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

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

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

Example

$$\mathbf{strict}(u\colon \mathbb{N}_\perp \to \mathbb{N}_\perp) \stackrel{\mathsf{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

PCF+$\mathbf{strict}$ is implementable

Implementability

We can not implement

$$\mathrm{PCF}+\mathbf{strict}+\mathbf{pconv}$$

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?

$\Uparrow$

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

Main Results

Thm

The full subCCC of $\mathbf{Cpo}$ 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 $\mathbf{Coh}$ generated by $\mathbb{N}_\perp$ is a full subcategory of $\mathbf{Asm}(\mathbf{wRel}(\mathbb{N},\mathbb{N}))$

$$= \{0,1,\ldots,\infty\}^{\mathbb{N}\times\mathbb{N}}$$

c.f. [Oosten 99]
The full subCCC of $\mathbf{HCoh}$ generated by $\mathbb{N}_\perp$ is a full subcategory of $\mathbf{Asm}(\mathbf{Pfn}(\mathbb{N},\mathbb{N}))$

Related Works

Characterizations of $\mathbf{HCoh}$ in terms of

– GoI [Oosten 99]

– Game semantics [Melliès 05]

Embeddings of game semantics into $\mathbf{Coh}$

– [Melliès 13]

– [Calderon & McCusker 10]

