

Geometry of Interaction and Coherence Spaces

Naohiko Hoshino
Kyoto University
(CRECOGI 2016)

Computable Functions

Church-Turing thesis

- Turing machines
- Recursive functions
- The untyped lambda calculus

How about higher order functions?

PCF

PCF consists of

- The simply typed lambda calculus
- Recursion

PCF miss some “computable” functions

Example

$$\mathbf{pconv}(x : \mathbb{N}_\perp, y : \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

It is easy to implement $\text{PCF} + \mathbf{pconv}$

Example

strict($u: \mathbb{N}_\perp \rightarrow \mathbb{N}_\perp$)

$$\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

PCF+strict is implementable

Implementability

We can not implement

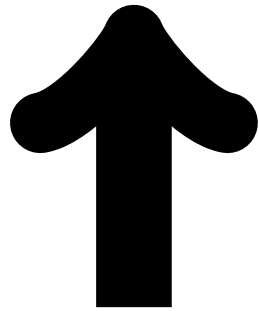
PCF+strict+pconv

because there is

$$f(x : \mathbb{N}_\perp) = \begin{cases} 0 & \text{if } x \neq \perp \\ 1 & \text{if } x = \perp \end{cases}$$

Question

How many notions of
higher order computability are there?



Geometry of Interaction (GoI)
would provide a uniform approach.

Main Results

Thm

The full subCCC of **Cpo** generated by \mathbb{N}_\perp is a full subcategory of $\mathbf{Asm}(\mathbf{Rel}(\mathbb{N}, \mathbb{N}))$
 $= \{0, 1\}^{\mathbb{N} \times \mathbb{N}}$

Thm

The full subCCC of **Coh** generated by \mathbb{N}_\perp is a full subcategory of $\mathbf{Asm}(\mathbf{wRel}(\mathbb{N}, \mathbb{N}))$
 $= \{0, 1, \dots, \infty\}^{\mathbb{N} \times \mathbb{N}}$

c.f. [Oosten 99]

The full subCCC of **HCoh** generated by \mathbb{N}_\perp is a full subcategory of $\mathbf{Asm}(\mathbf{Pfn}(\mathbb{N}, \mathbb{N}))$

Related Works

Characterizations of **HCoh** in terms of

- **Gol** [Oosten 99]
- **Game semantics** [Melliès 05]

Embeddings of game semantics into **Coh**

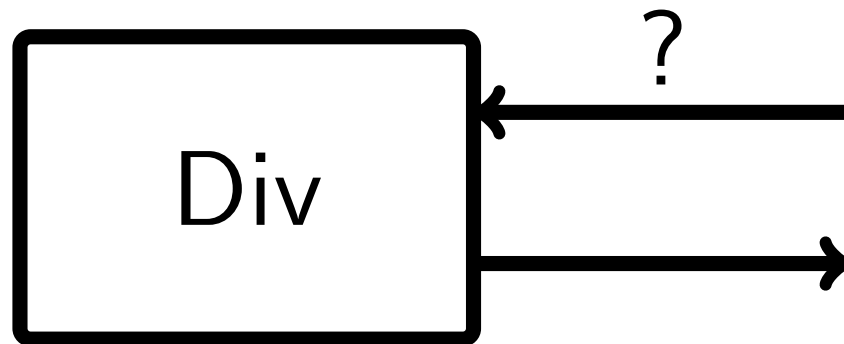
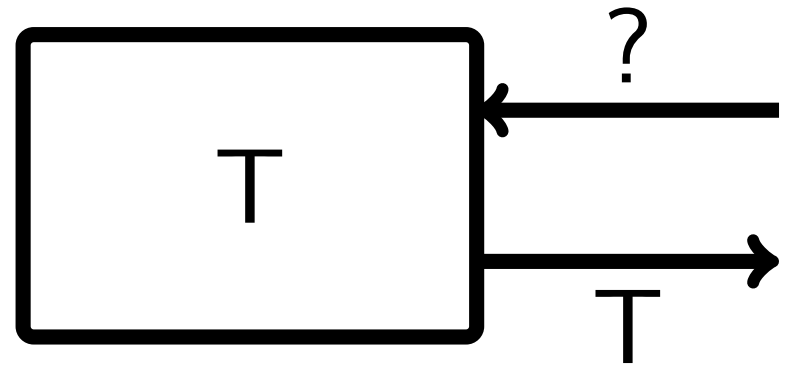
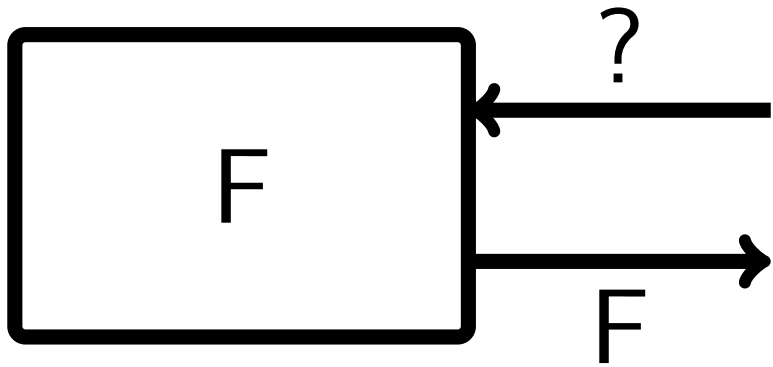
- [Melliès 13]
- [Calderon & McCusker 10]

