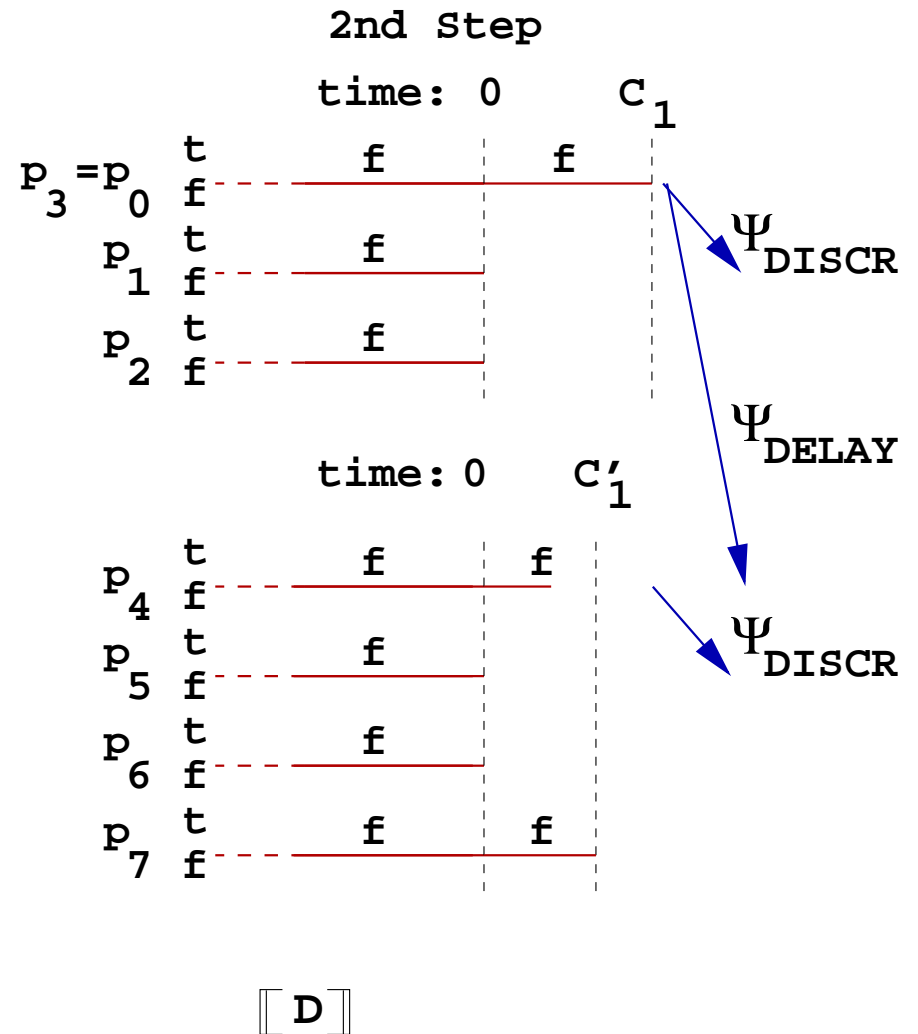
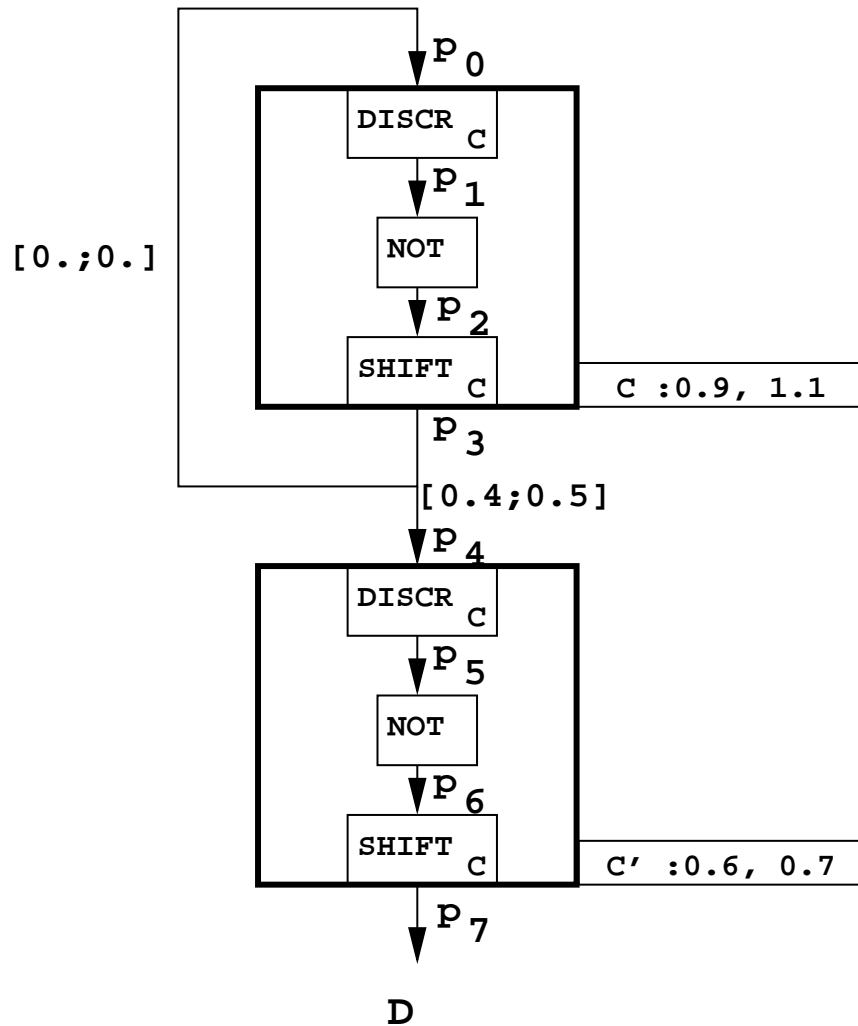
$$so_1 \triangleq \Psi_{DISCR_c}(si_1)$$