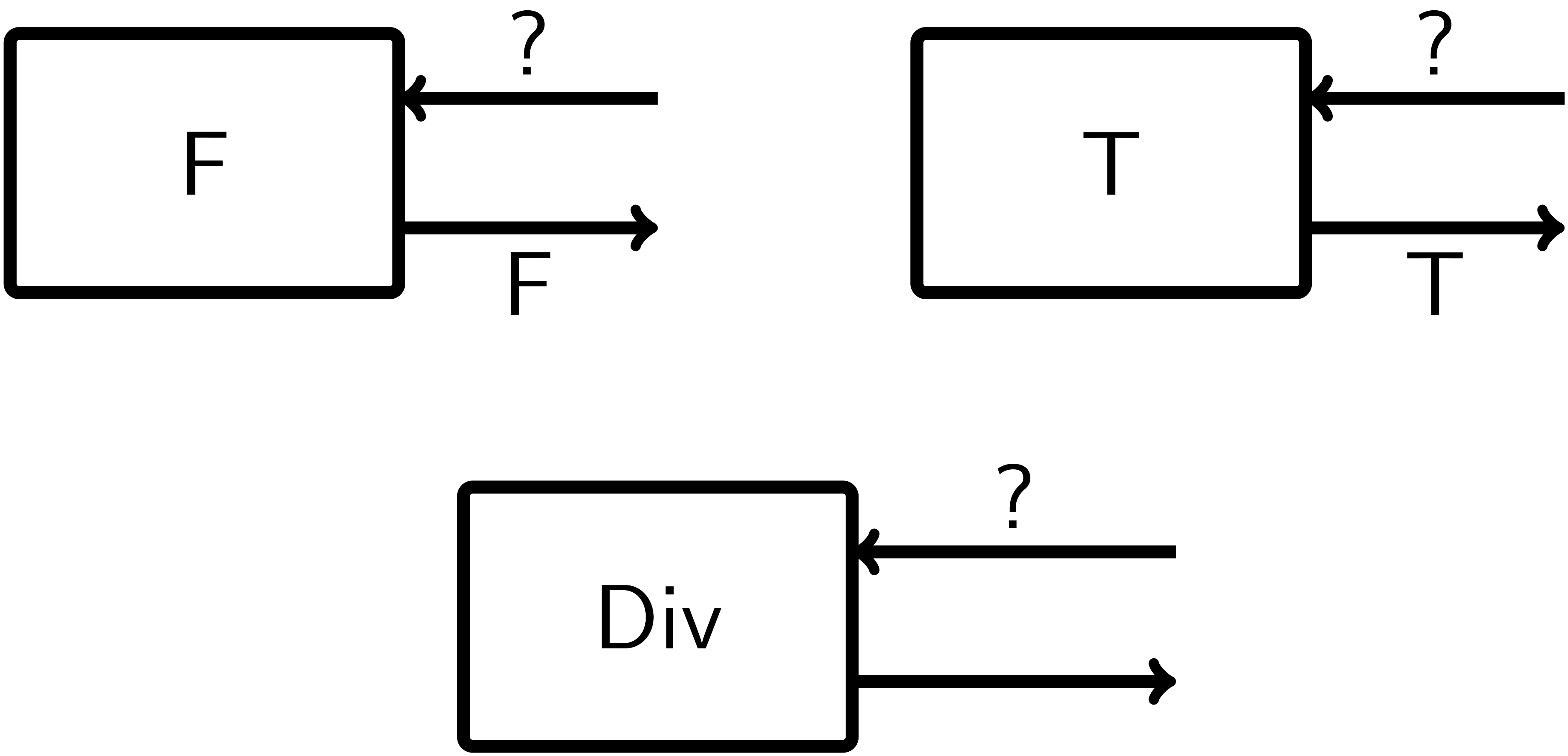
Geometry of Interaction

Models of lambda calculi

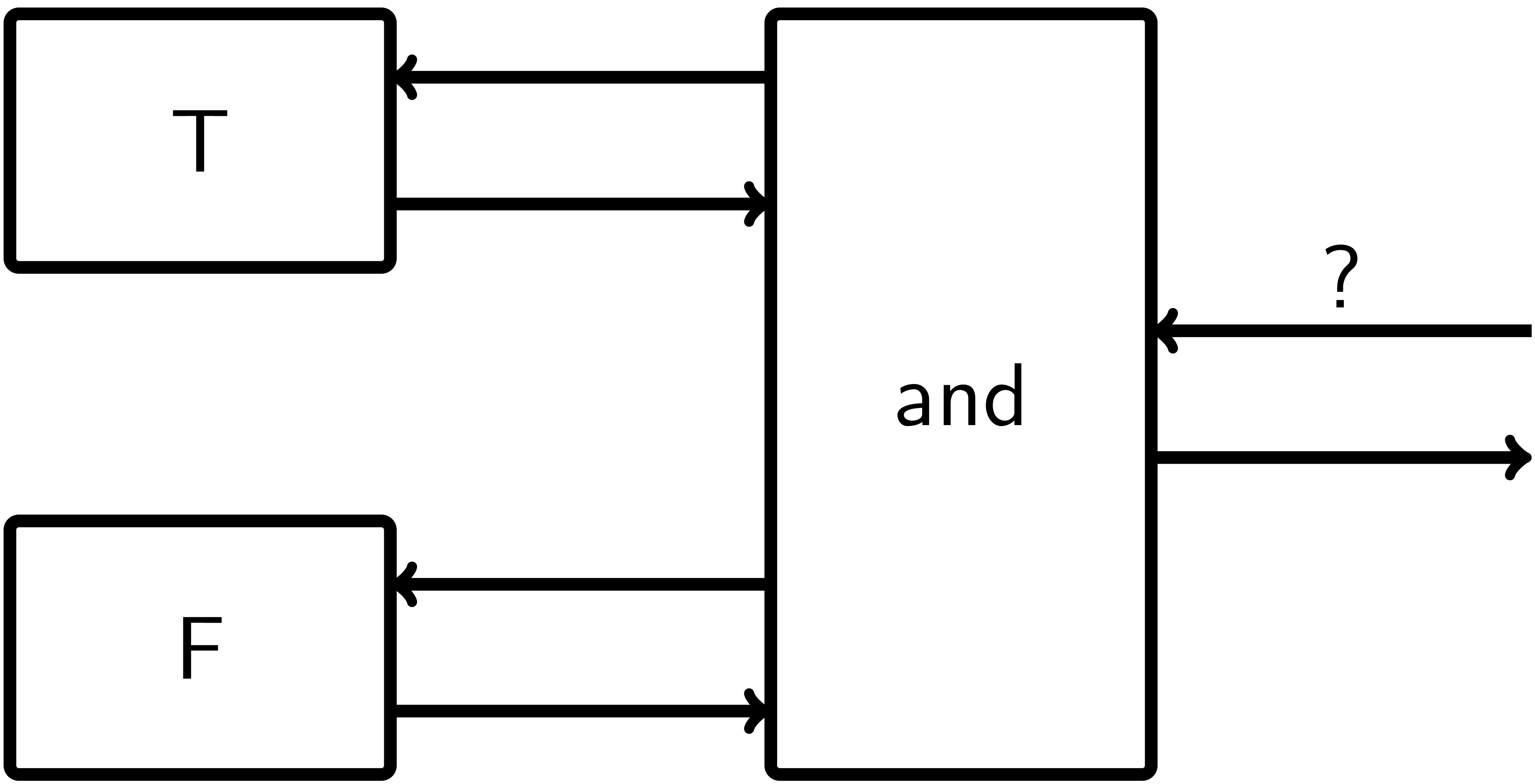
– Programs as token machines

– Evaluation as execution

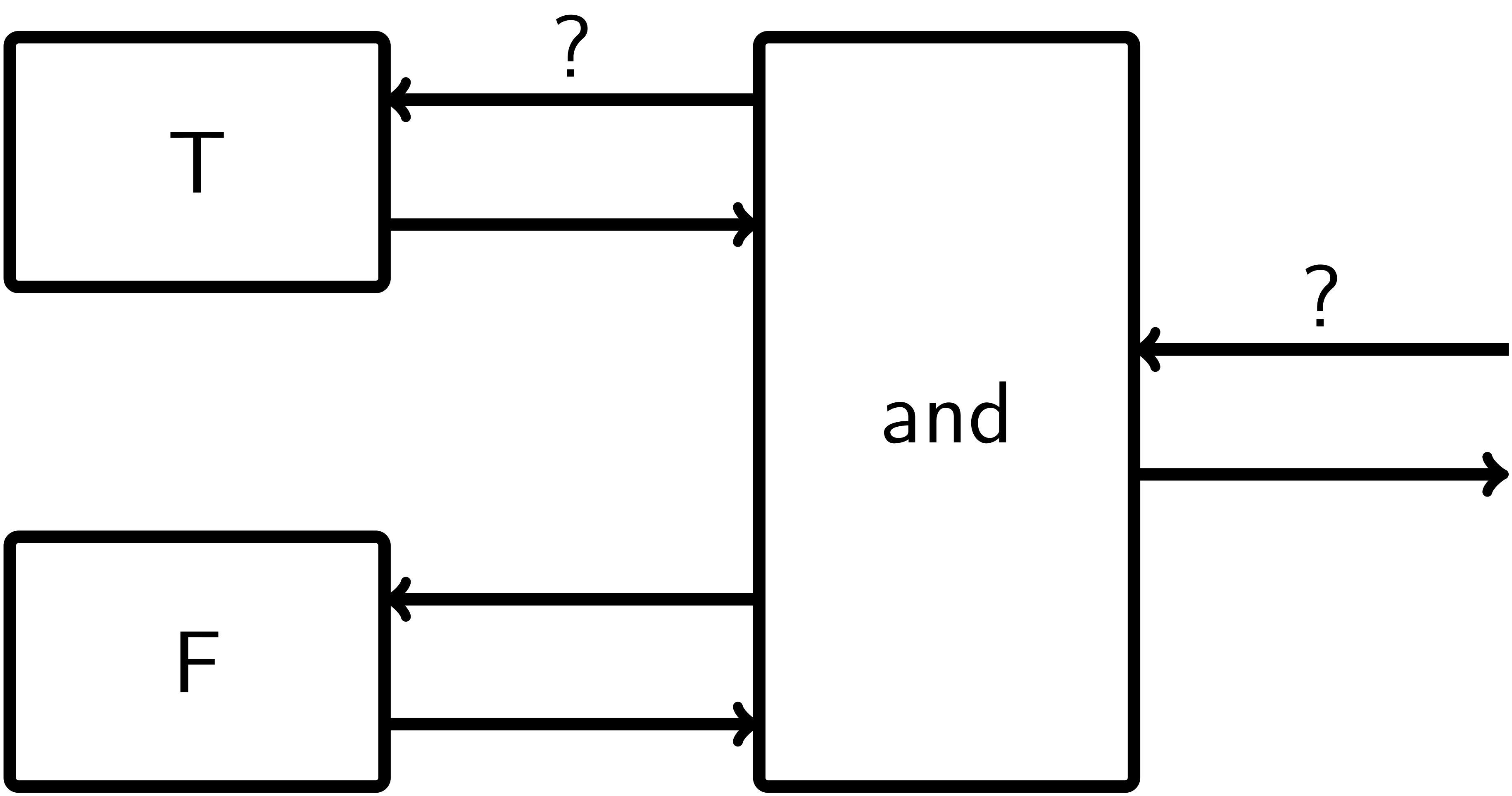
Bool



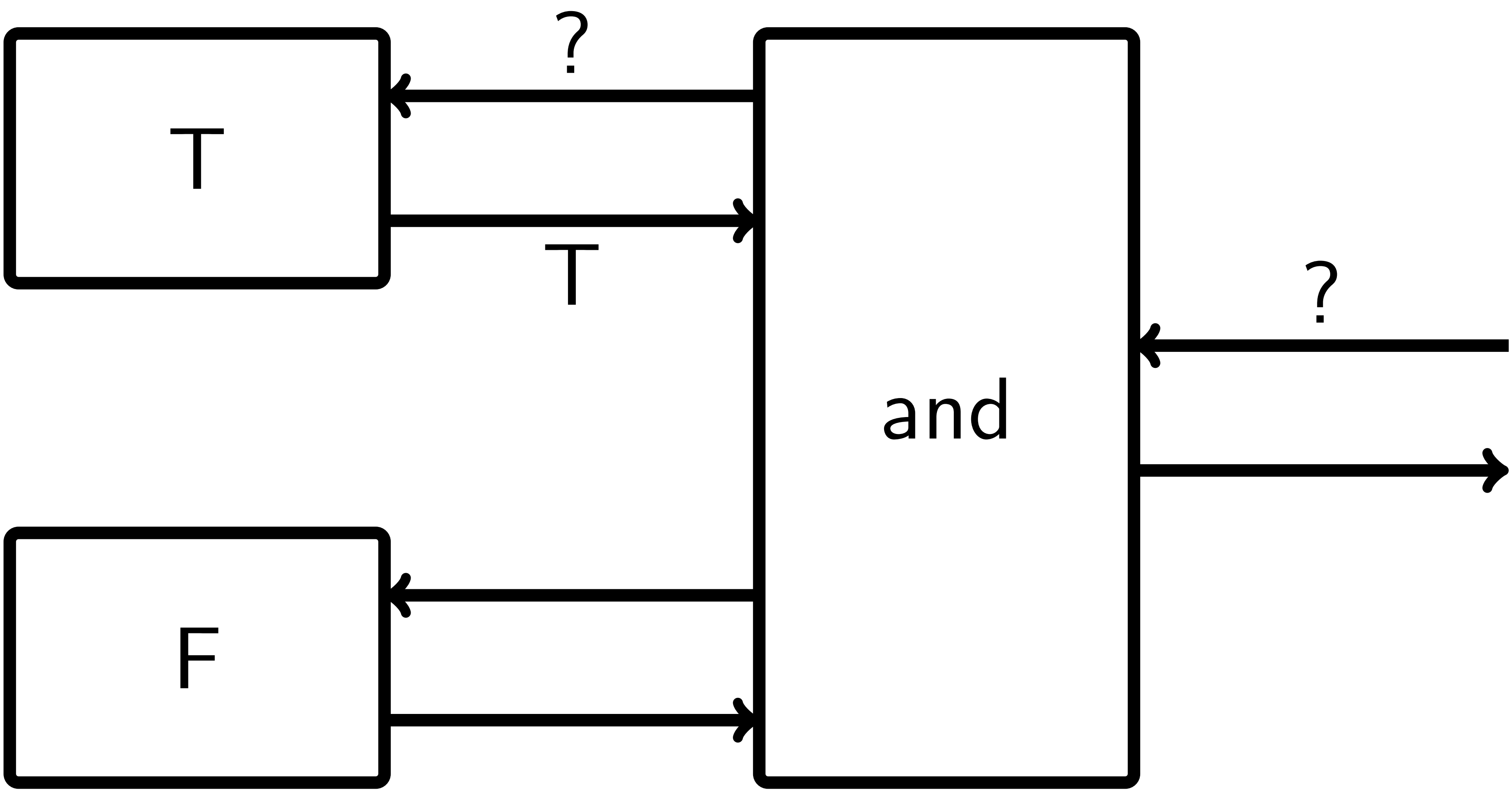
Conjunction



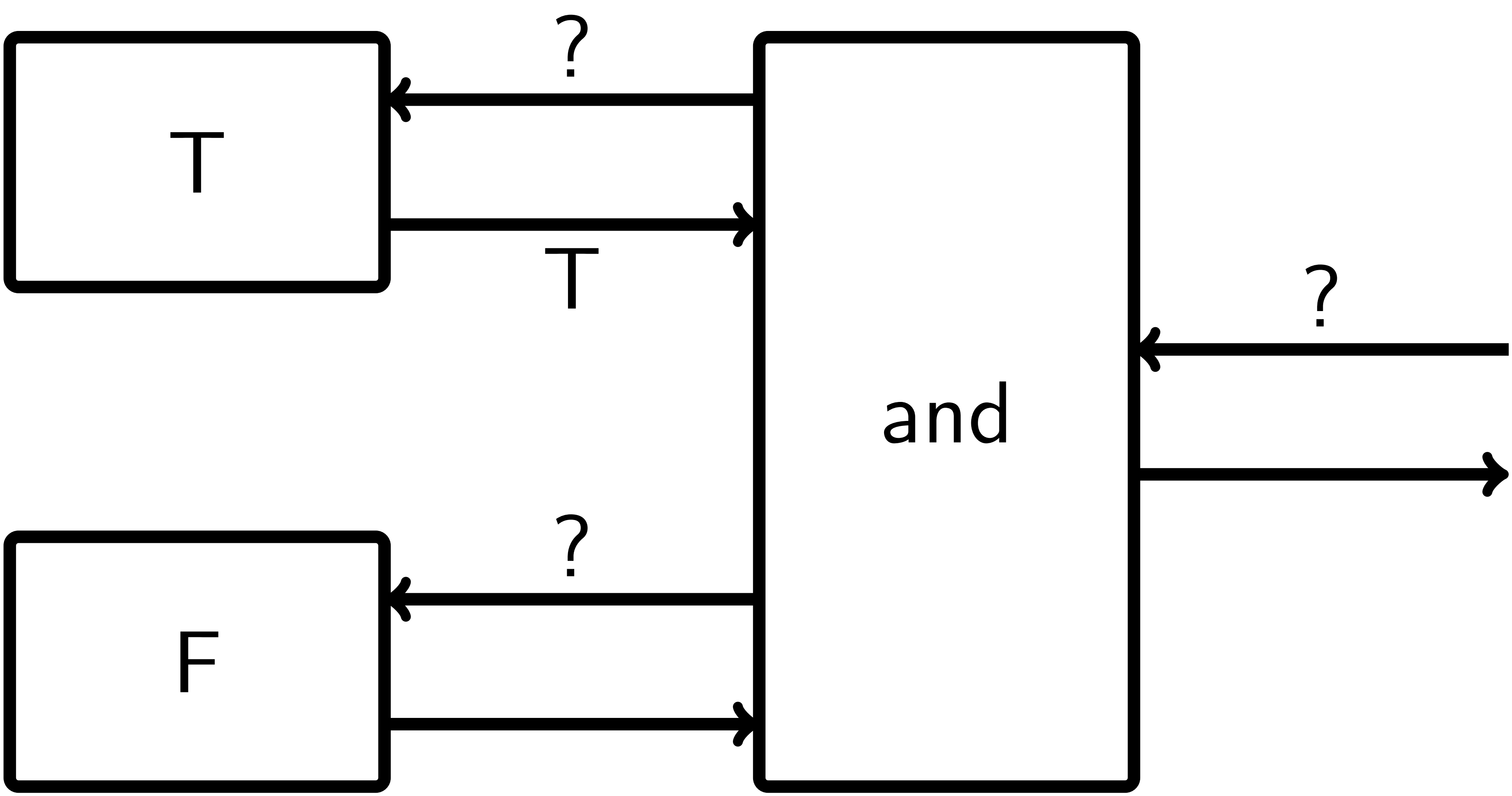
Conjunction



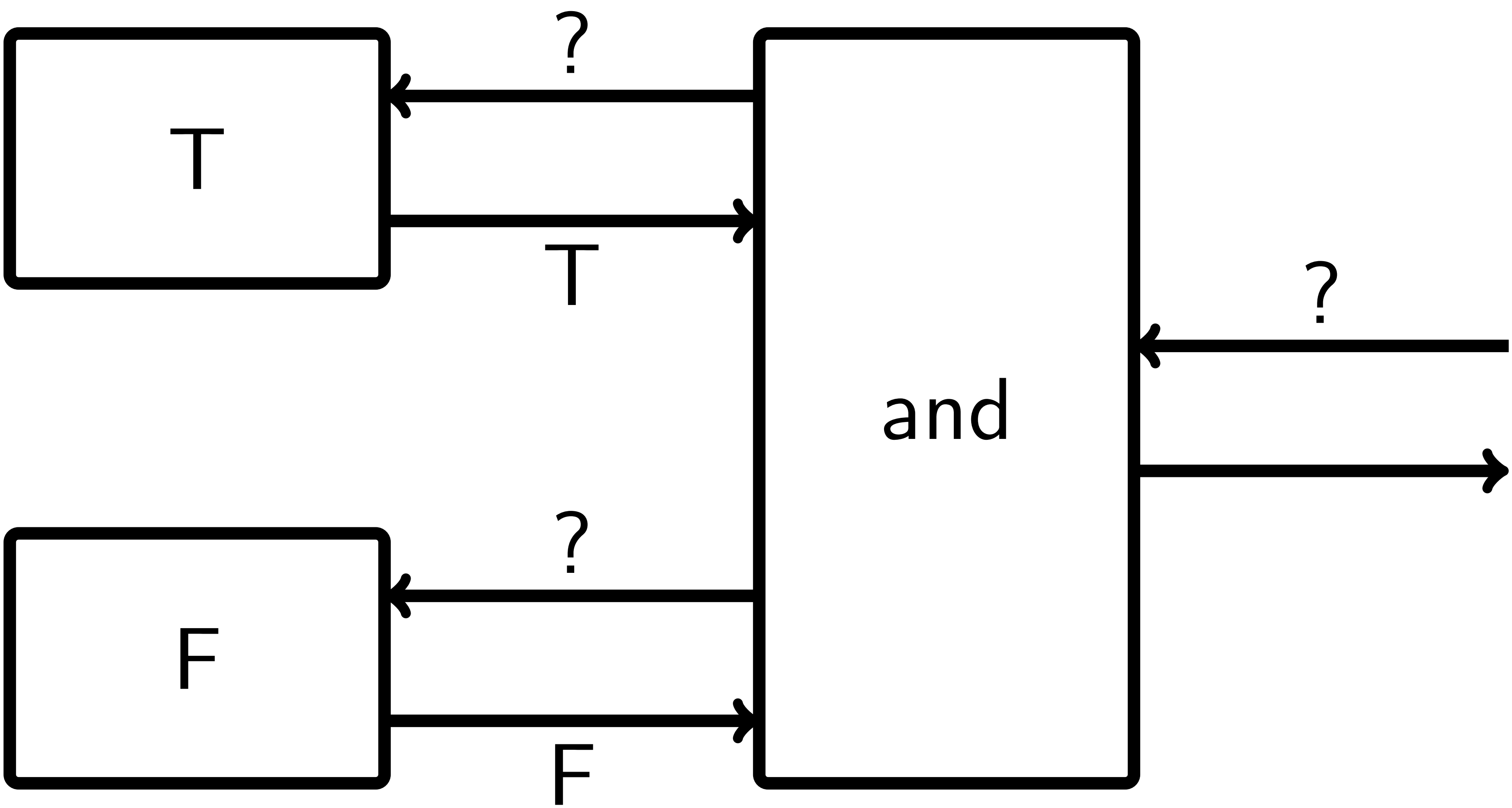
Conjunction



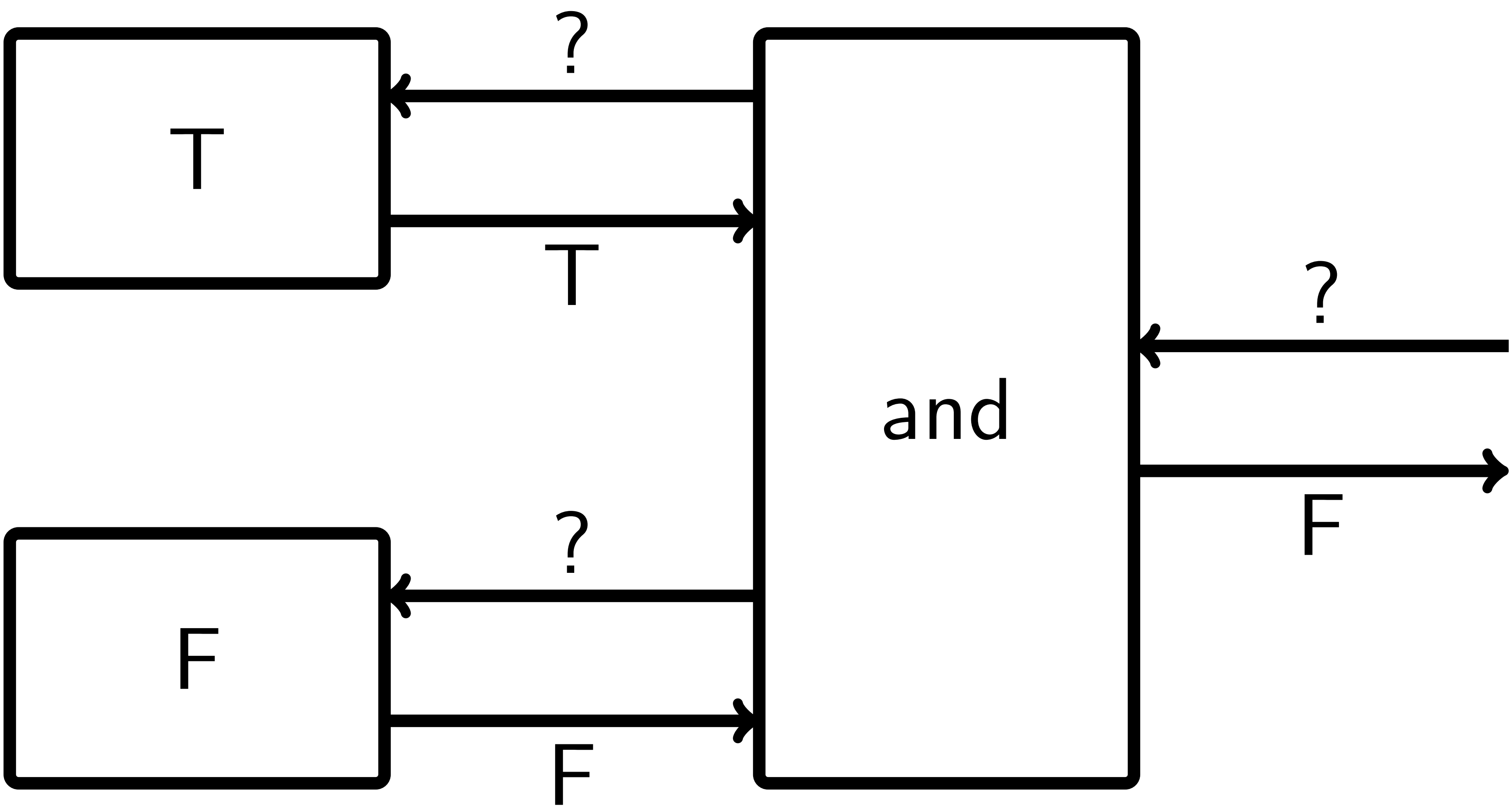
Conjunction



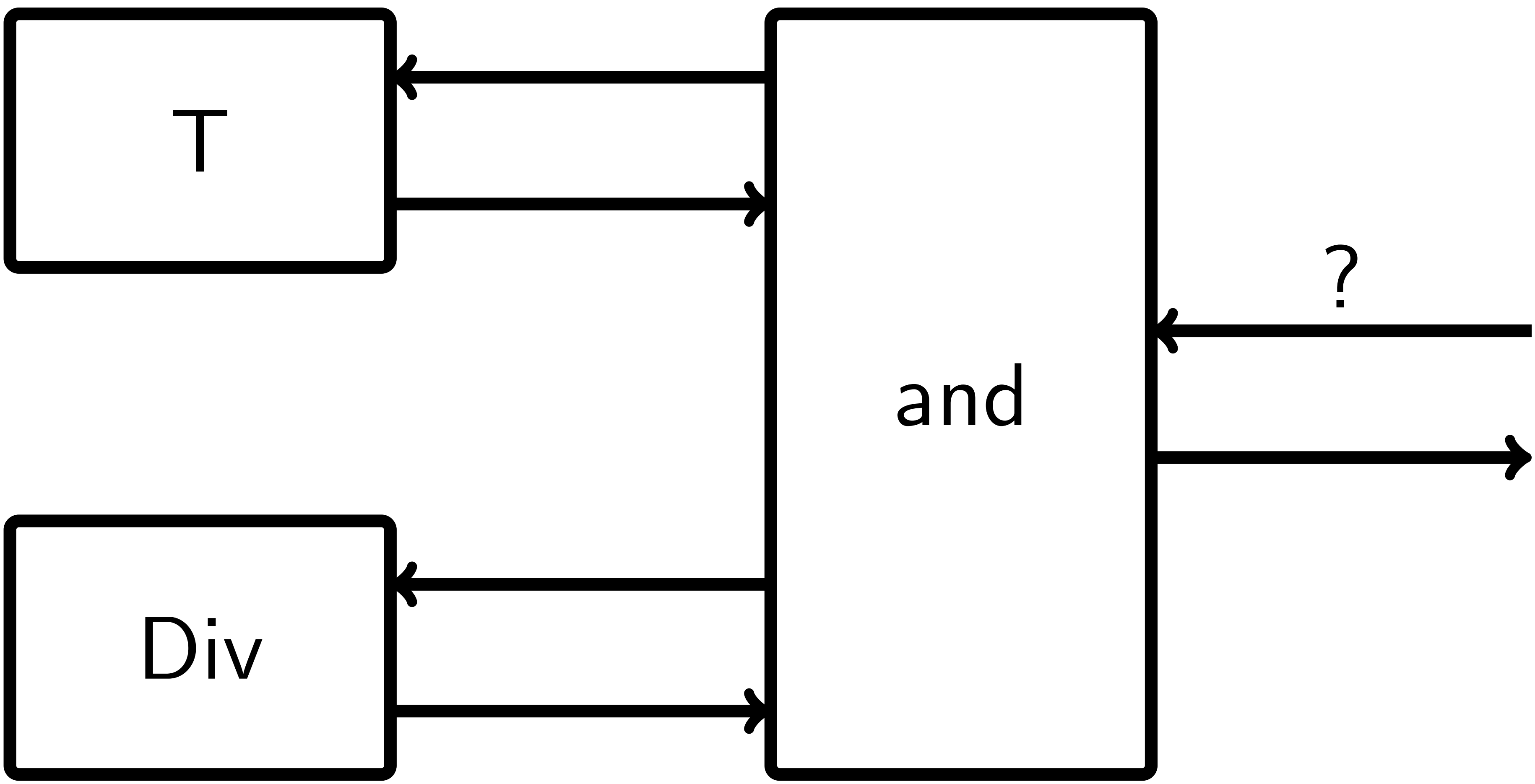
Conjunction