Geometry of Interaction

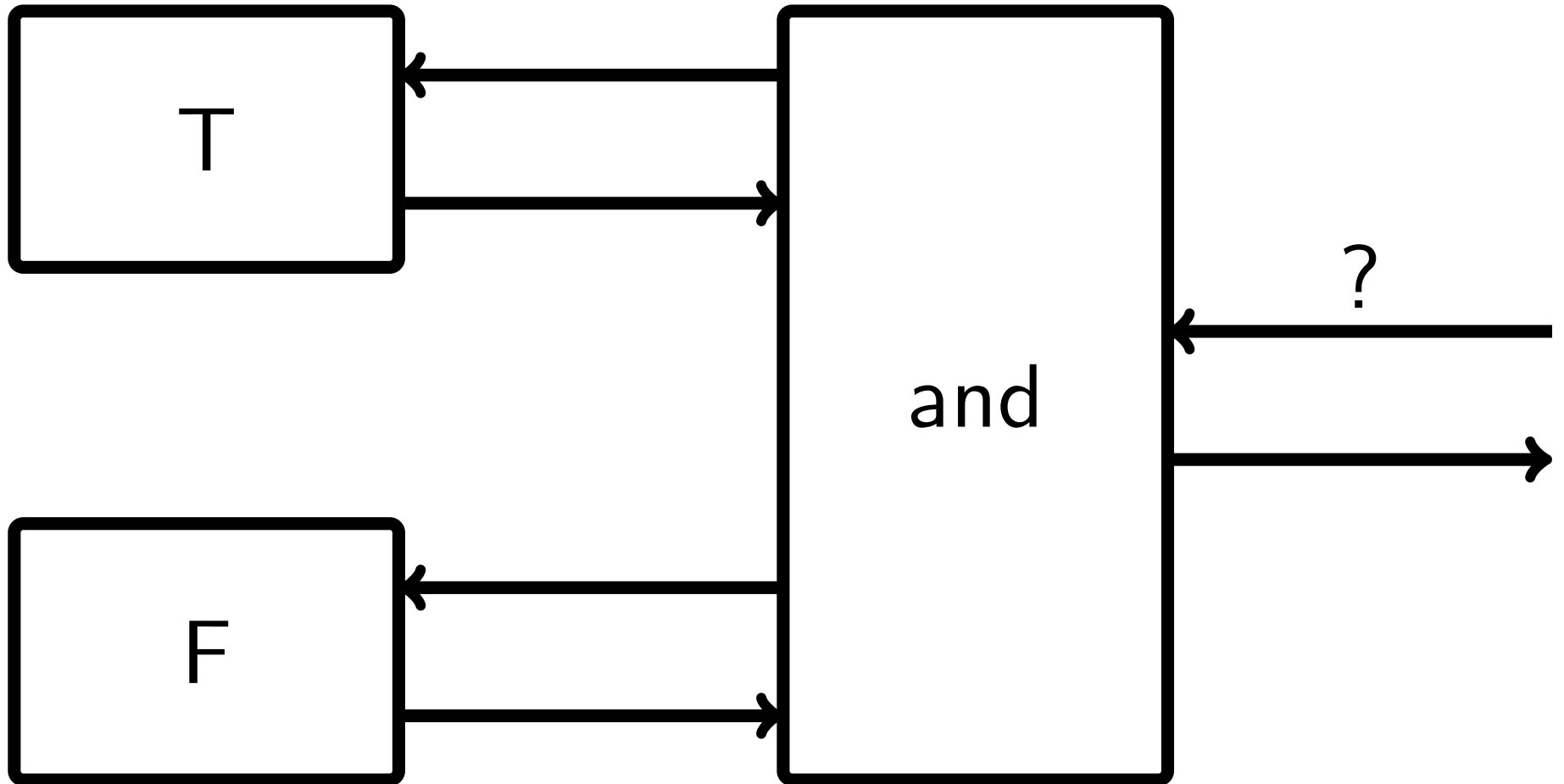
Models of lambda calculi

- Programs as token machines
- Evaluation as execution

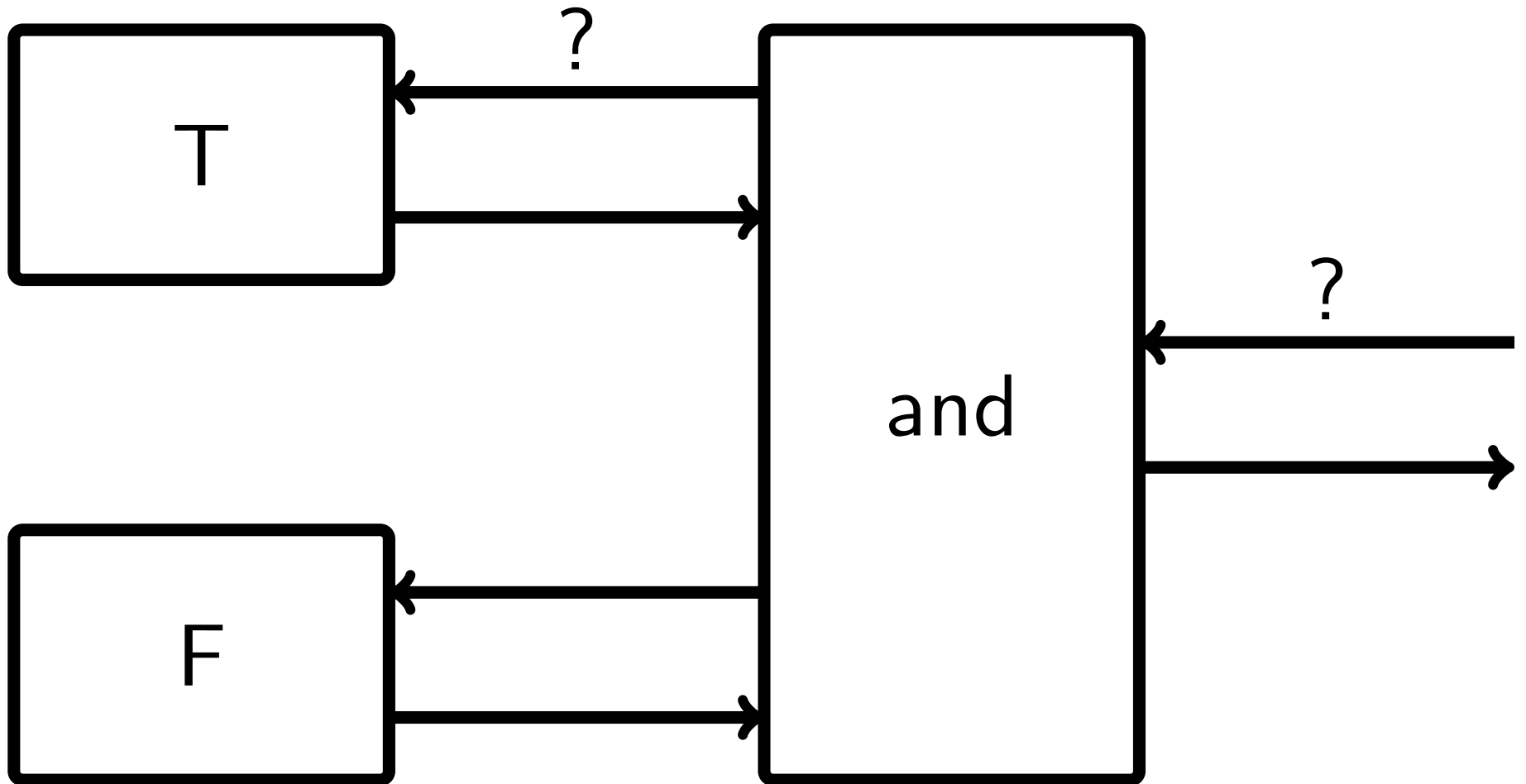
Bool



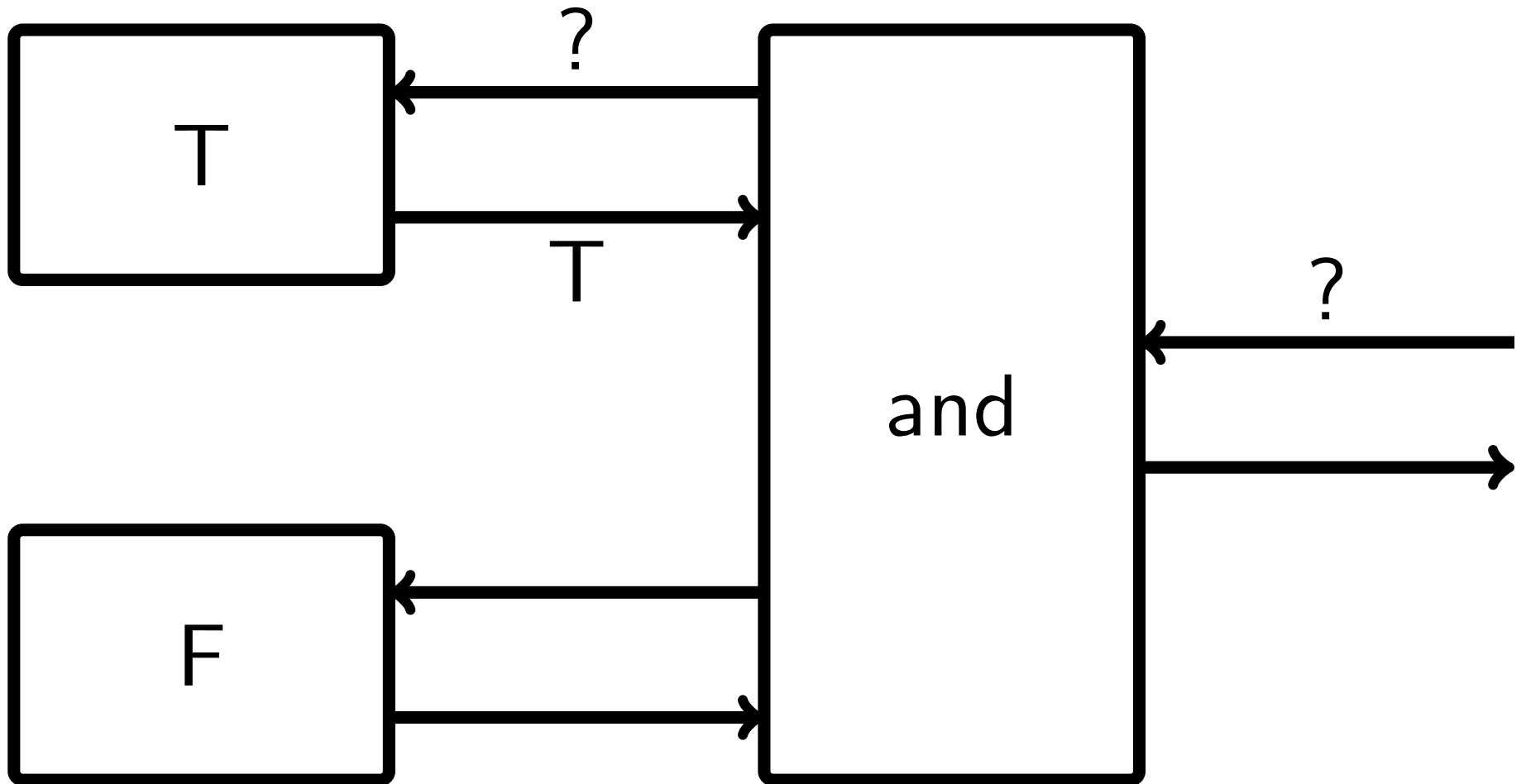
Conjunction



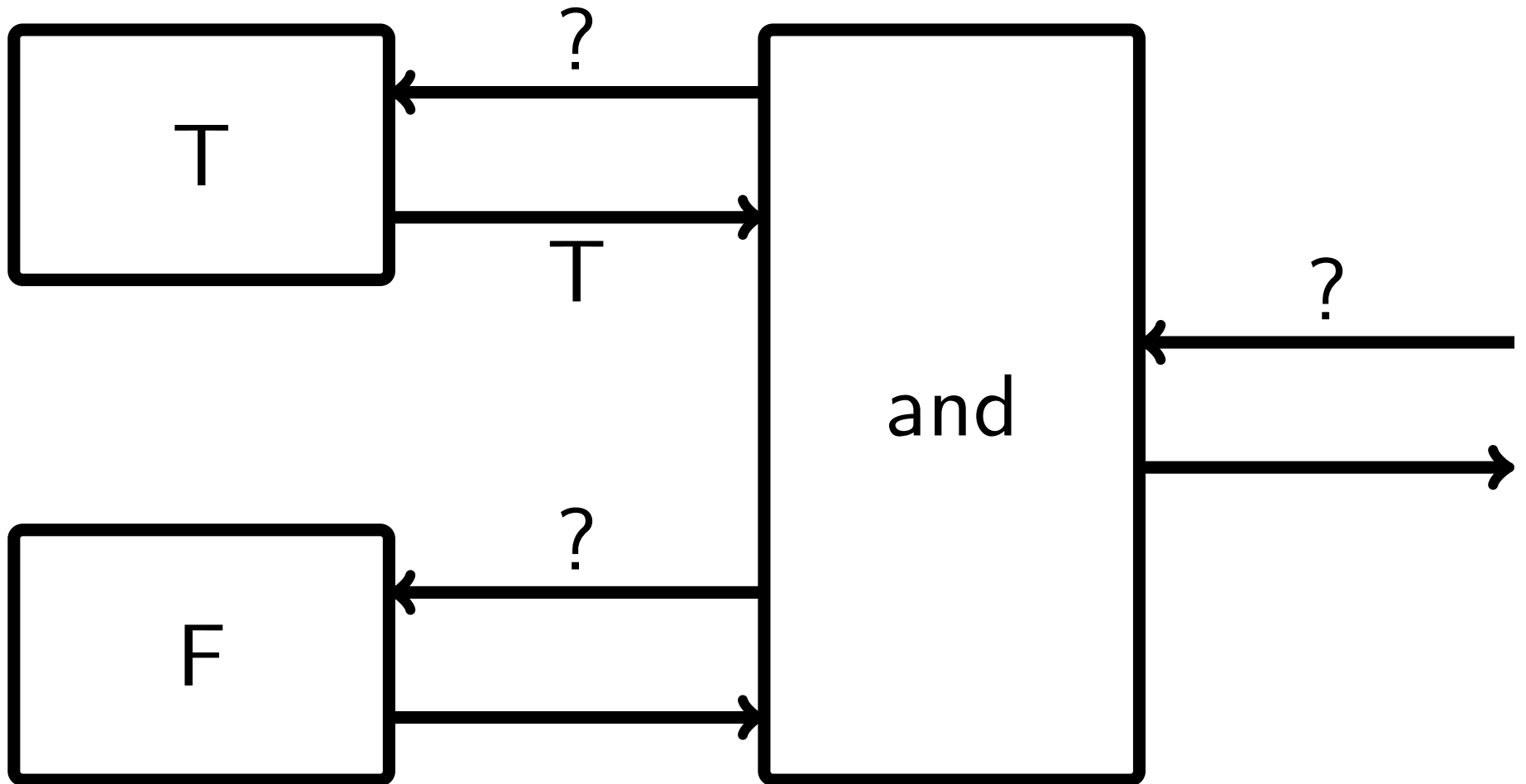
Conjunction



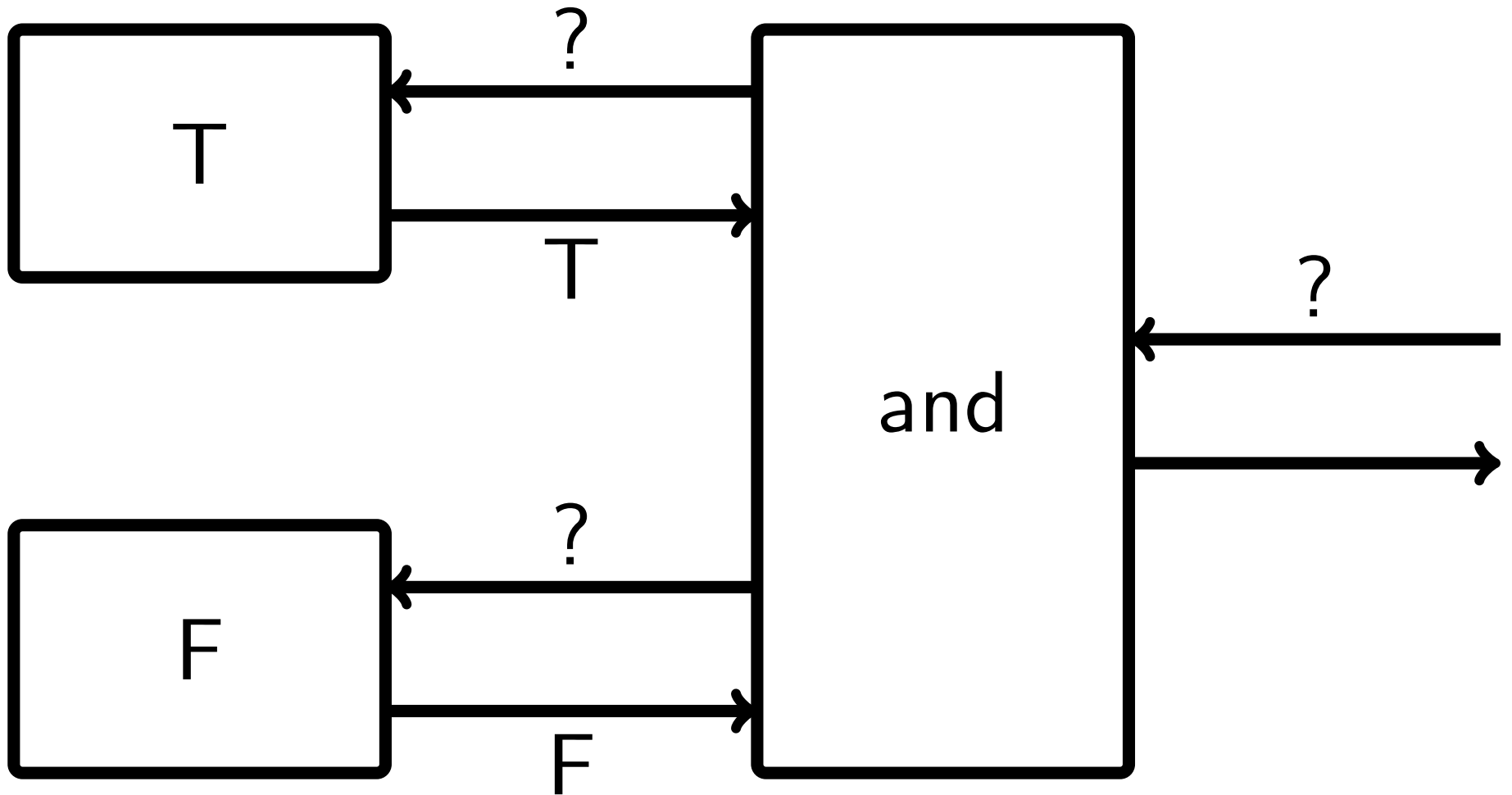
Conjunction



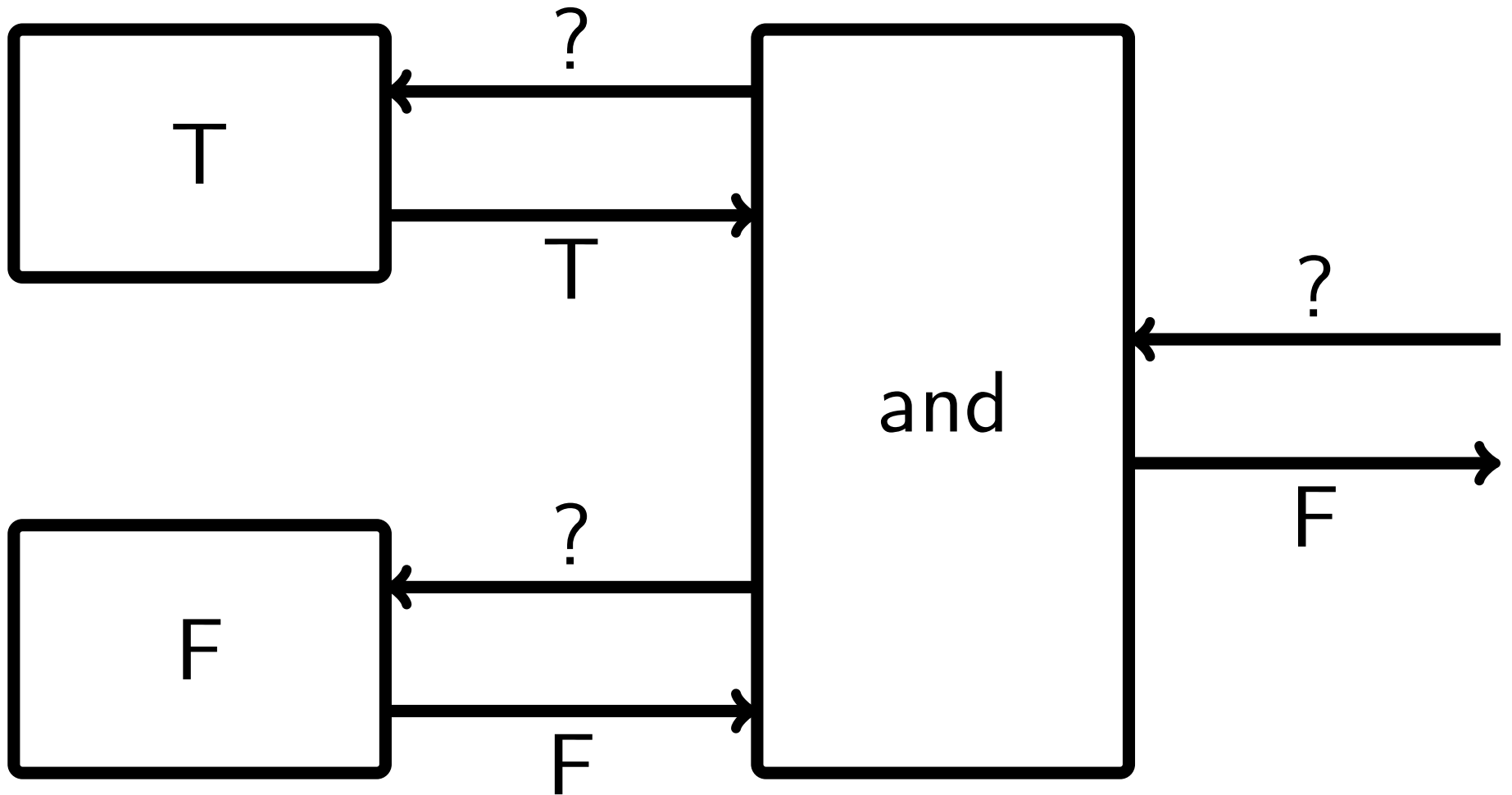
Conjunction



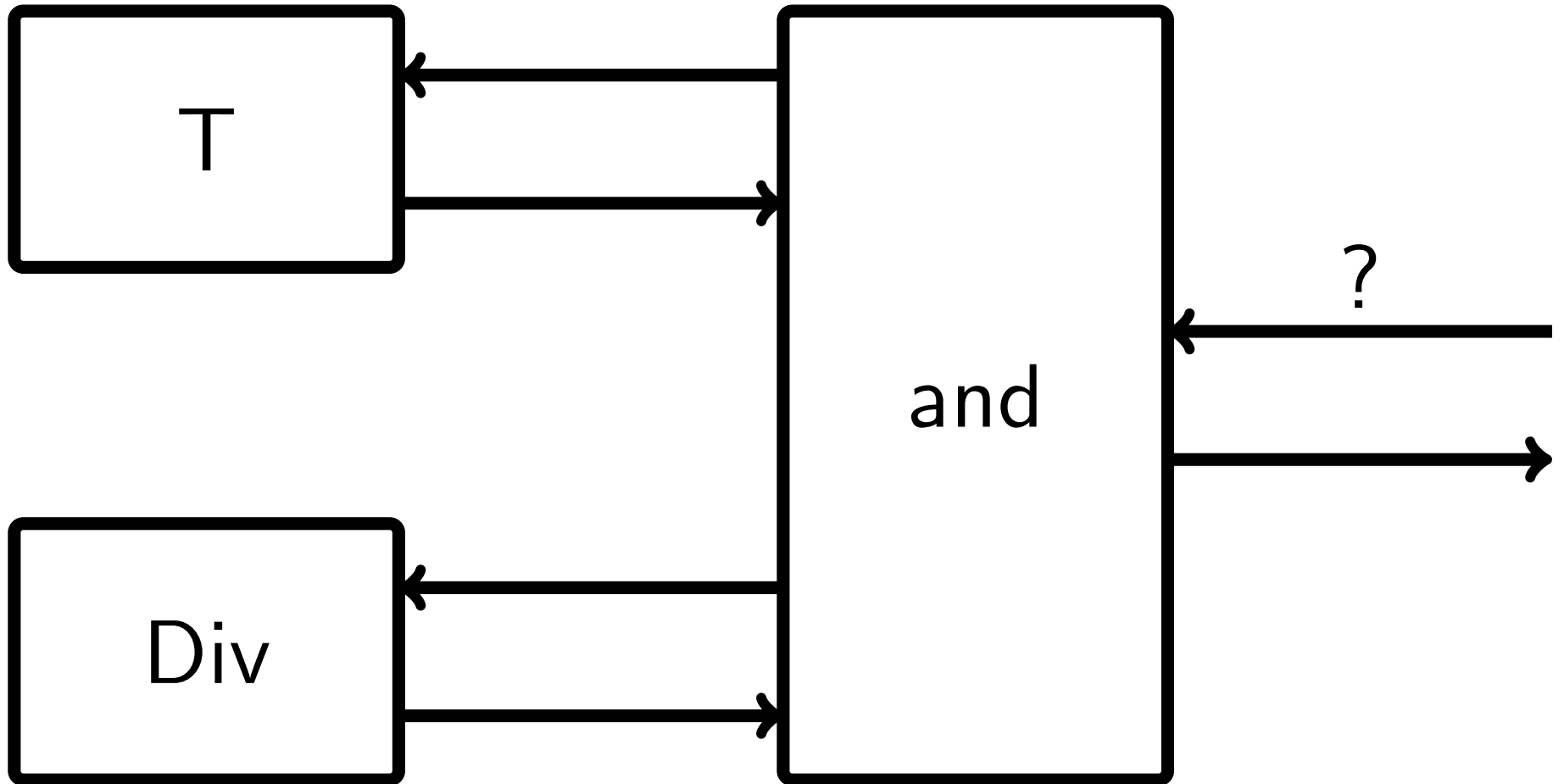
Conjunction