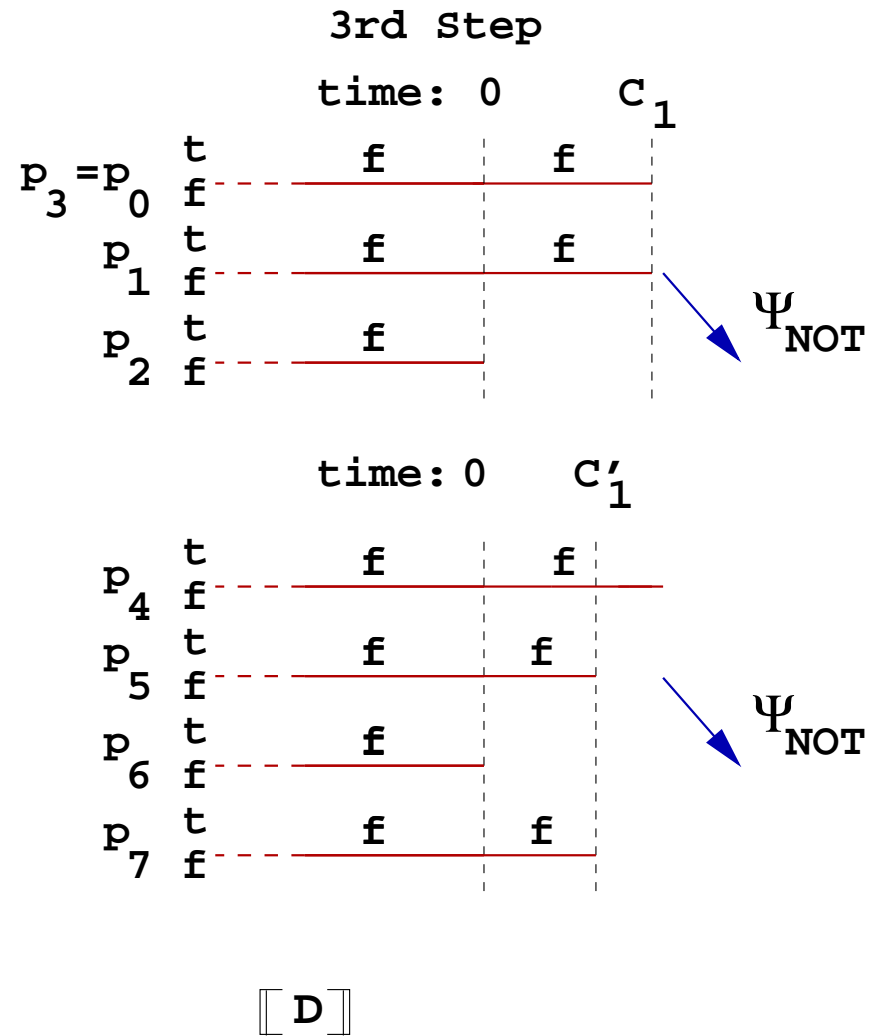
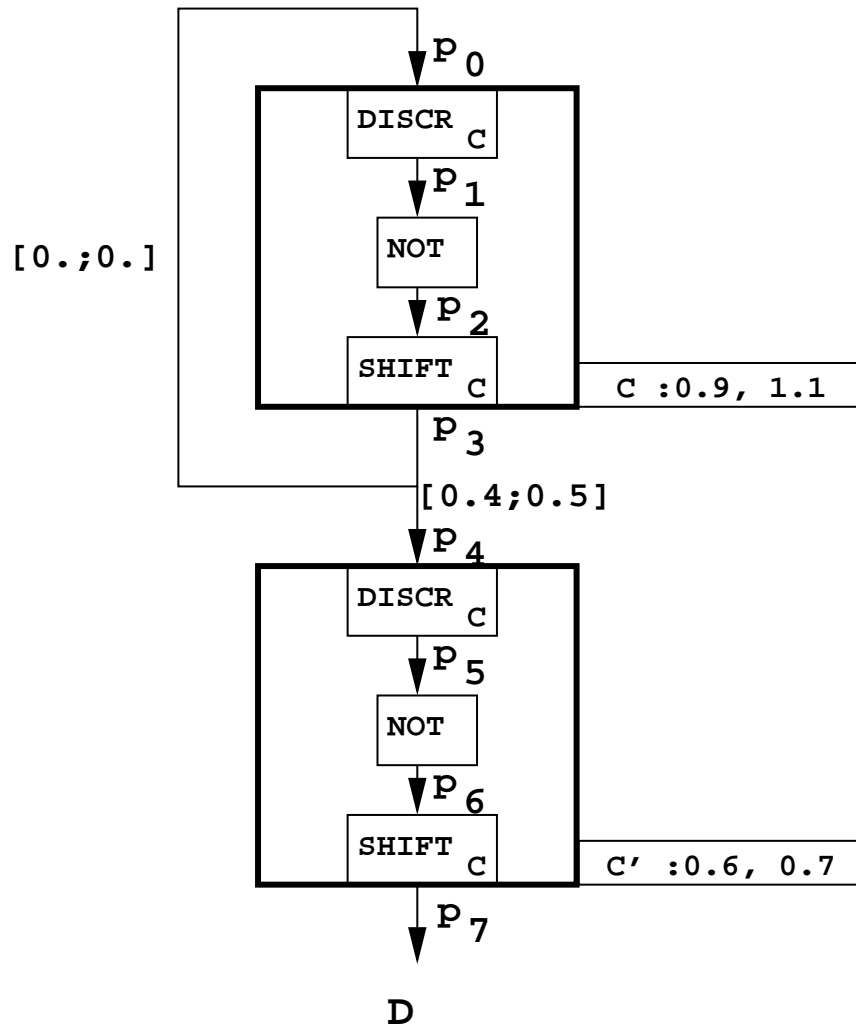
# Syntax and semantics



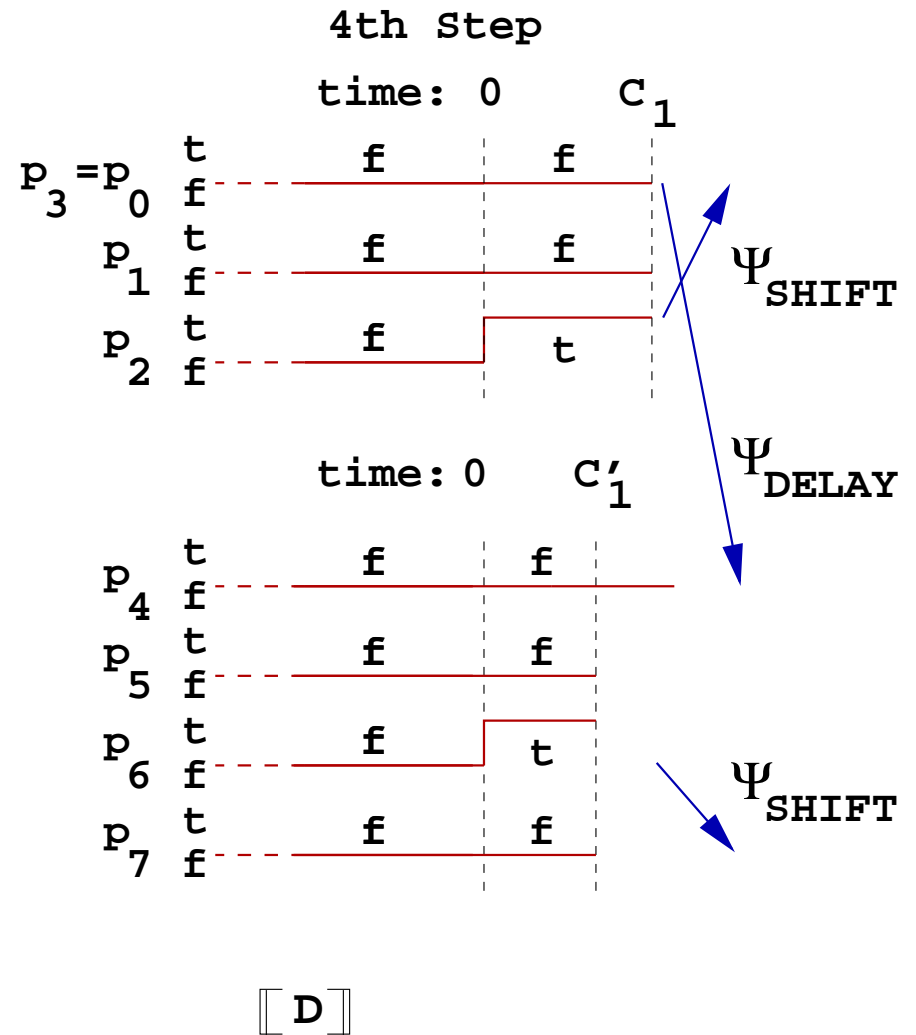
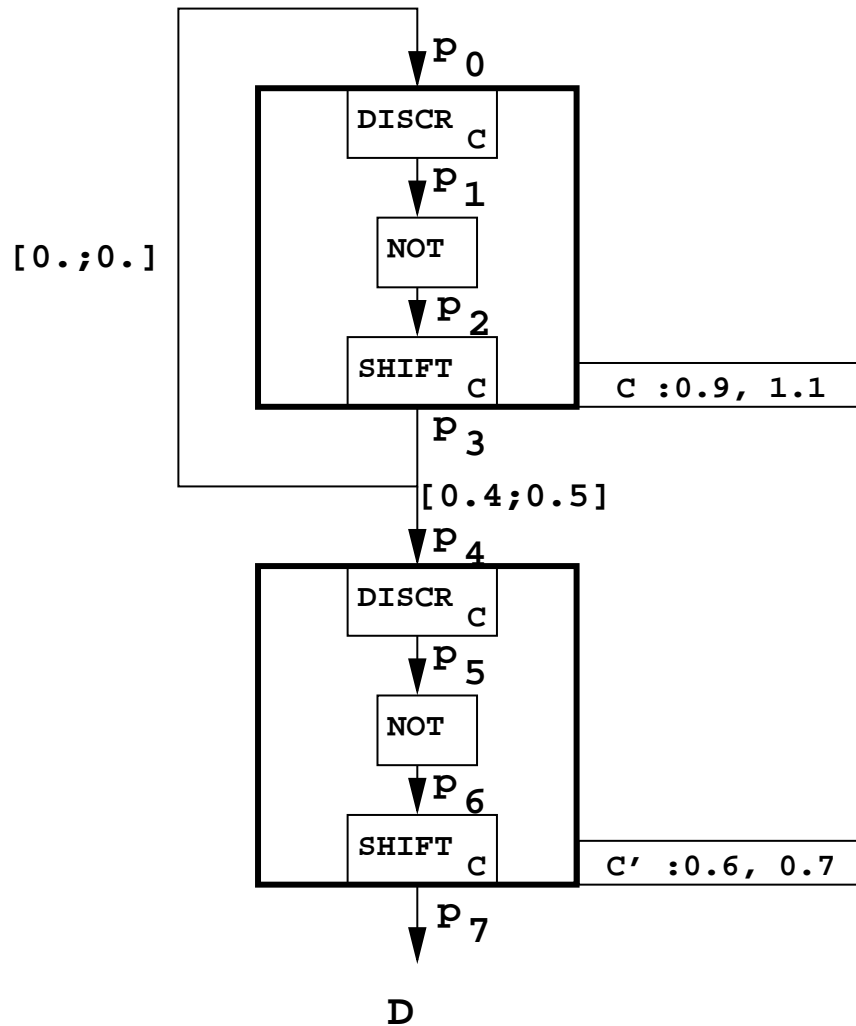
# Syntax and semantics