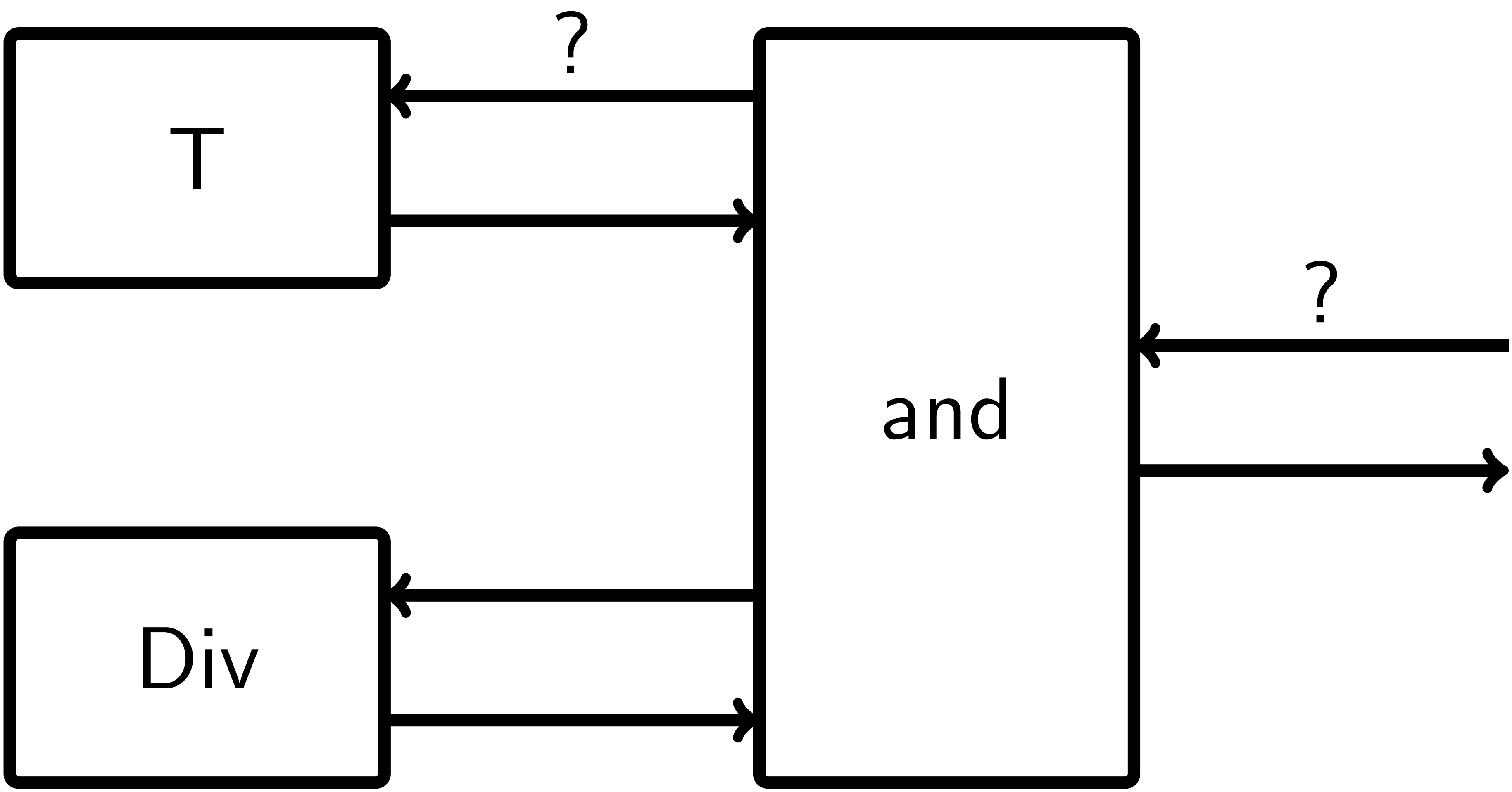
Conjunction



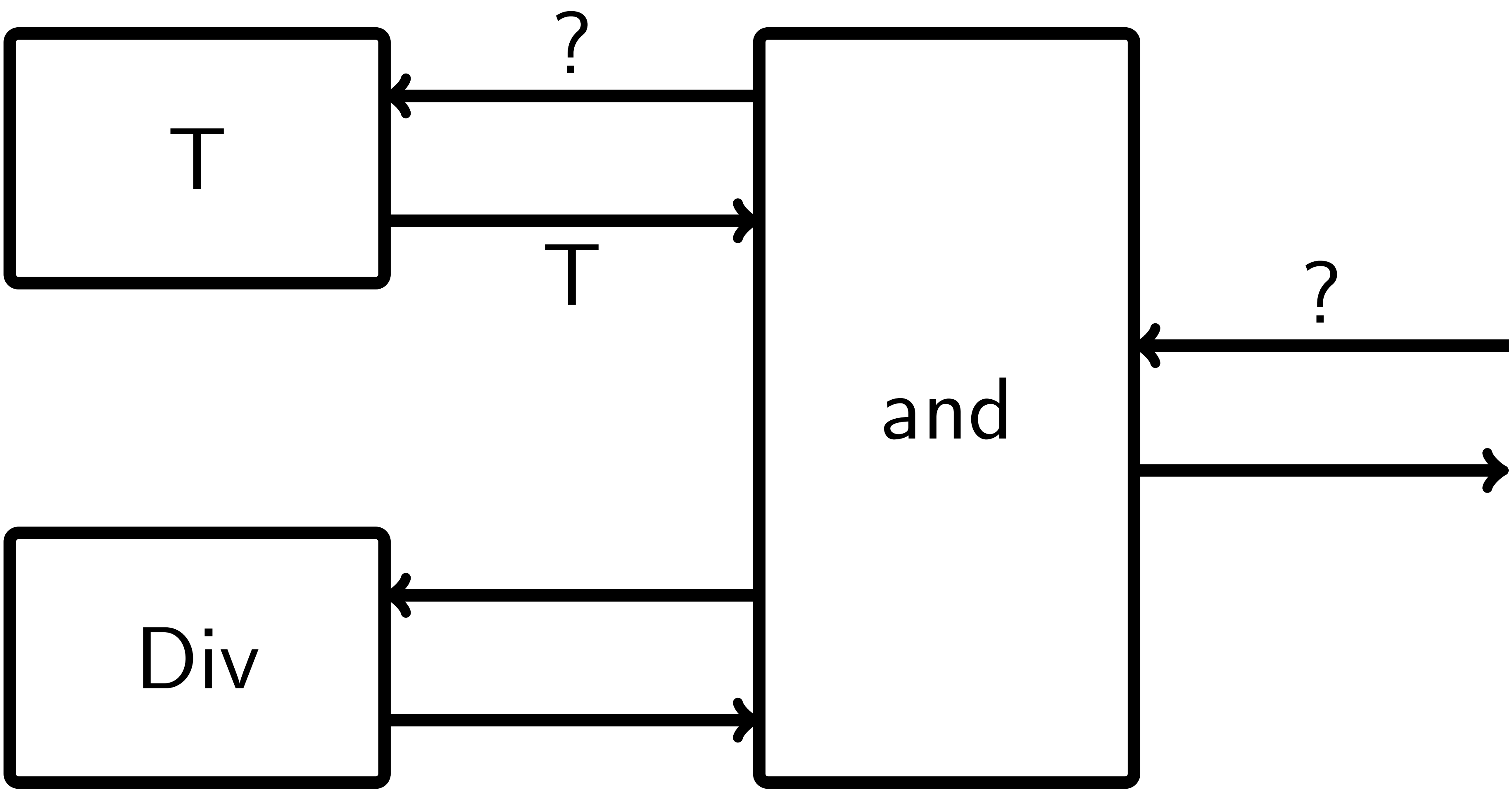
Conjunction



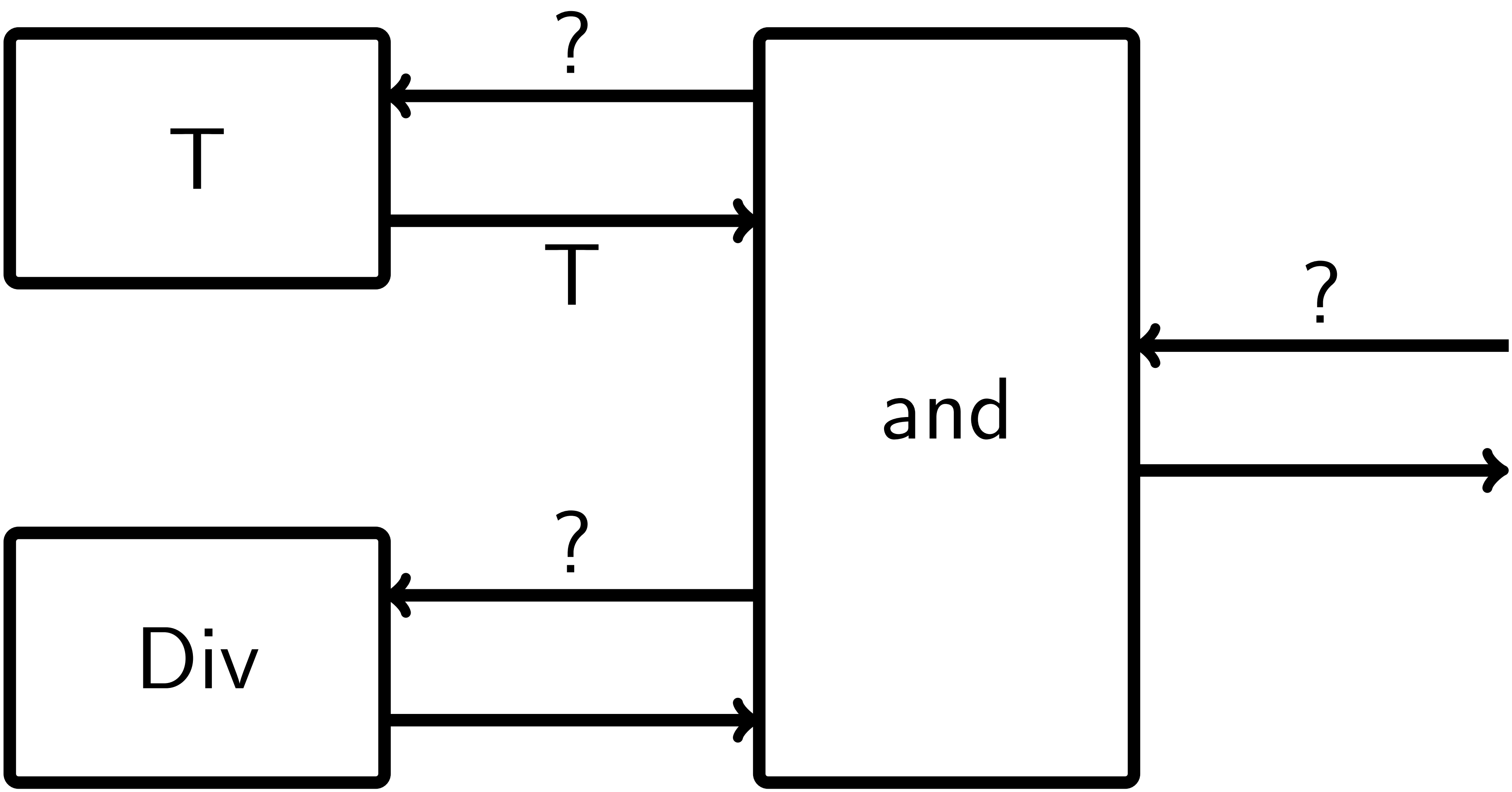
Conjunction



Conjunction



Conjunction



GoI-Rel
GoI-Pfn
GoI-wRel

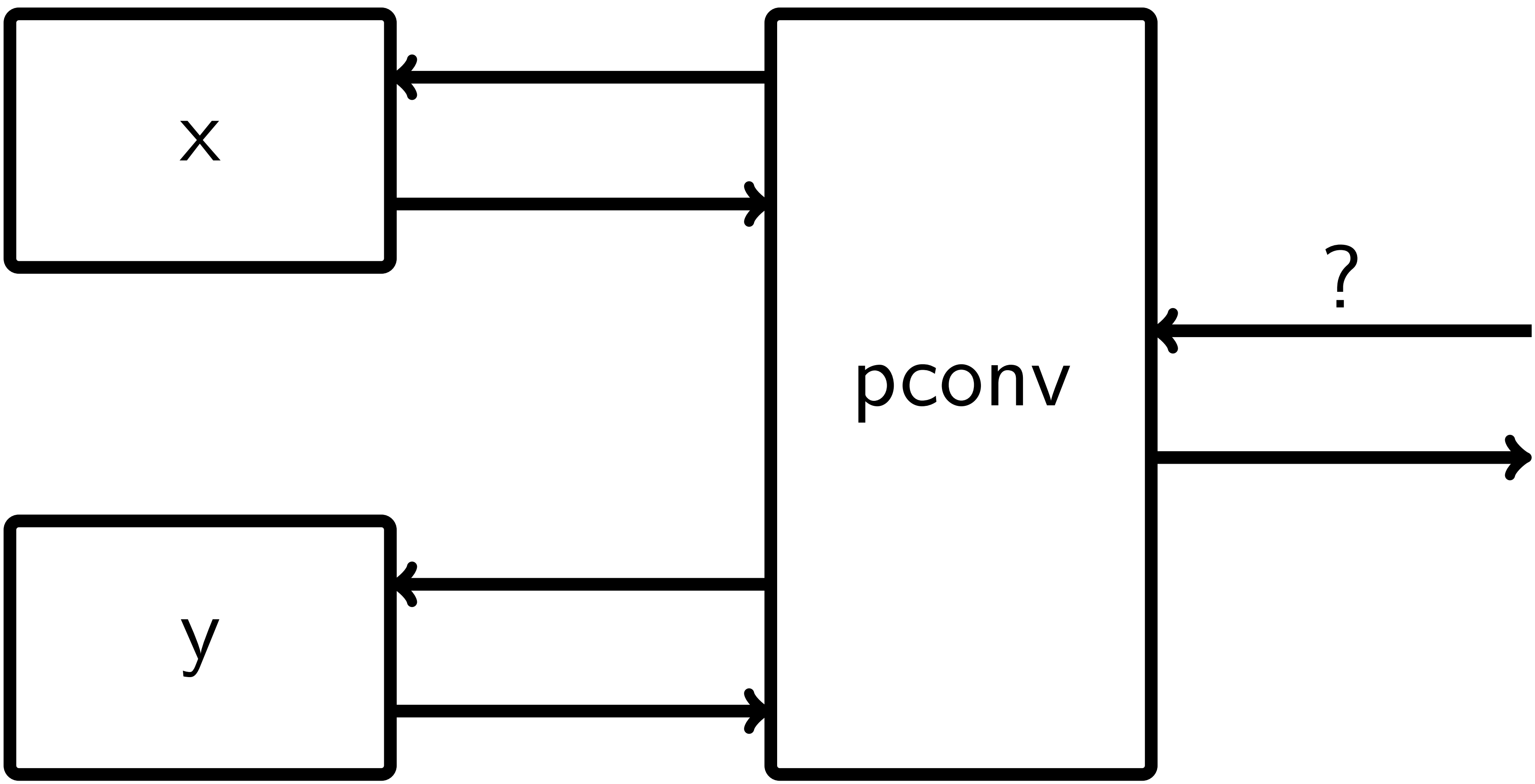
GoI-Rel

We can implement $\mathbf{pconv}$
by sending multiple messages

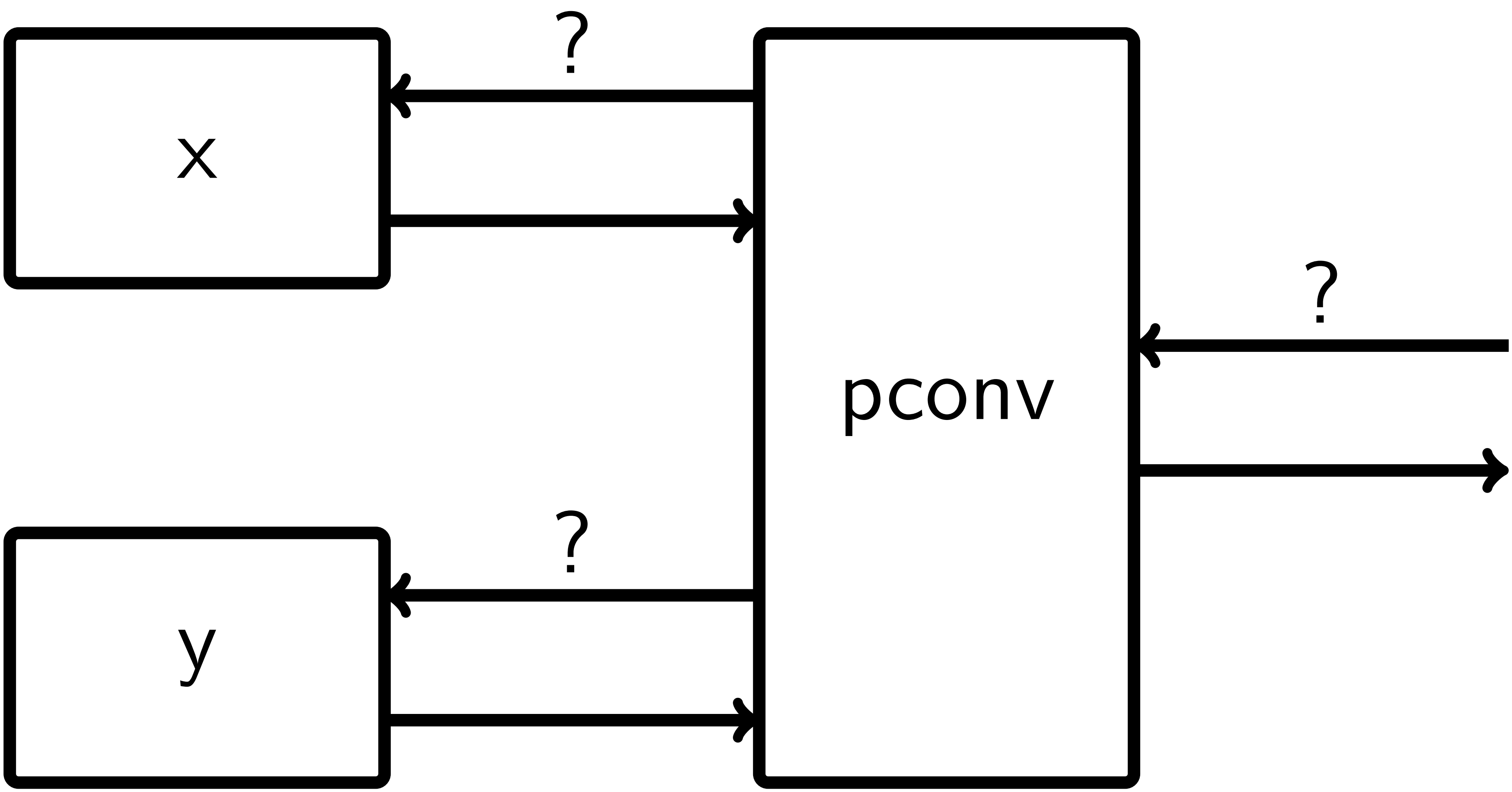
$$\mathbf{pconv}(x\colon \mathbb{N}_\perp, y\colon \mathbb{N}_\perp) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{pconv} \in \mathbf{Cpo})$