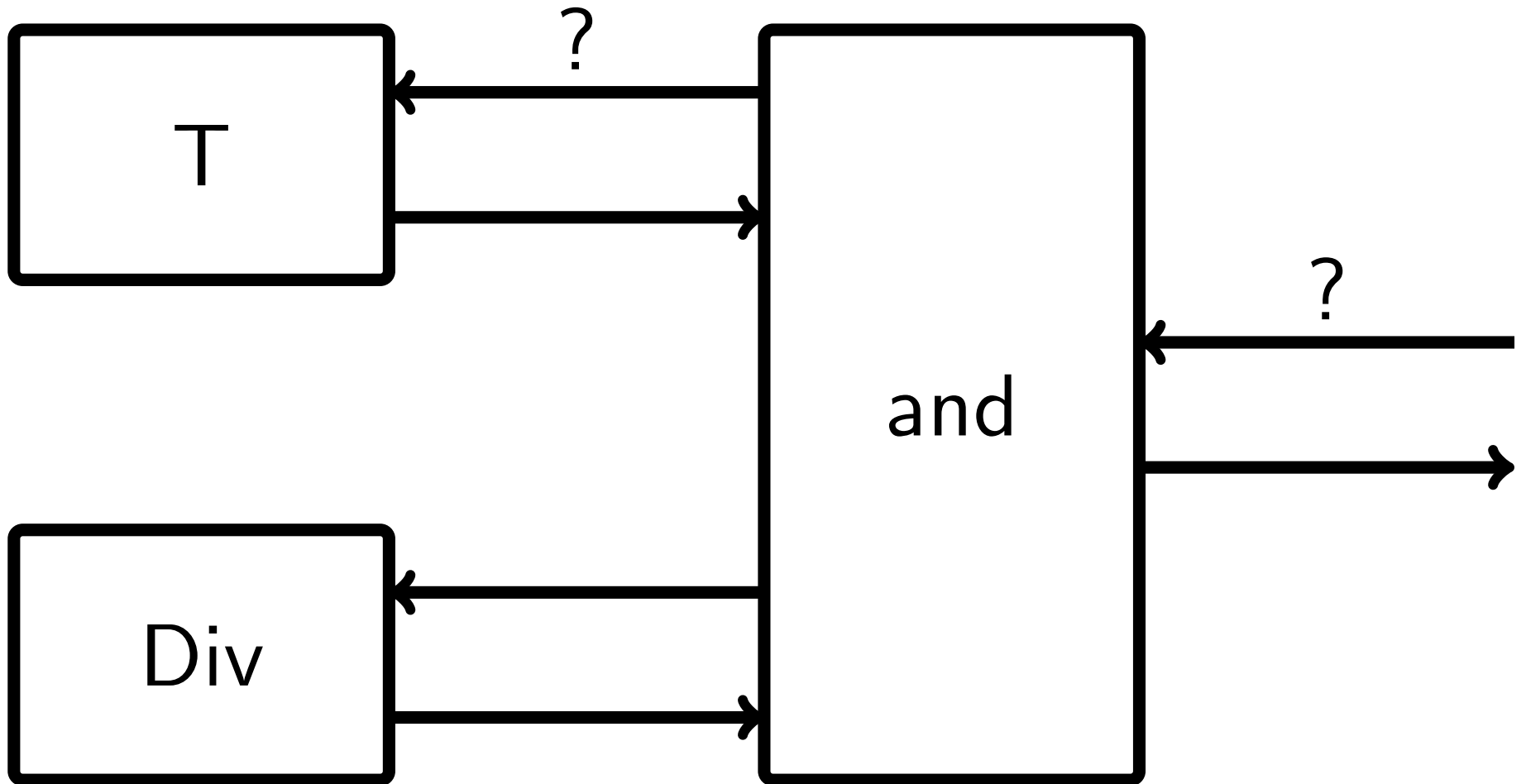
Conjunction



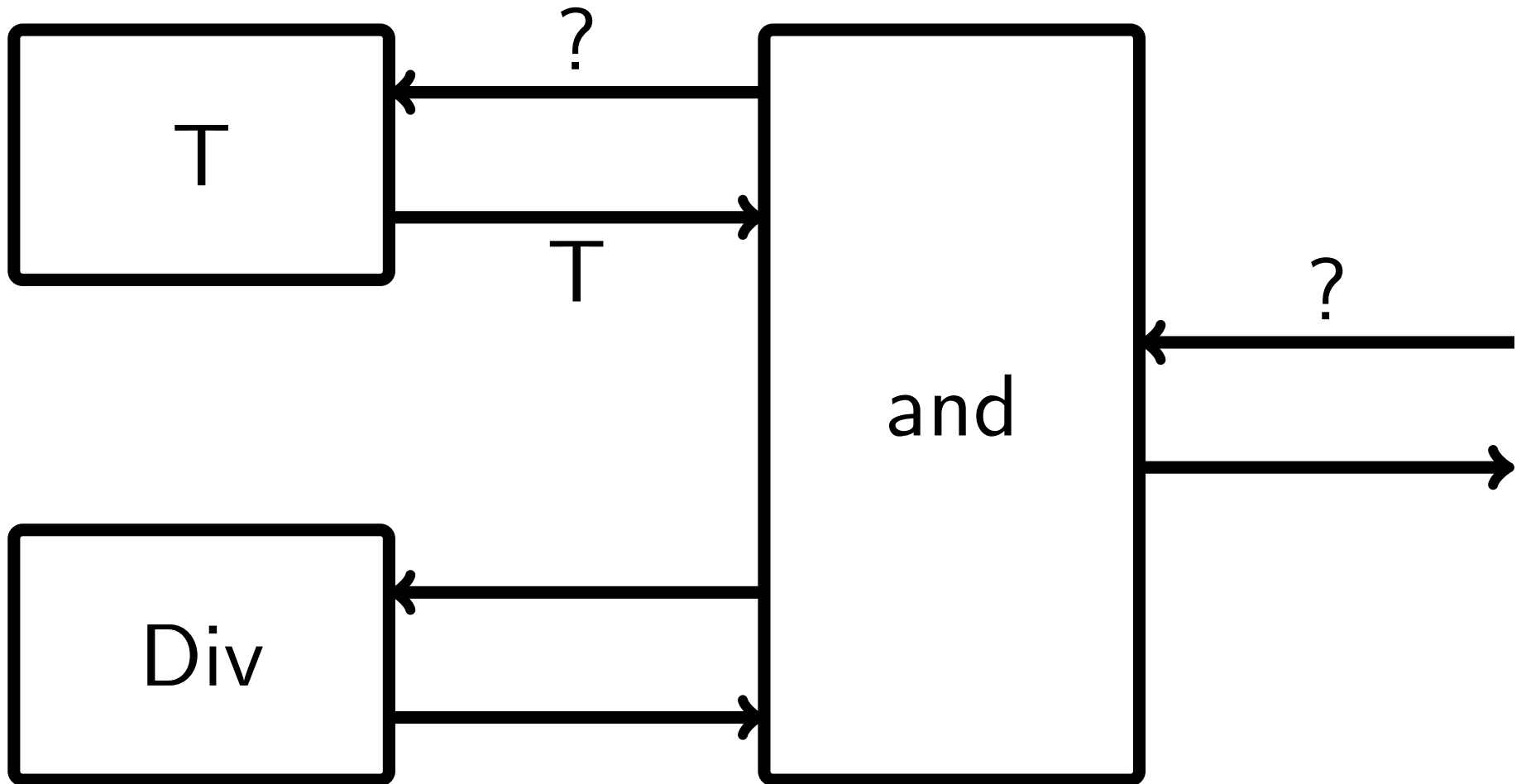
Conjunction



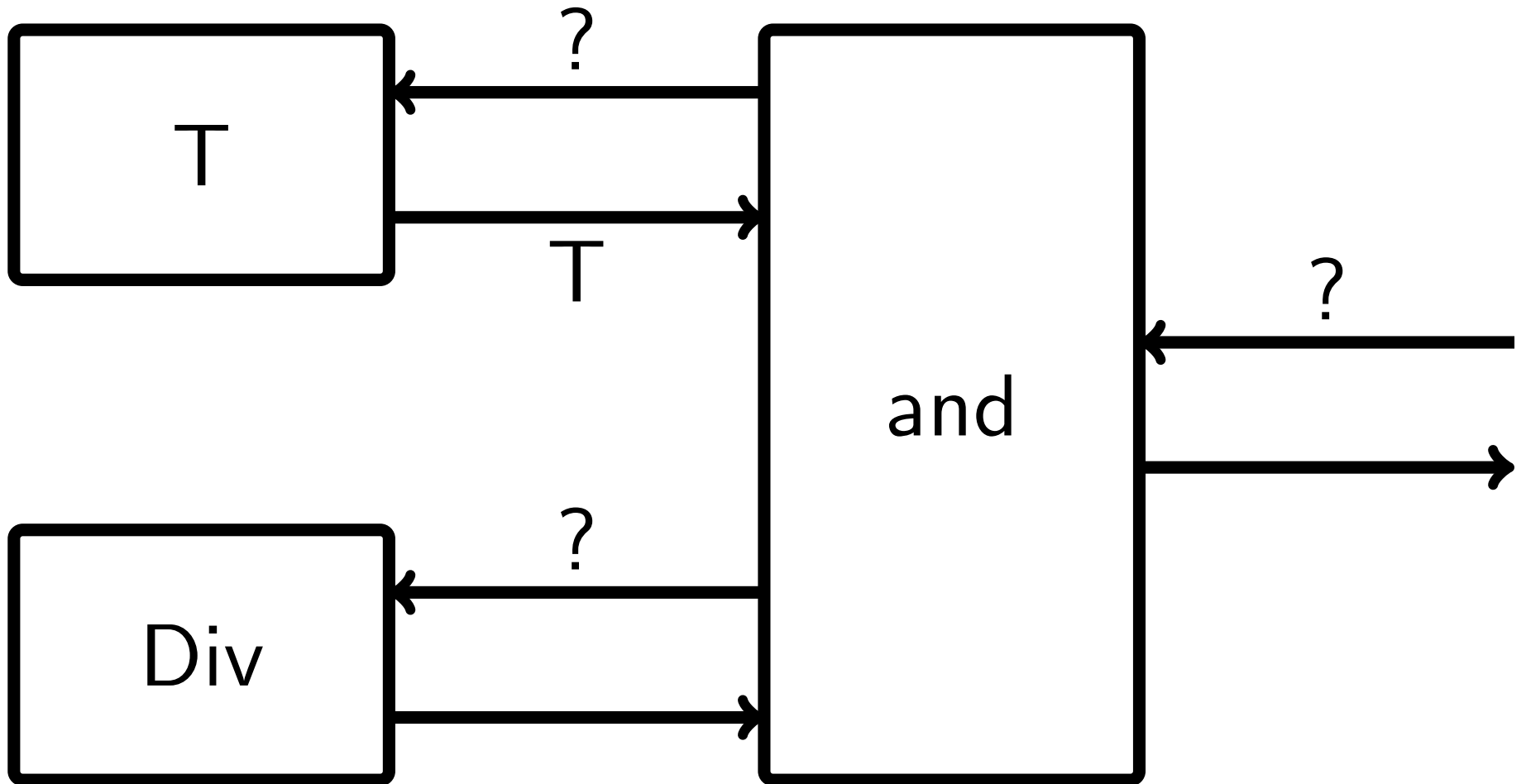
Conjunction



Conjunction



Conjunction



Gol-Rel
Gol-Pfn
Gol-wRel

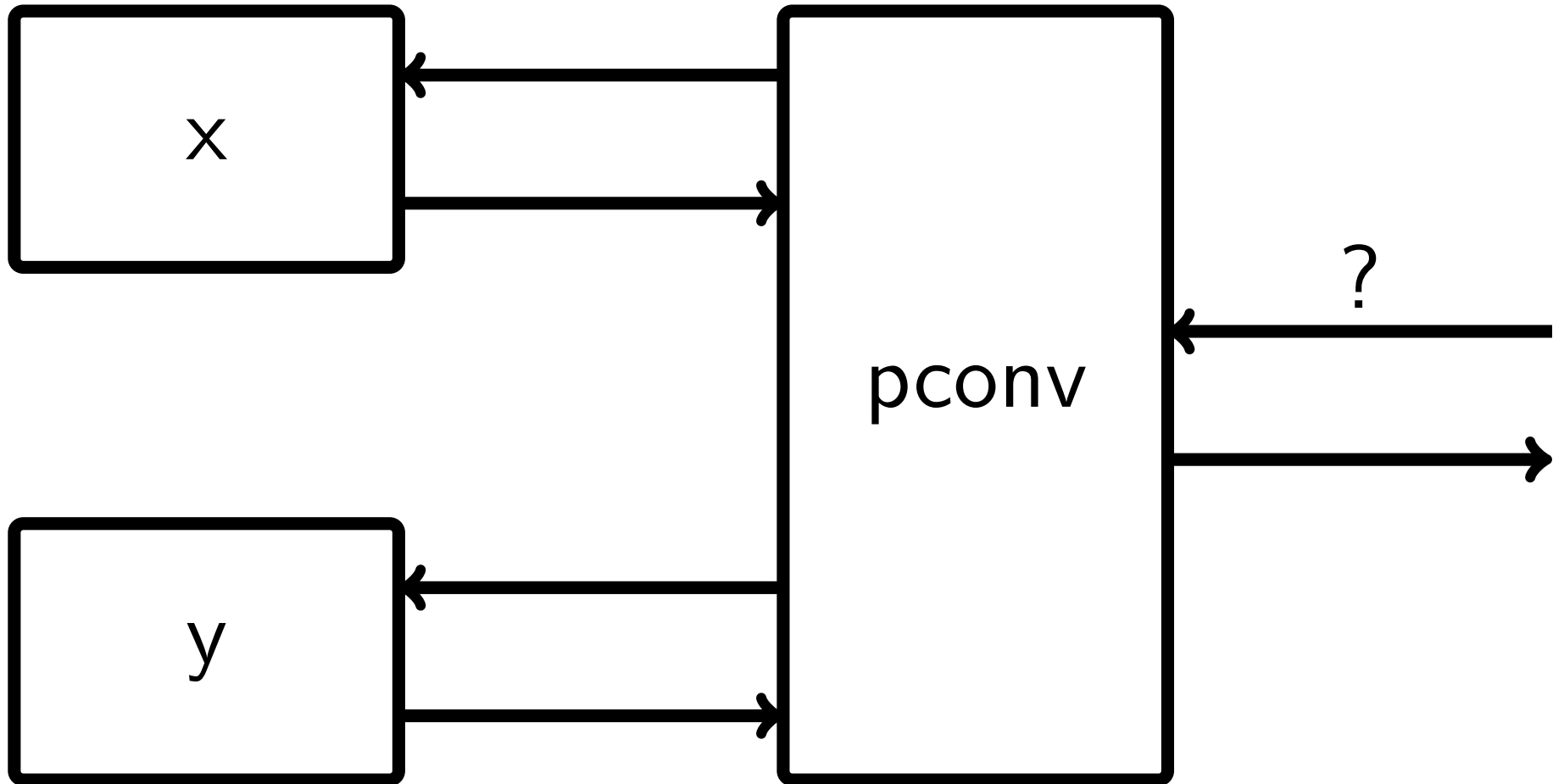
GoI-Rel

We can implement **pconv**
by sending multiple messages

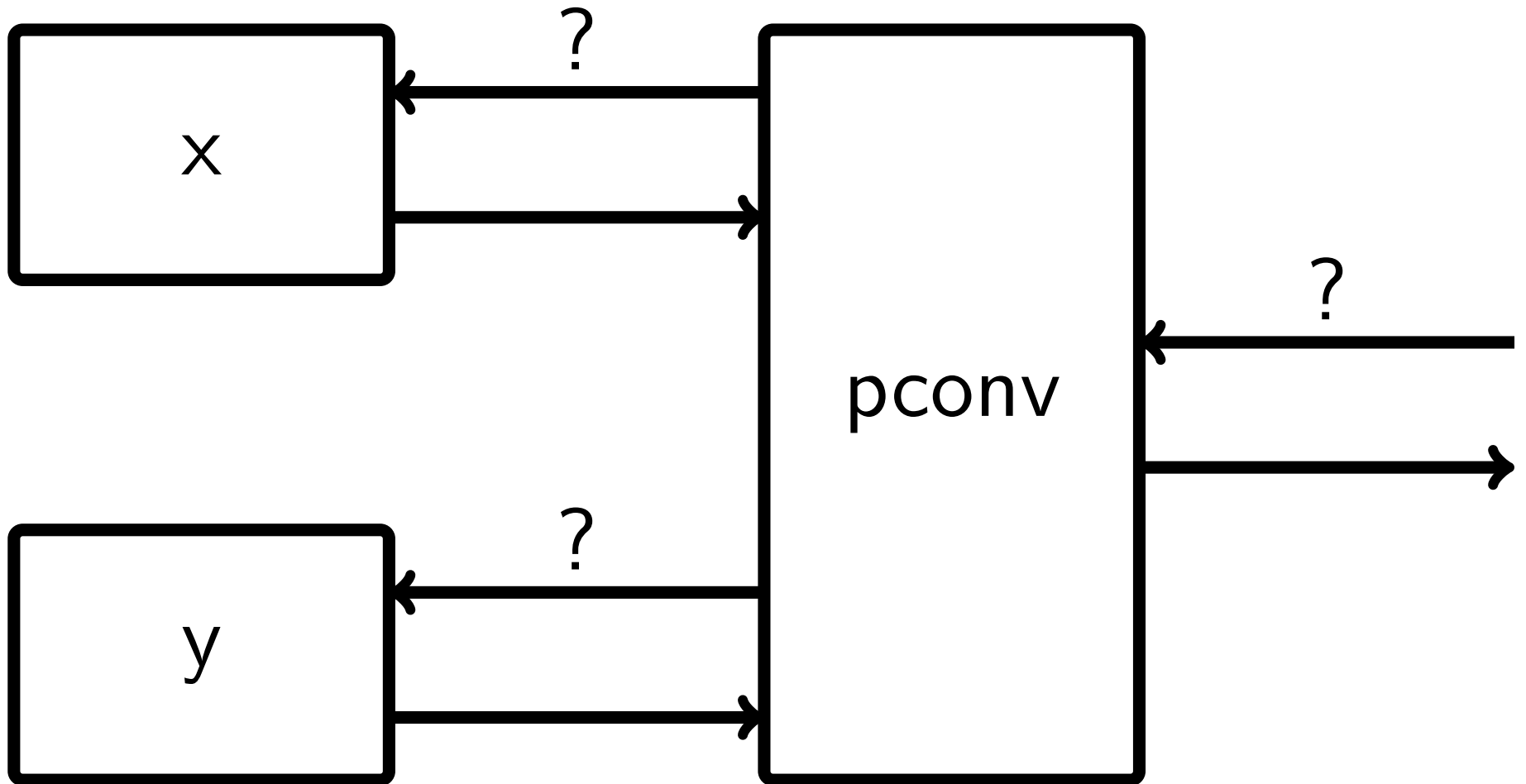
$$\mathbf{pconv}(x : \mathbb{N}_{\perp}, y : \mathbb{N}_{\perp}) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

(**pconv** \in **Cpo**)