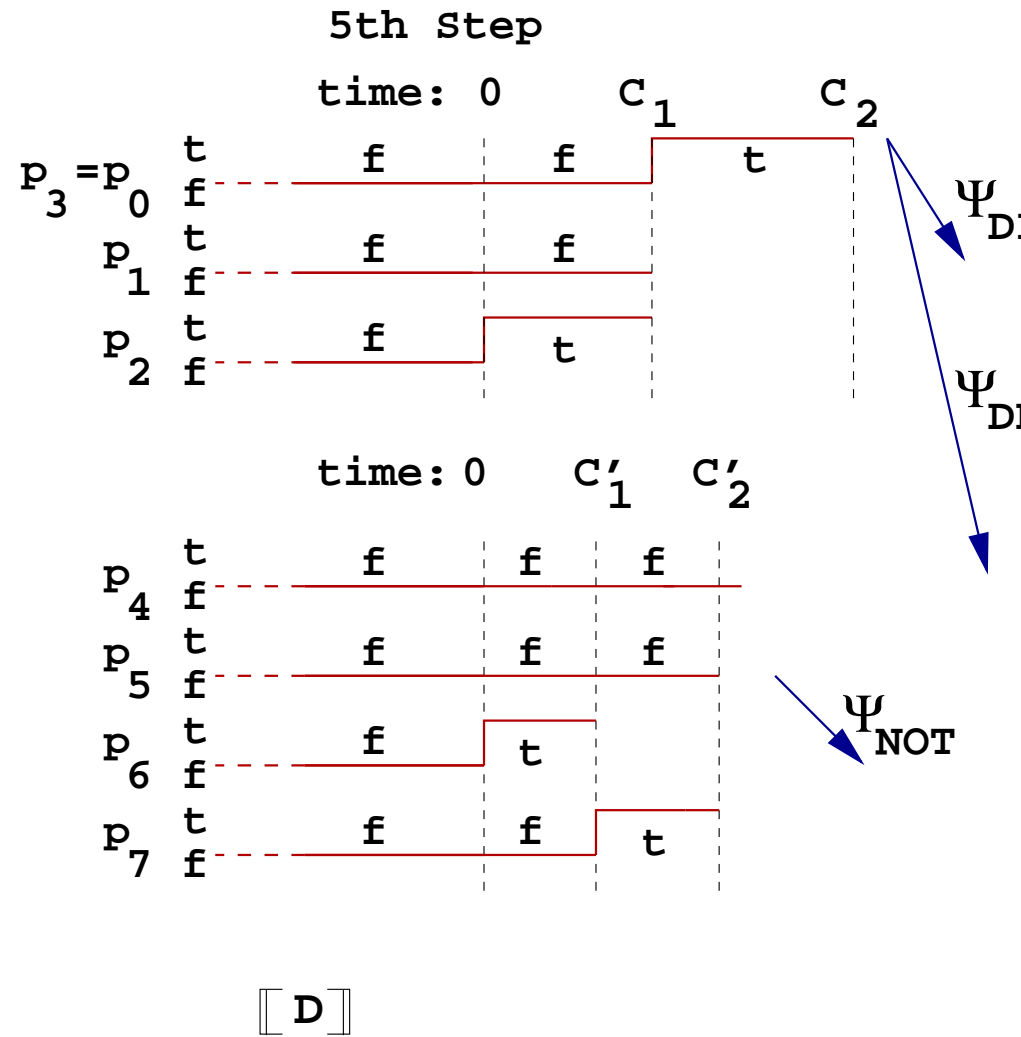
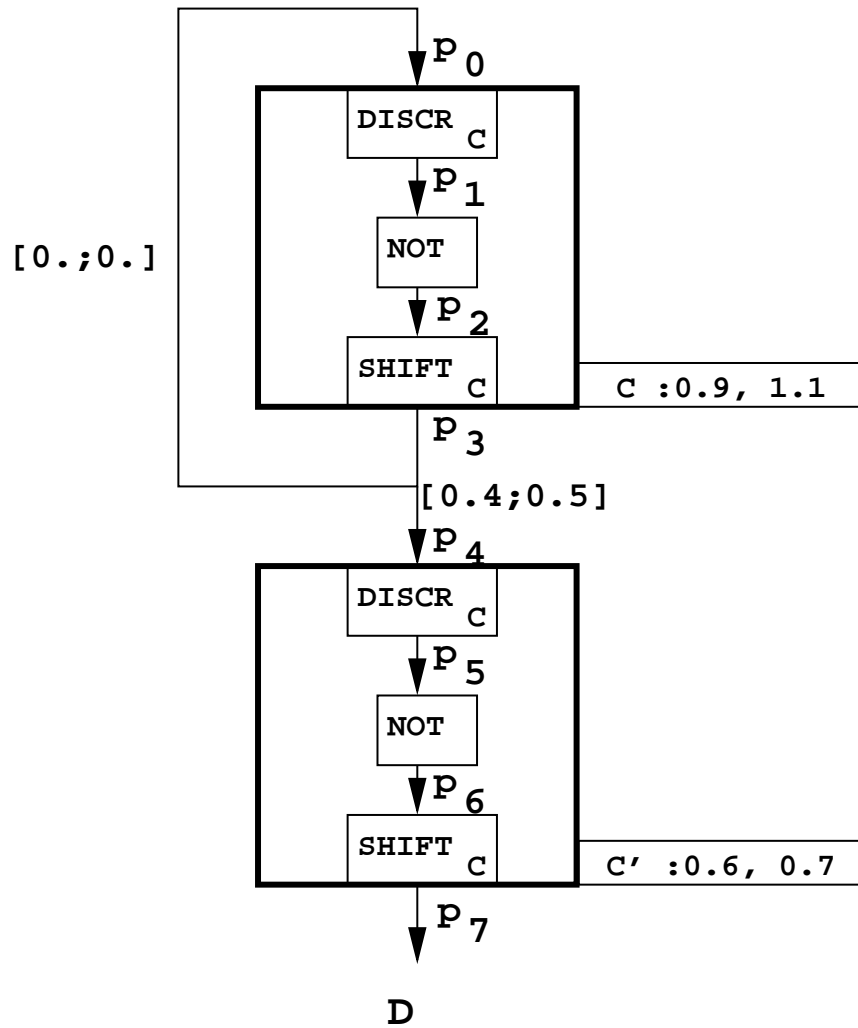
# Syntax and semantics