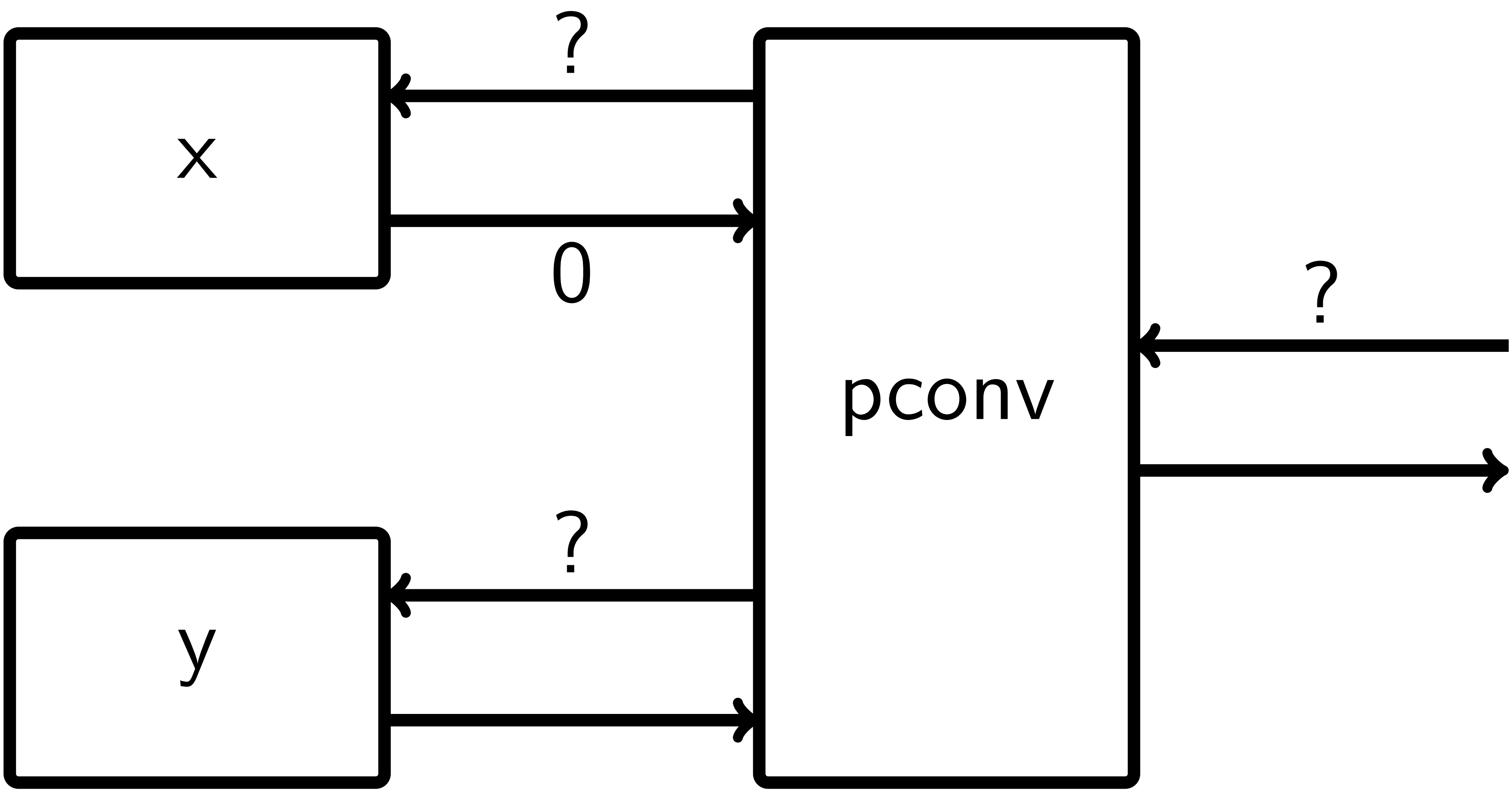
Implementation



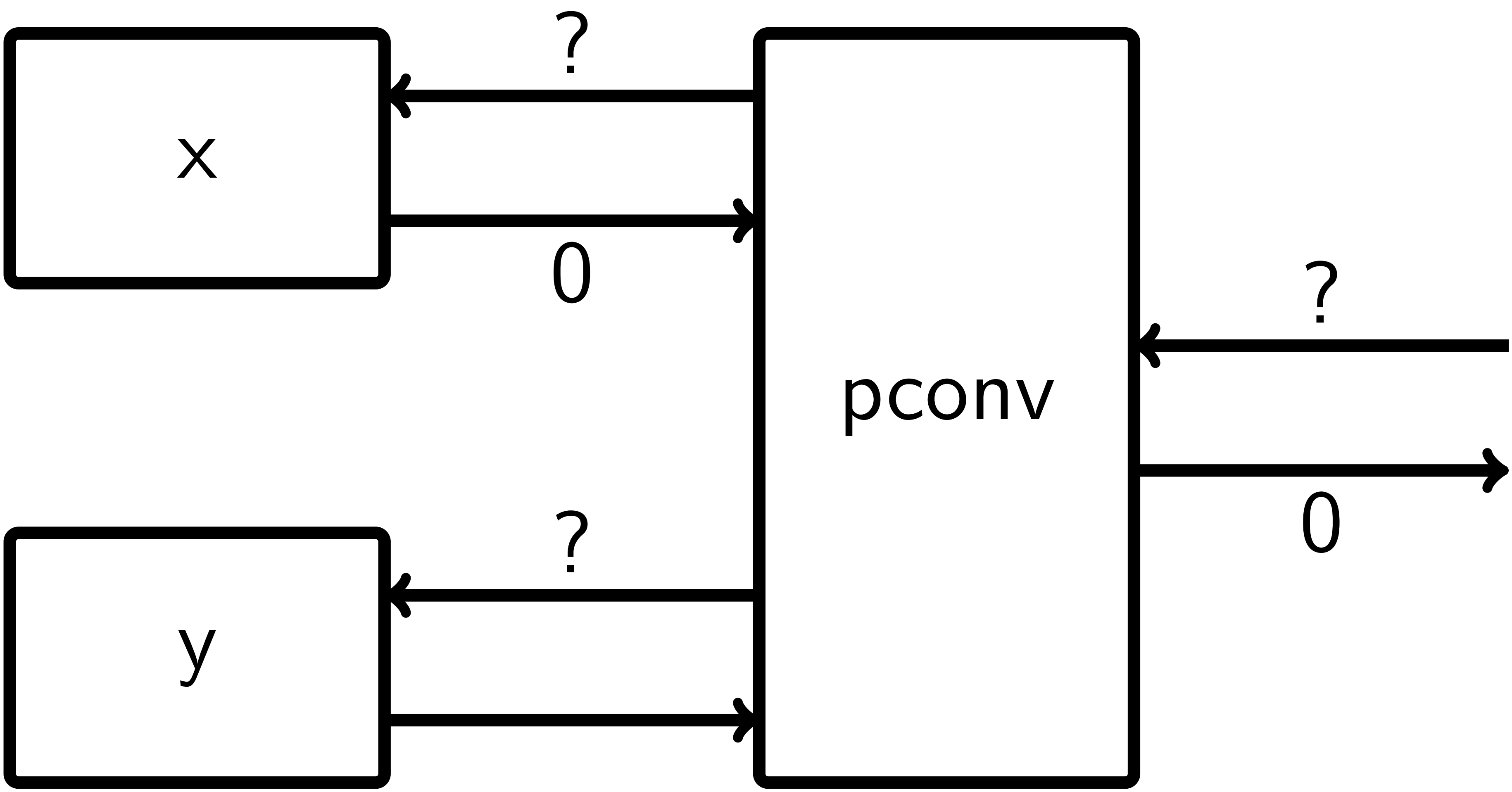
Implementation



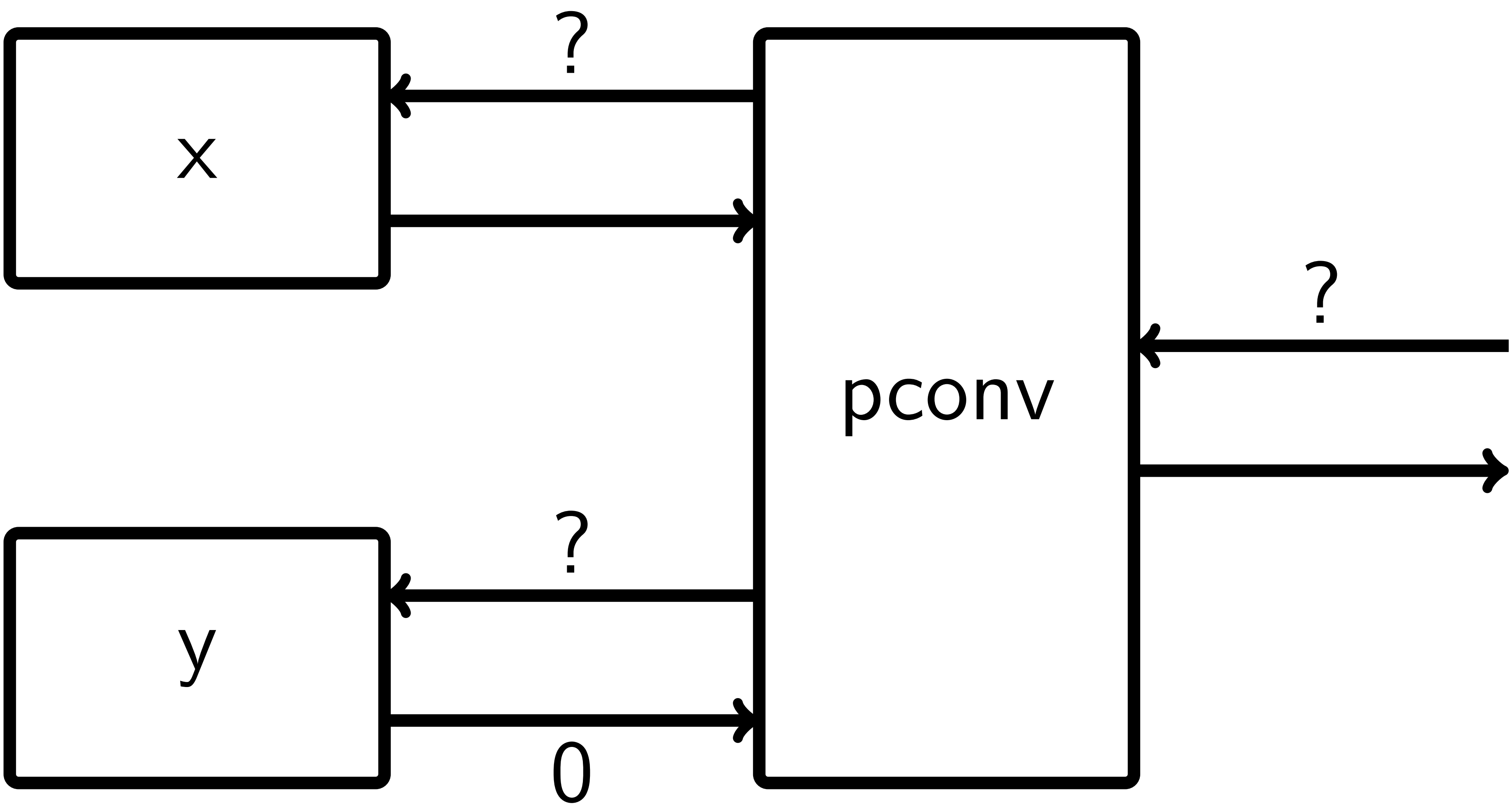
Implementation



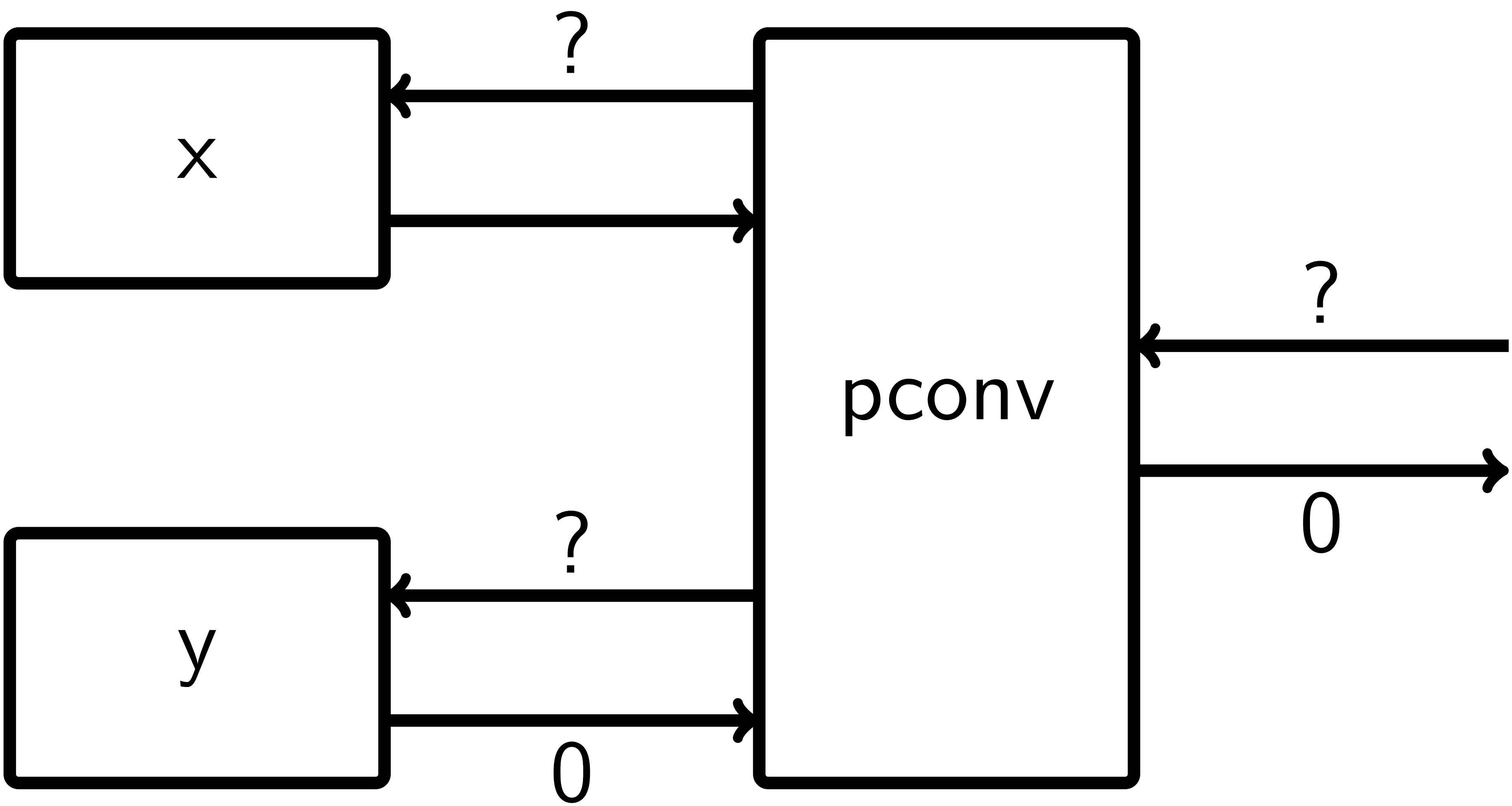
Implementation



Implementation



Implementation



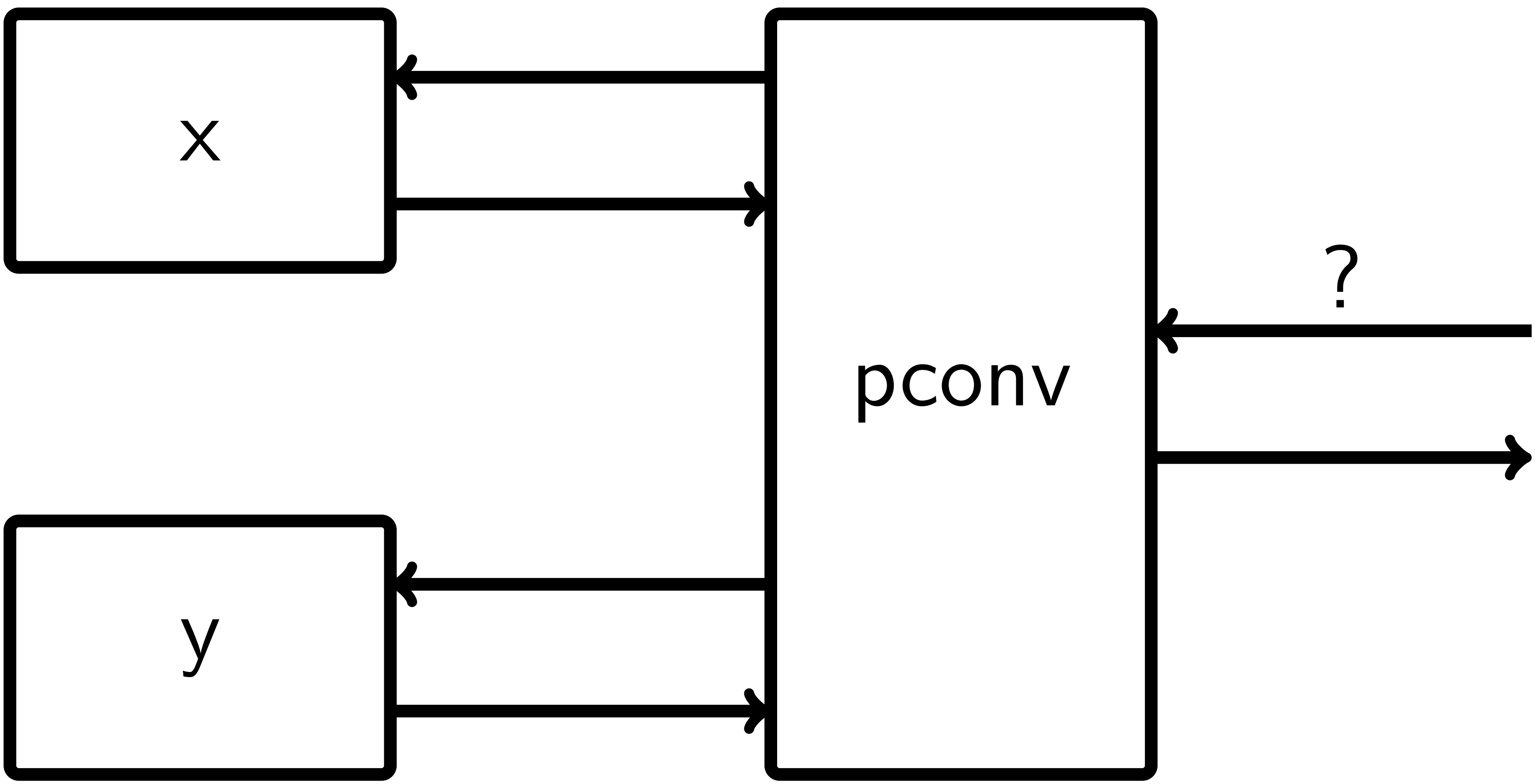
GoI-Pfn

We can't implement $\mathbf{pconv}$
if we can't send multiple messages

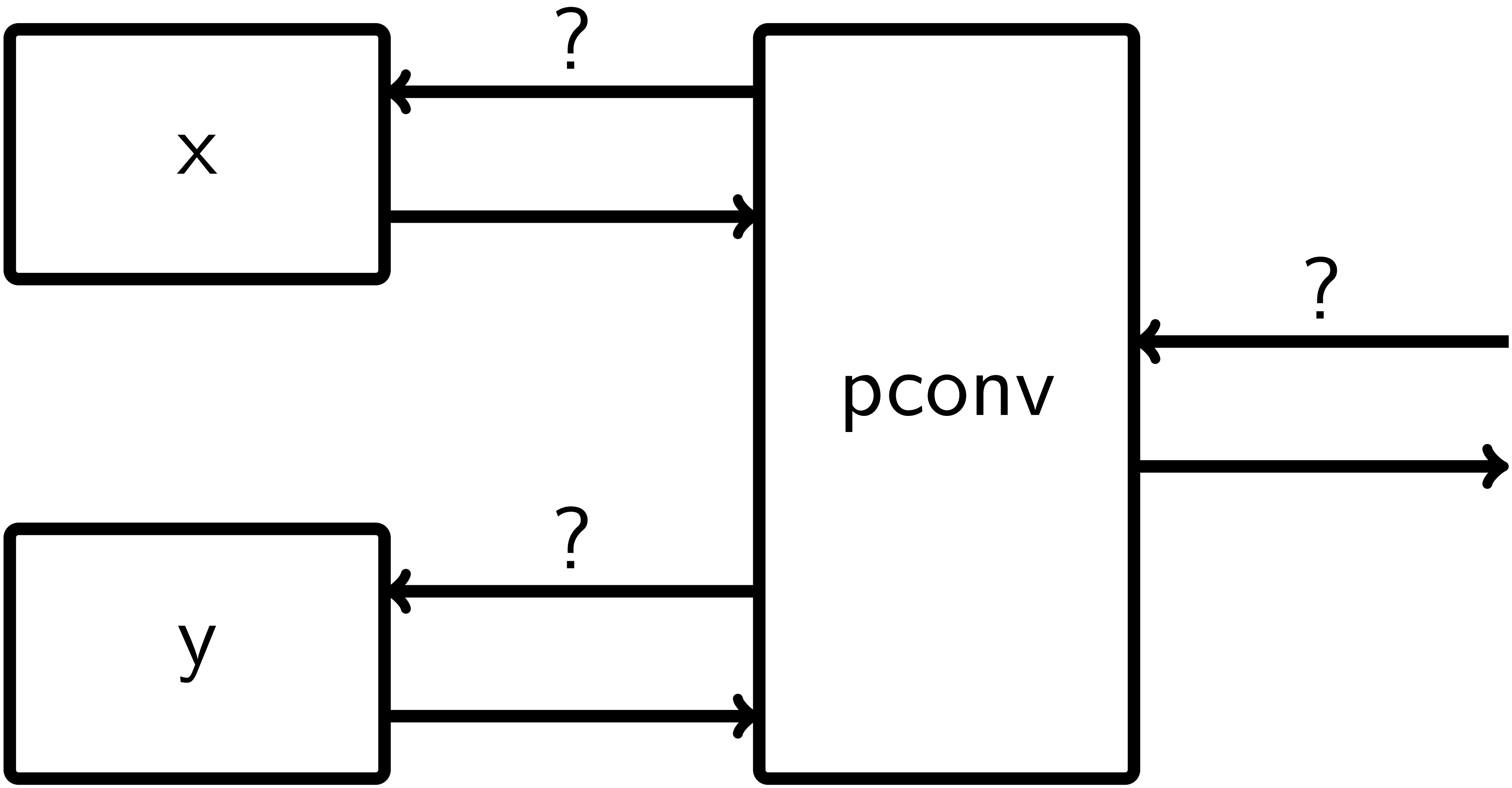