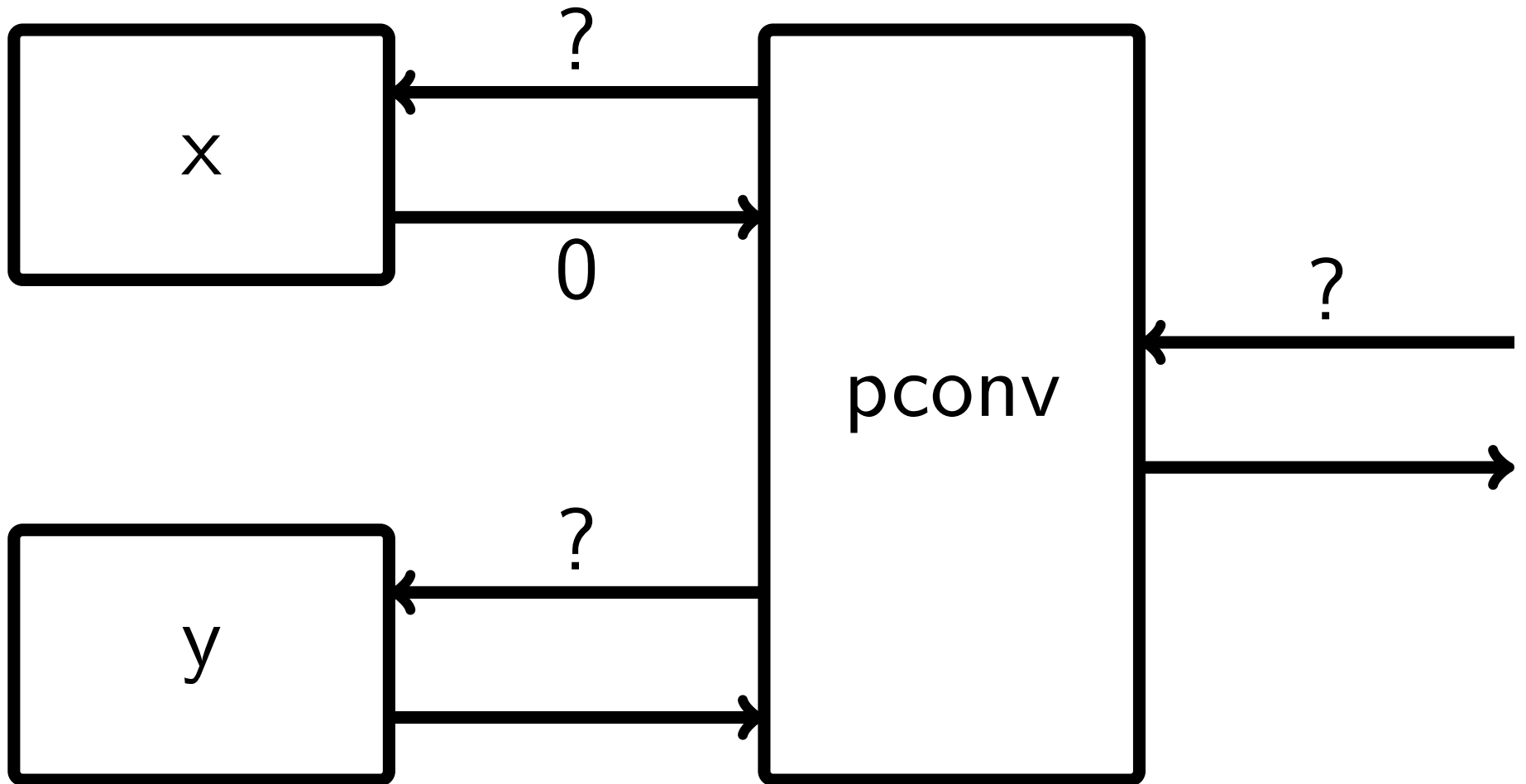
Implementation



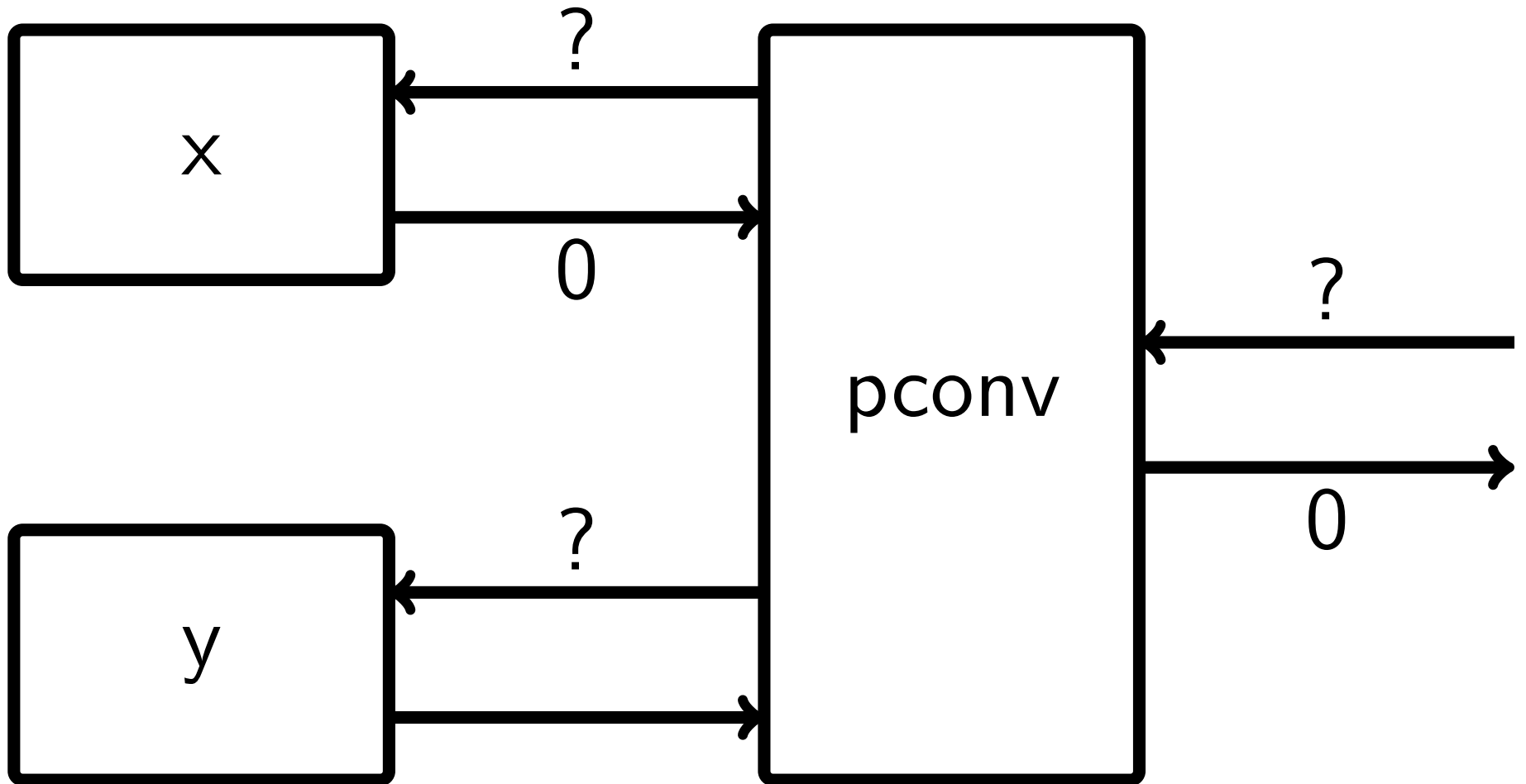
Implementation



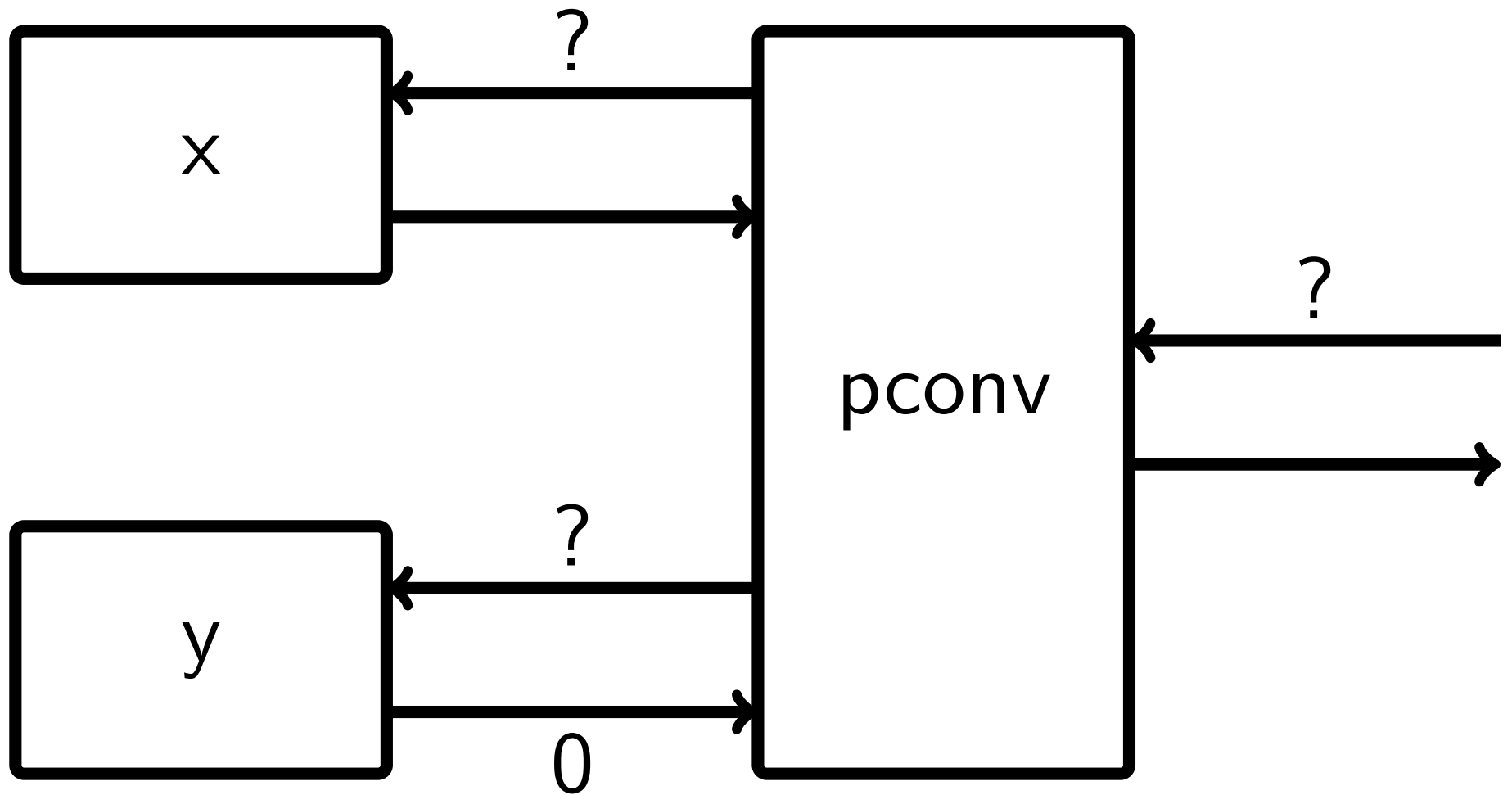
Implementation



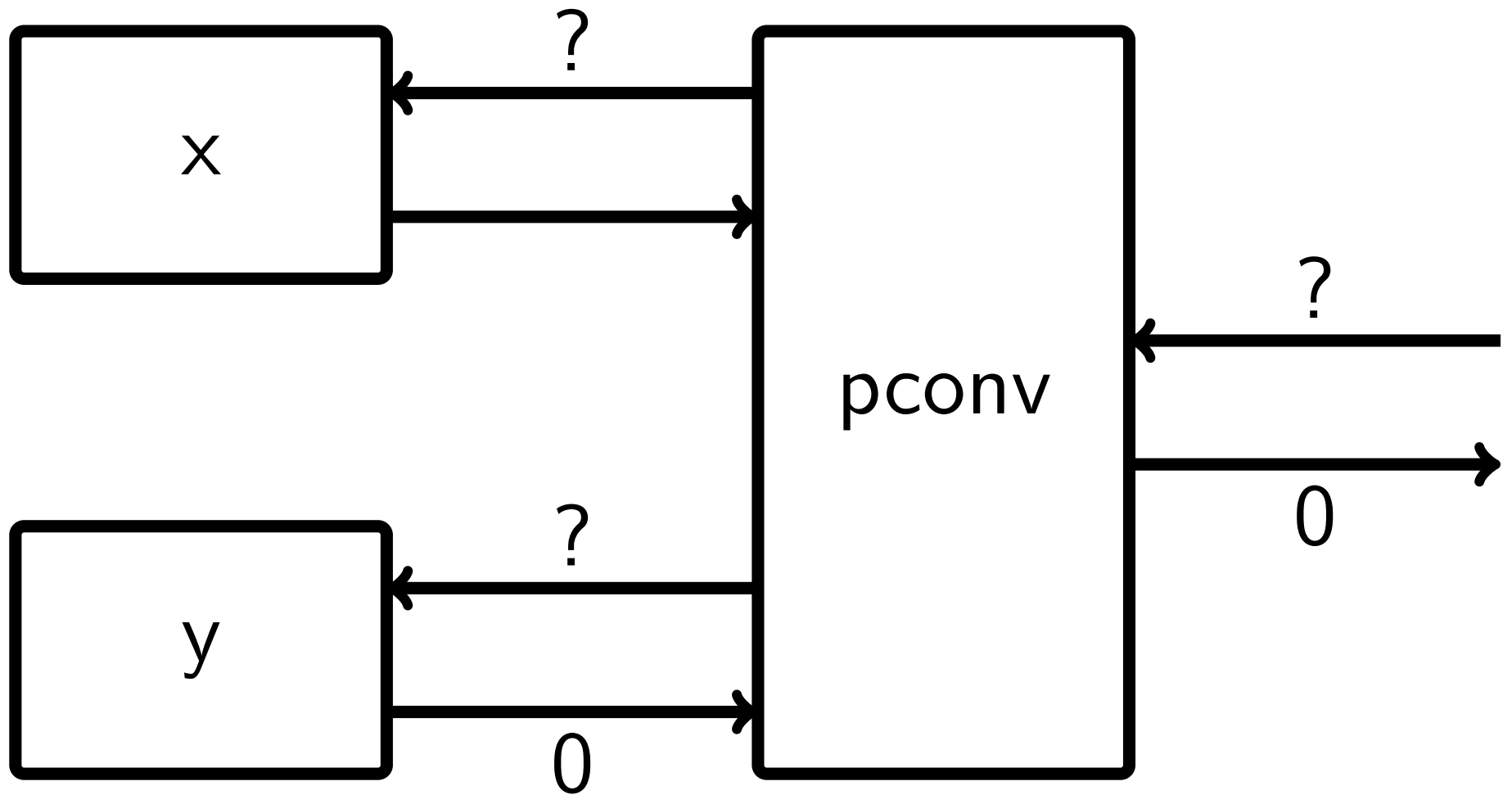
Implementation



Implementation



Implementation



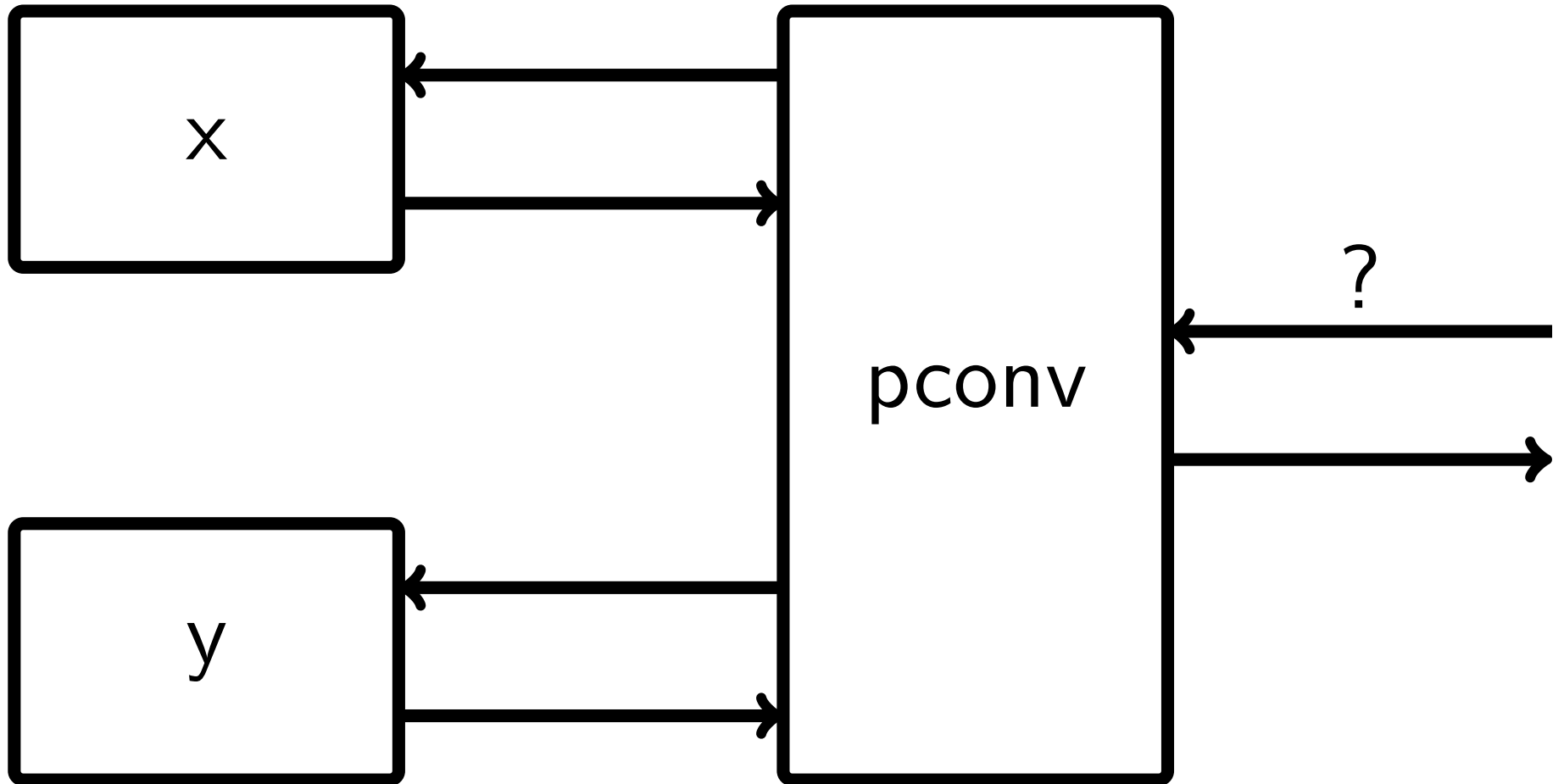
GoI-Pfn

We can't implement **pconv**
if we can't send multiple messages

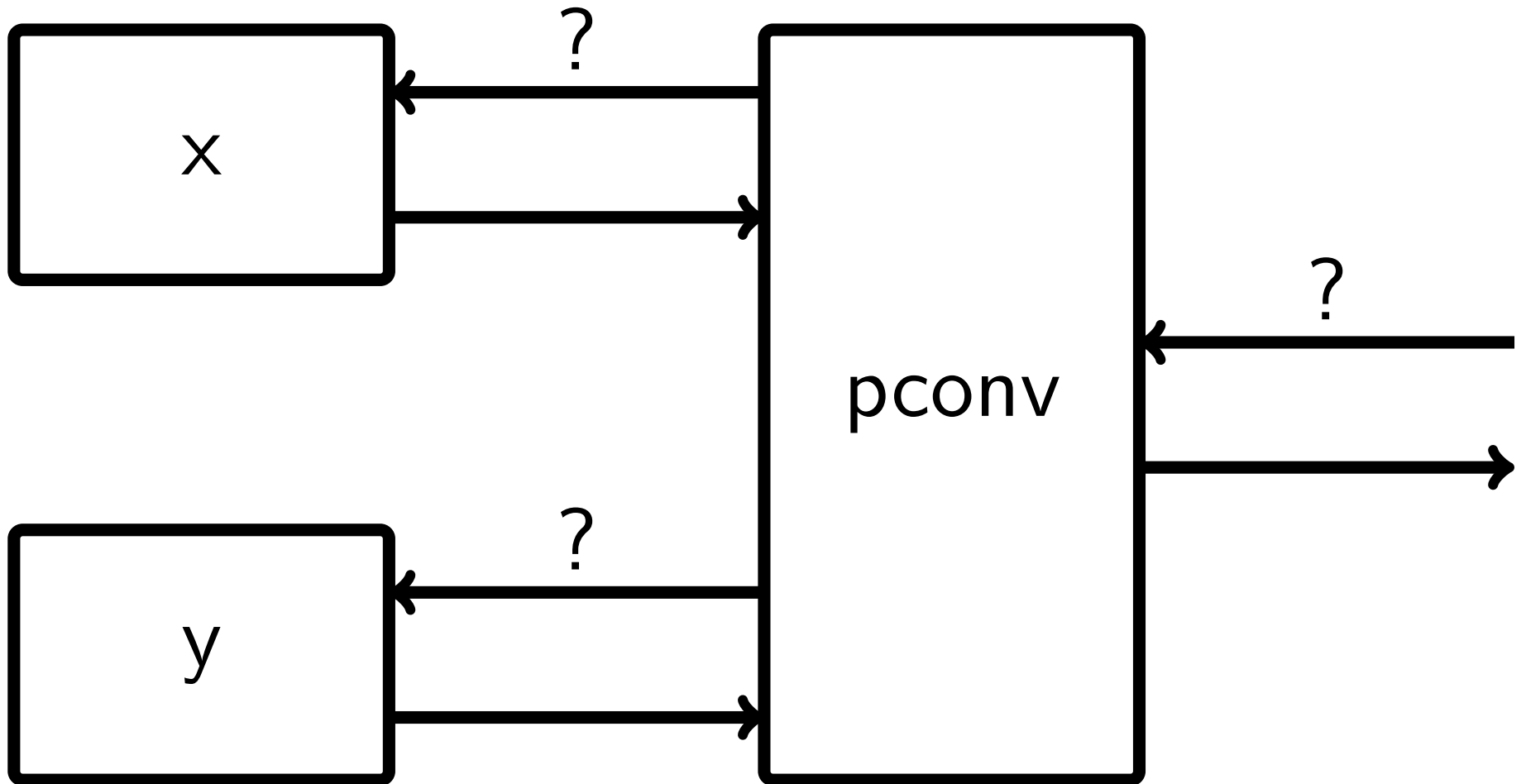
$$\mathbf{pconv}(x : \mathbb{N}_\perp, y : \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

(**pconv** \notin HCoh)

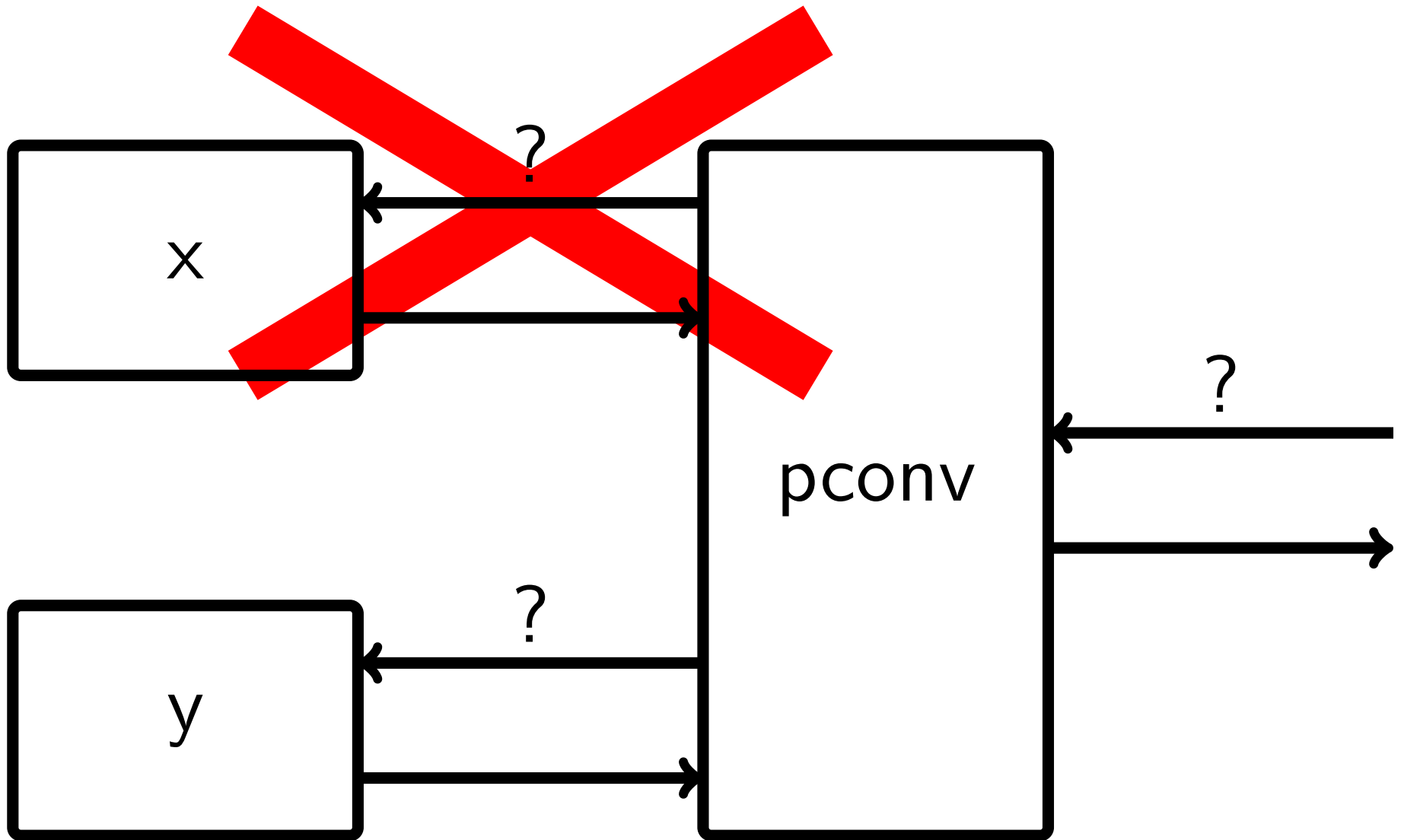
Gol-Pfn



Gol-Pfn



Gol-Pfn



Gol-Pfn

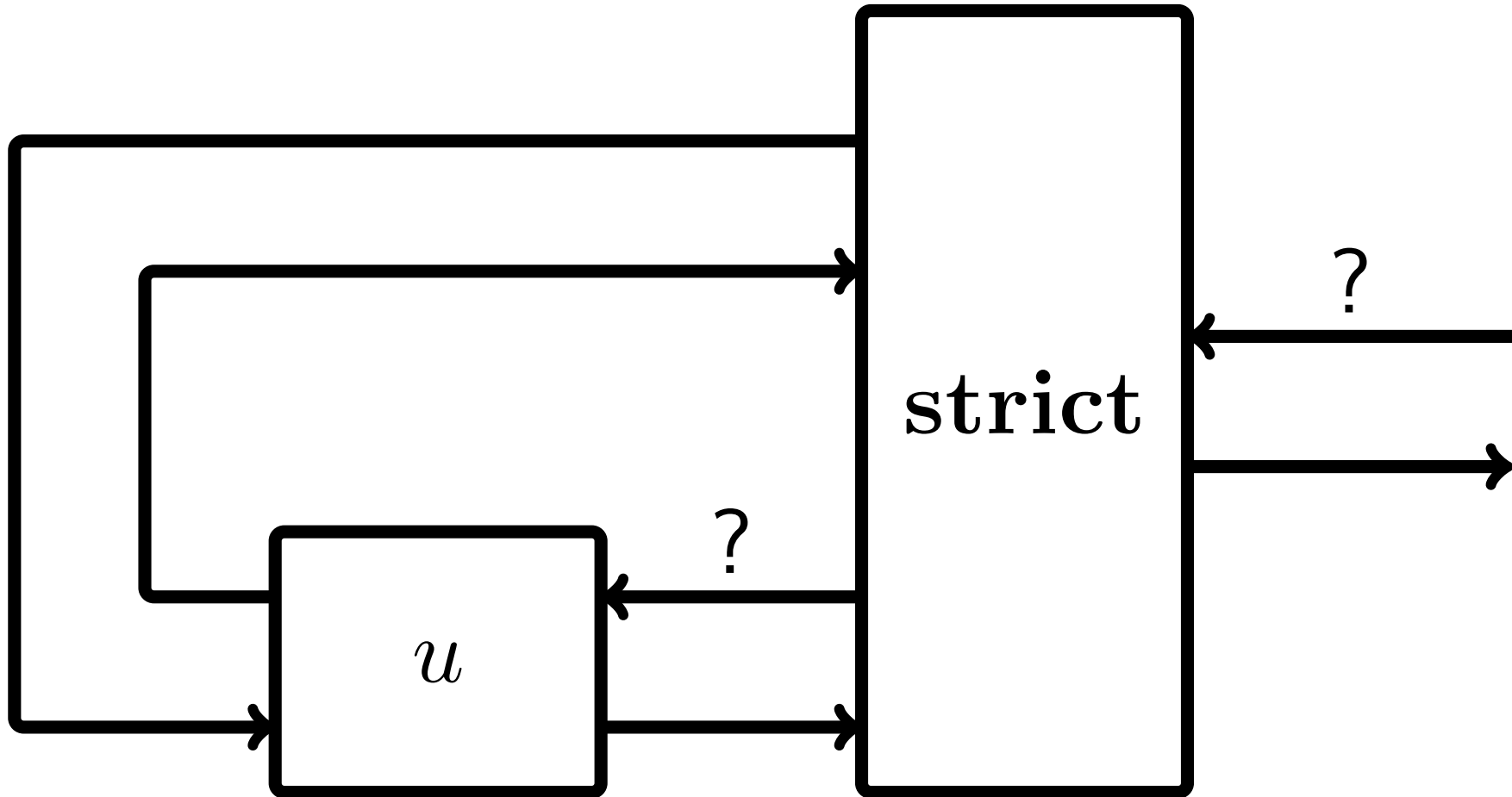
We can implement **strict**
if u don't send multiple messages

strict($u: \mathbb{N}_\perp \rightarrow \mathbb{N}_\perp$)

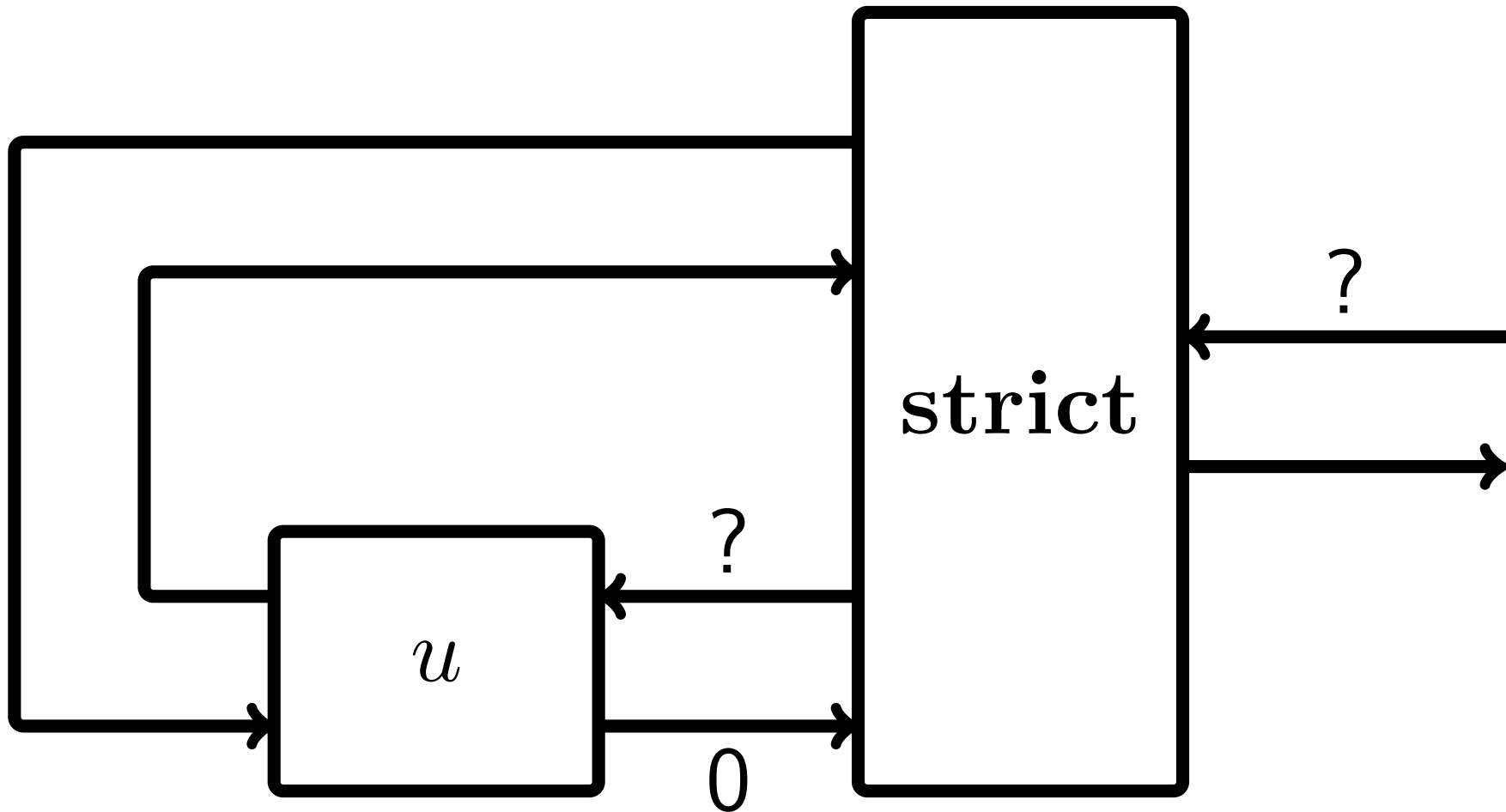
$$\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

(**strict** \in **HCoh**)