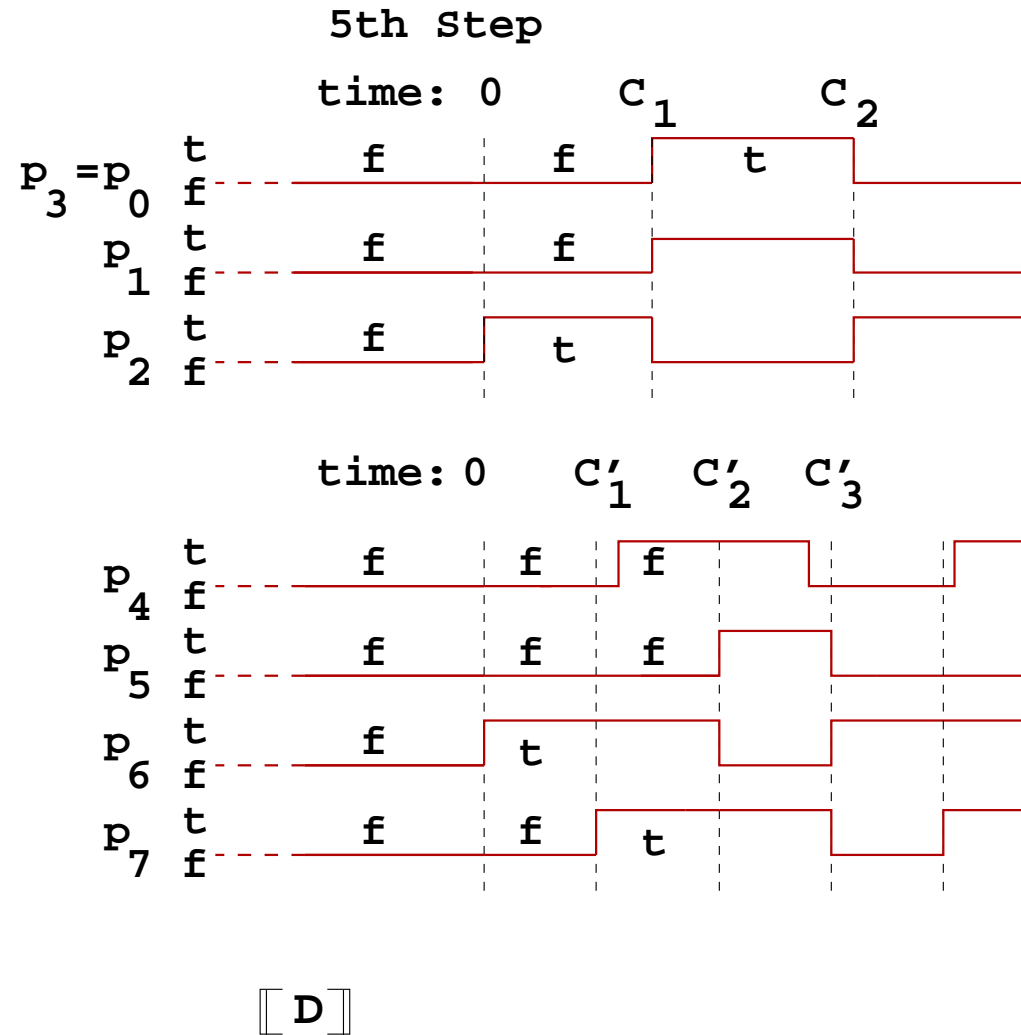
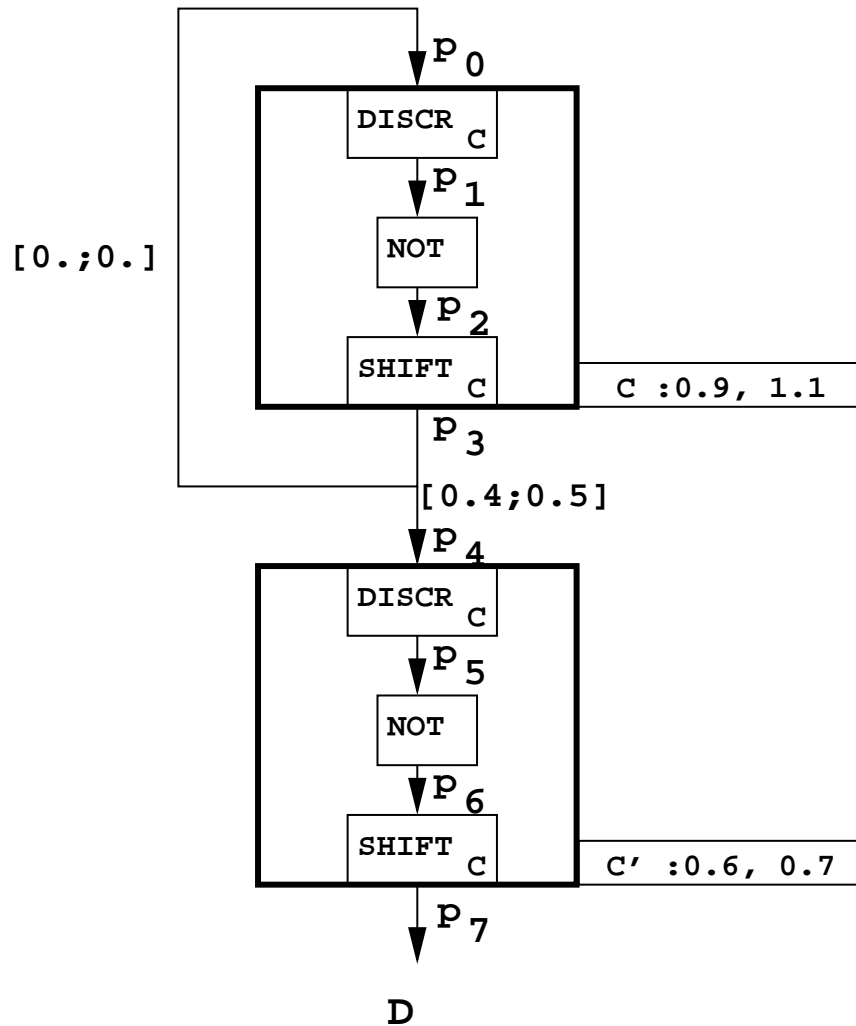
# Syntax and semantics



# Syntax and semantics



# Syntax and semantics



# Abstract interpretation based analysis

---

- $\llbracket D \rrbracket$  is the semantics of a set  $D$  of systems.
- $[P]$  is the set of behaviors satisfying a property  $P$ .
- **Former goal** : Prove that  $\llbracket D \rrbracket \subseteq [P]$ .
- **Now** :  $(\Psi \cap Id)(\llbracket D \rrbracket \cap [\neg P]) \subseteq \llbracket D \rrbracket \cap [\neg P]$
- **Thus** :

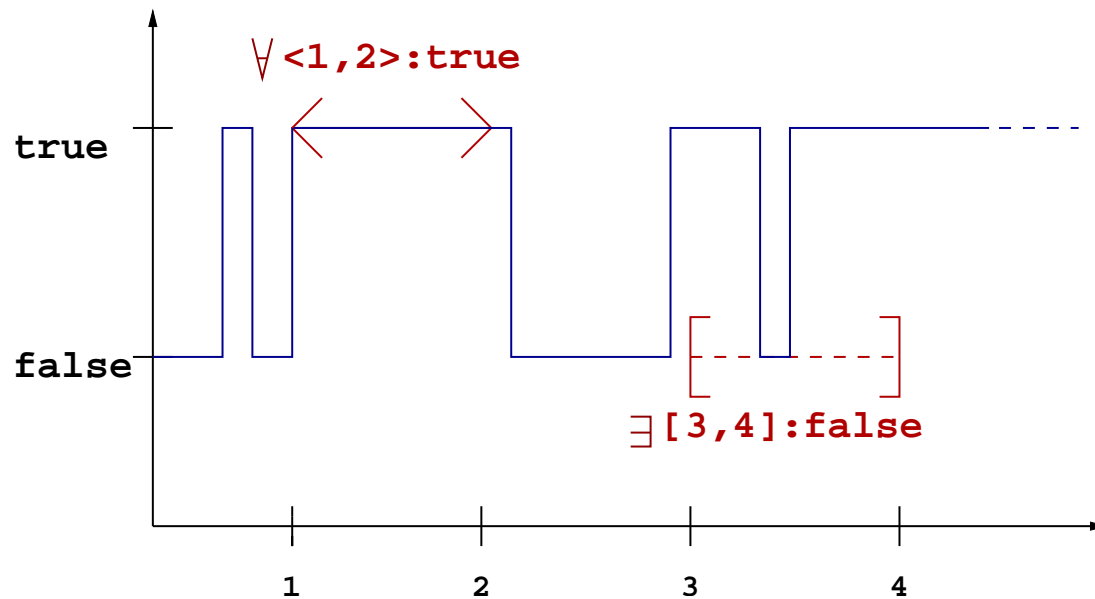
$$\llbracket D \rrbracket \cap [\neg P] \subseteq \text{gfp}_{[\neg P]}(\Psi \cap Id) \subseteq? \emptyset$$

- **True if (not iff)** :

$$\text{gfp}_{[\neg P]}^{\#}(\Psi \cap Id) \subseteq^{\#} \emptyset^{\#} = \perp$$