$$\mathbf{pconv}(x\colon \mathbb{N}_\perp, y\colon \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{pconv} \notin \mathbf{HCoh})$

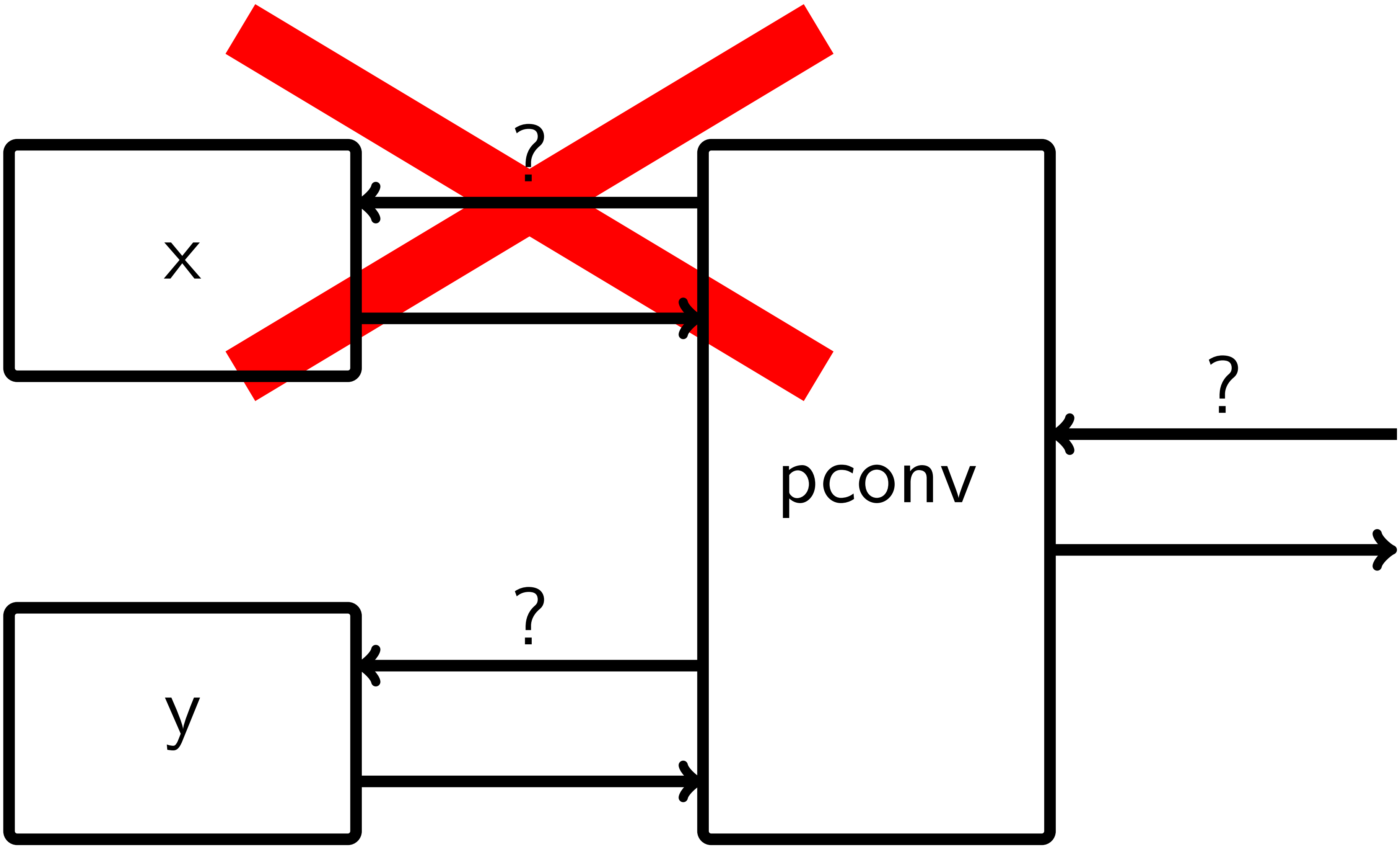
GoI-Pfn



GoI-Pfn



GoI-Pfn



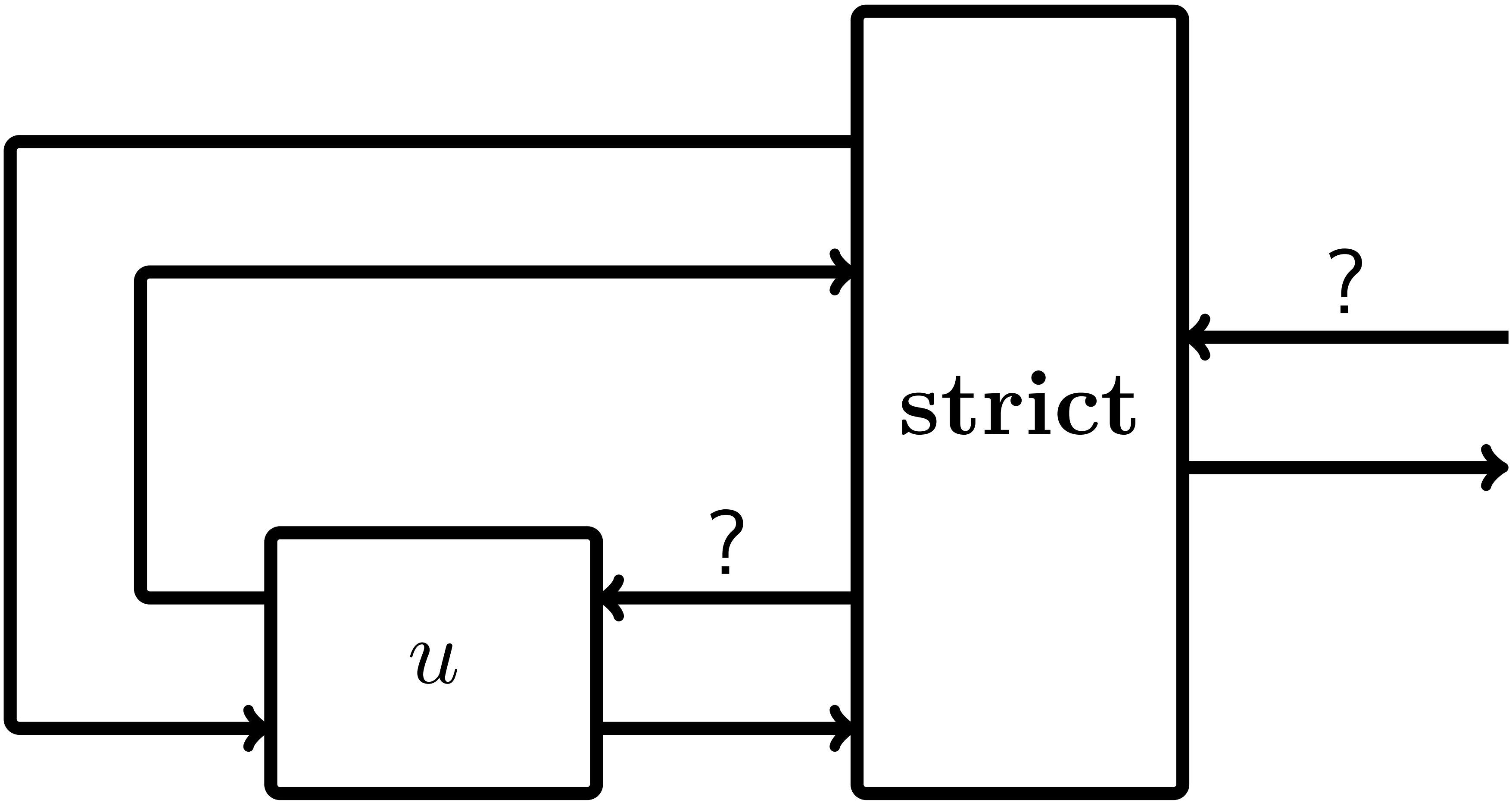
GoI-Pfn

We can implement $\mathbf{strict}$
if u don't send multiple messages

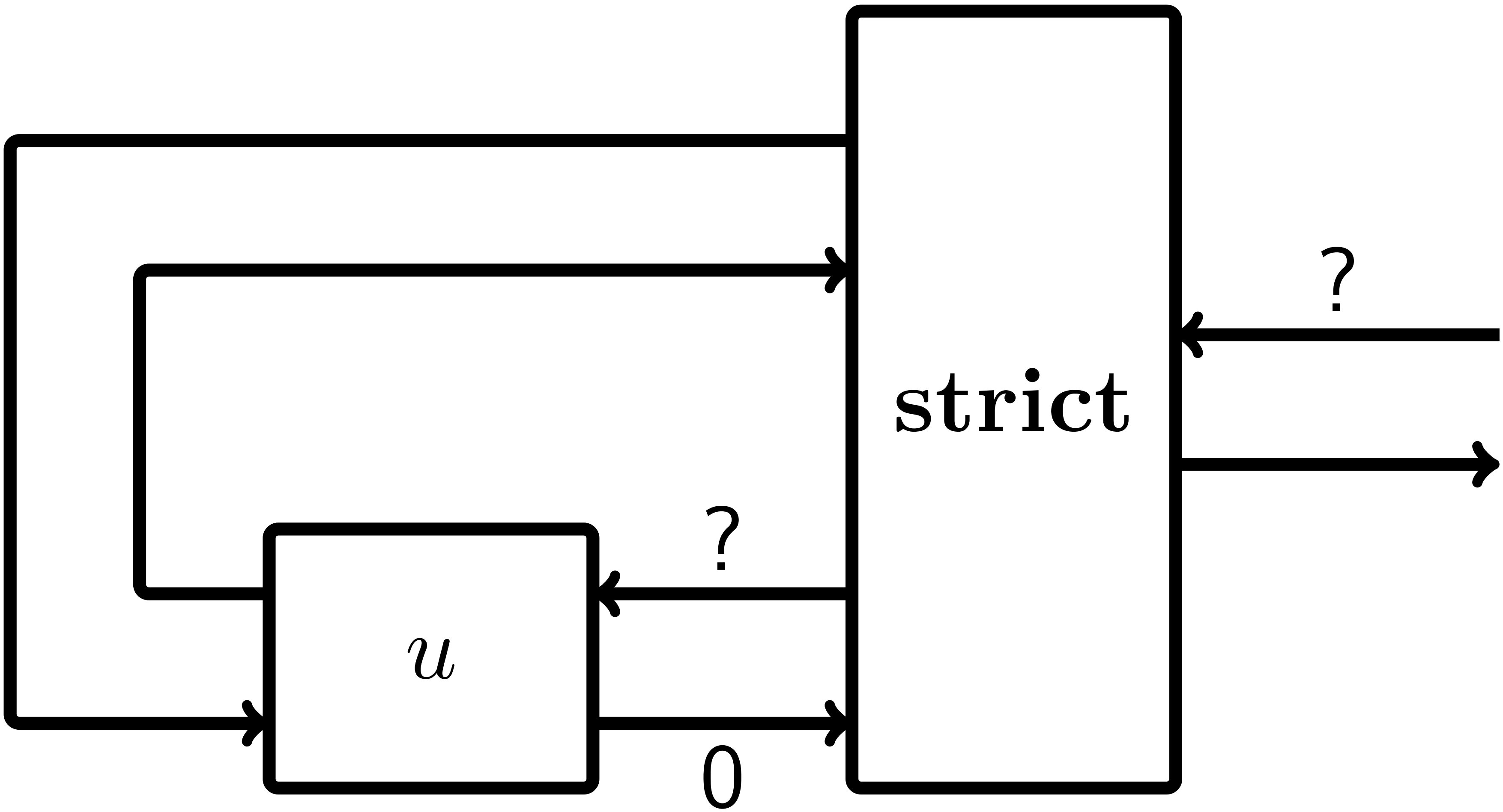
$$\mathbf{strict}(u\colon \mathbb{N}_\perp \to \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{strict} \in \mathbf{HCoh})$