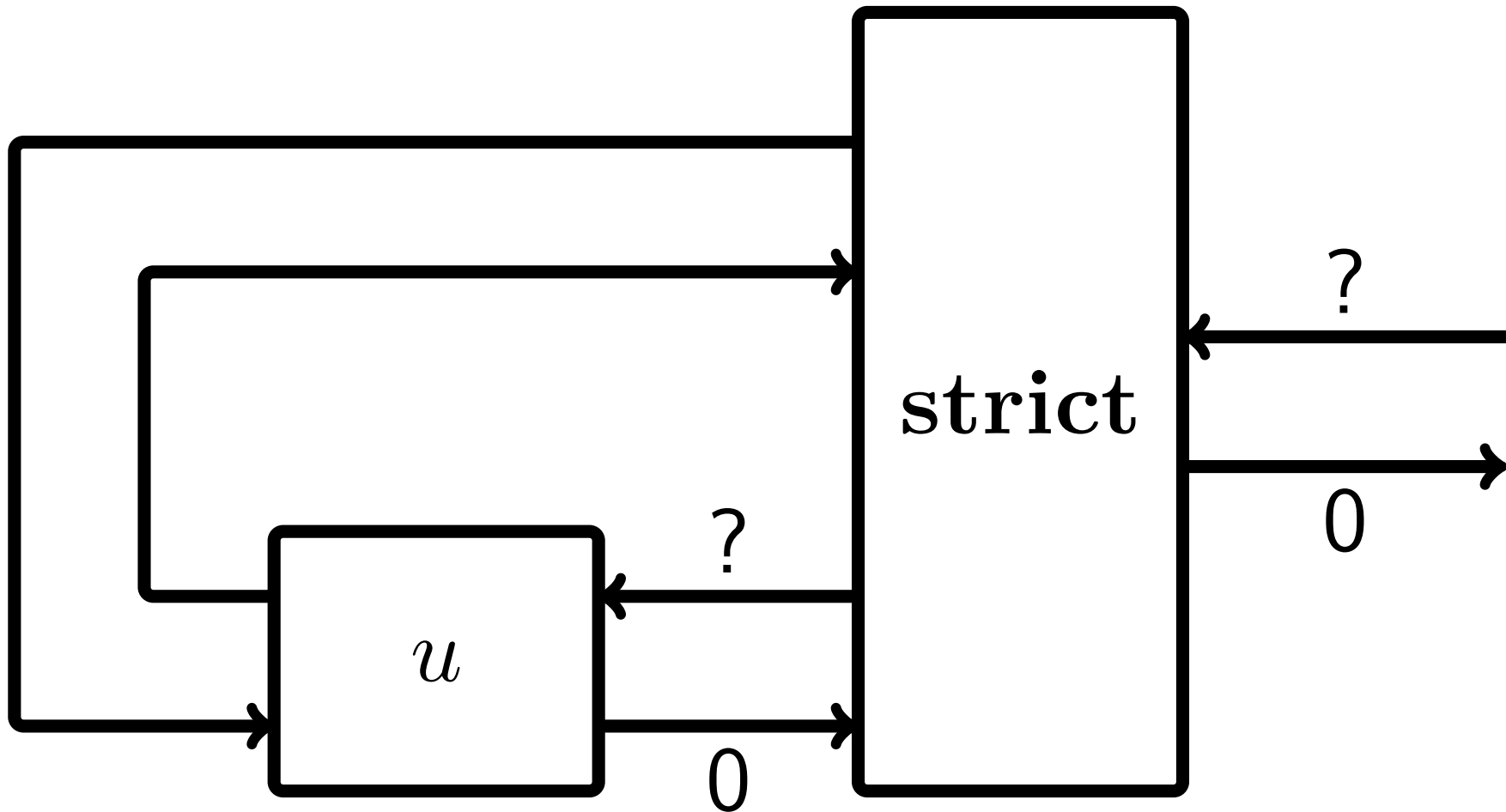
Implementation



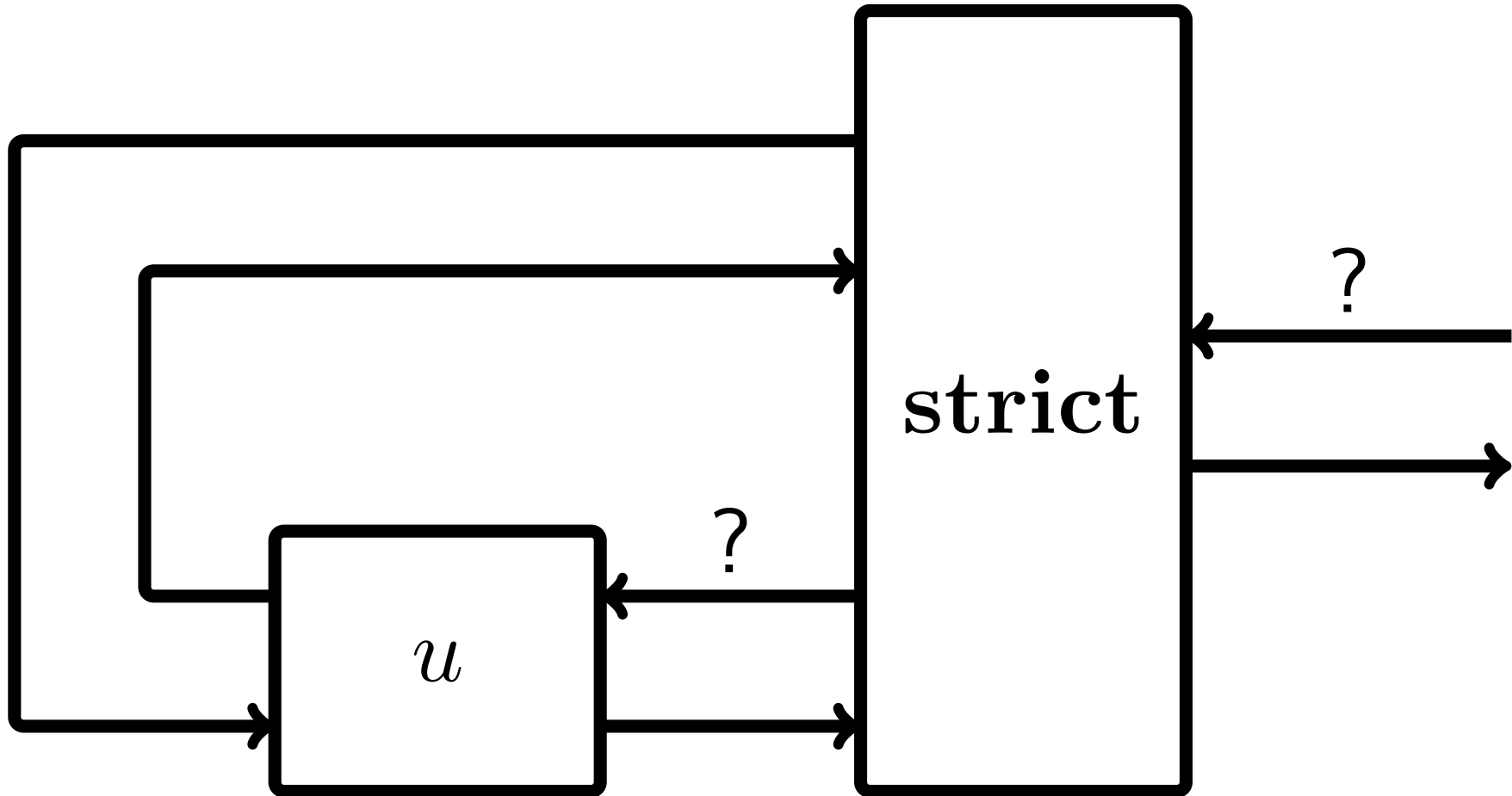
Implementation



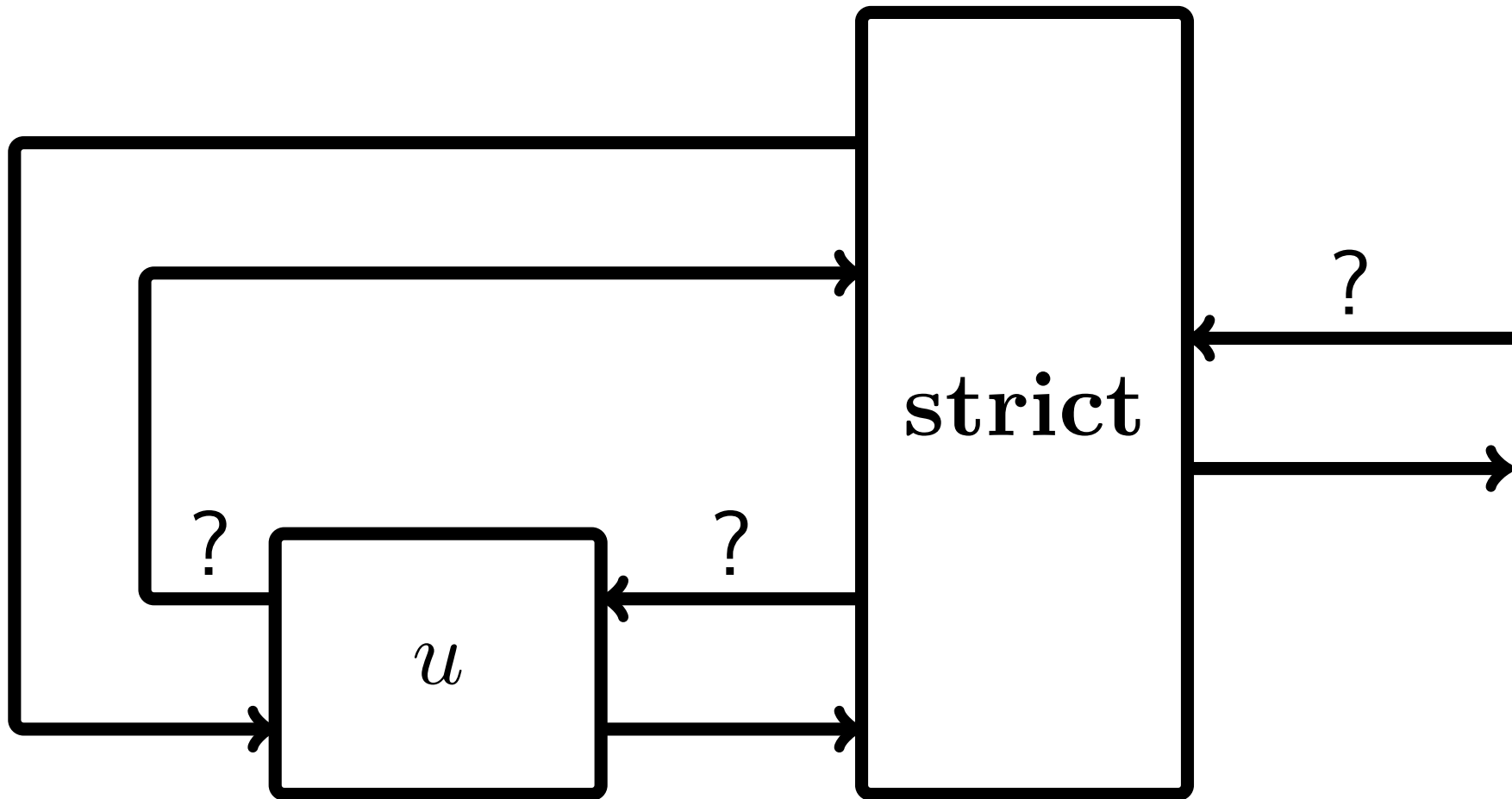
Implementation



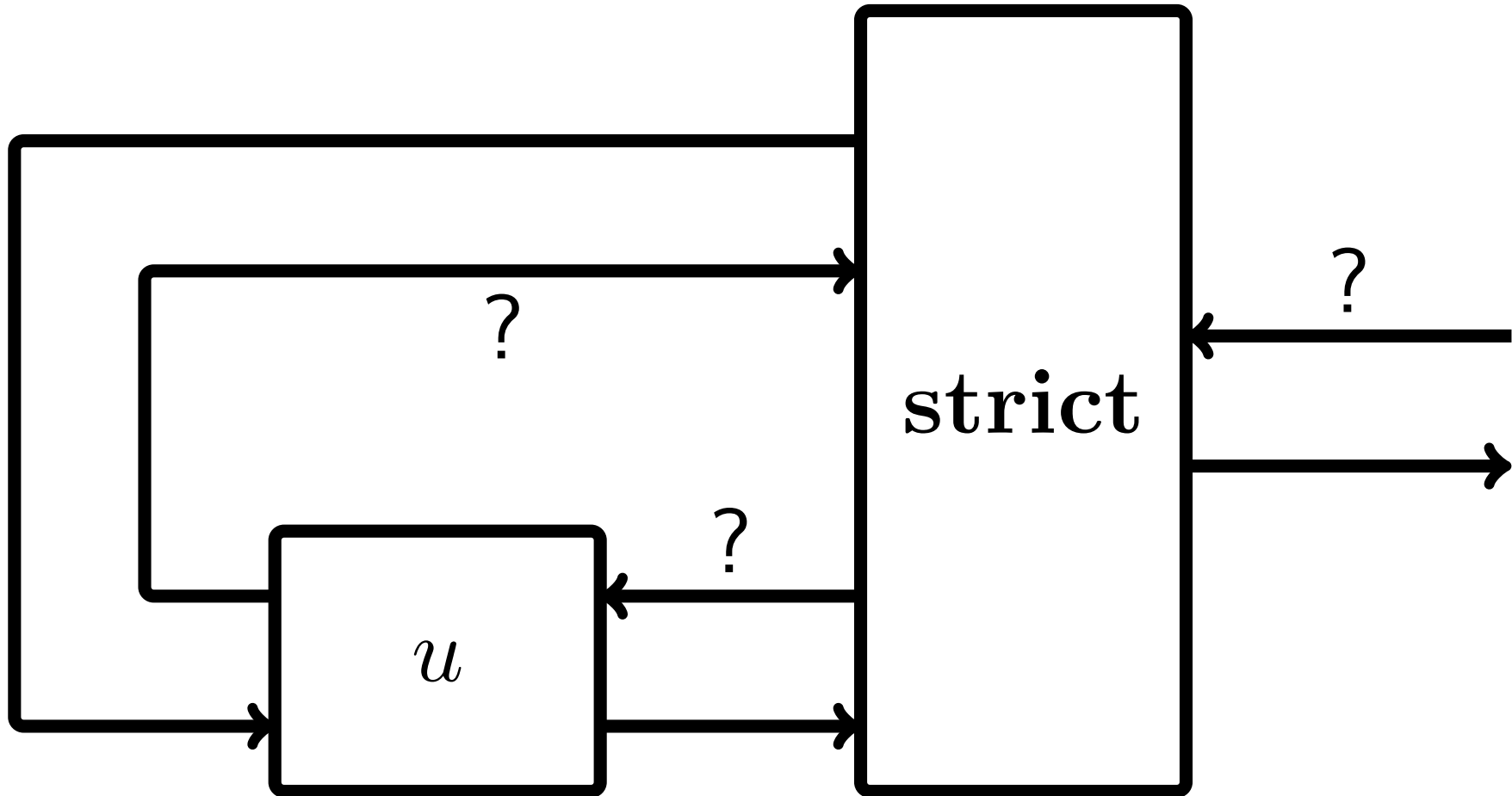
Implementation



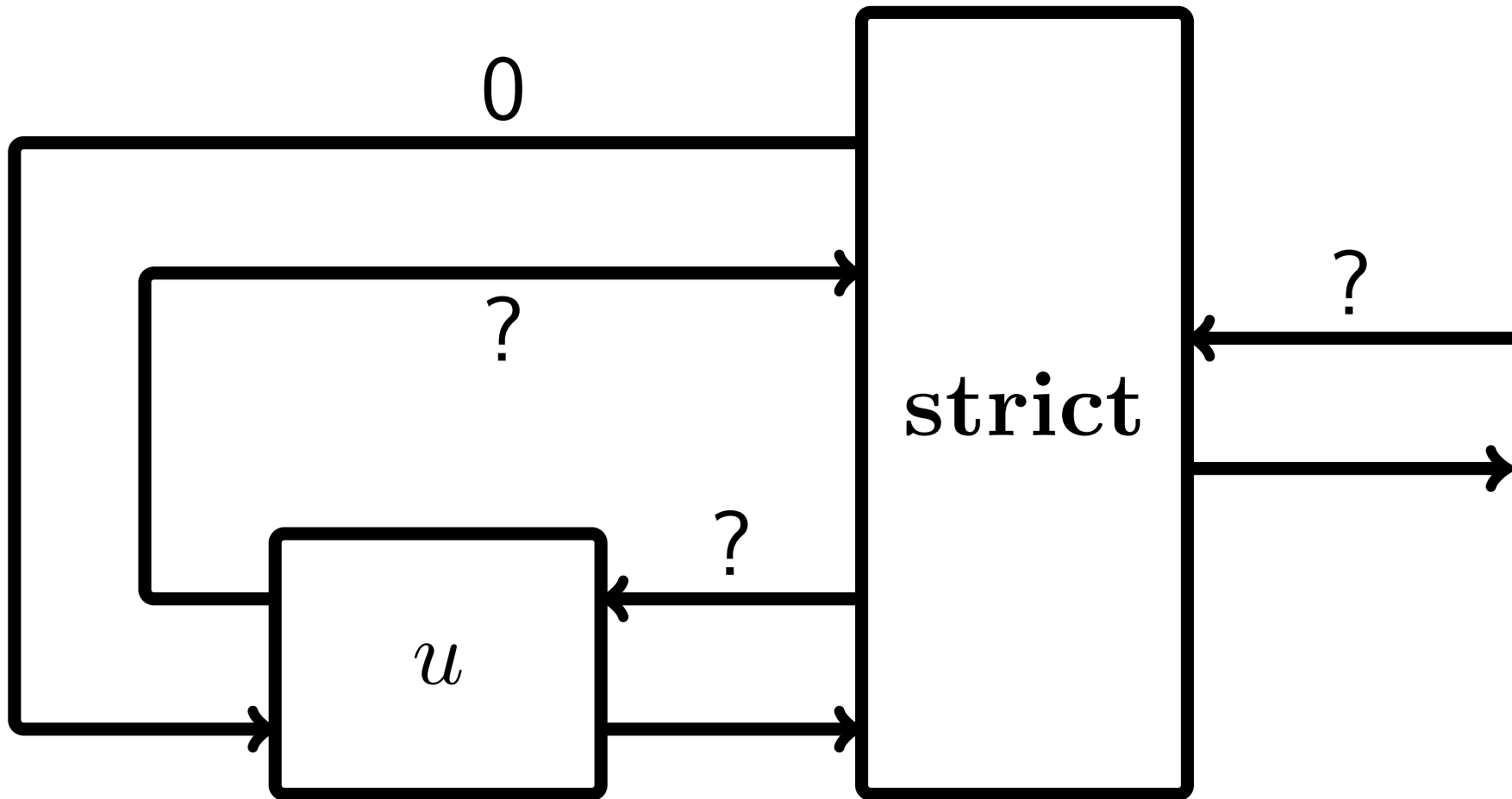
Implementation



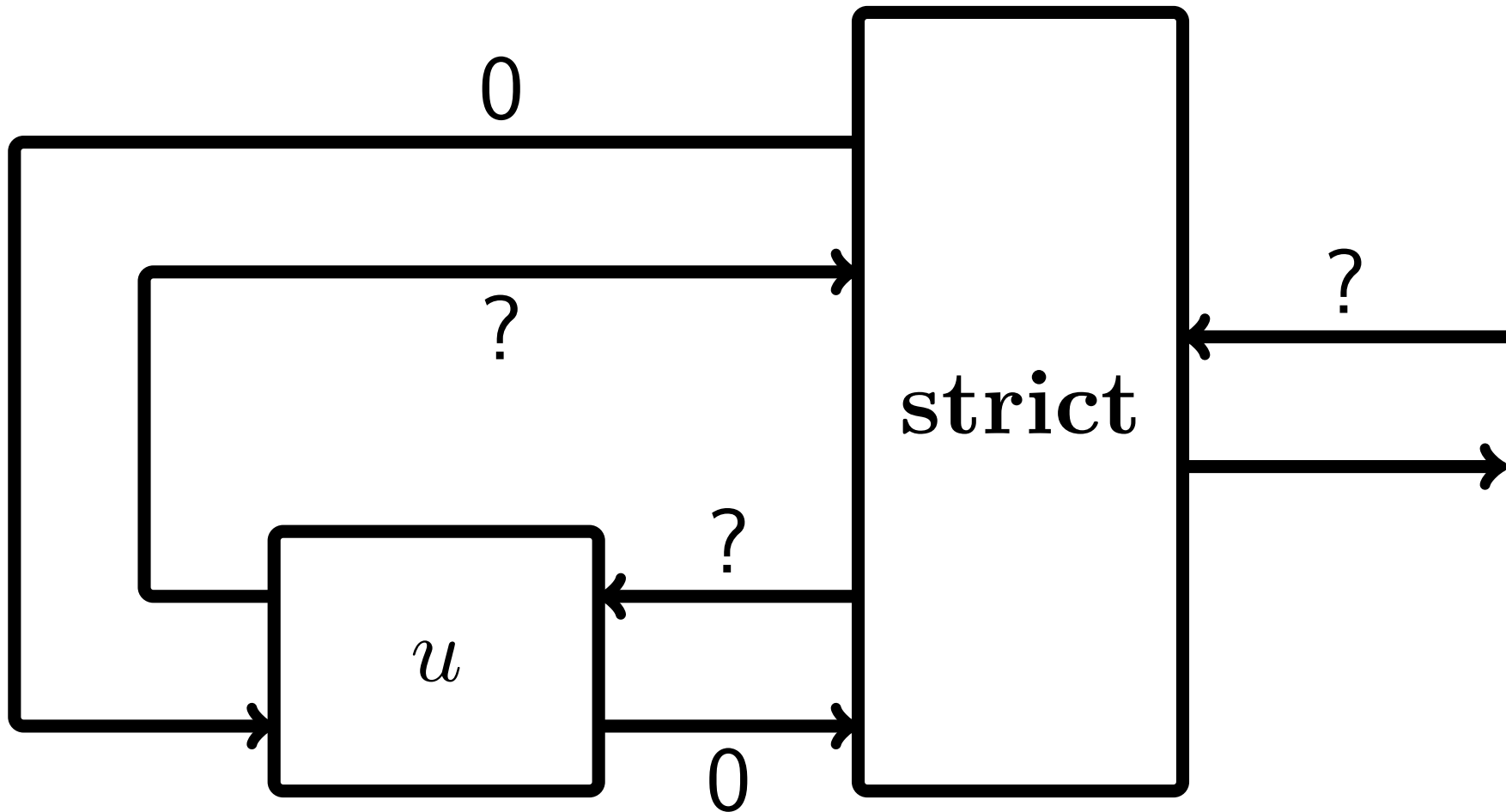
Implementation



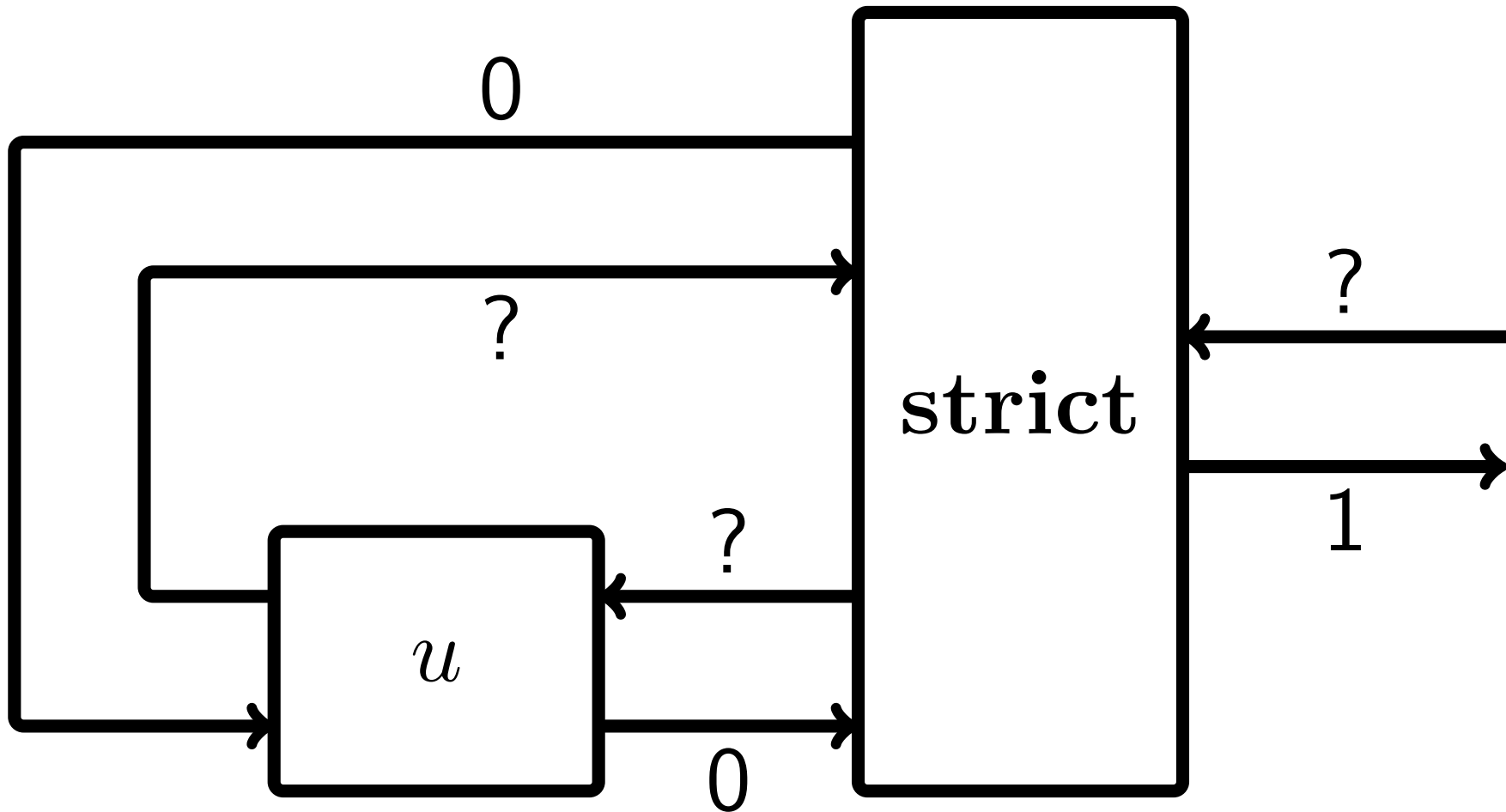
Implementation



Implementation



Implementation



GoI-Rel

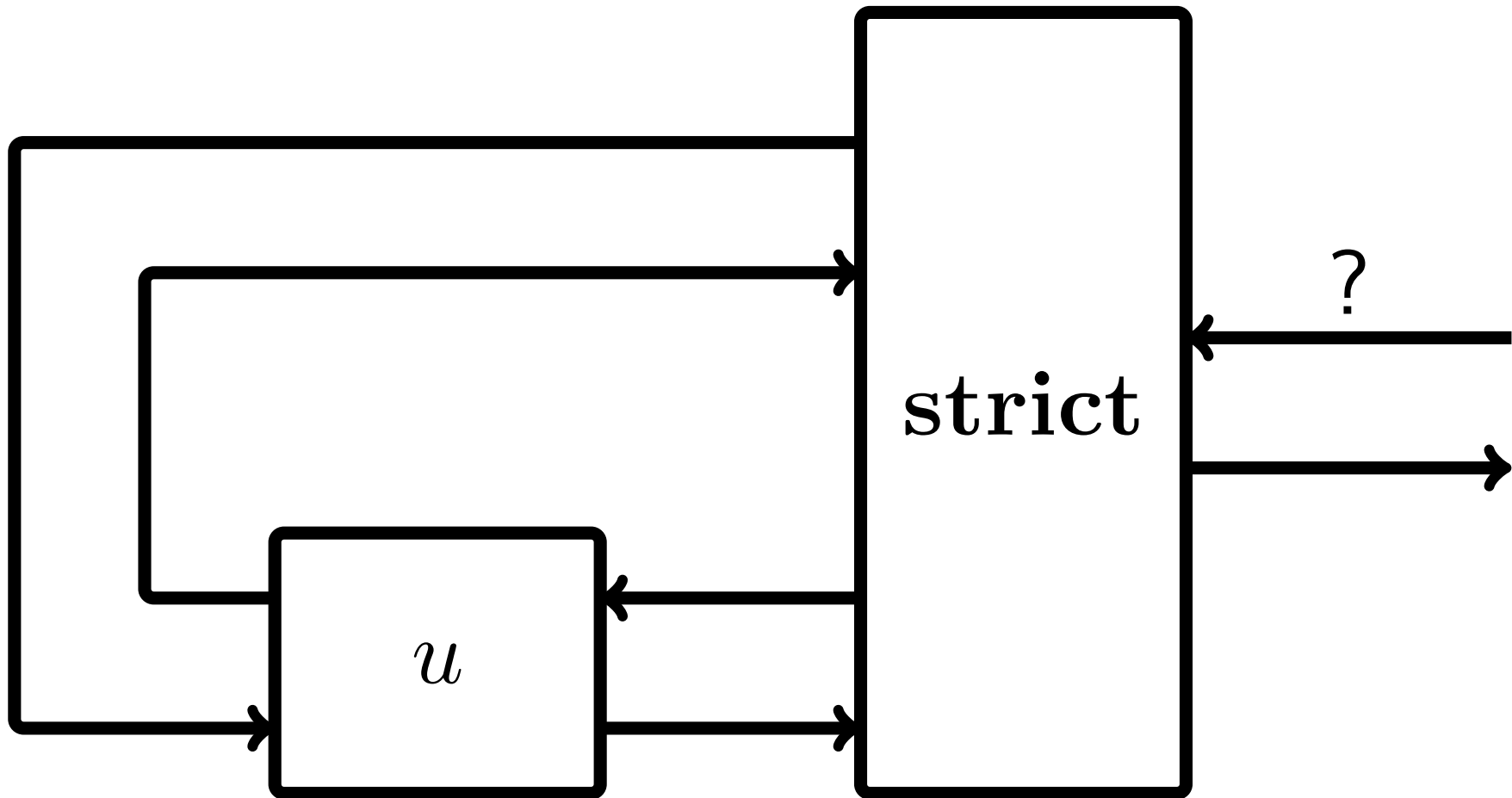
We can't implement **strict**
if u may send multiple messages