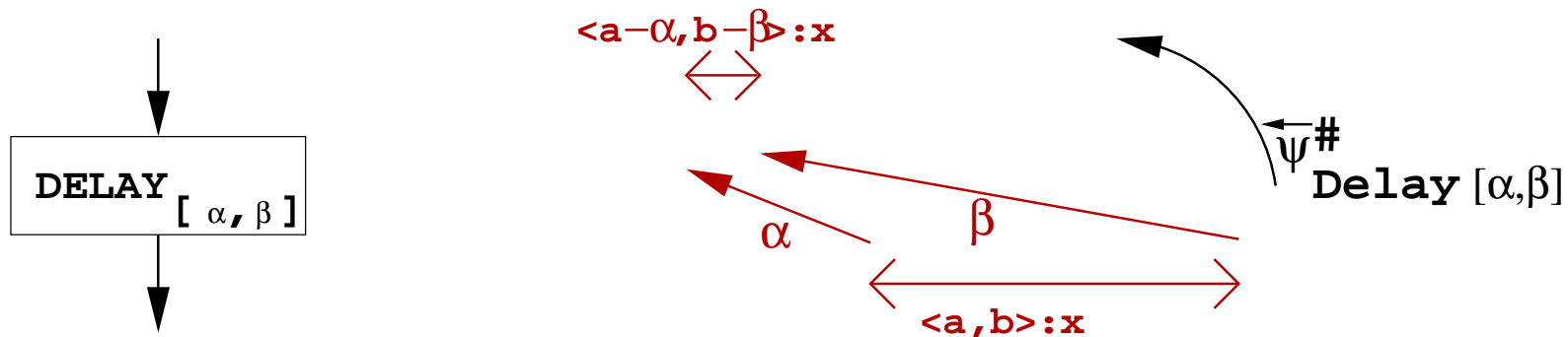
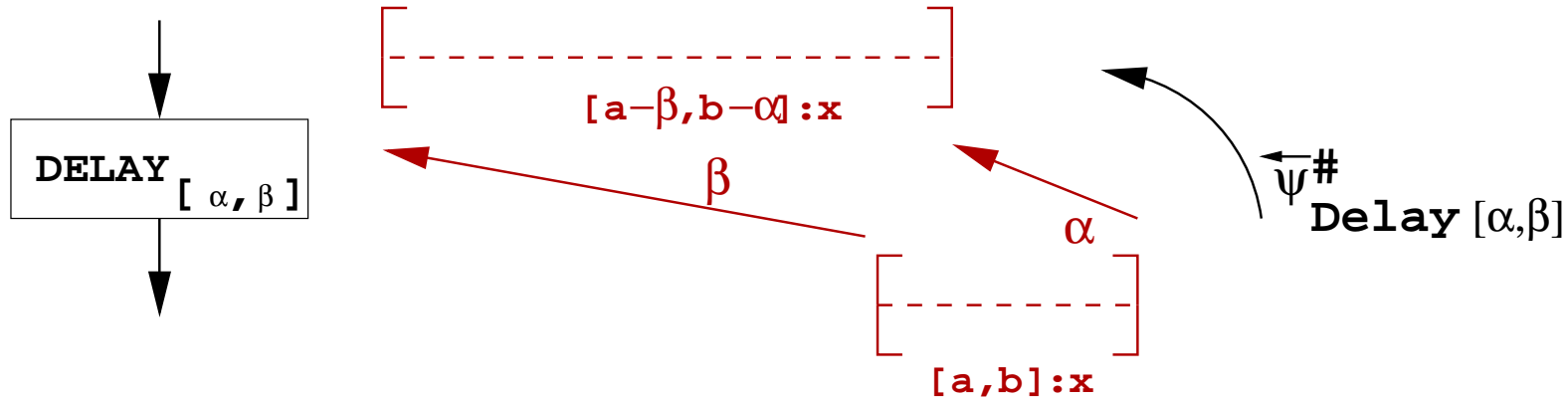
# 1st abstract domain

---



- A constraint  $\exists [a; b] : x$  forces signals to be equal to  $x$  **at least once** during  $[a; b]$ .
- A constraint  $\forall \langle a; b \rangle : x$  forces signals to be equal to  $x$  **during the whole**  $[a; b]$ .

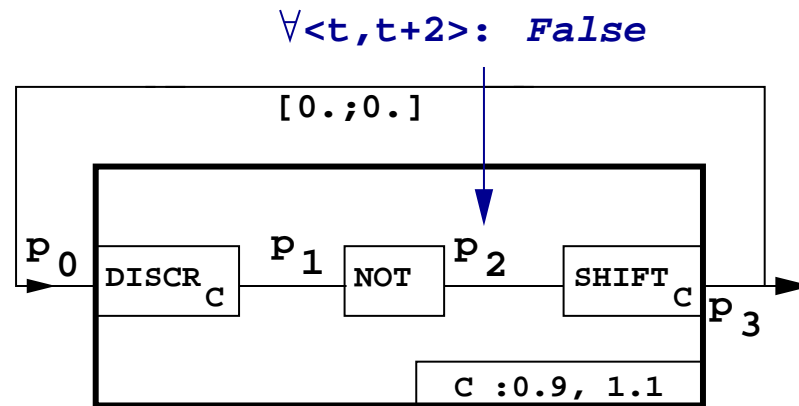
# Abstract Operators and Constraints : an example



- $\overleftarrow{\Psi}_{\text{DELAY}[\alpha, \beta]}^{\#} (\exists [a; b] : x) \triangleq \exists [a - \beta; b - \alpha]$
- $\overleftarrow{\Psi}_{\text{DELAY}[\alpha, \beta]}^{\#} (\forall \langle a; b \rangle : x) \triangleq \forall \langle a - \alpha; b - \beta \rangle$

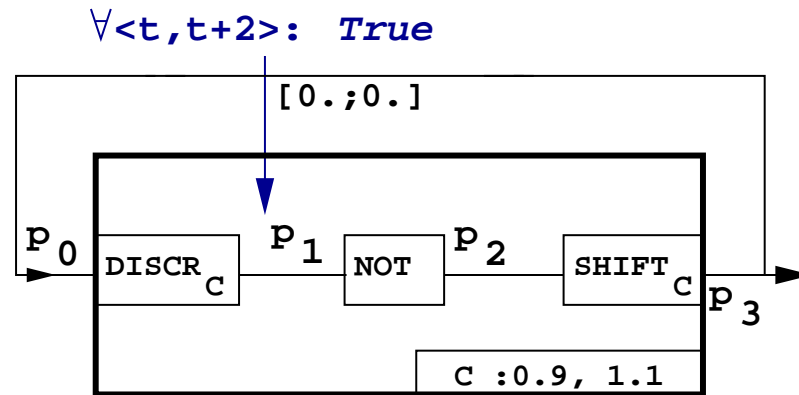
# Analysis in the abstract domain of Constraints

---



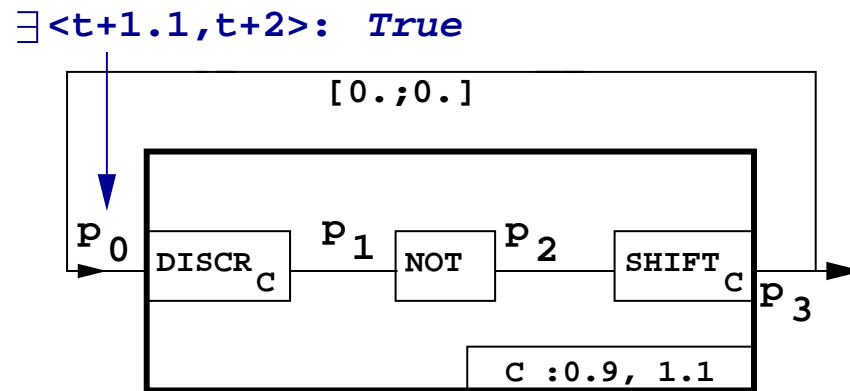
# Analysis in the abstract domain of Constraints

---



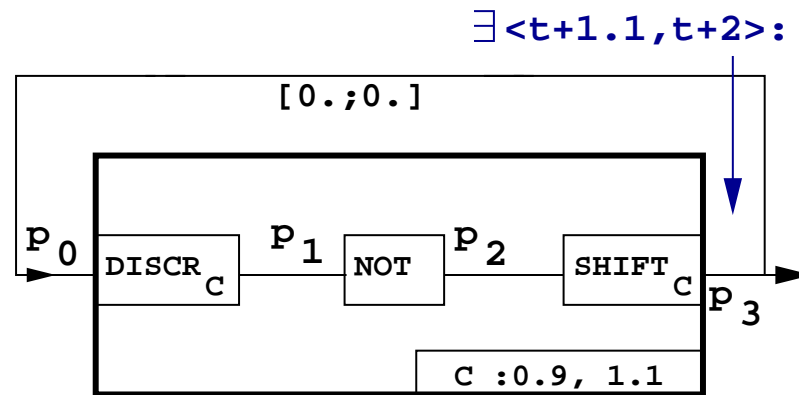
# Analysis in the abstract domain of Constraints

---



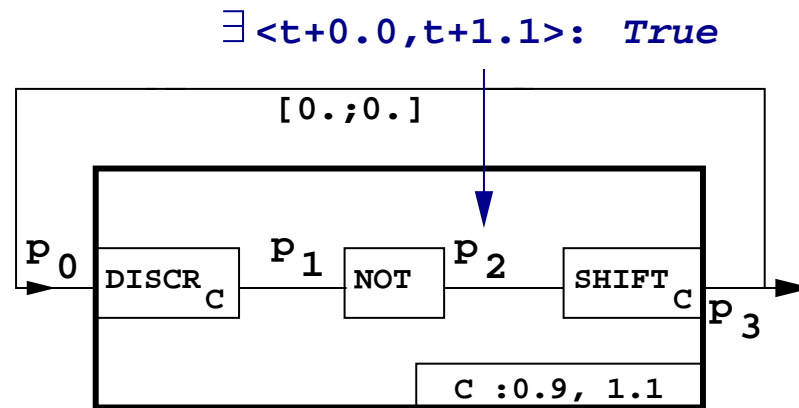
# Analysis in the abstract domain of Constraints