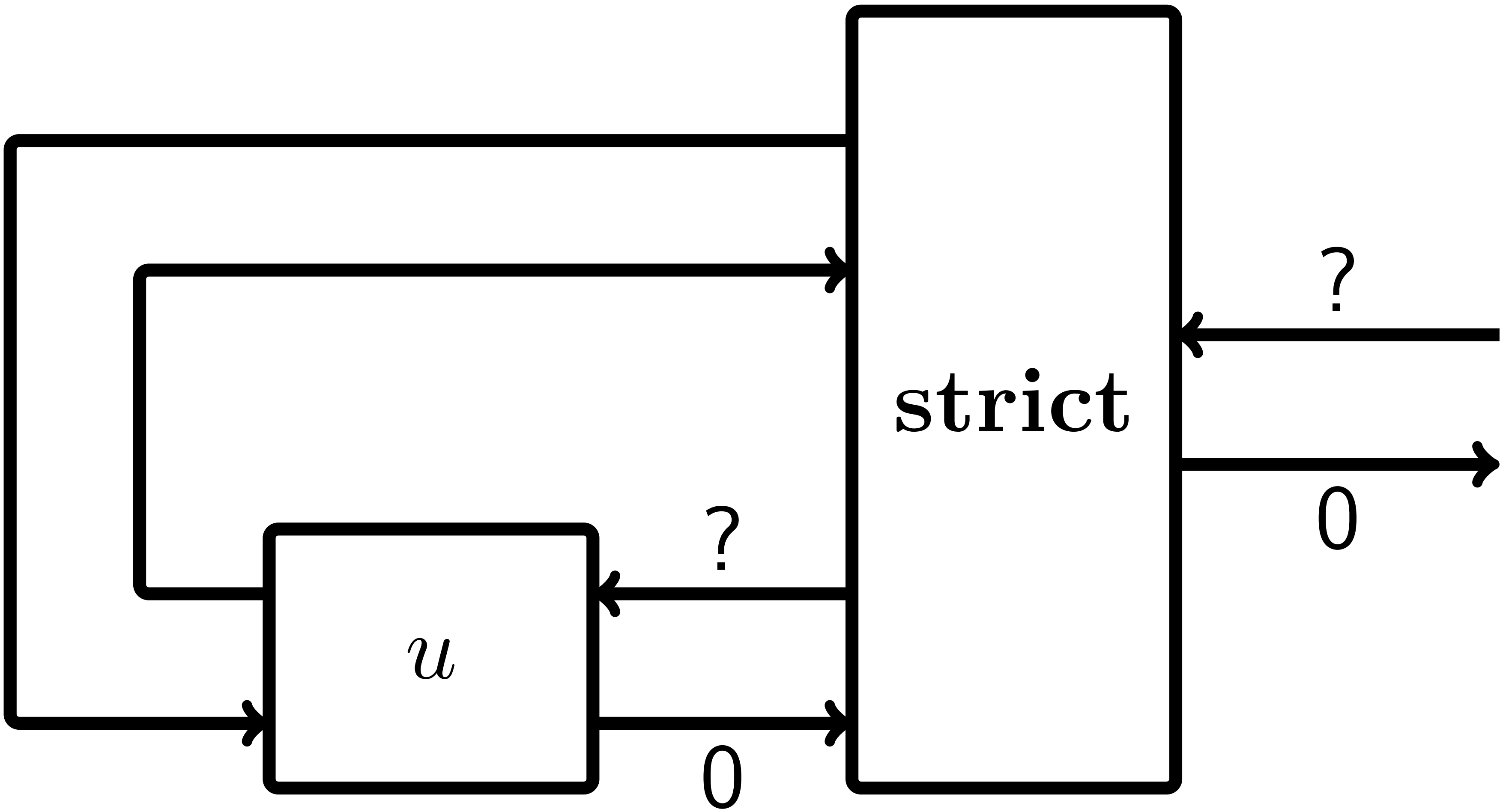
Implementation



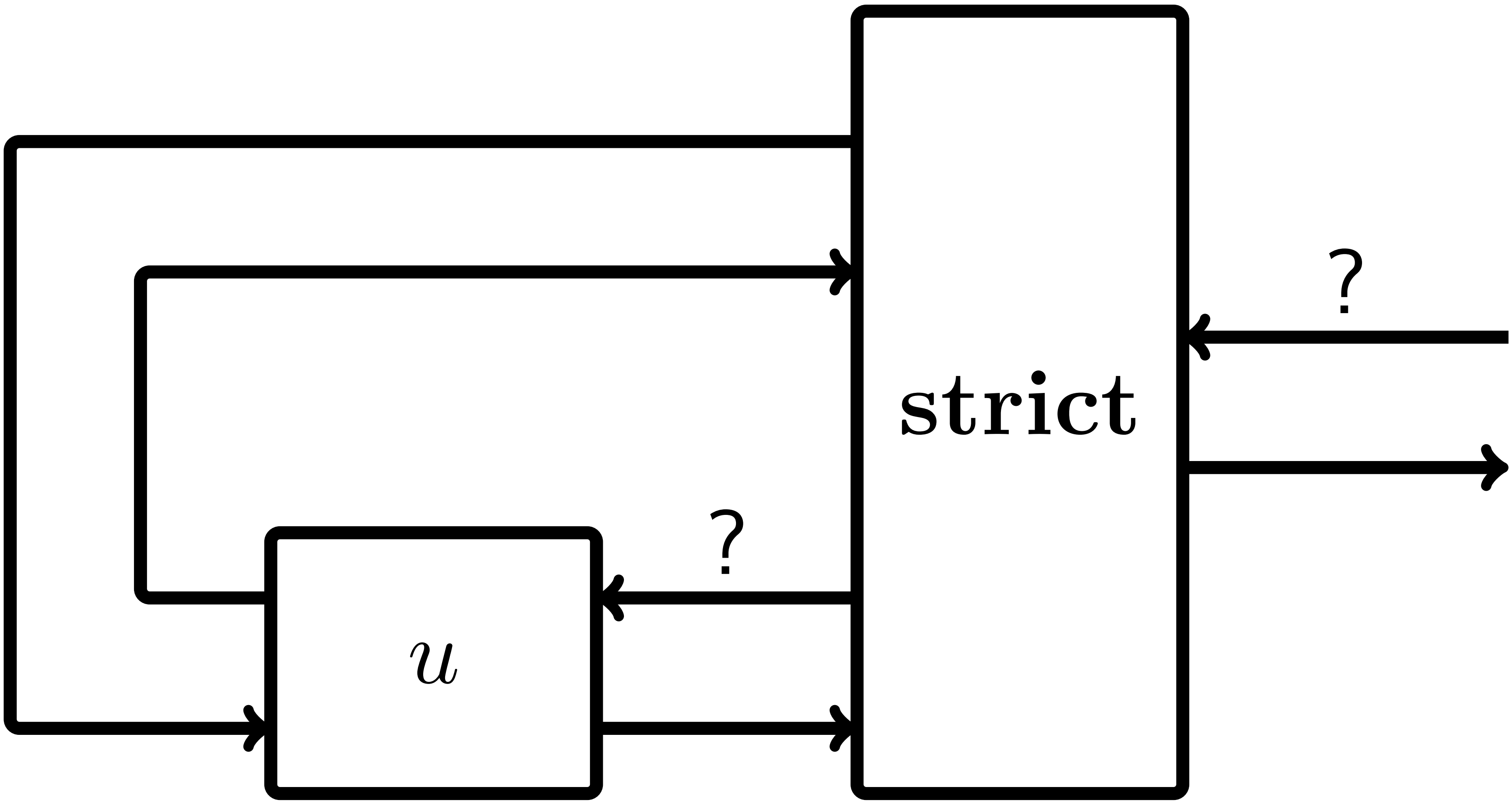
Implementation



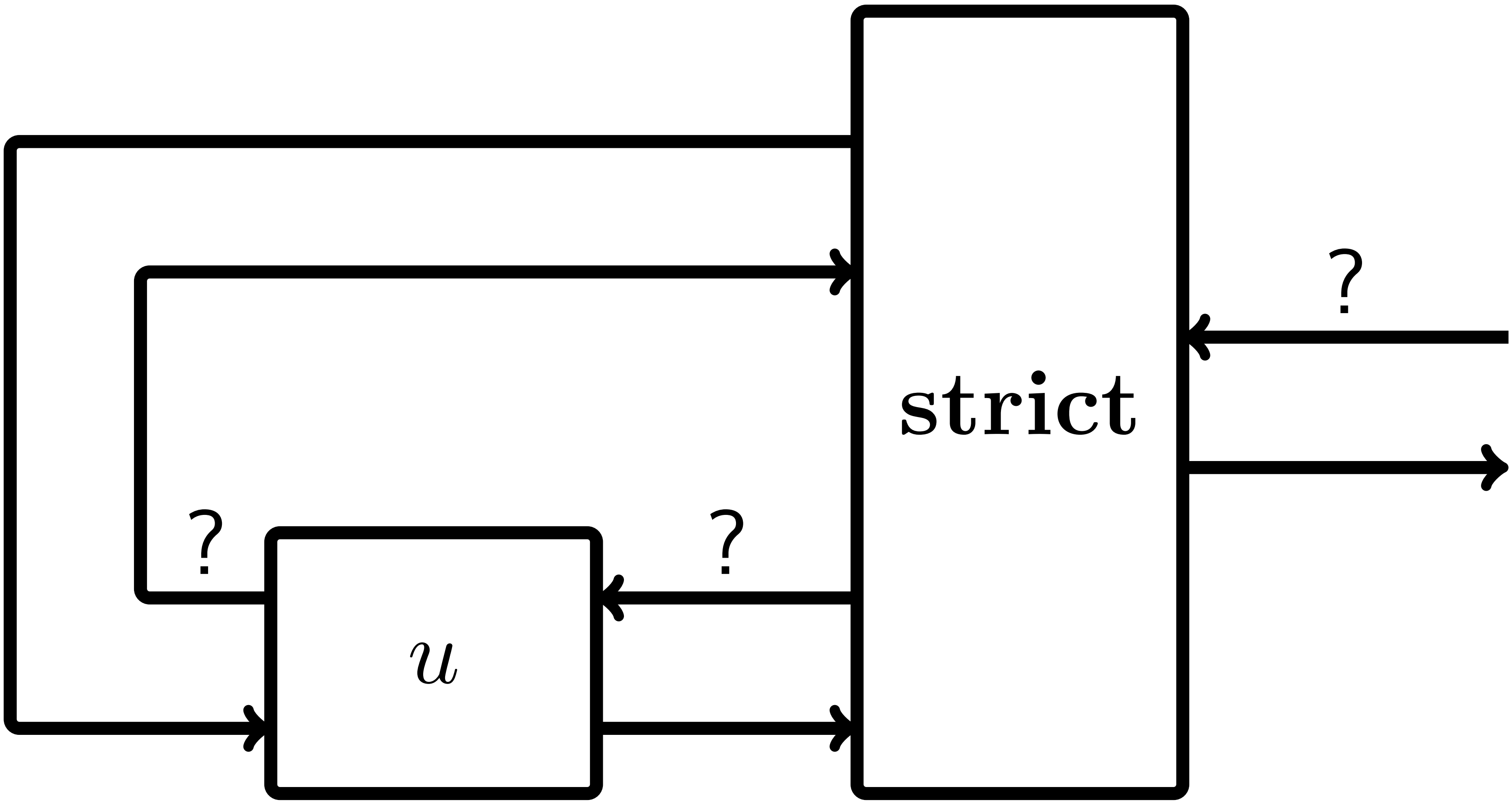
Implementation



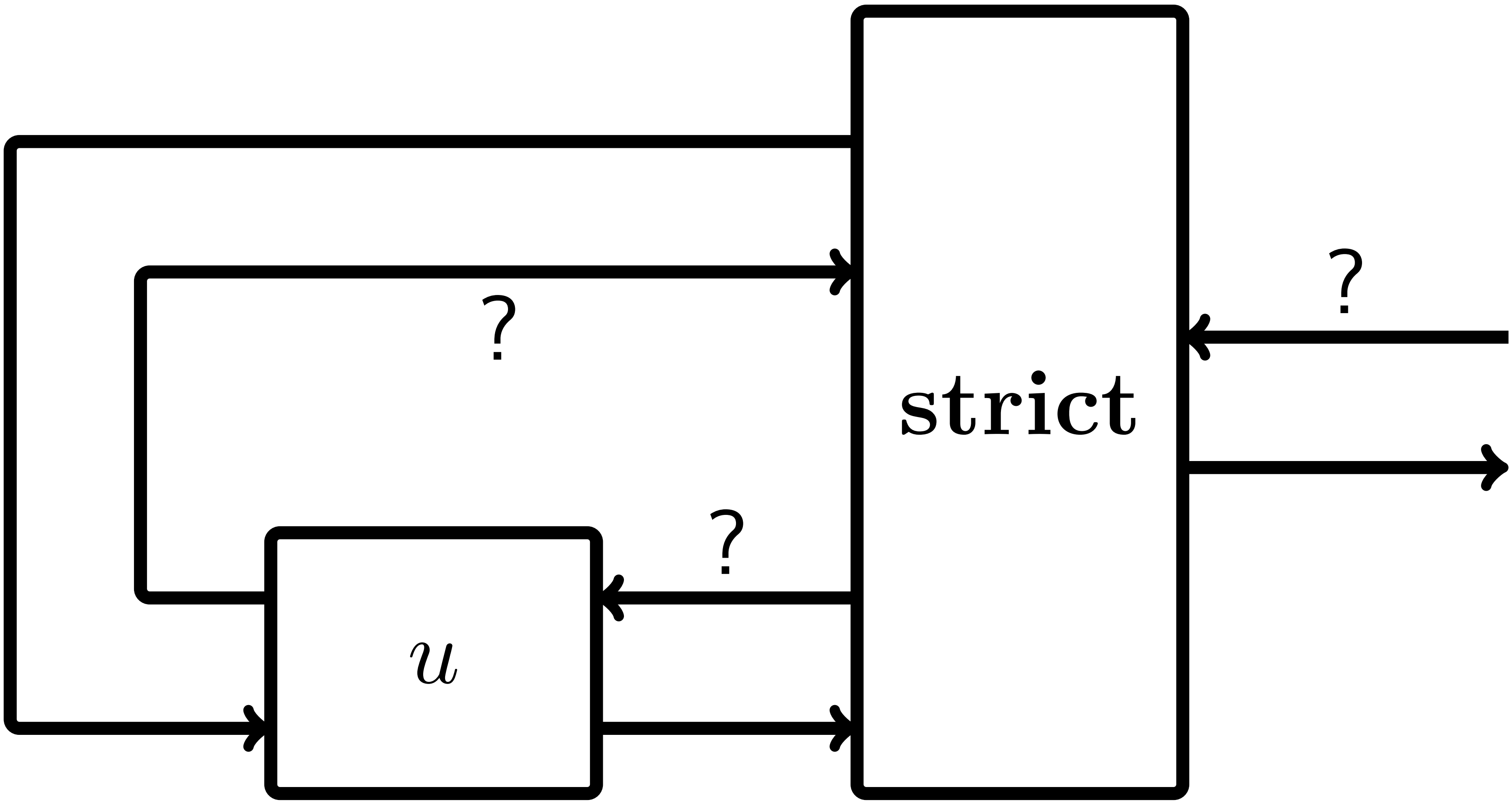
Implementation



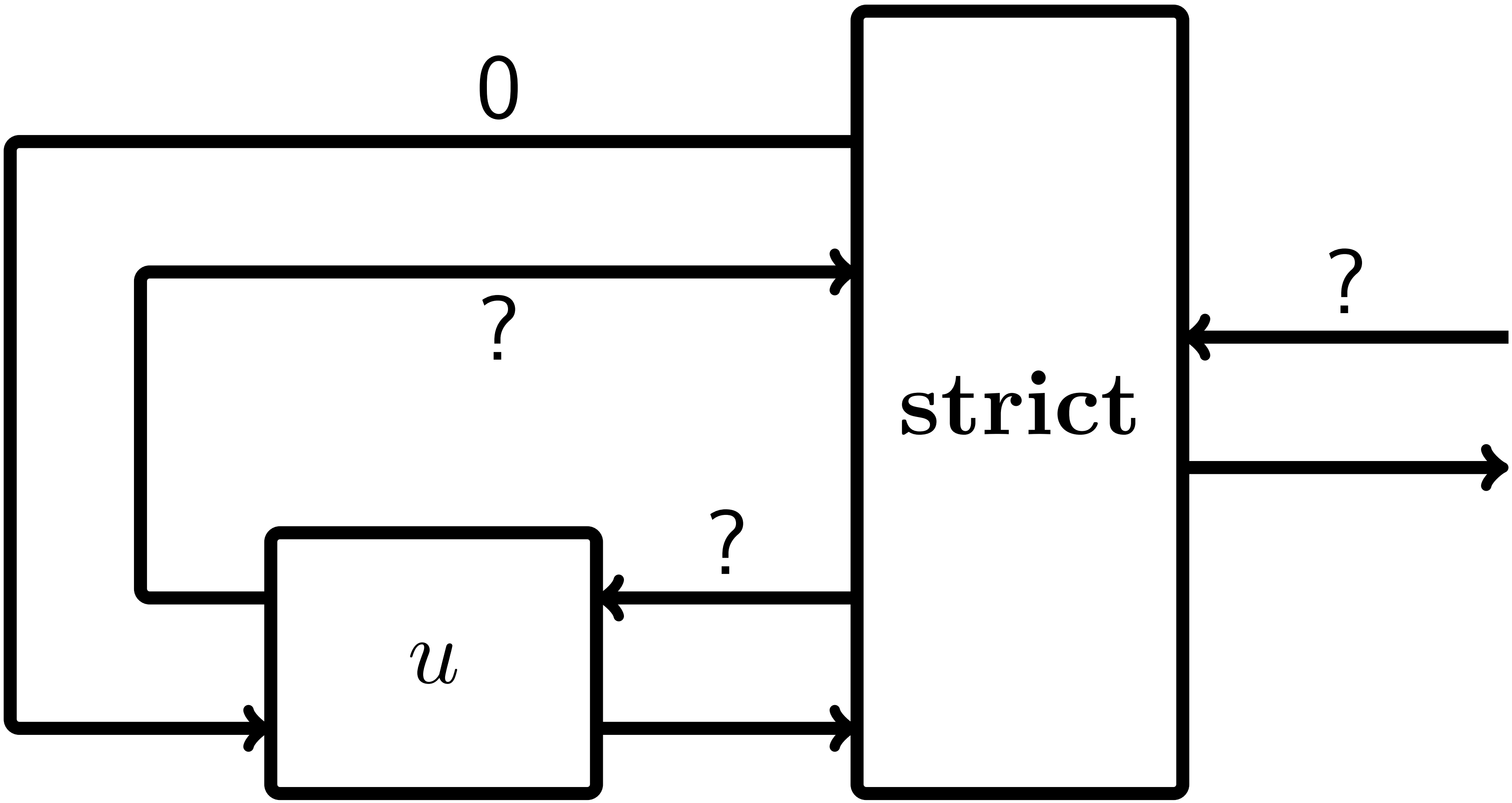
Implementation



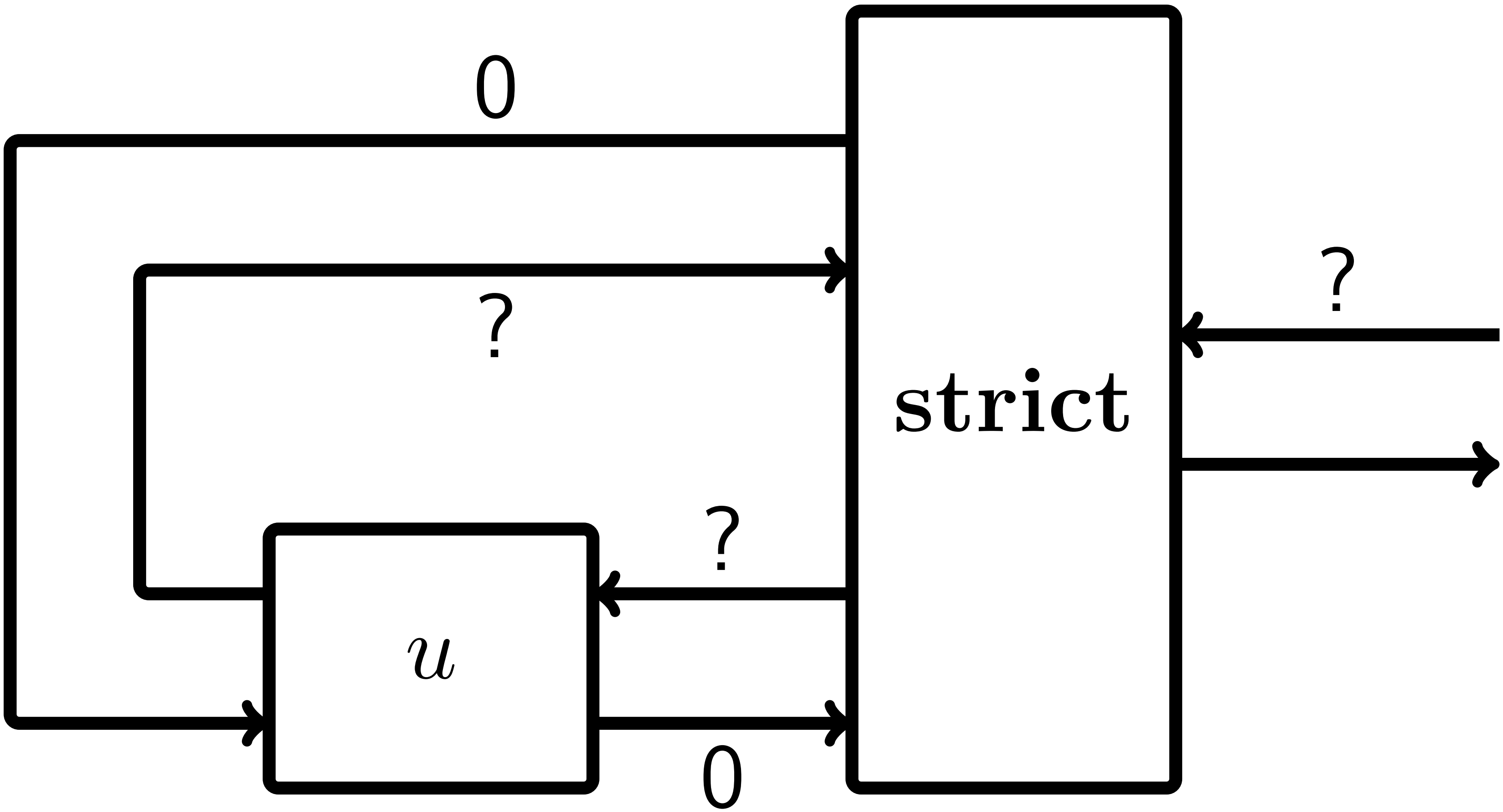
Implementation



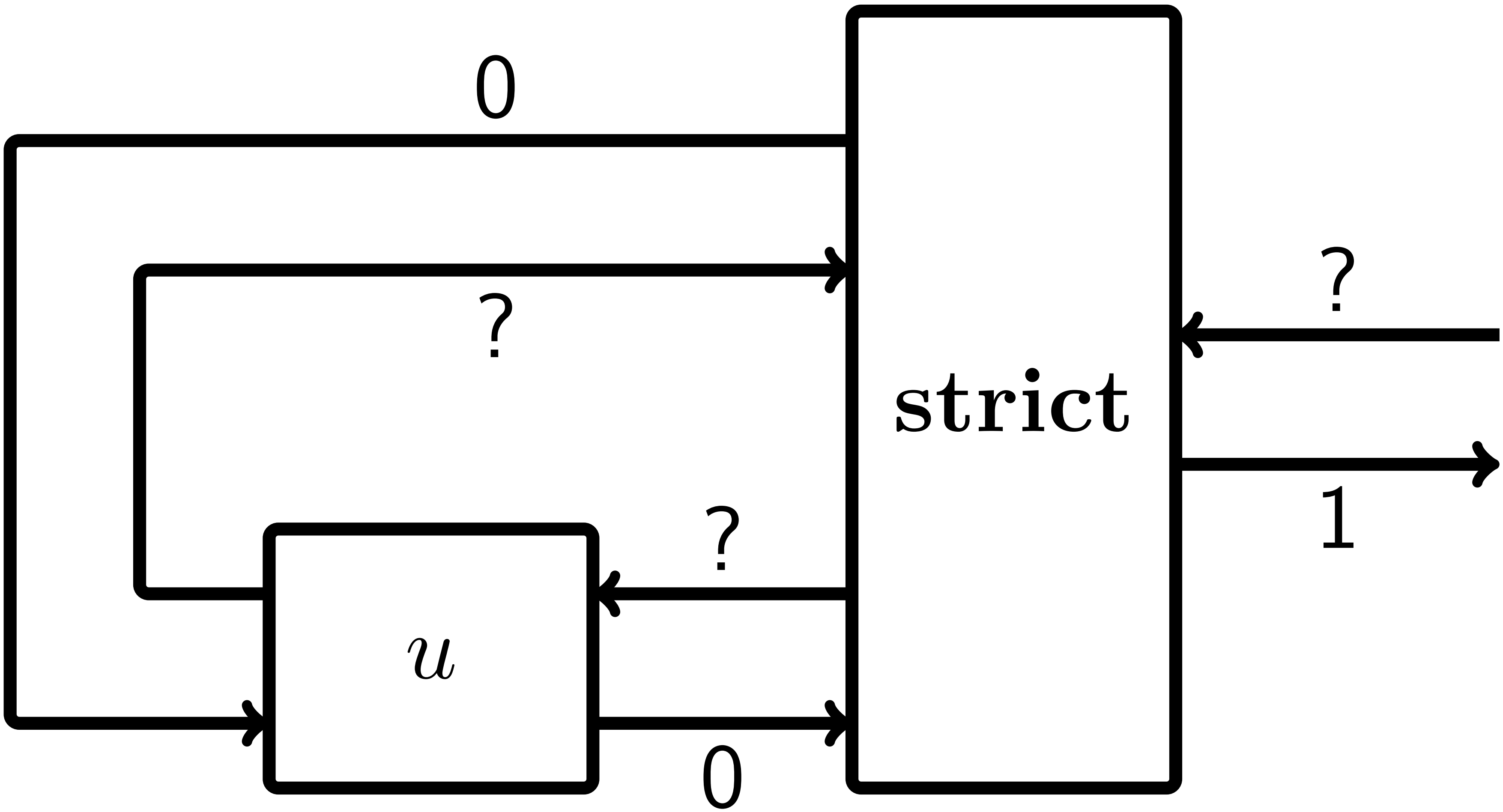
Implementation



Implementation



Implementation



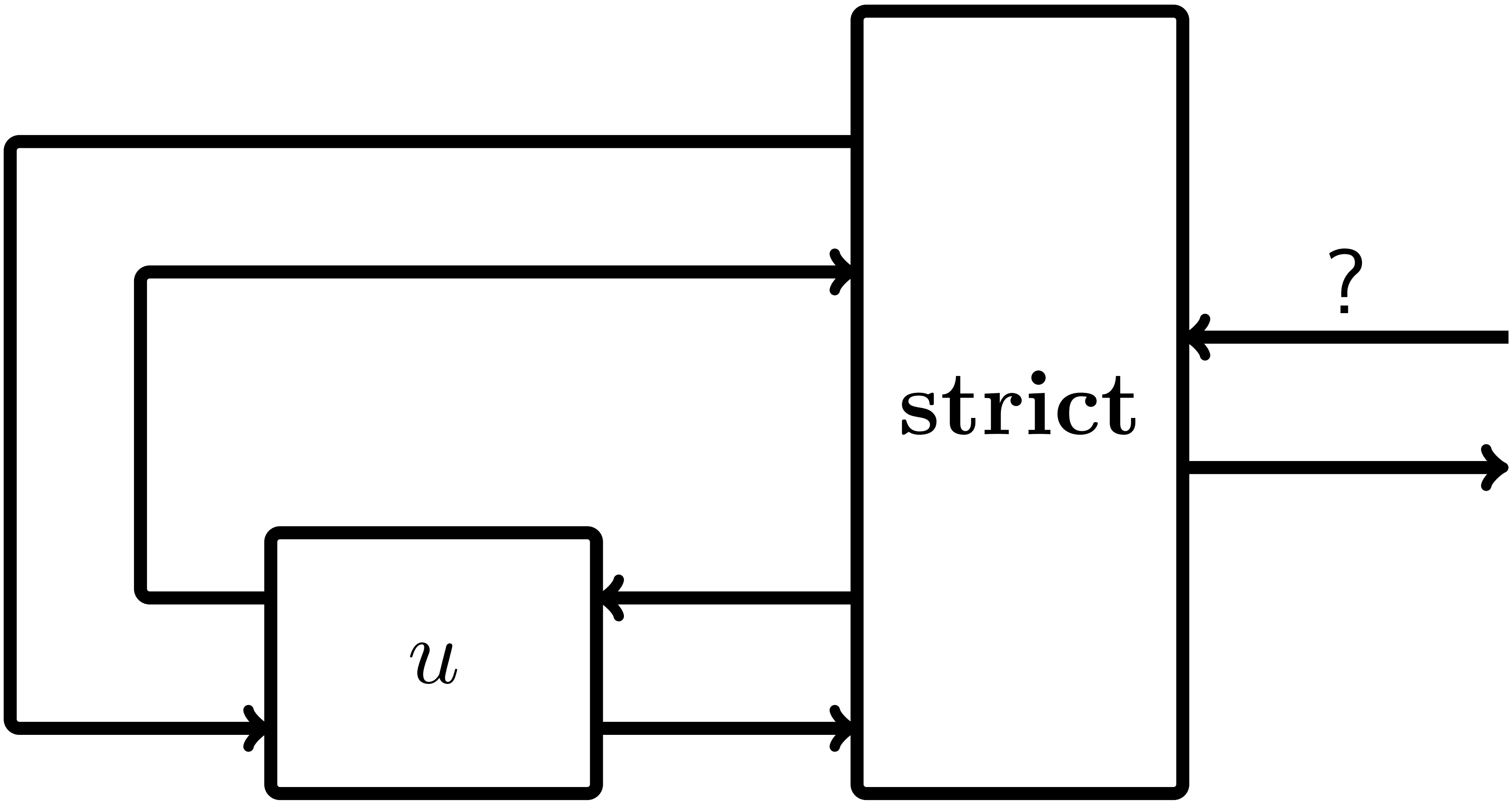
GoI-Rel

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

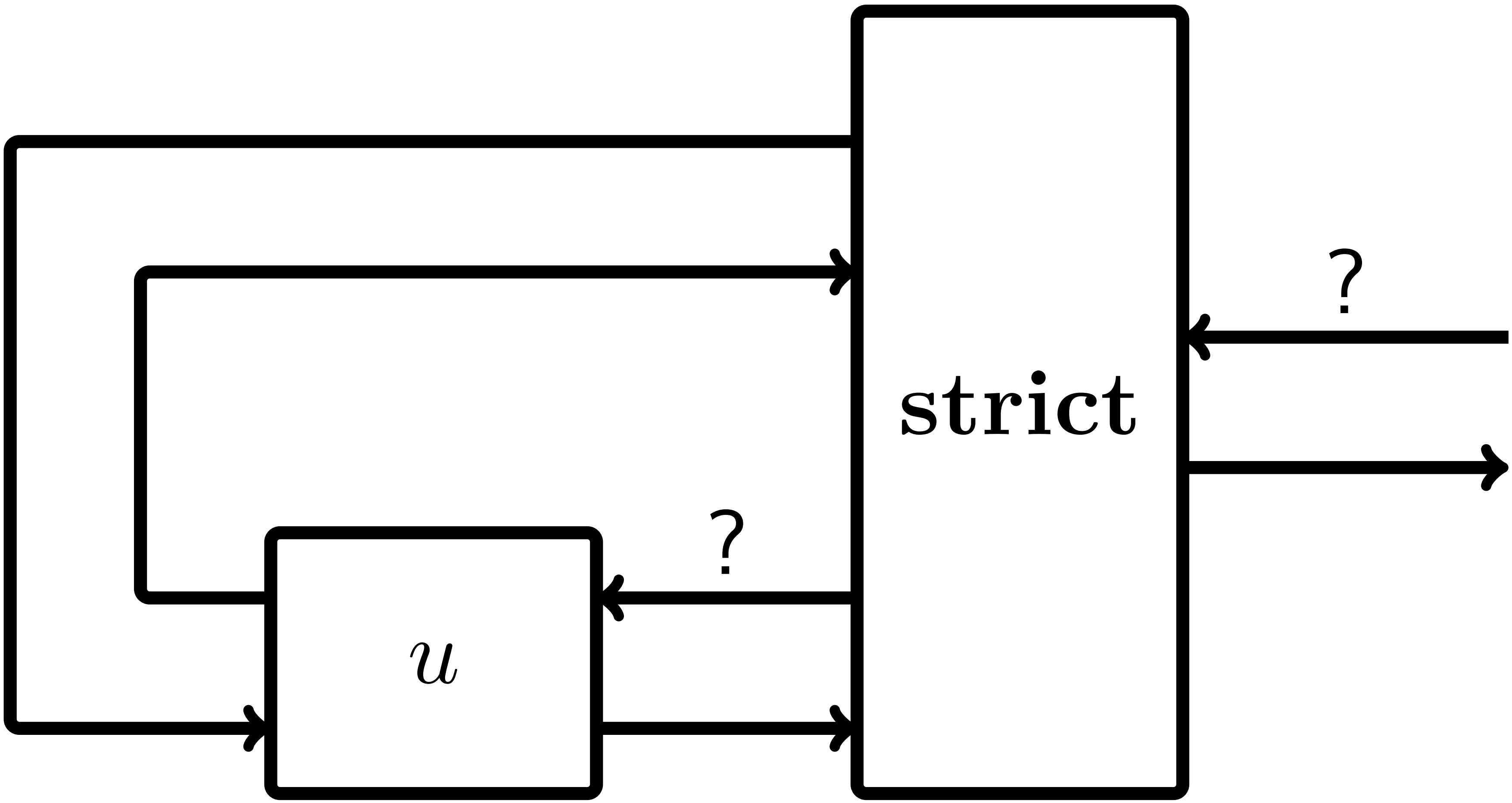
$$\mathbf{strict}(u \colon \mathbb{N}_\perp \to \mathbb{N}_\perp) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

($\mathbf{strict} \notin \mathbf{Cpo}$)