strict($u: \mathbb{N}_\perp \rightarrow \mathbb{N}_\perp$)

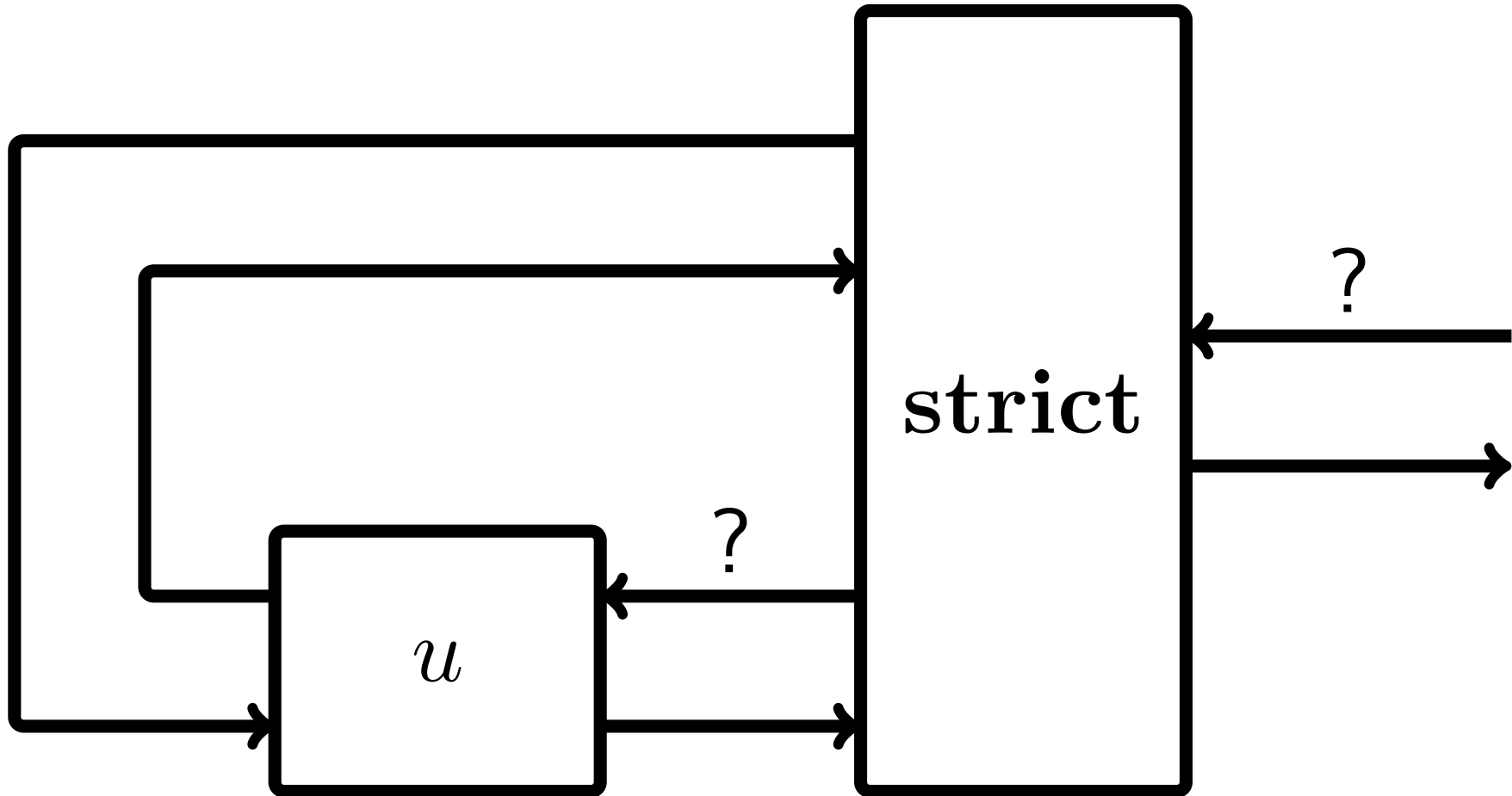
$$\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

(**strict** \notin **Cpo**)