---



# Analysis in the abstract domain of Constraints

---



At point  $p_2$  : 2 constraints should be satisfied :

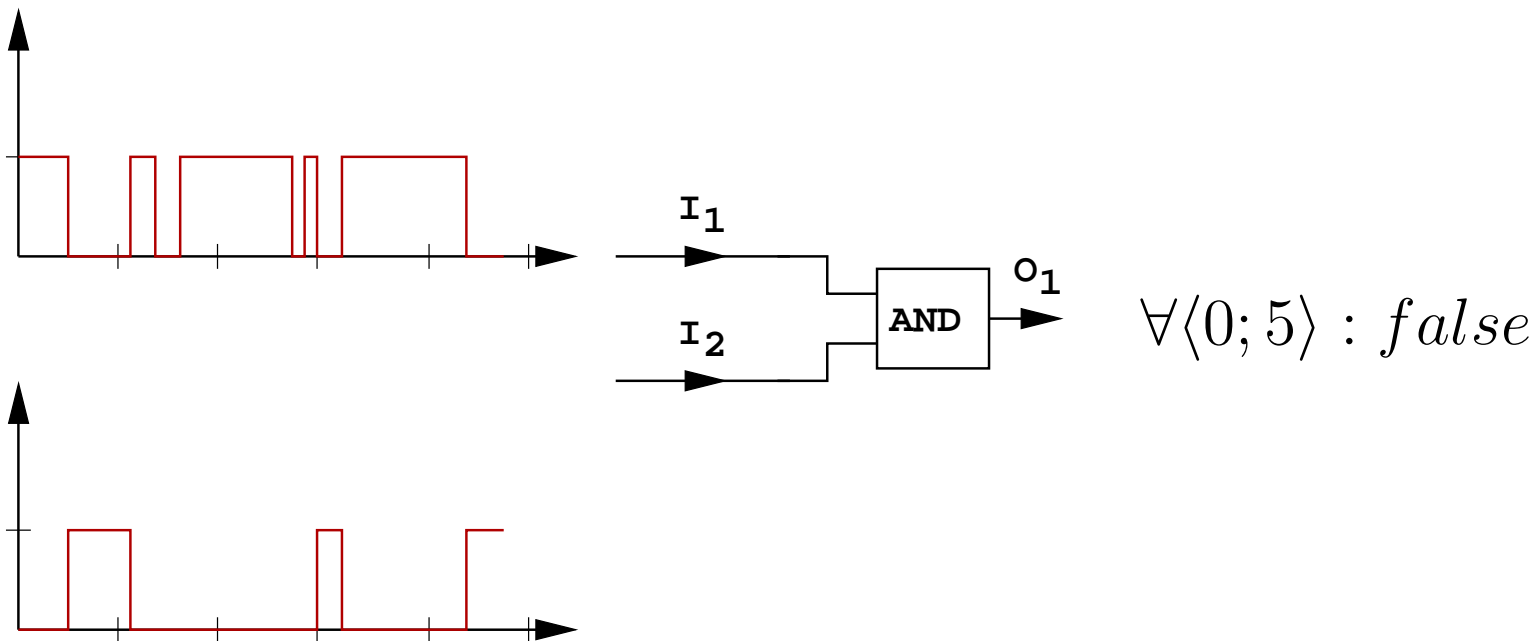
- $\forall \langle t; t + 2 \rangle : False$
- $\exists [t + 0.0; t + 1.1] : True$

which is clearly impossible and therefore invalidates our hypothesis.

# Weaknesses of the Constraints domain

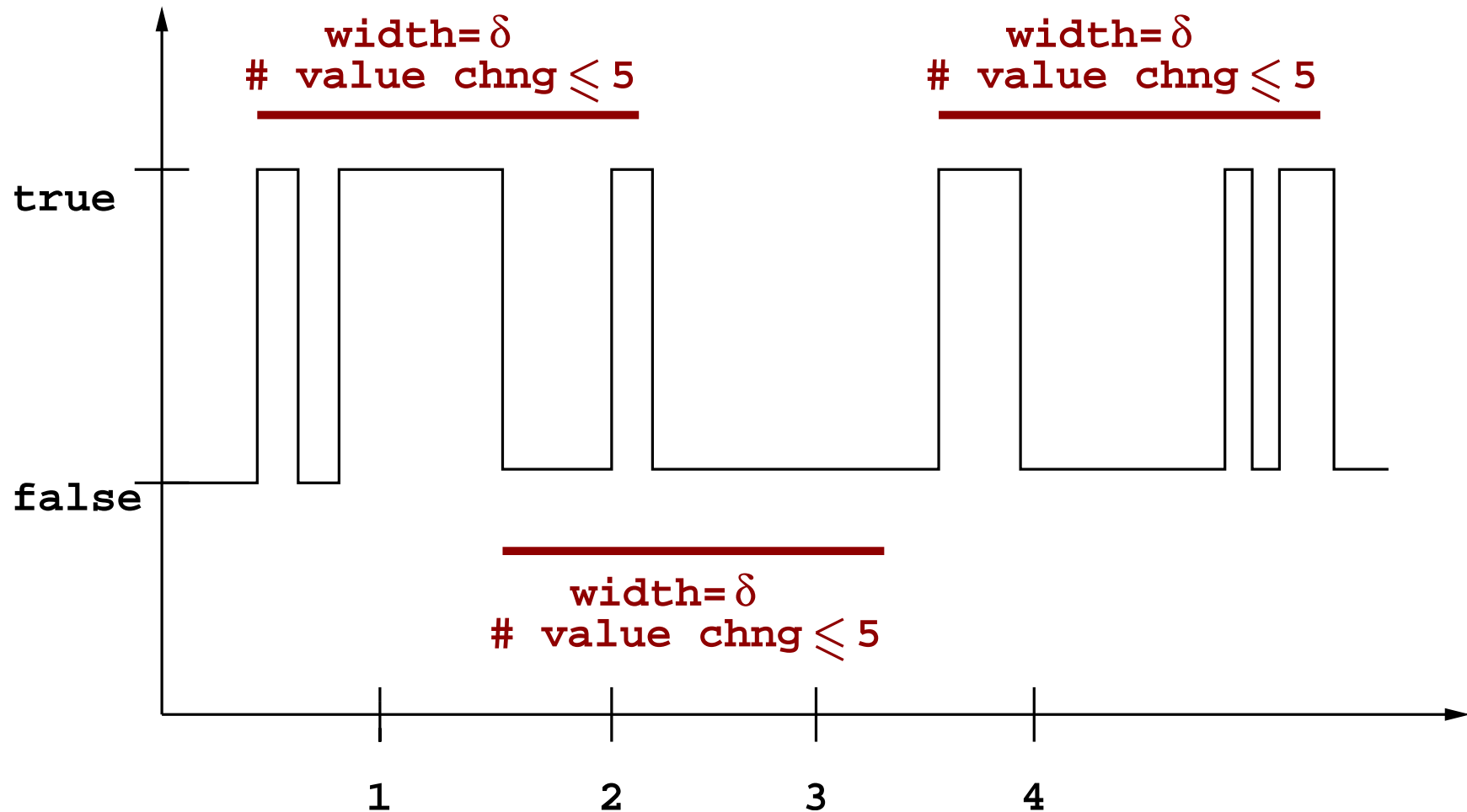
---

- Weak loss of precision in the case of : DELAY, DISCR, SHIFT, NOT,
- Unwished loss of precision in the case of : AND, OR, XOR



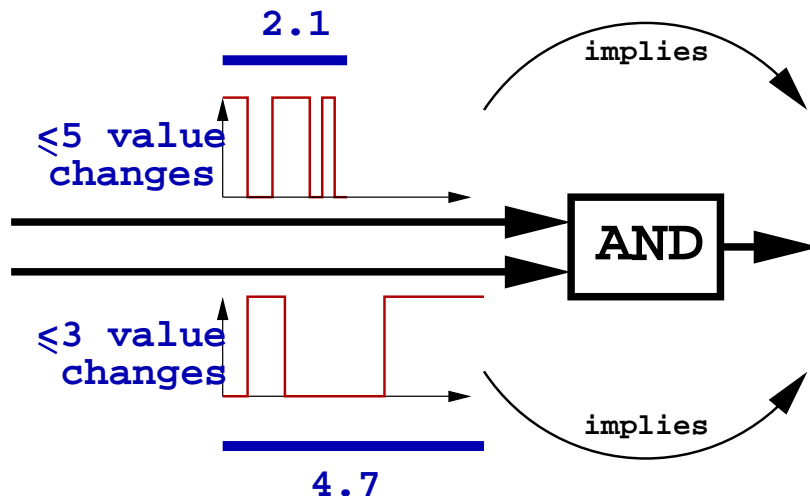
# 2nd Abstract Dom. : Changes Counting Dom.