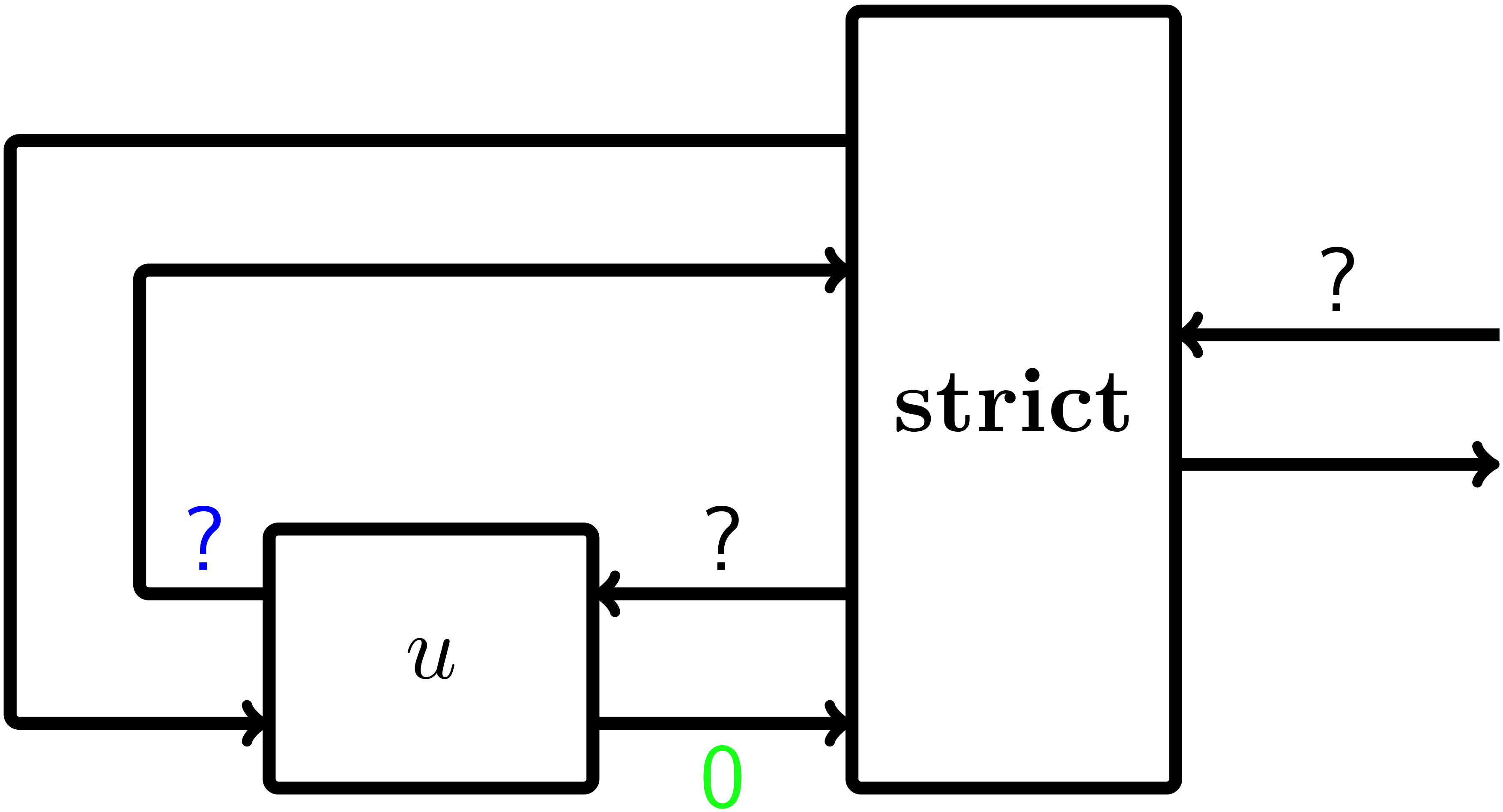
GoI-Rel



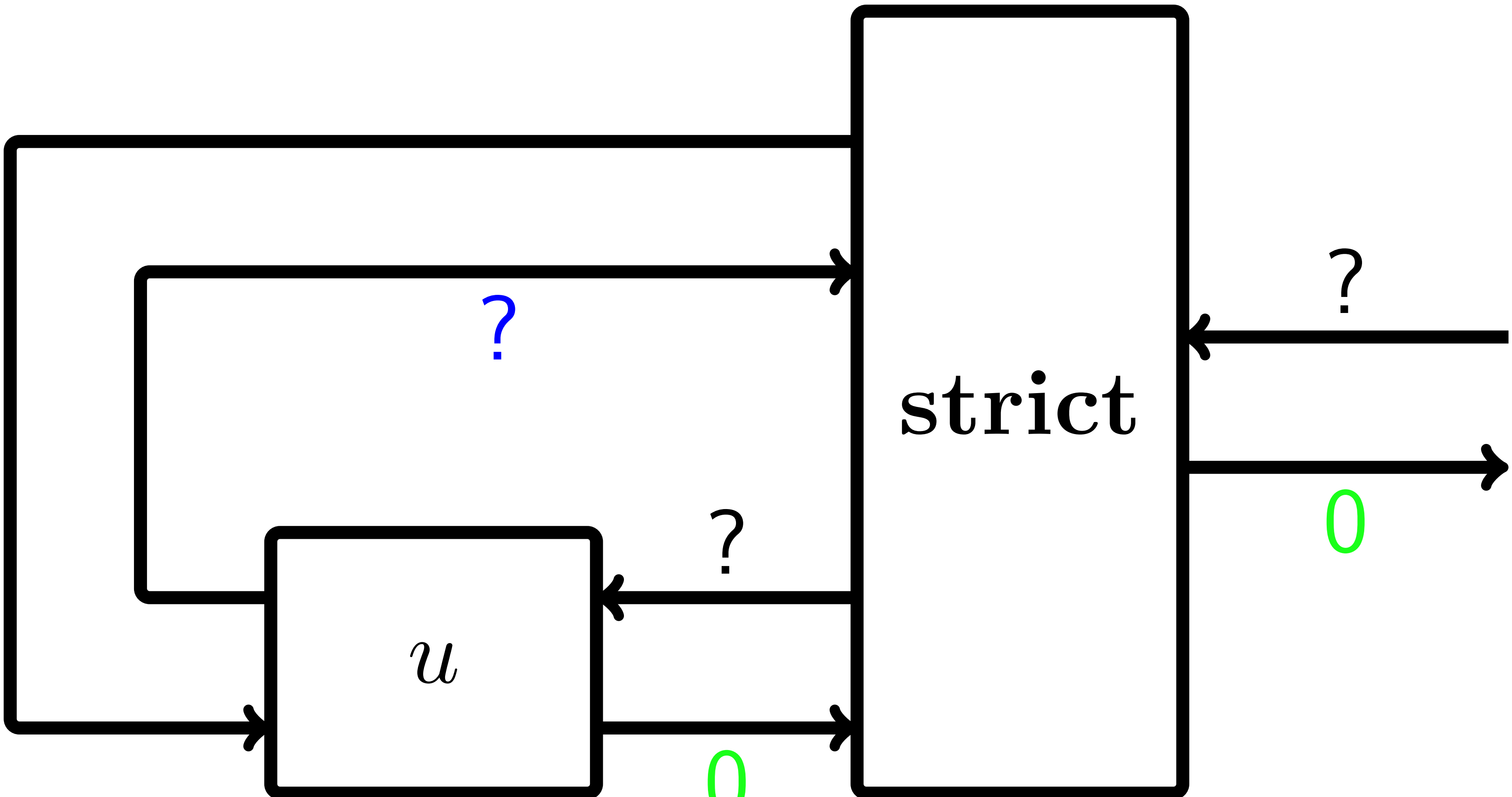
GoI-Rel



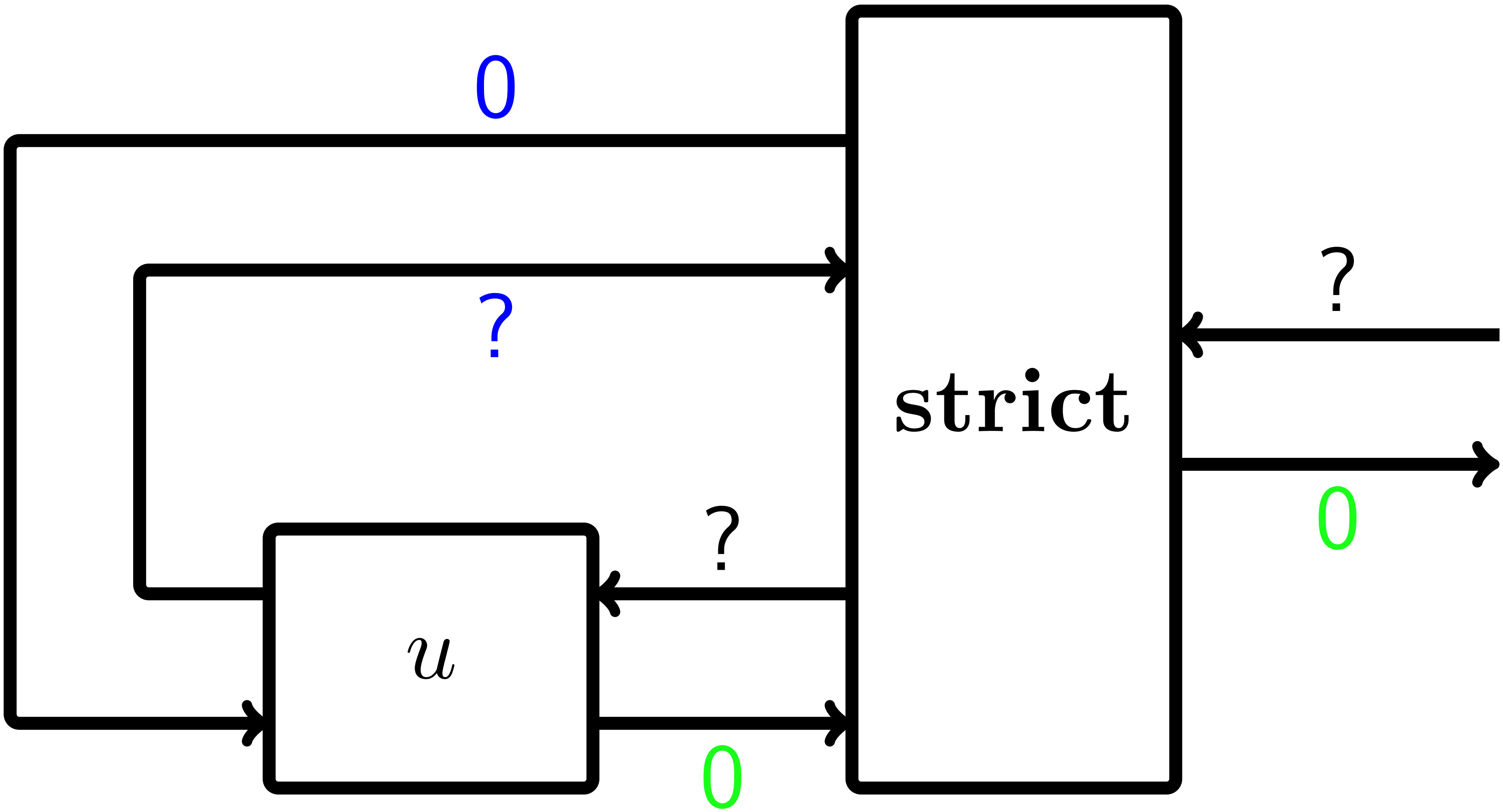
GoI-Rel



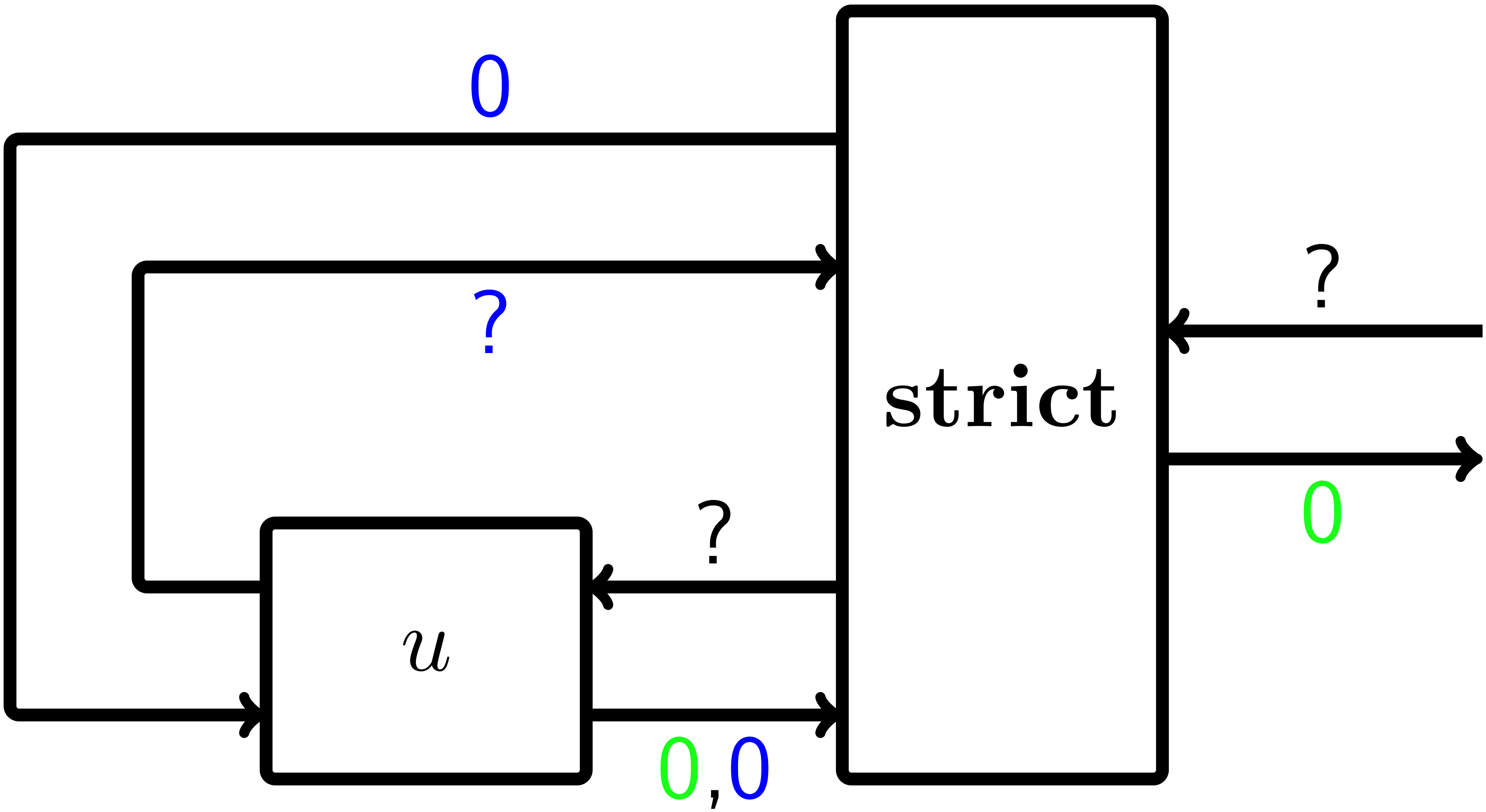
GoI-Rel



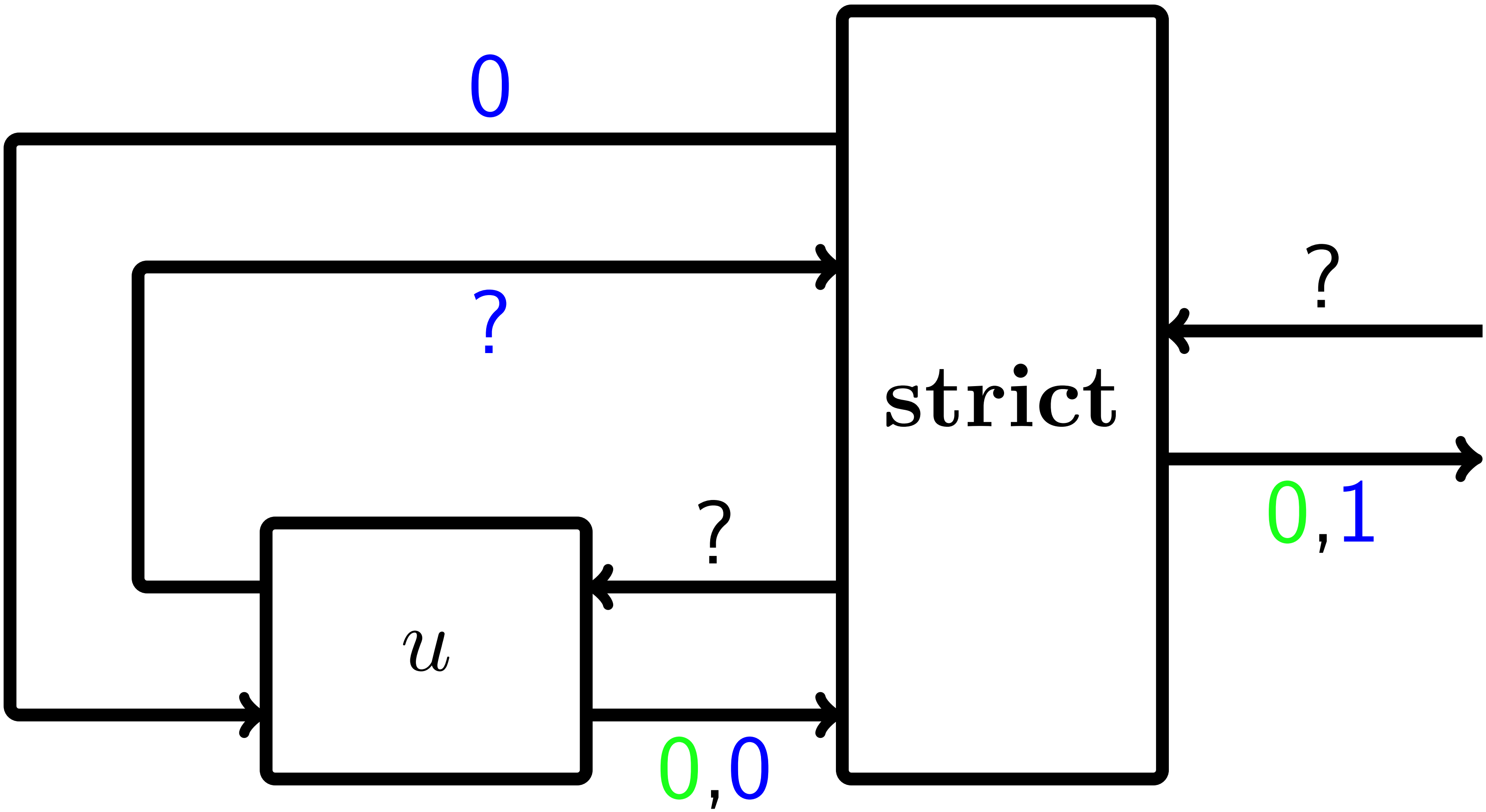
GoI-Rel



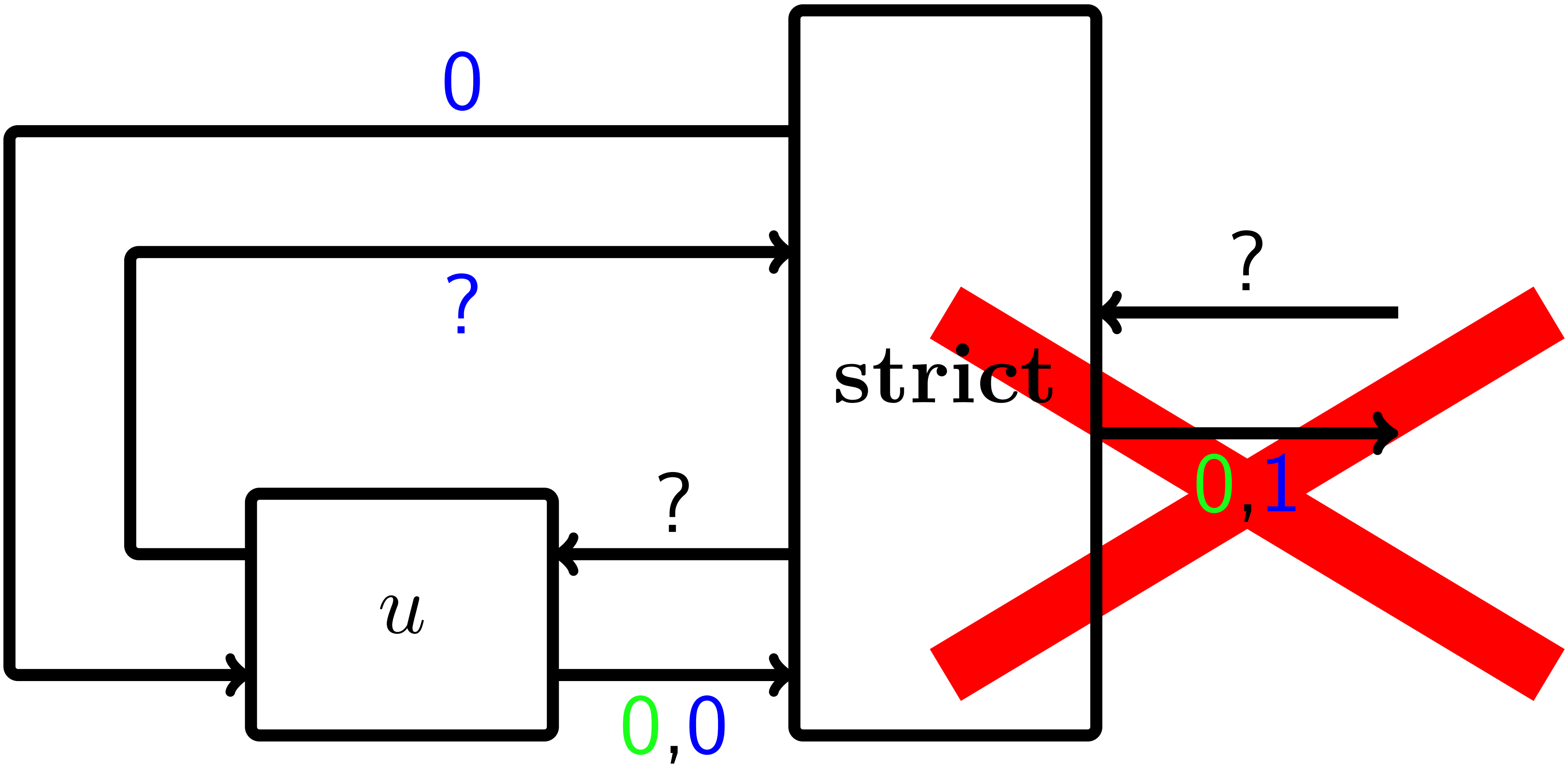
GoI-Rel



GoI-Rel

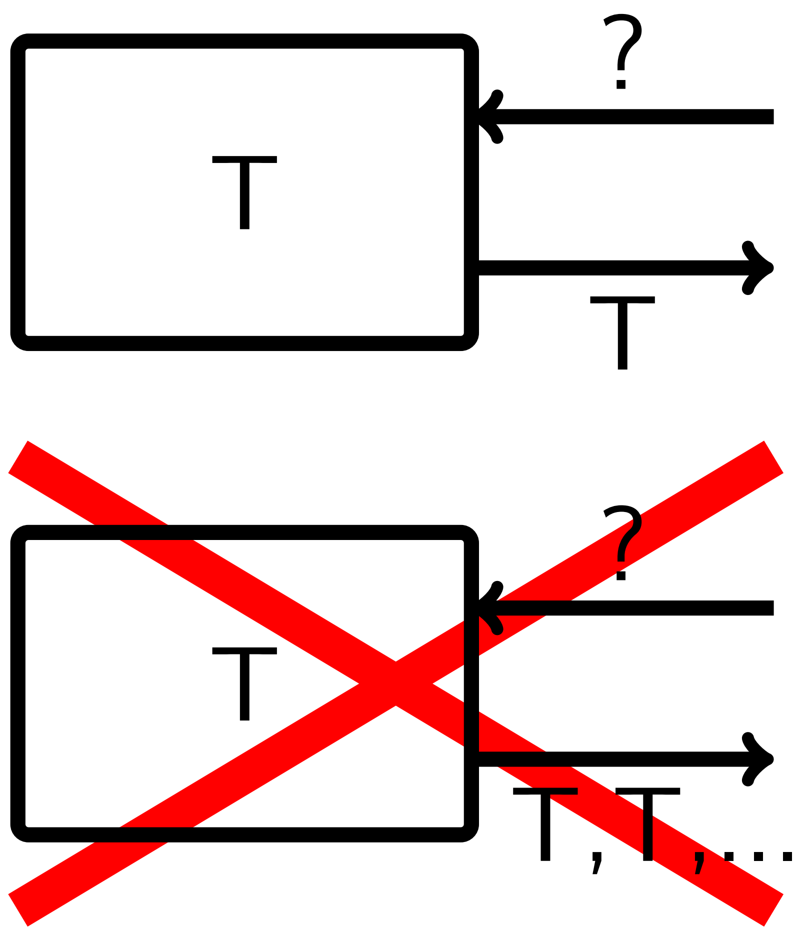


GoI-Rel



GoI-wRel

We can send multiple messages.
BUT, each number must be presented by



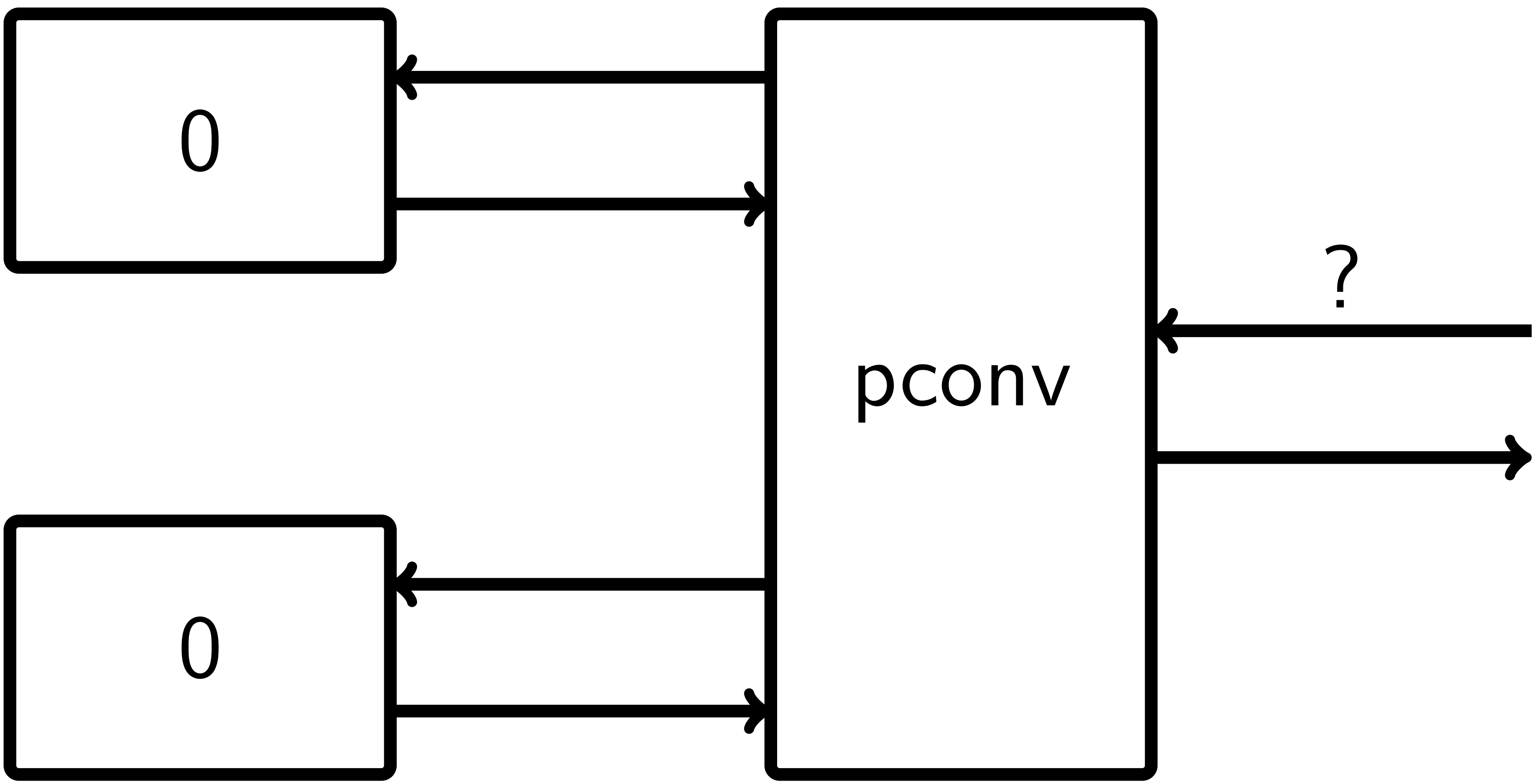
GoI-wRel

We can't quantitatively implement $\mathbf{pconv}$ even if we can send multiple messages

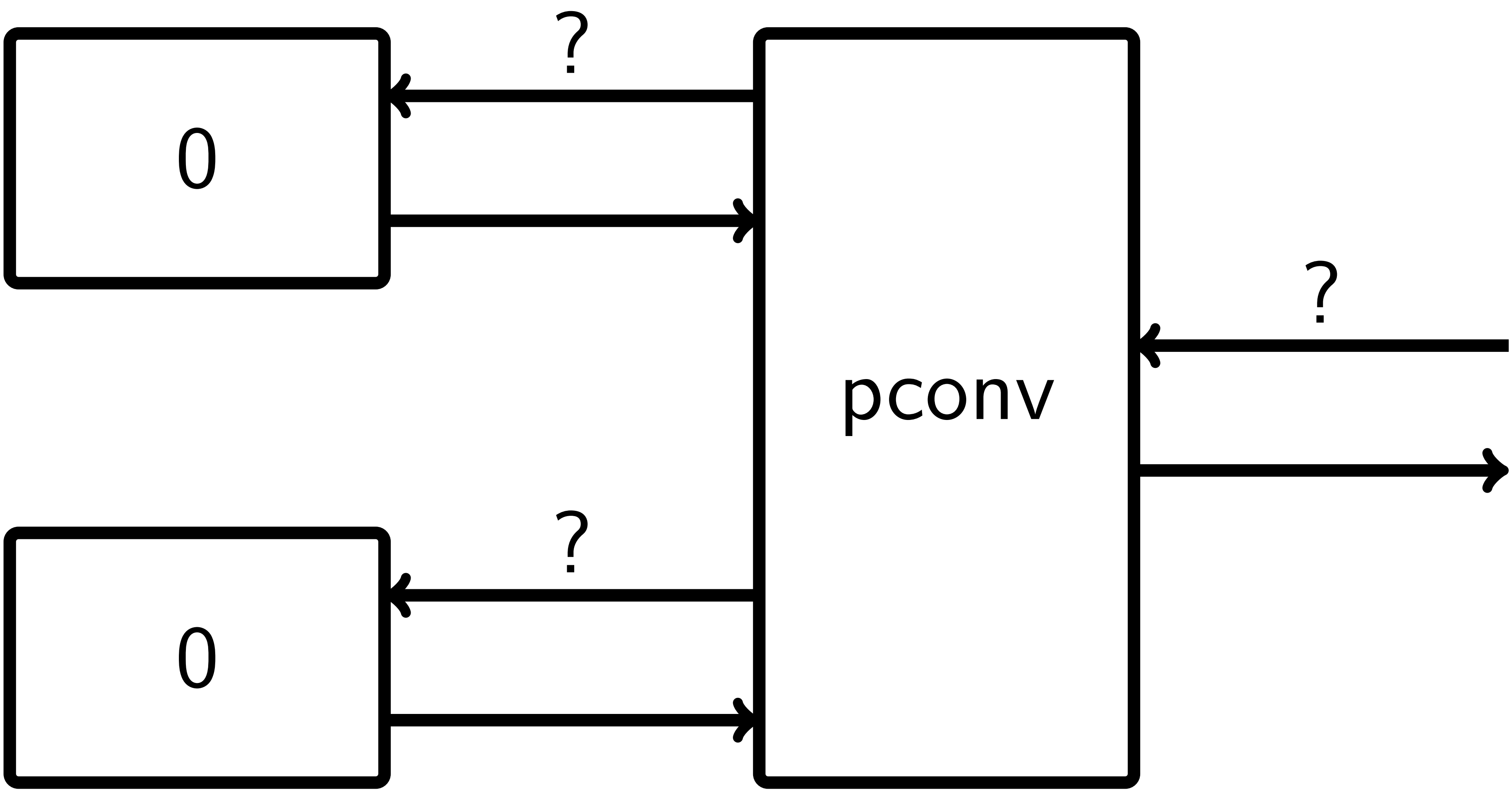
$$\mathbf{pconv}(x\colon \mathbb{N}_\perp, y\colon \mathbb{N}_\perp) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } x \neq \perp \\ 0 & \text{if } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{pconv} \notin \mathbf{Coh})$