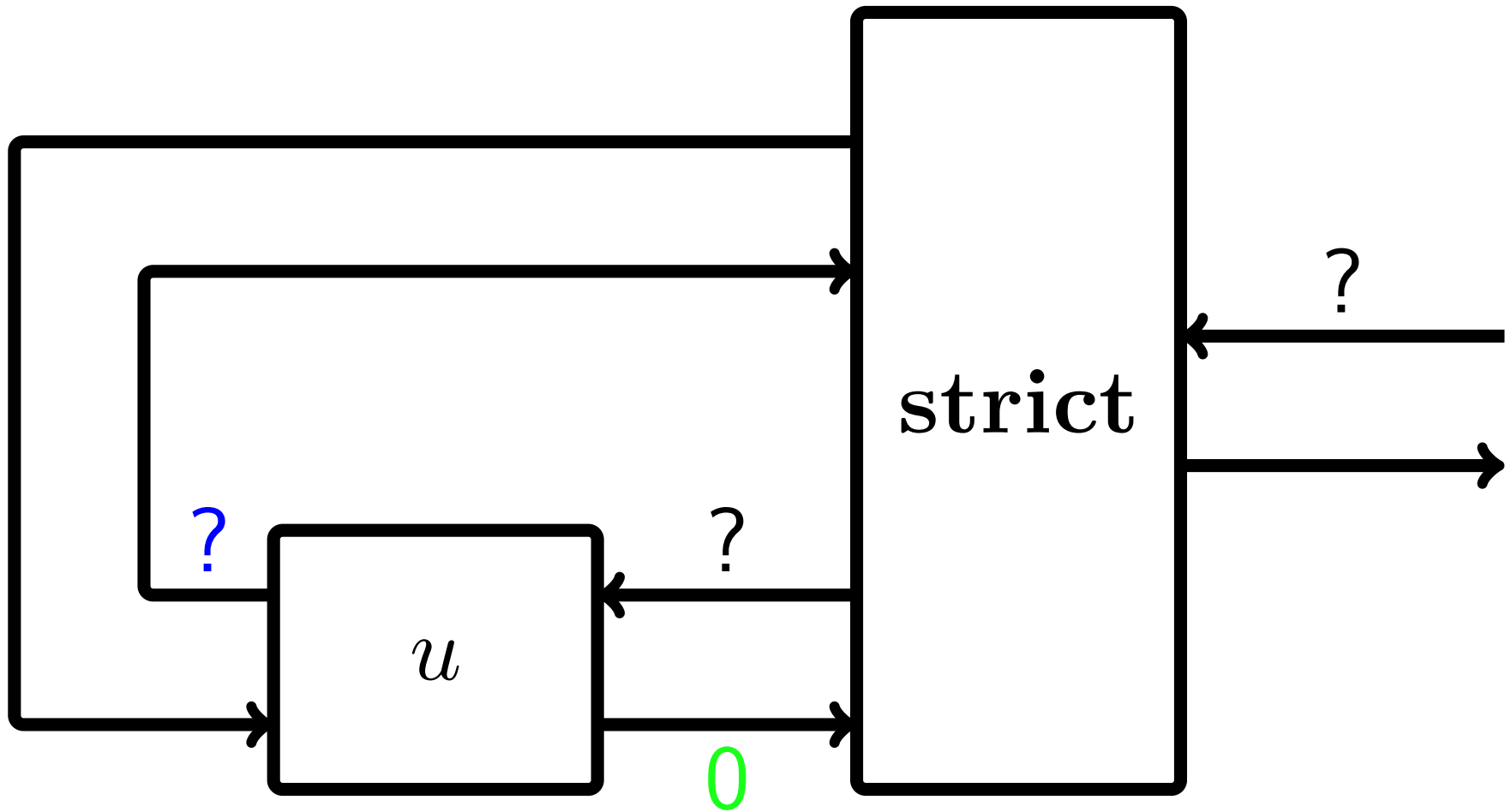
Gol-Rel



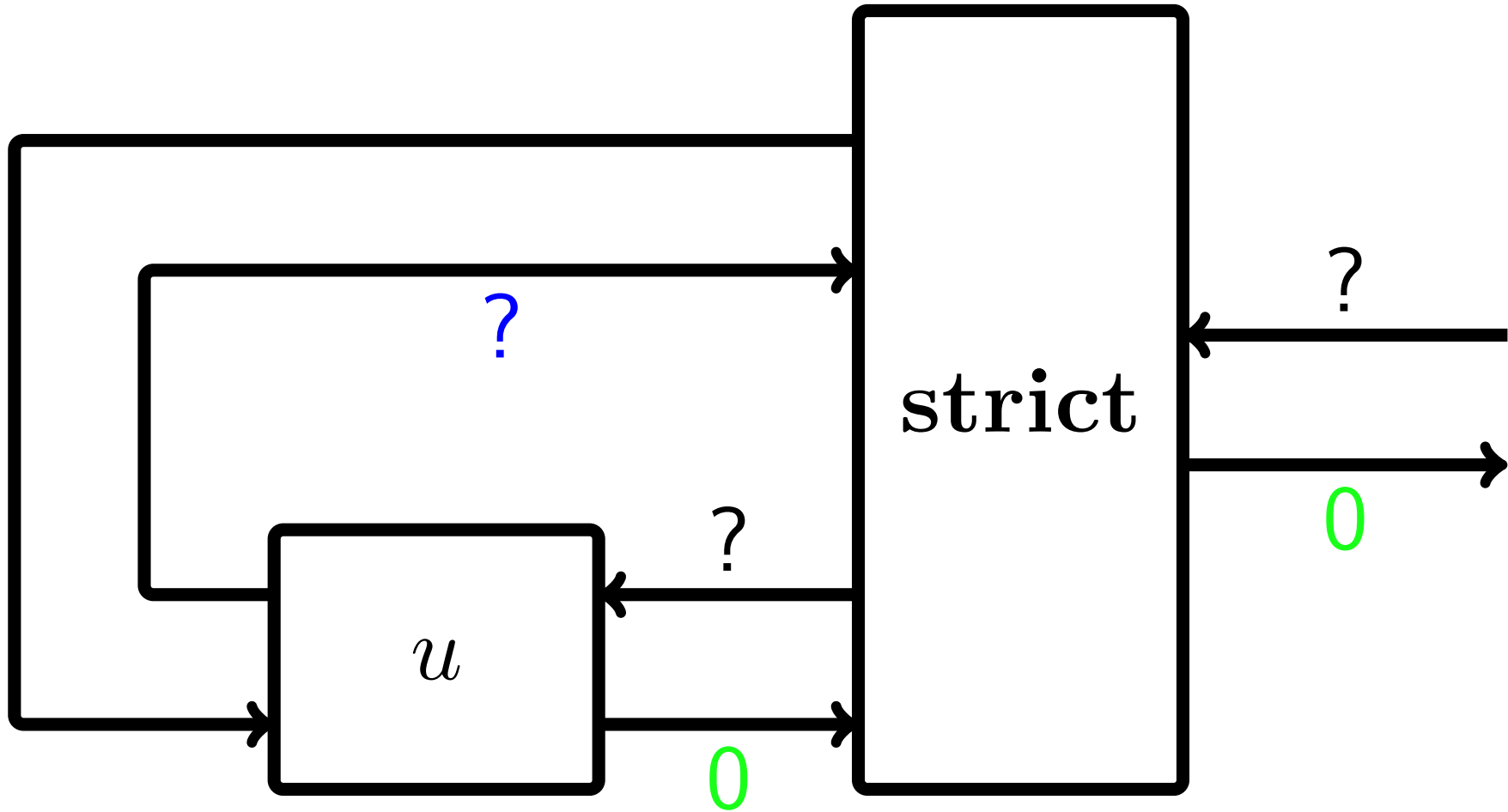
Gol-Rel



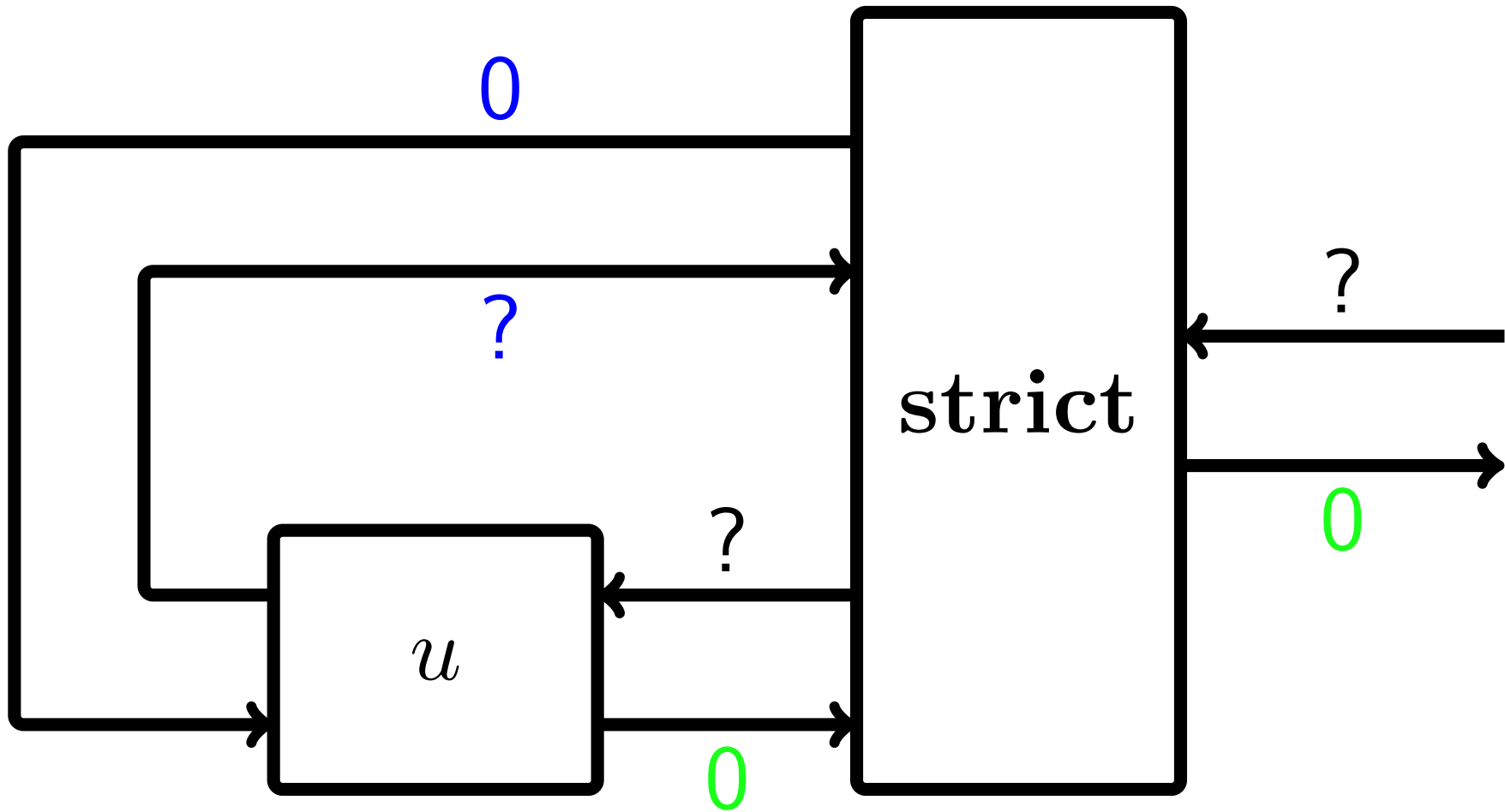
GoI-Rel



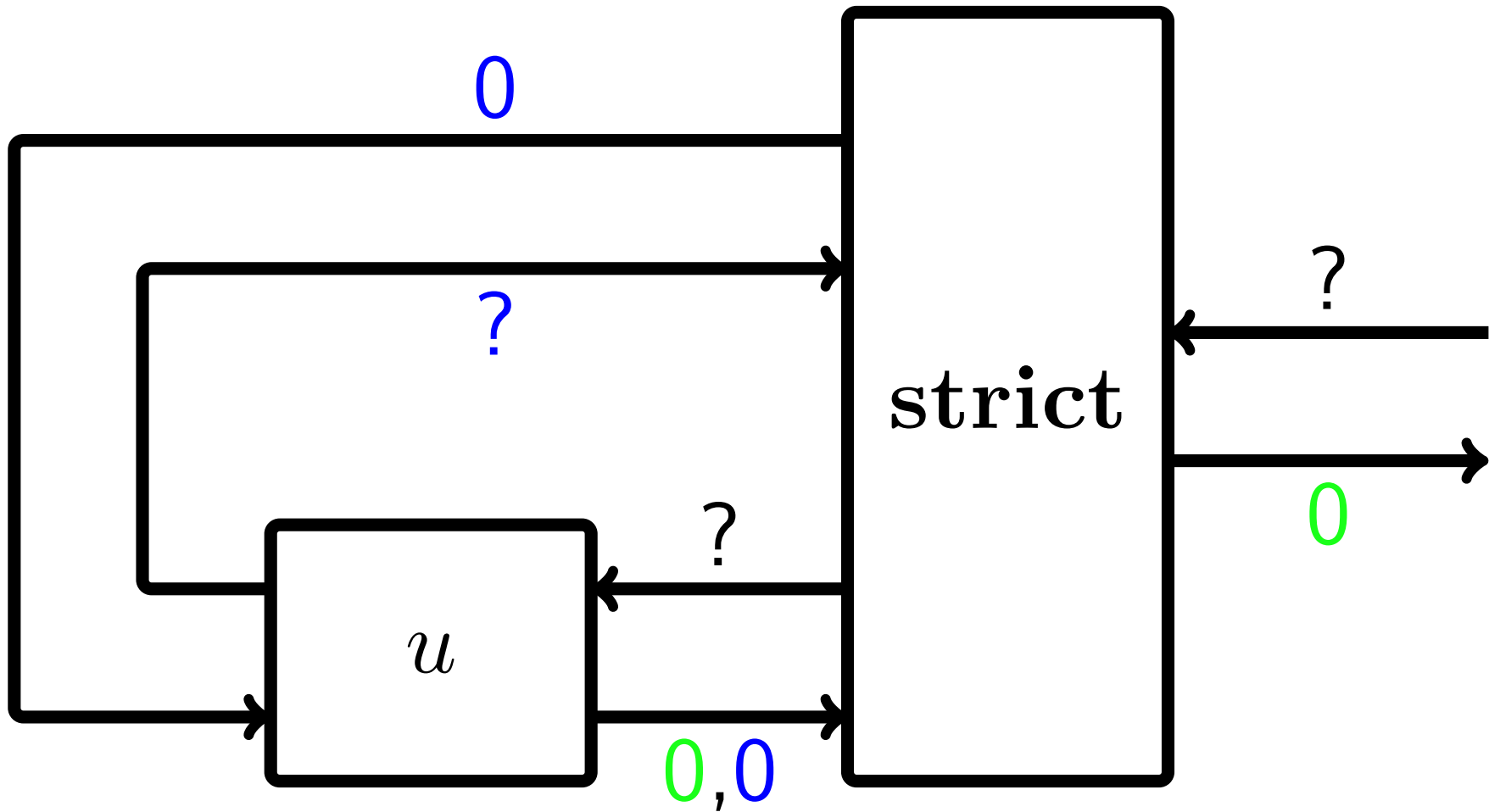
GoI-Rel



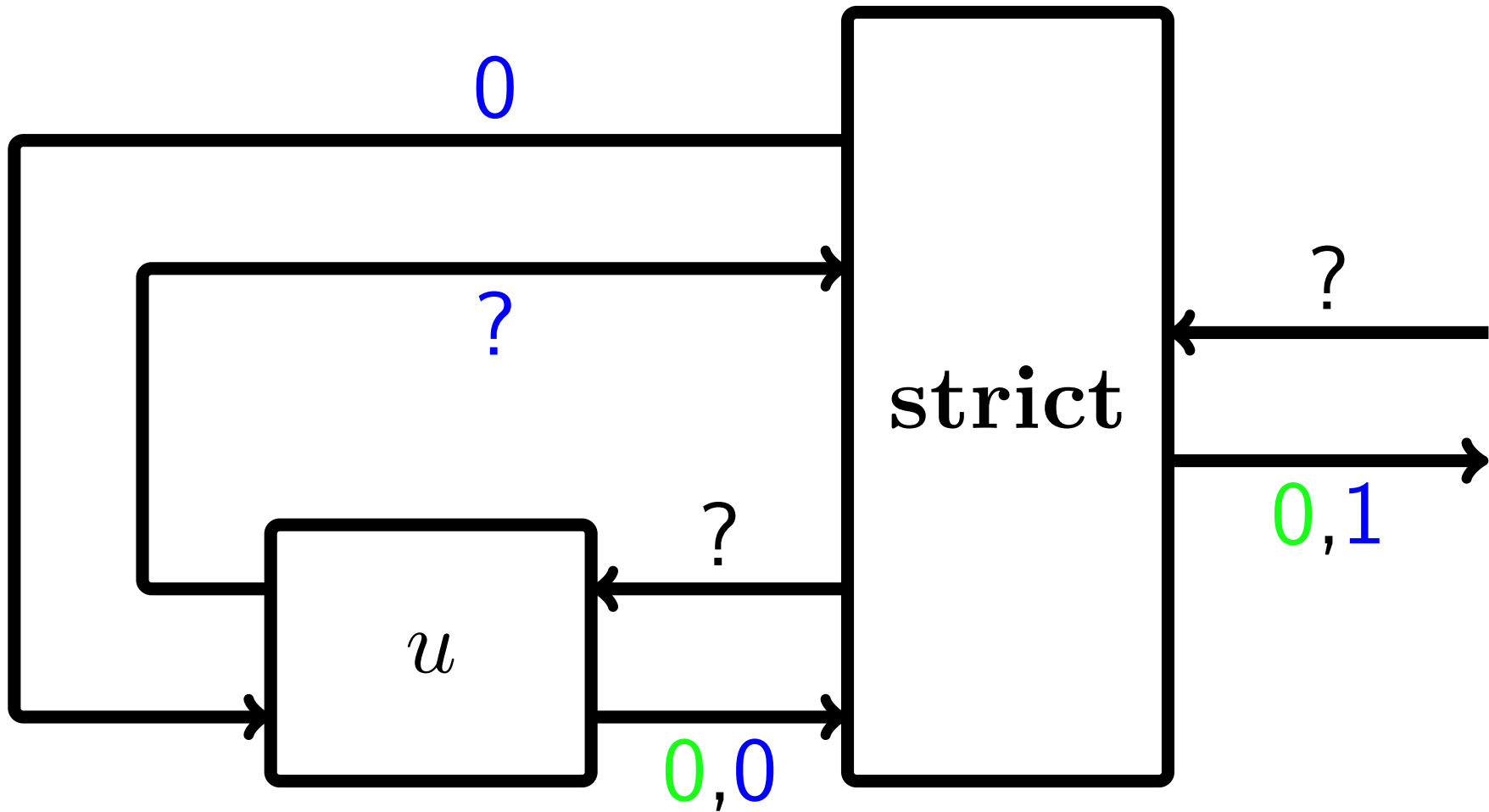
GoI-Rel



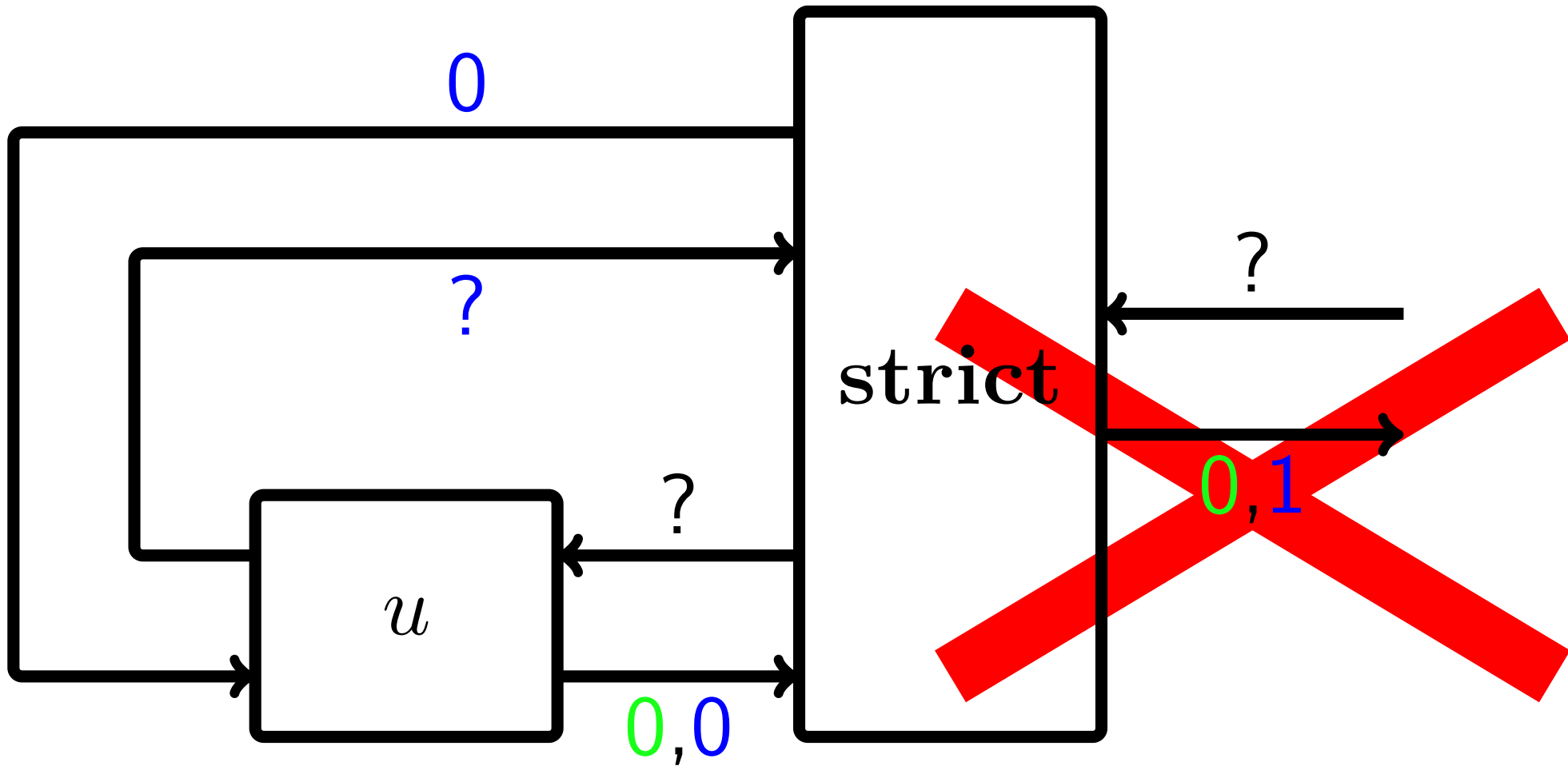
GoI-Rel



GoI-Rel



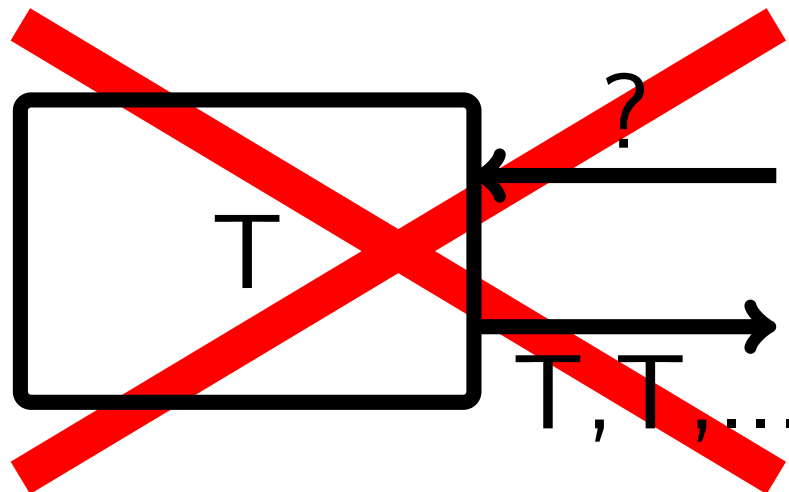
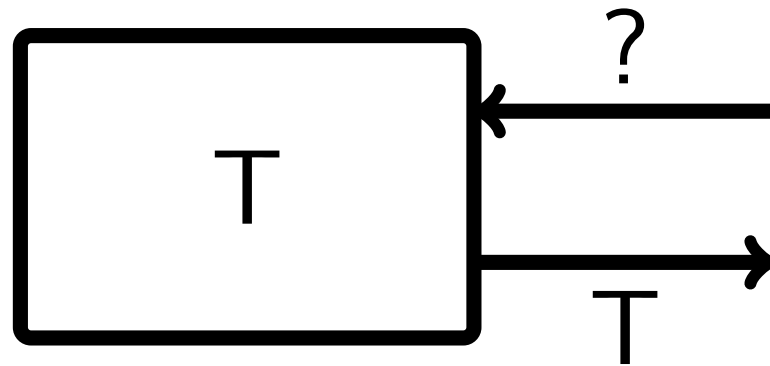
GoI-Rel



GoI-wRel

We can send multiple messages.

BUT, each number must be presented by



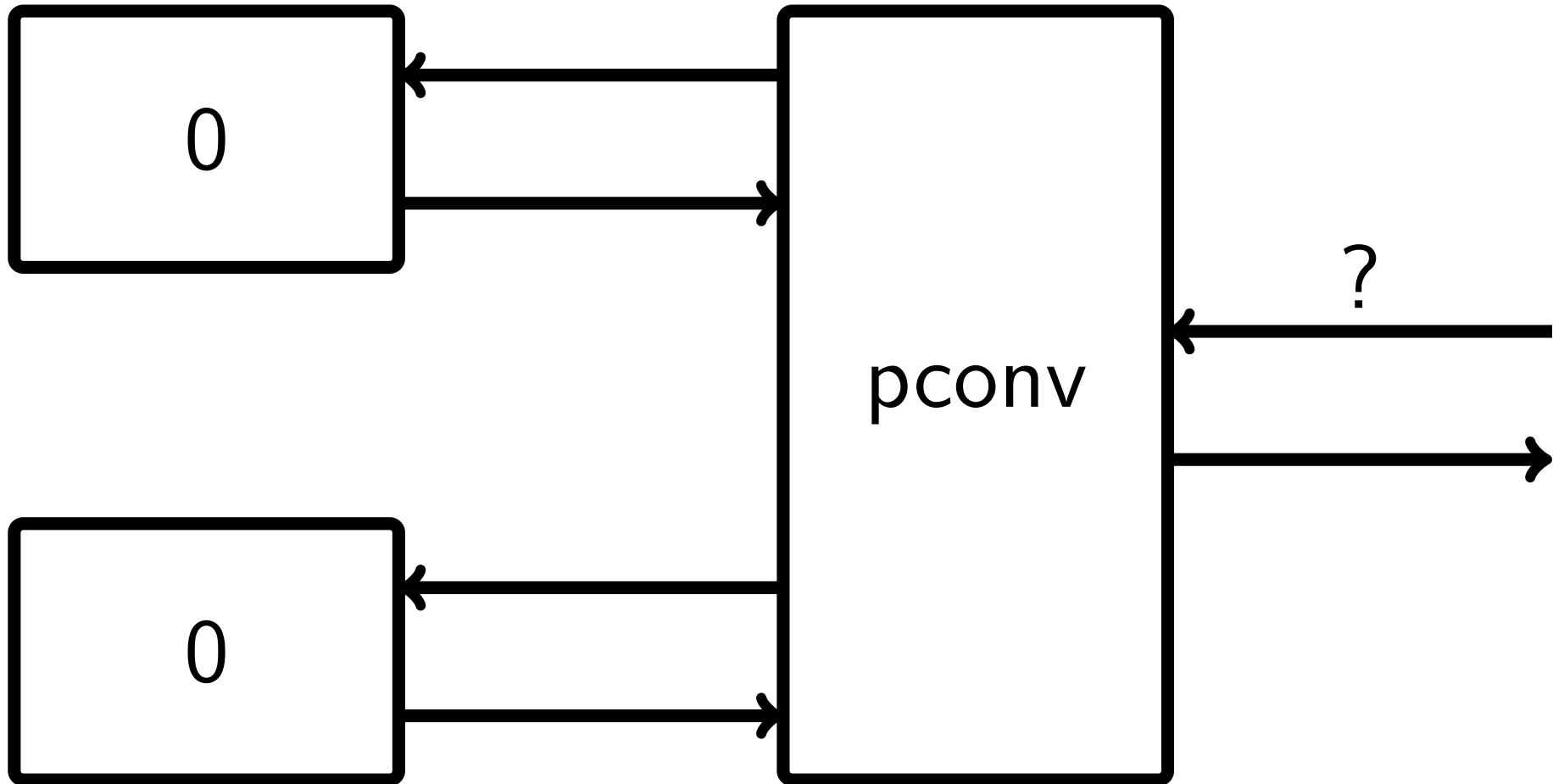
GoI-wRel

We can't **quantitatively** implement **pconv** even if we can send multiple messages

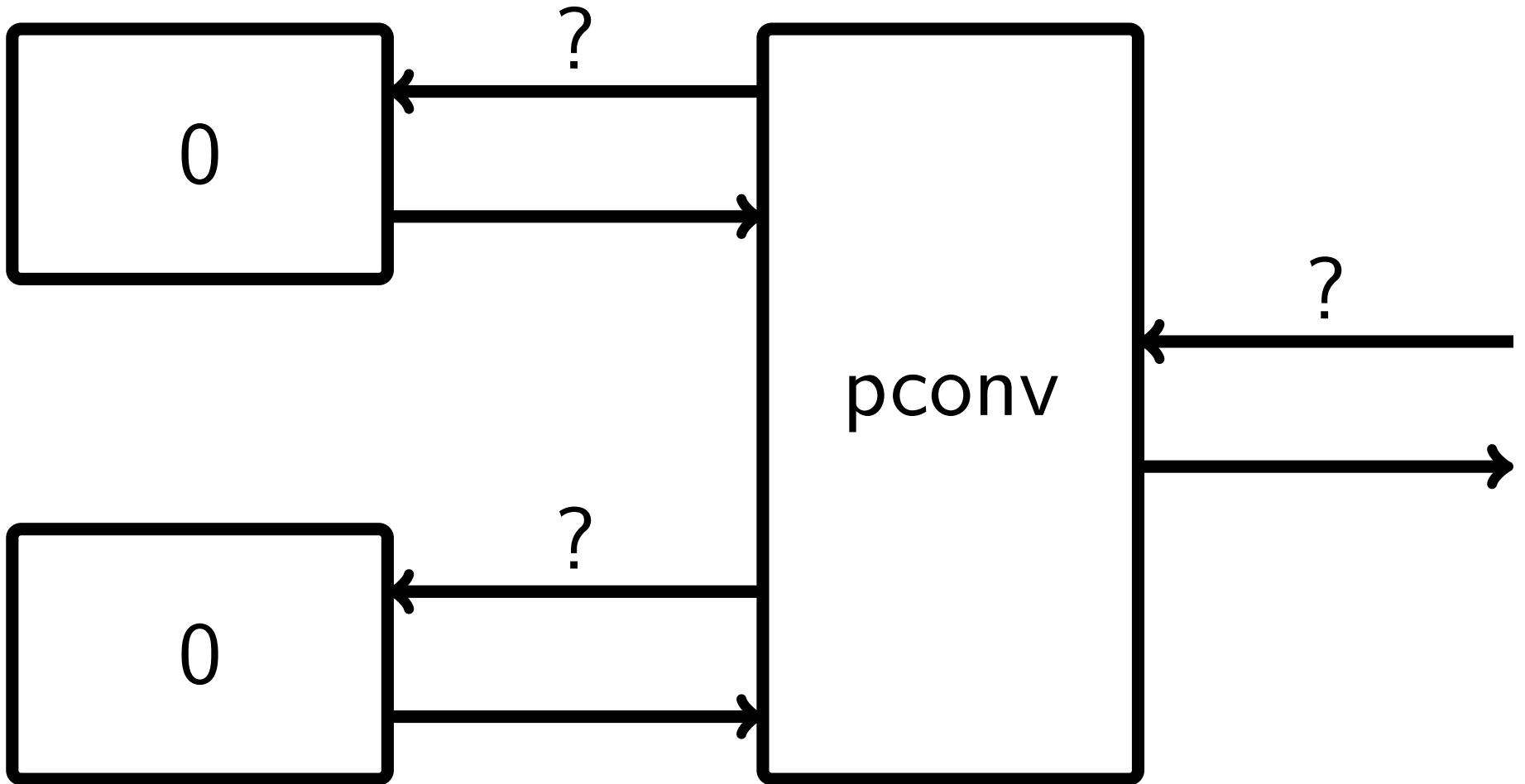
$$\mathbf{pconv}(x : \mathbb{N}_\perp, y : \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

(**pconv** \notin **Coh**)