---

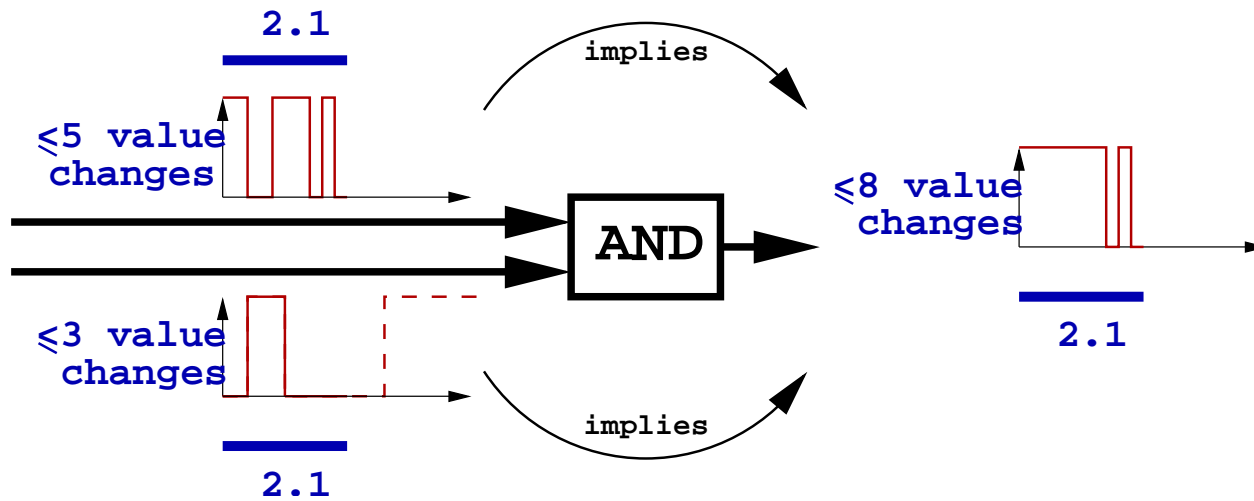


# Time-independent Abstract Operators inside the Changes Counting Domain

---

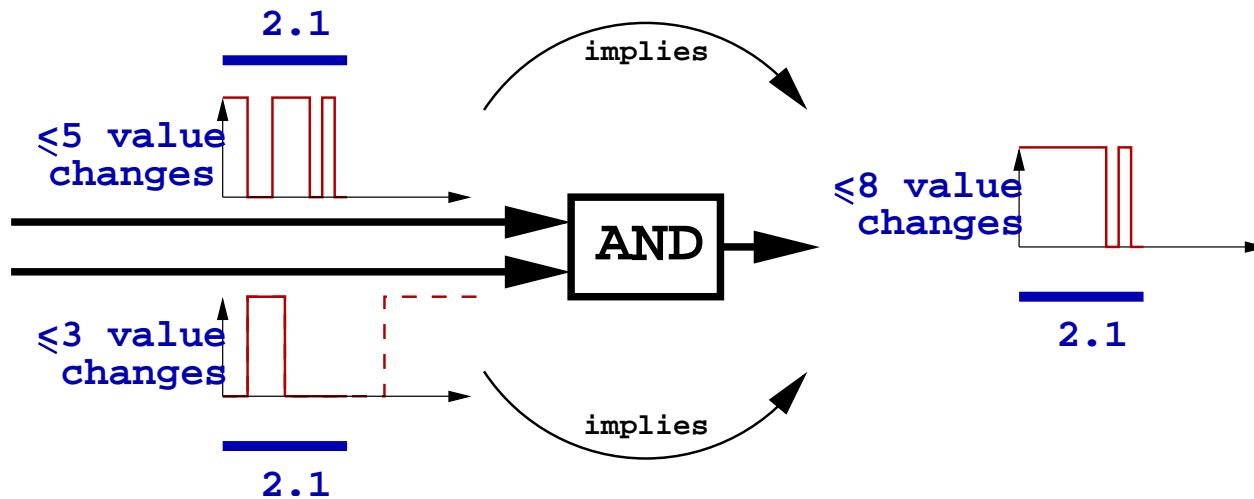


# Time-independent Abstract Operators inside the Changes Counting Domain



$$\overrightarrow{\Psi}_{\text{AND}}^{\#}((n_1, \delta_1), (n_2, \delta_2)) \triangleq (\tilde{n}_1 + \tilde{n}_2, \tilde{\delta}_1)$$

# Time-independent Abstract Operators inside the Changes Counting Domain

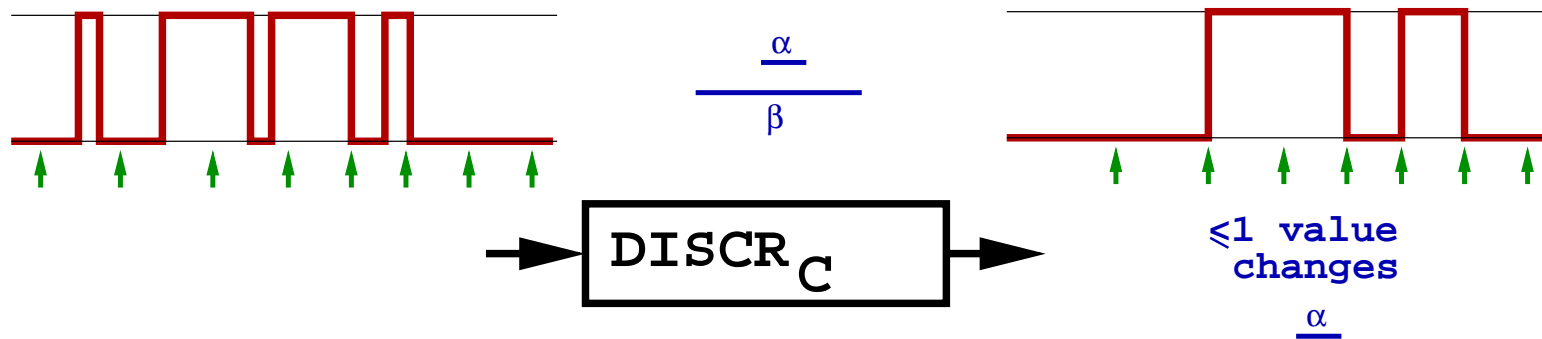


$$\overrightarrow{\Psi}_{\text{AND}}^{\#}((n_1, \delta_1), (n_2, \delta_2)) \triangleq (\tilde{n}_1 + \tilde{n}_2, \tilde{\delta}_1)$$

- $\varphi$  is a reframing function and
- $\varphi((n_1, \delta_1), (n_2, \delta_2)) = ((\tilde{n}_1, \tilde{\delta}_1), (\tilde{n}_2, \tilde{\delta}_1))$ .

# Time-dependent Abstract Operators inside the Changes Counting Domain

---

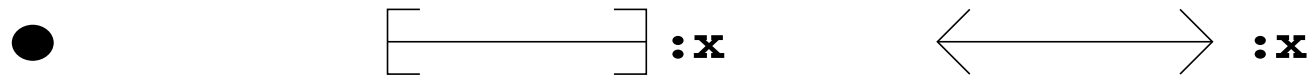


–  $[\alpha, \beta]$  parameter of clock  $C$

–  $\overrightarrow{\Psi}_{\text{DISCR}_{[\alpha, \beta]}}^{\#}(-) \triangleq (1, \alpha)$