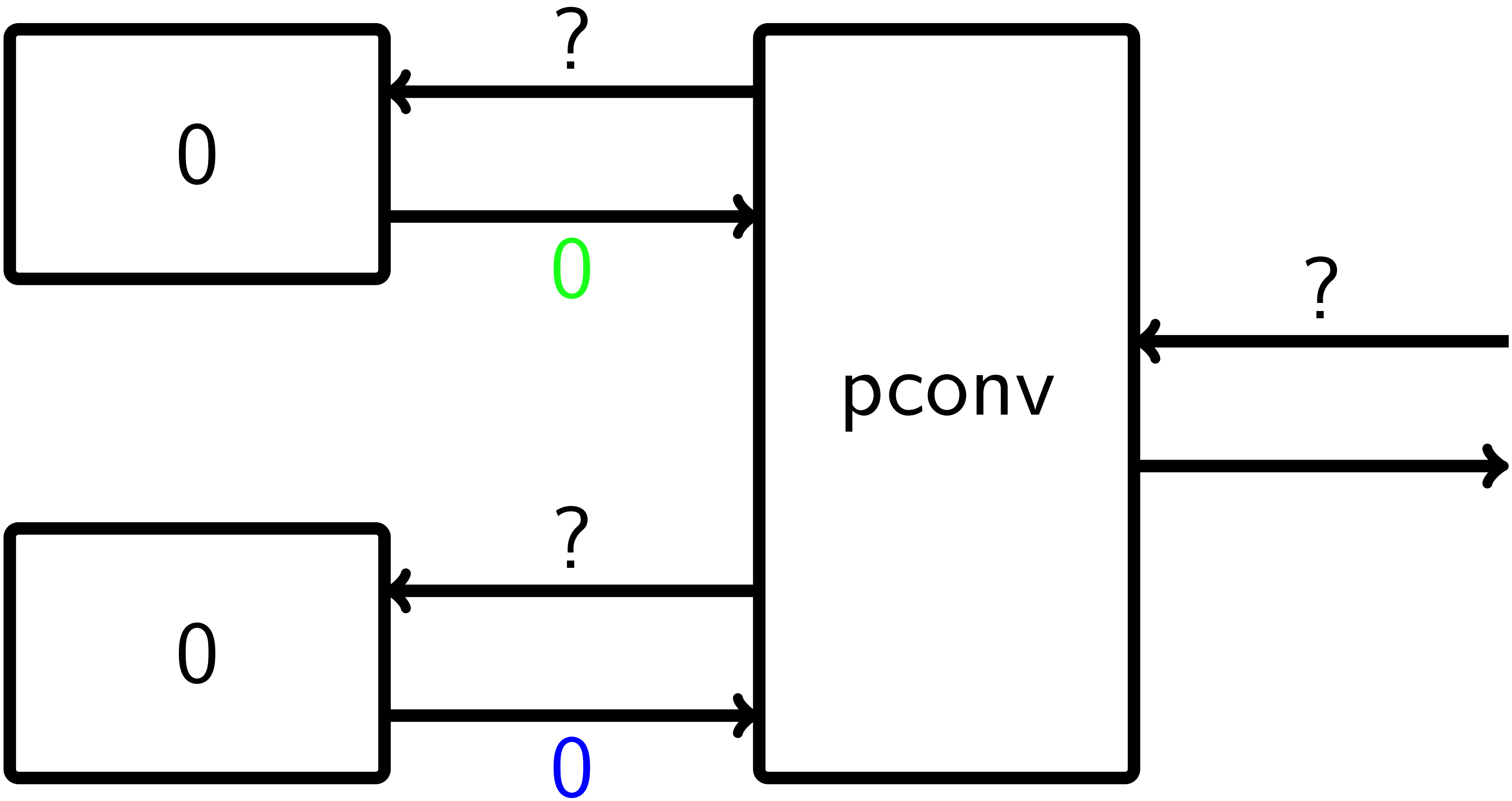
GoI-wRel



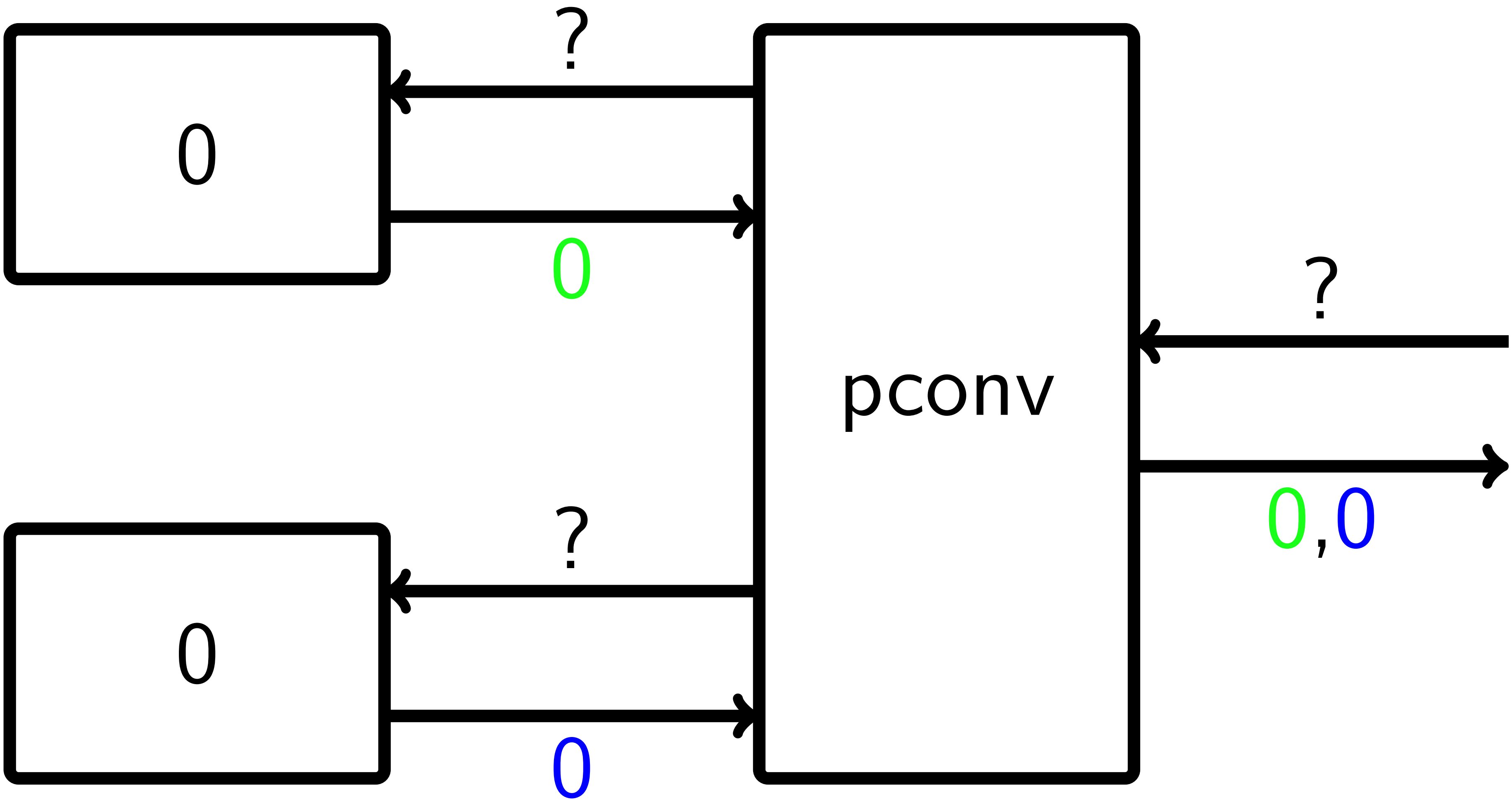
GoI-wRel



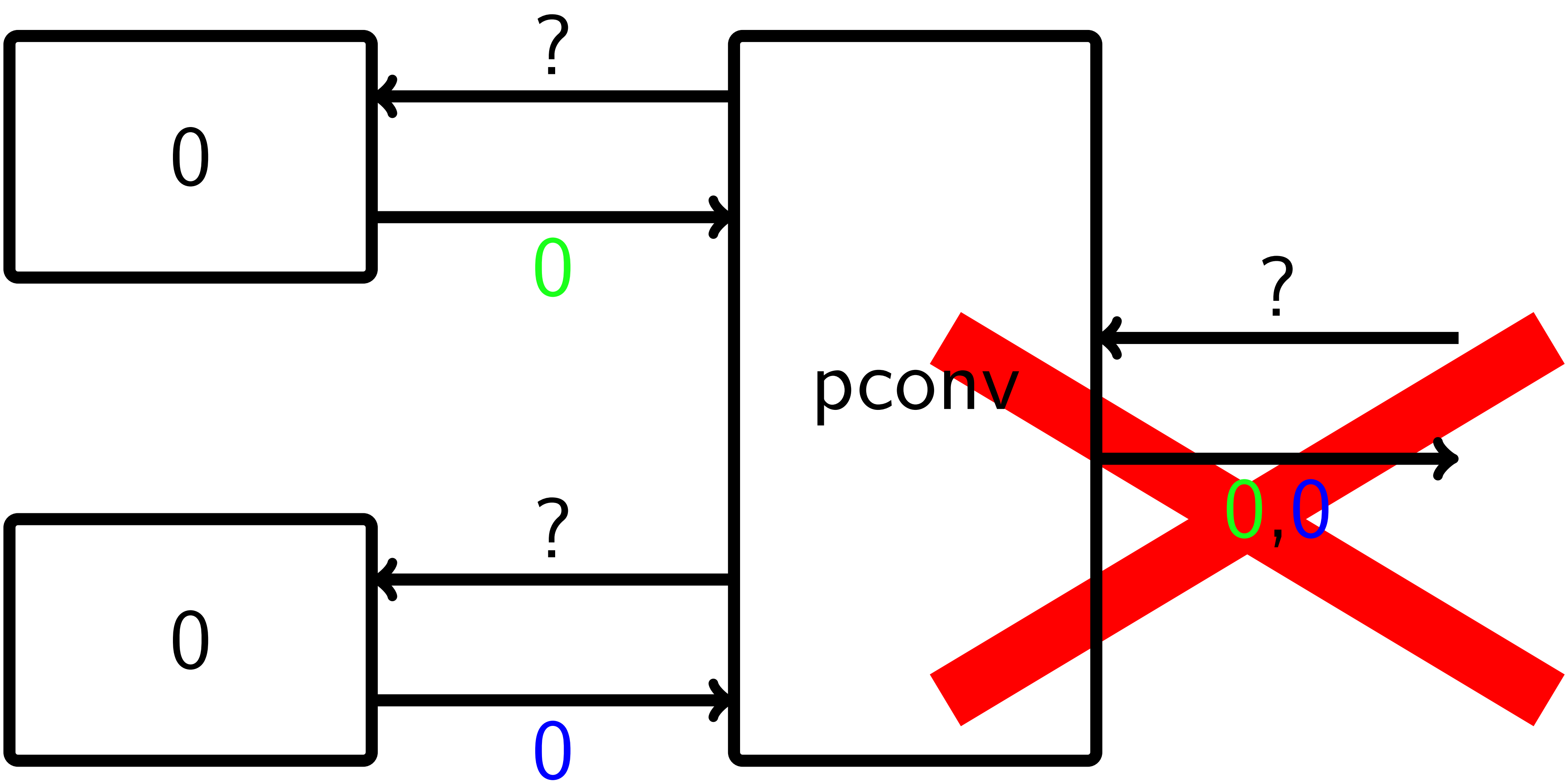
GoI-wRel



GoI-wRel



GoI-wRel



GoI-wRel

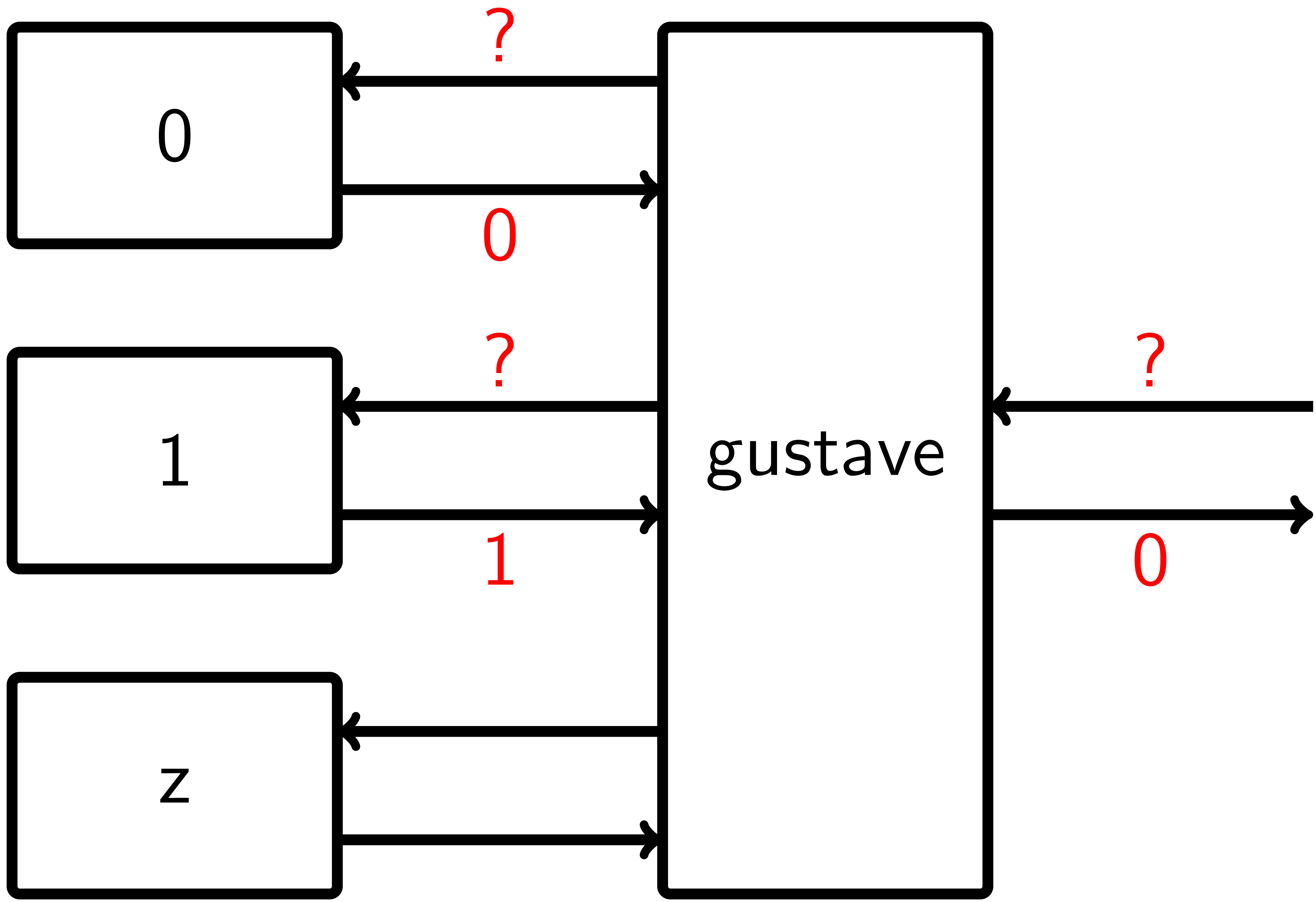
We can quantitatively implement $\mathbf{gustave}$ if we can send multiple messages

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

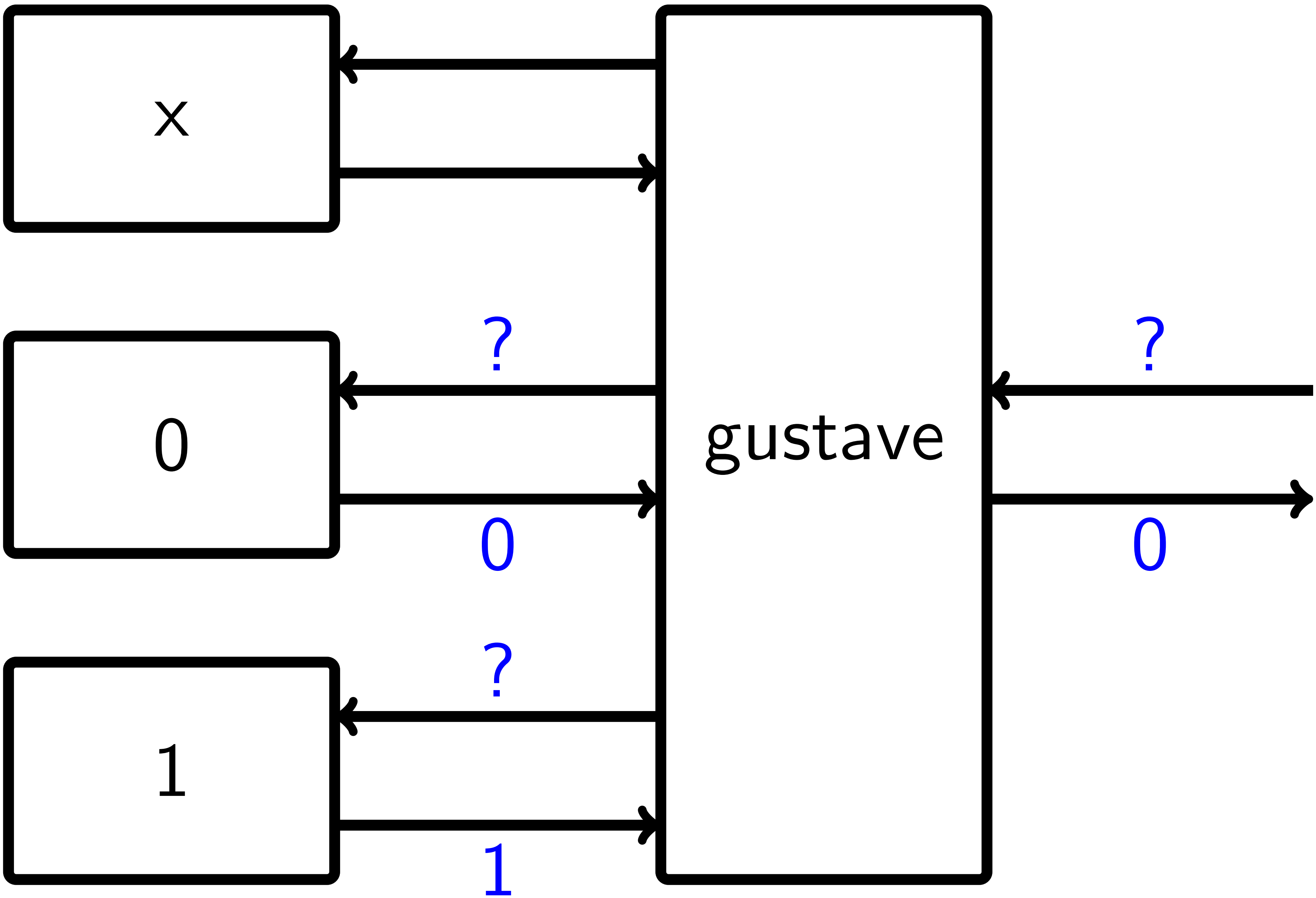
$$\stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } (x,y) = (0,1) \\ 0 & \text{if } (y,z) = (0,1) \\ 0 & \text{if } (z,x) = (0,1) \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{gustave} \in \mathbf{Coh})$

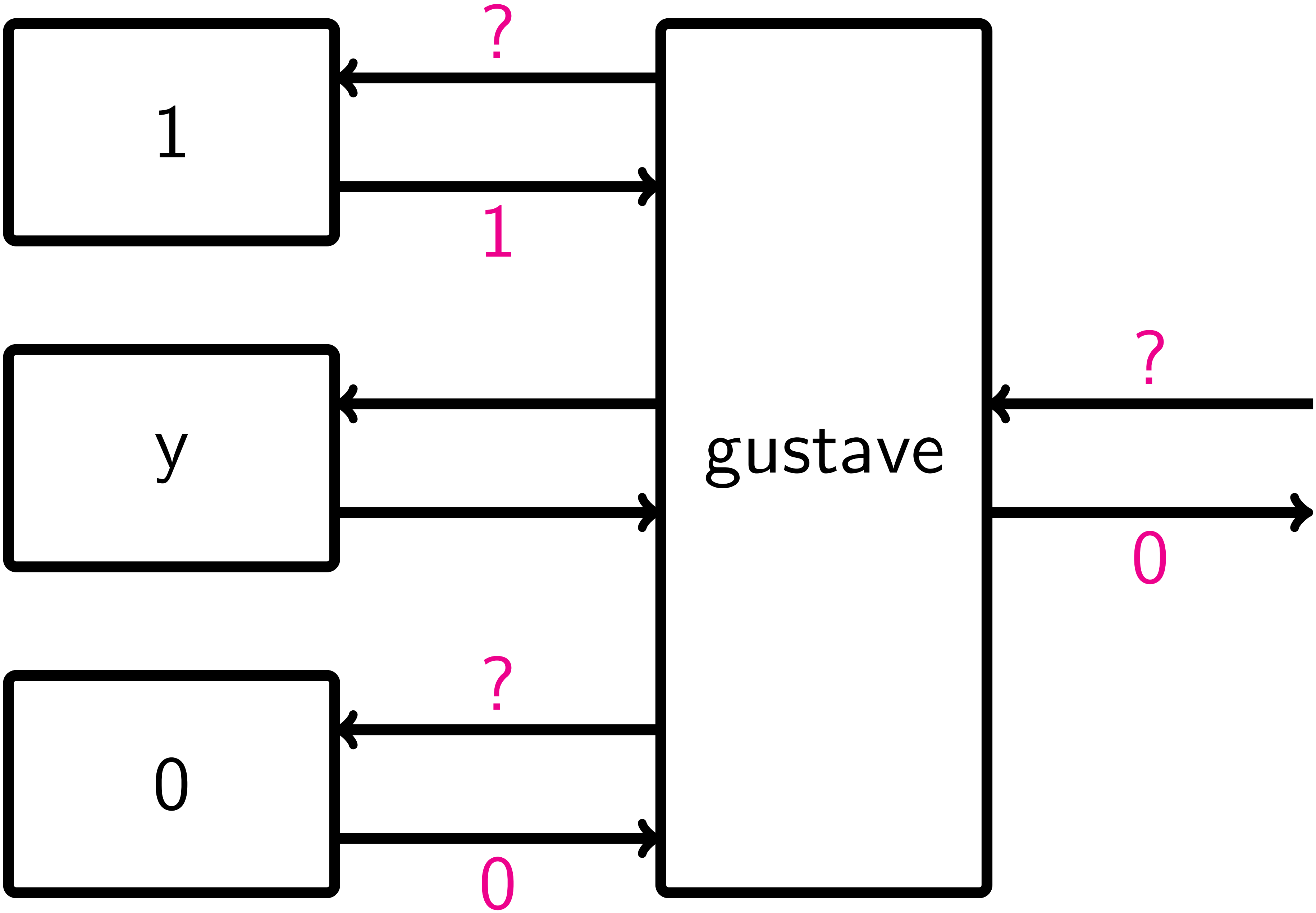
Implementation



Implementation



Implementation



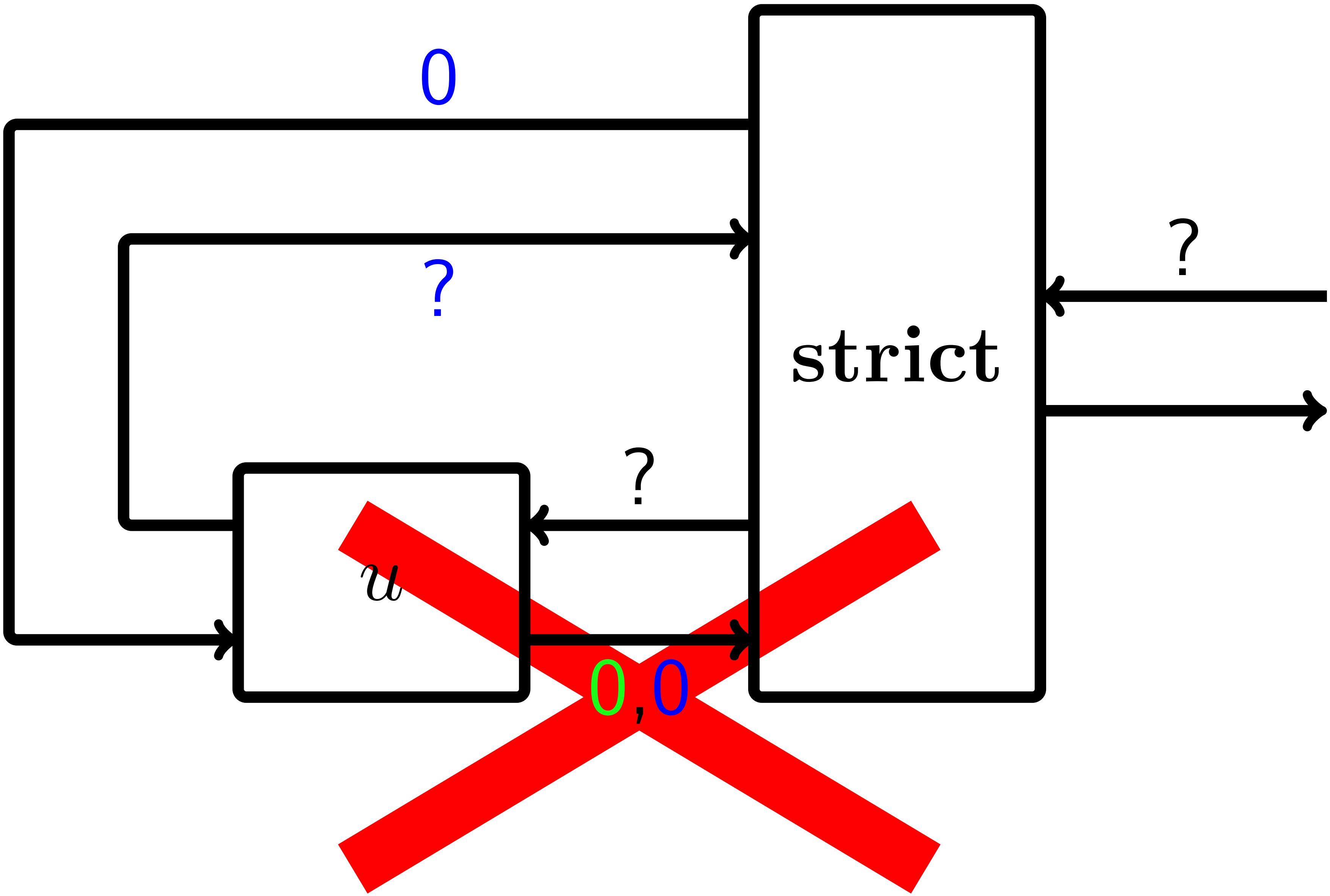
GoI-wRel

We can quantitatively implement $\mathbf{strict}$ even if u may send multiple messages

$$\mathbf{strict}(u\colon \mathbb{N}_\perp \to \mathbb{N}_\perp) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } u(\perp) = 0 \\ 1 & \text{if } u(\perp) = \perp \text{ and } u(0) = 0 \\ \perp & \text{otherwise} \end{cases}$$

$(\mathbf{strict} \in \mathbf{Coh})$

Arg can't deceive



Question

Is there any uniform approach?

Thank you