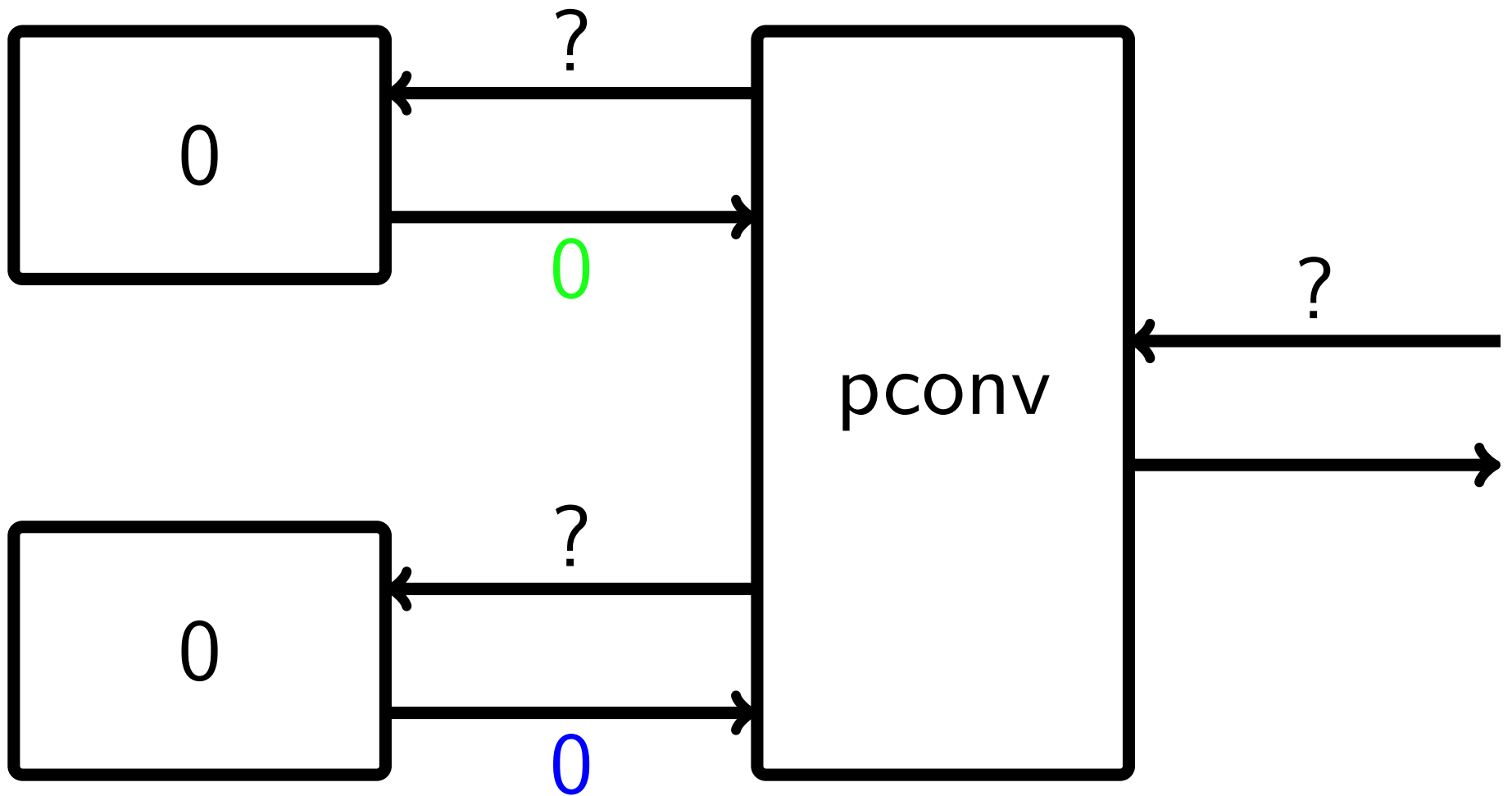
Gol-wRel



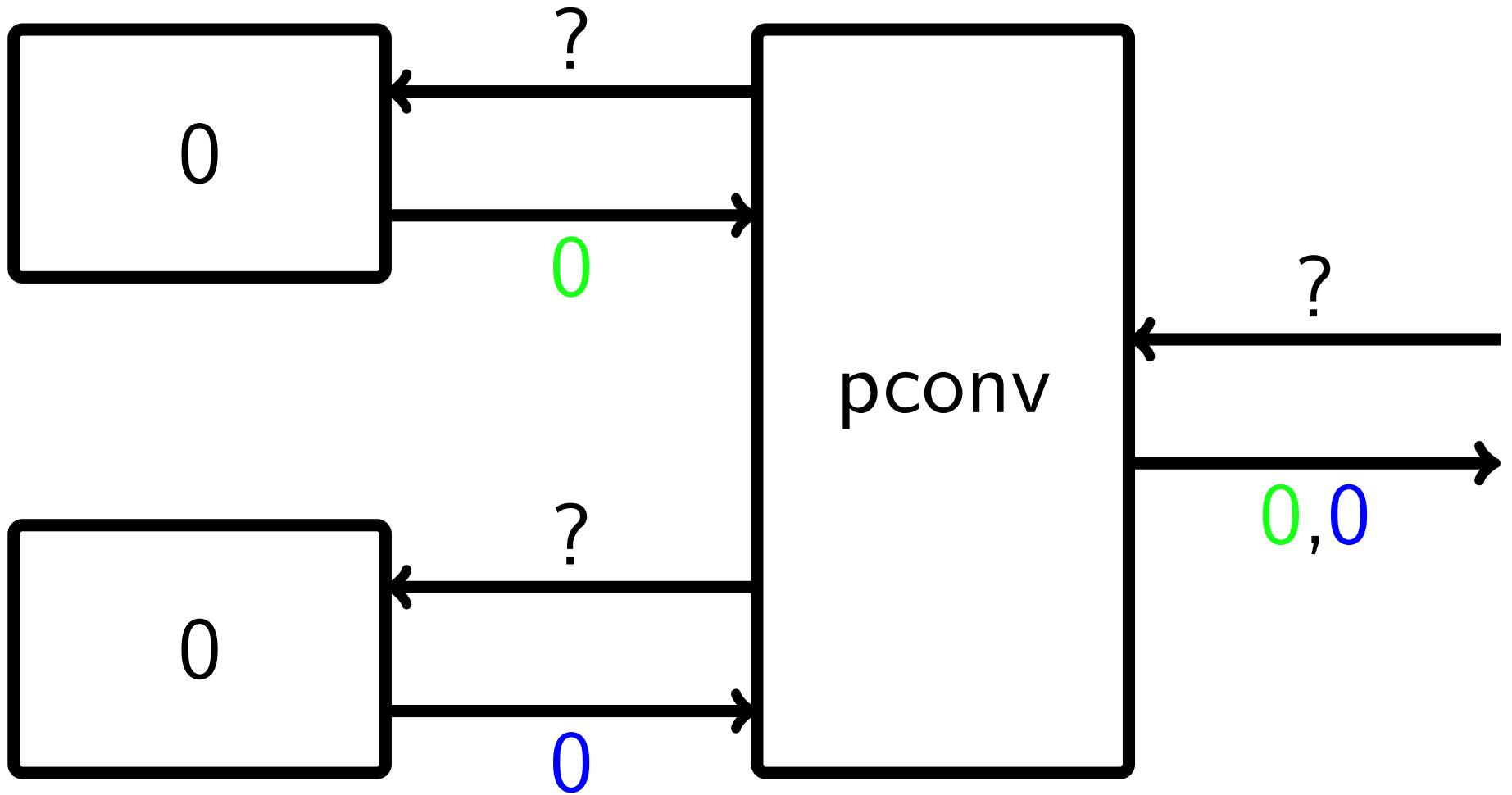
Gol-wRel



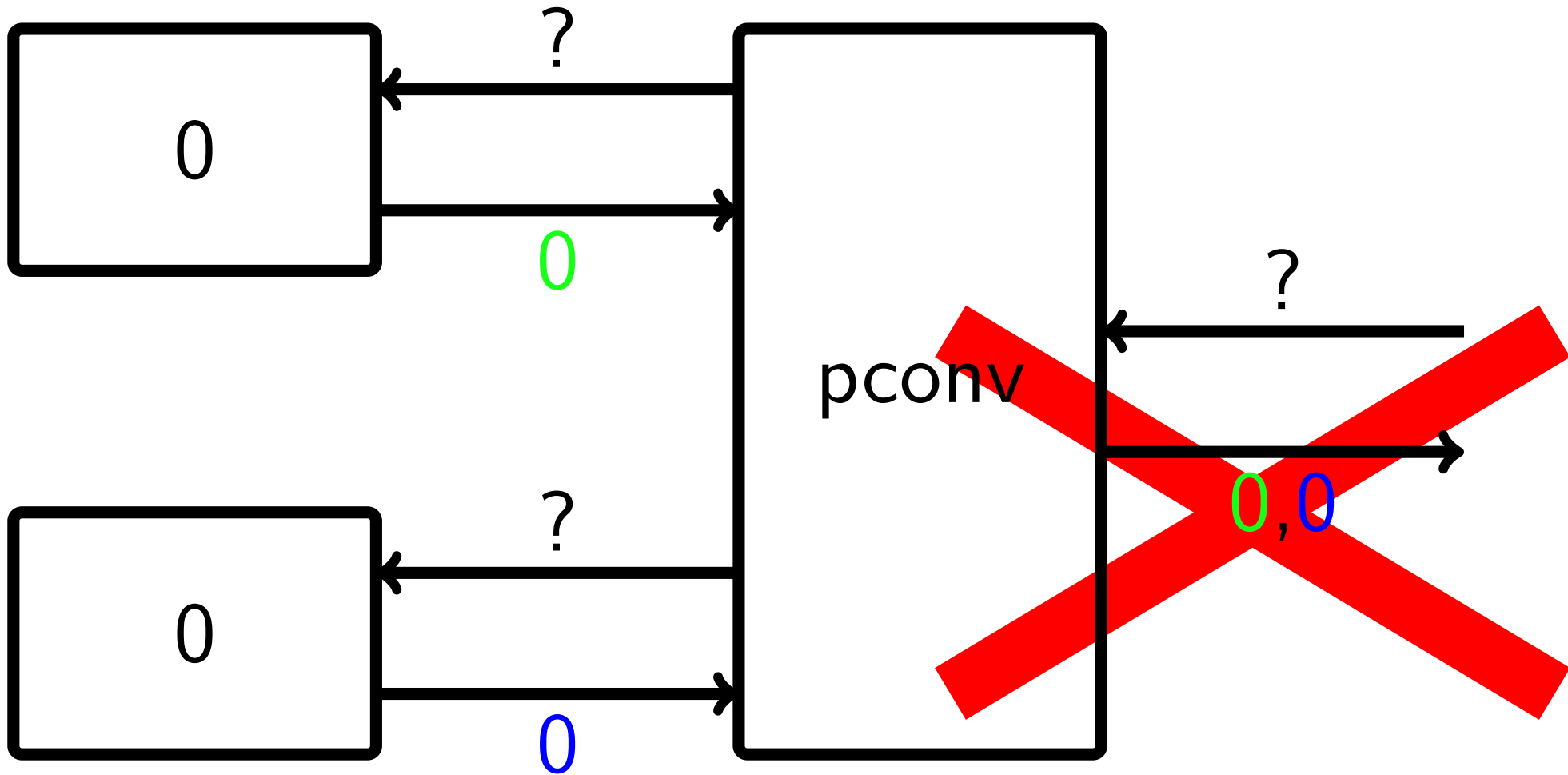
Gol-wRel



Gol-wRel



Gol-wRel



GoI-wRel

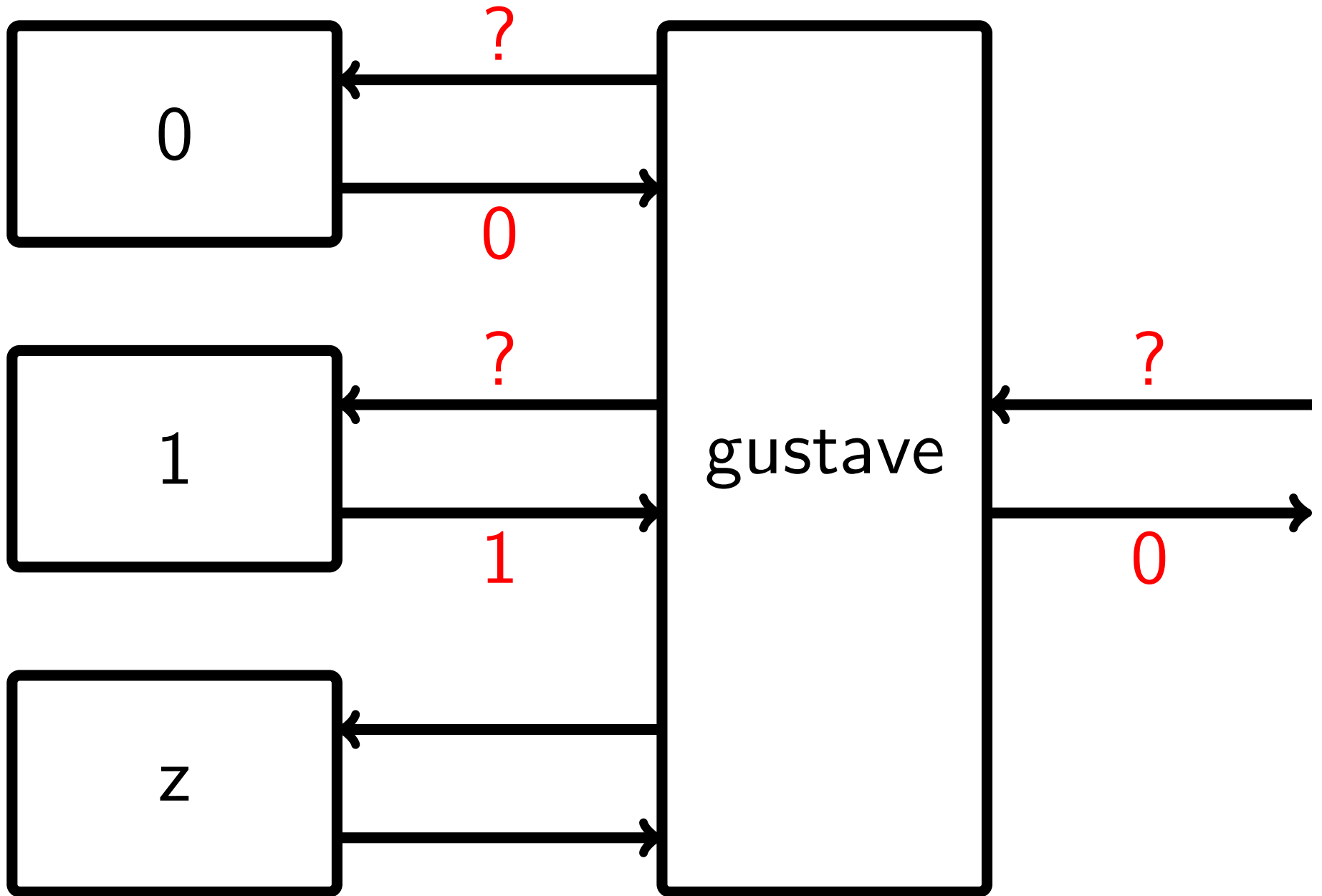
We can **quantitatively** implement **gustave** if we can send multiple messages

gustave($x : \mathbb{N}_\perp, y : \mathbb{N}_\perp, z : \mathbb{N}_\perp$)

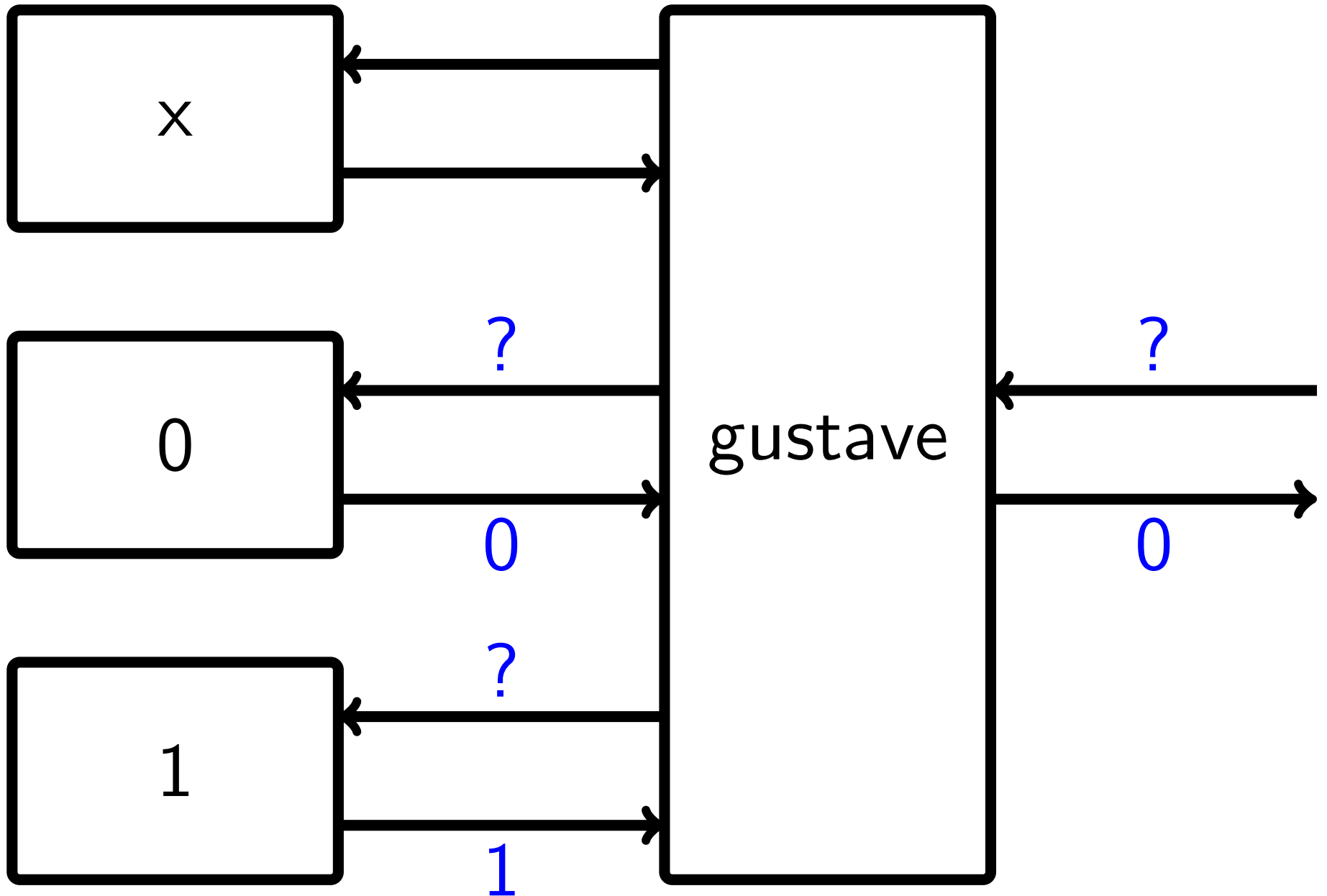
$$\underline{\underline{\text{def}}} \left\{ \begin{array}{ll} 0 & \text{if } (x, y) = (0, 1) \\ 0 & \text{if } (y, z) = (0, 1) \\ 0 & \text{if } (z, x) = (0, 1) \\ \perp & \text{otherwise} \end{array} \right.$$

(**gustave** \in **Coh**)

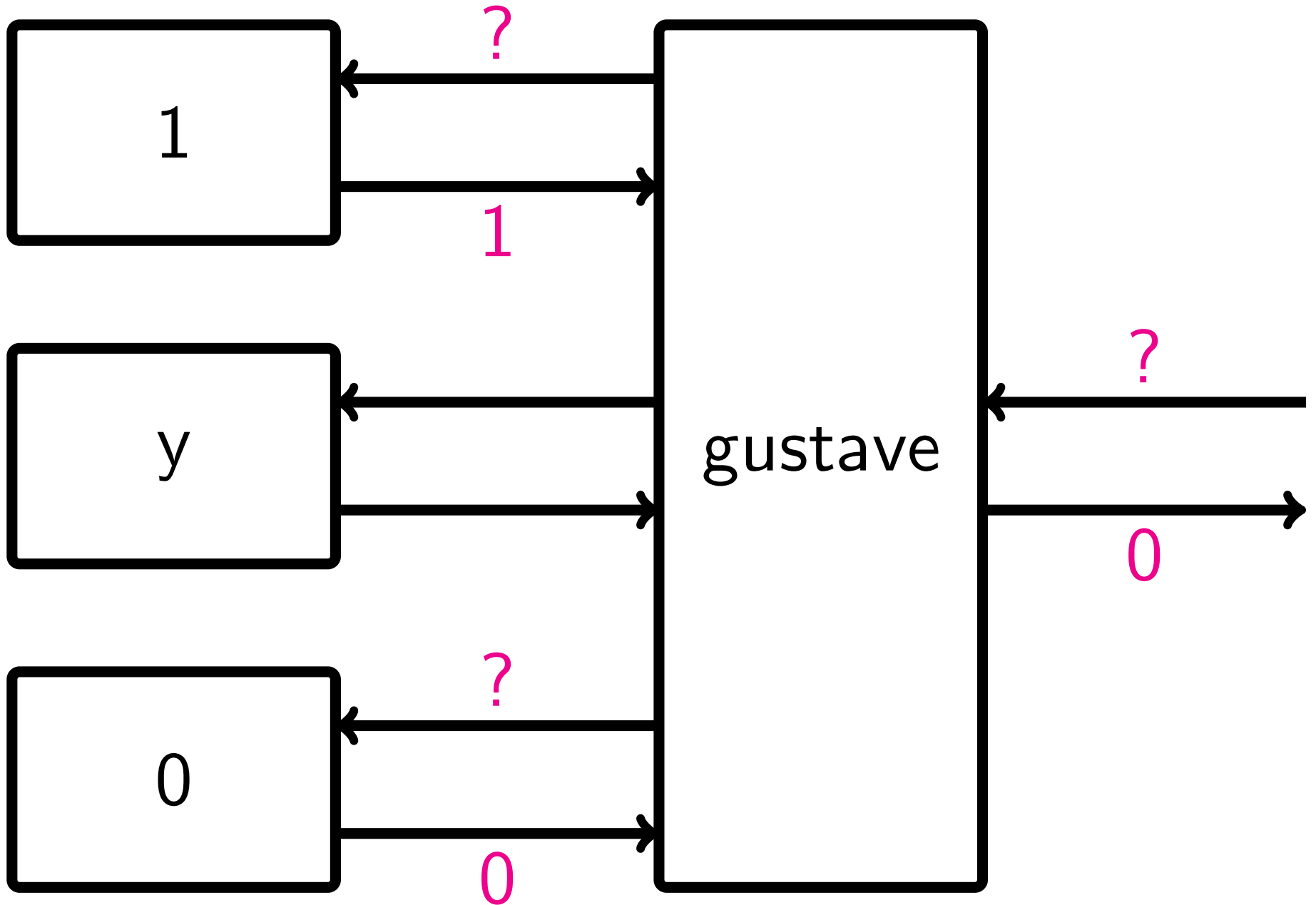
Implementation



Implementation



Implementation



Gol-wRel

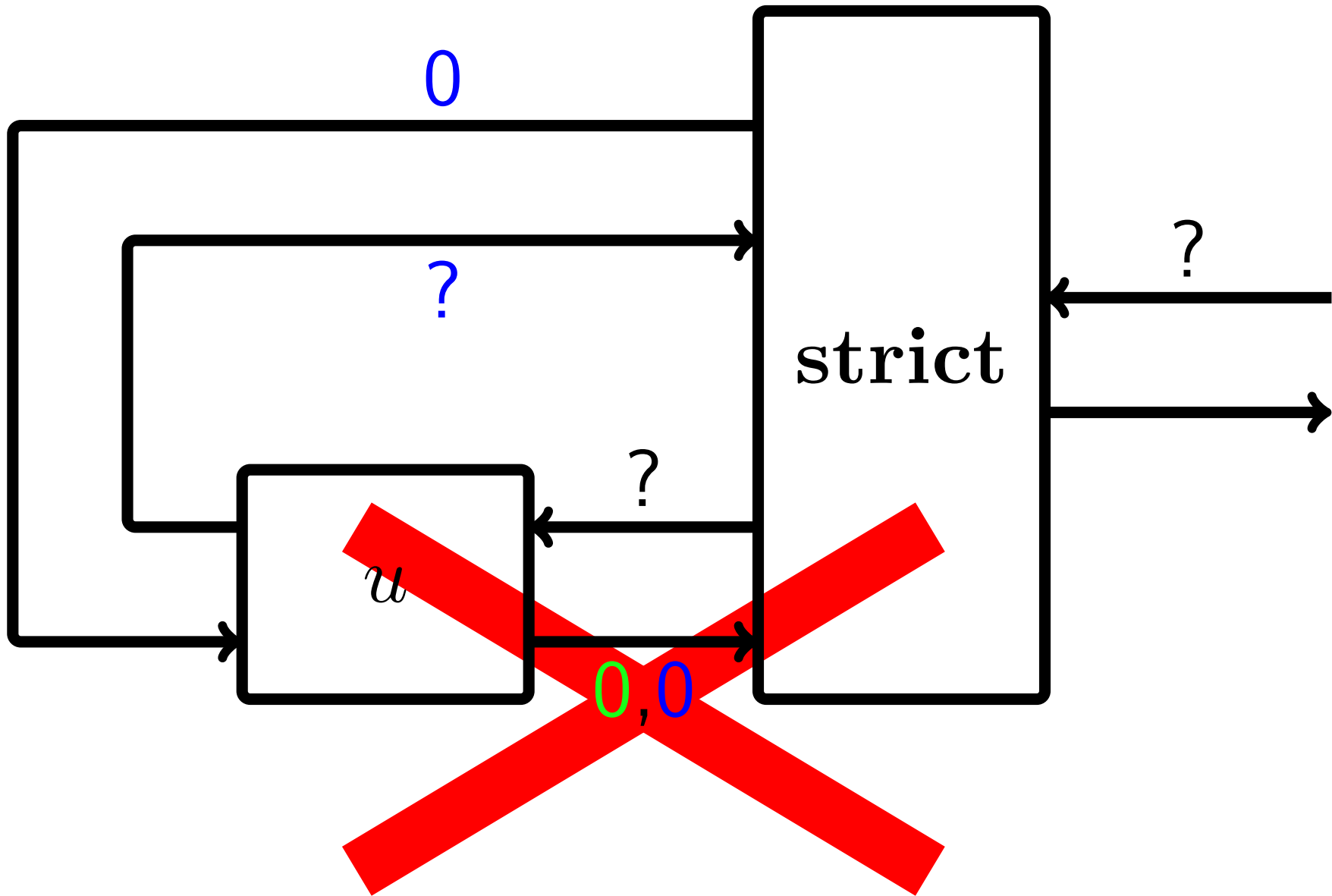
We can **quantitatively** implement **strict** even if u may send multiple messages

strict($u: \mathbb{N}_\perp \rightarrow \mathbb{N}_\perp$)

$$\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

(**strict** \in **Coh**)

Arg can't deceive



Question

Is there any uniform approach?

Thank you