# Reduced Product Constraints-Changes Counting Domain

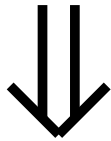
---



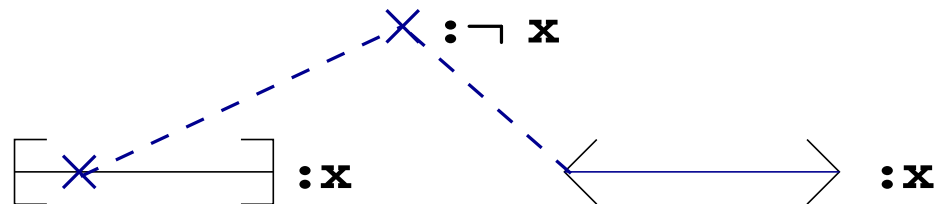
+

width =  $\delta$   
# value chng  $\leq 1$

●



●



# Reduced Product Constraints-Changes Counting Domain

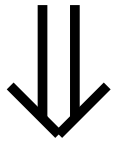
---

●  $\left[ \quad \right] :x$   $\longleftrightarrow$   $:x$

+

width =  $\delta$   
# value chng  $\leq 1$

●

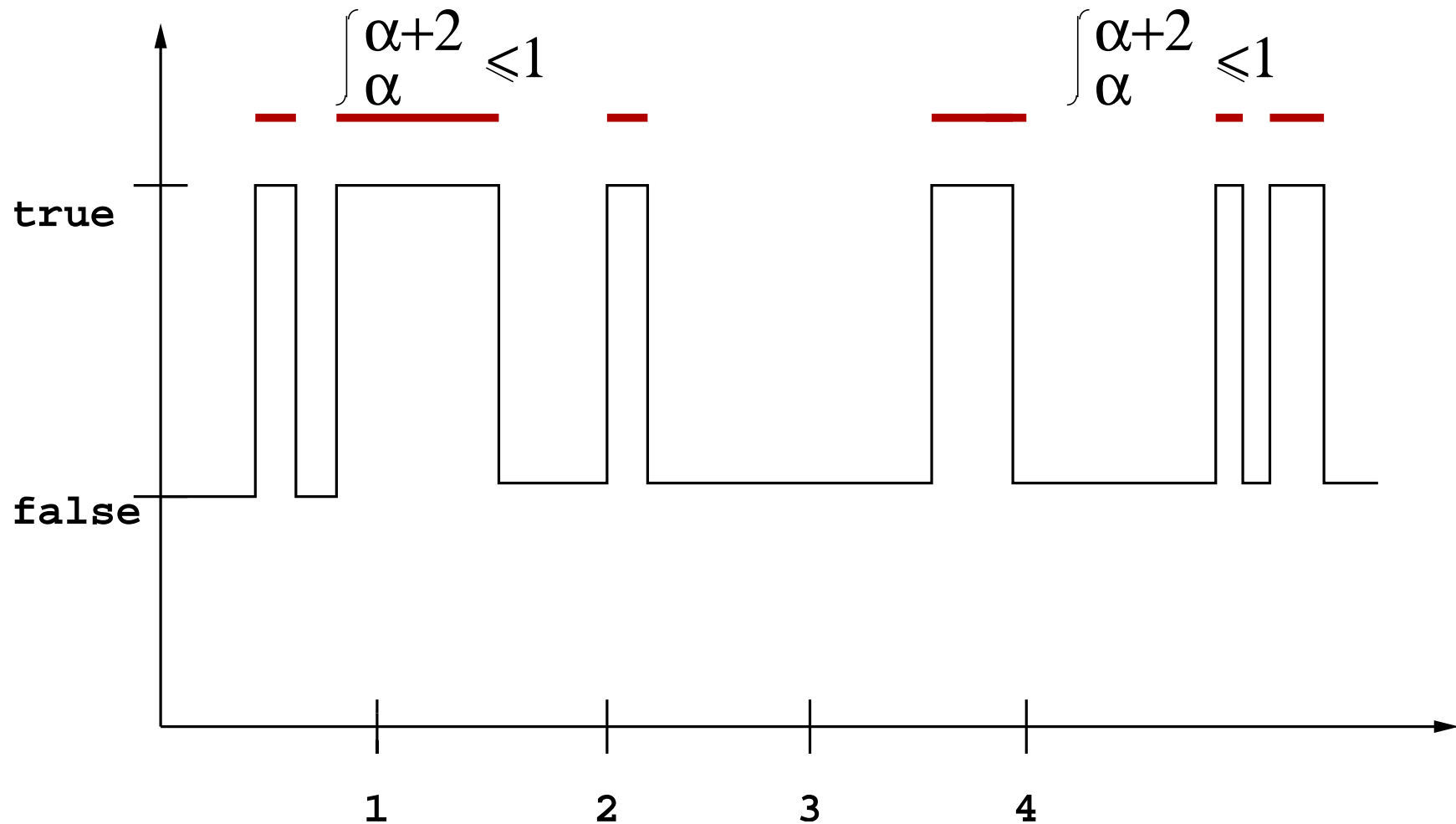


●

$\longleftrightarrow$   $:x$

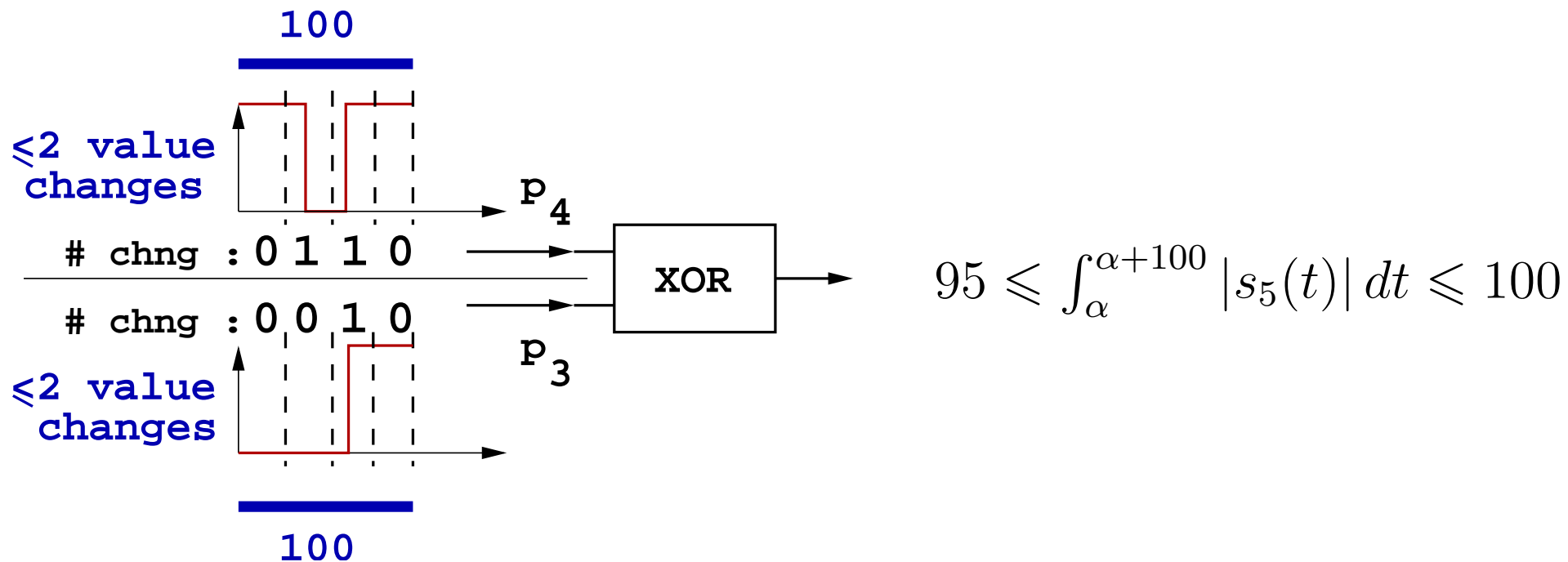
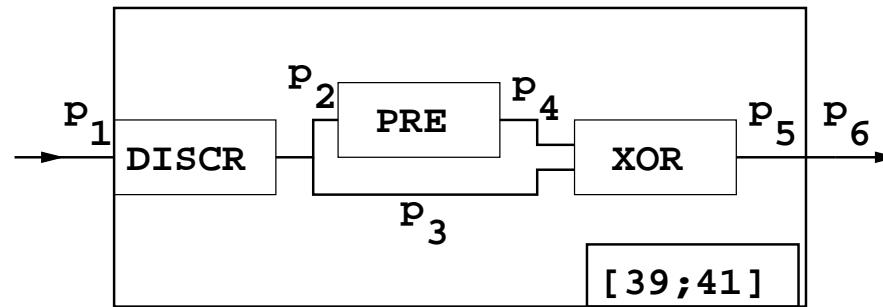
# 3rd abstract domain : Integral bounding Dom.

---



# Cooperation example of the 3 abstract dom.

width=100  
# value chng  $\leq 1$



# Conclusion

---

- Defined : **2 new abstract domains** handling sets of signals
- Defined : **corresponding sound** abstract operator for each primitive (backward and forward)
- **Strong cooperation** (reduced product) between the abstract domains
- Goal : prove robustness of critical software to :
  - **clock skew**
  - **non-constant delays in communication**
- Presently, without complex iteration strategy, simple temporal properties concerning redundancy may be proved.