From Linear Logic to Differential Linear Logic

Paolo Tranquilli

Dipartimento di Matematica
Università degli Studi Roma Tre

Preuves, Programmes et Systèmes
Université Denis-Diderot Paris 7

AILA – XXIII Incontro di Logica – 20/02/2008

Outline

Outline

Vector spaces and LL: is there a link?

- Following Pagani and Maieli's talk, you may have noted an extensive use of jargon borrowed from vector spaces and analysis:

  Linear, dual ($A^{\perp}$), exponential, tensor ($\otimes$), direct sum ($\oplus$)

  What is the catch?

Vector spaces and LL: is there a link?

- Following Maieli's talk, you may have noted an extensive use of jargon borrowed from vector spaces and analysis:

  Linear, dual ($A^{\perp}$), exponential, tensor ($\otimes$), direct sum ($\oplus$)

  What is the catch?

Vector spaces and LL: is there a link?

- Following Maieli's talk and hopefully Pagani's one this Friday, you may note an extensive use of jargon borrowed from vector spaces and analysis:

  Linear, dual ($A^{\perp}$), exponential, tensor ($\otimes$), direct sum ($\oplus$)

  What is the catch?

The link is there, though somewhat vague

- The ground for LL (Girard, 1987) was the study by Girard of System F (in fact, second order intuitionistic sequent calculus or natural deduction)
- Girard imagined formulas A as spaces (coherent spaces) given by atomic bits of information $|A|$
- Moreover, he imagined this bits to be a sort of basis for a vector space
- ...“sort of”?

The link is there, though somewhat vague

- The ground for LL (Girard, 1987) was the study by Girard of System F (in fact, second order intuitionistic sequent calculus or natural deduction)
- Girard imagined formulas A as spaces (coherent spaces) given by atomic bits of information $|A|$
- Moreover, he imagined this bits to be a sort of basis for a vector space
- ...“sort of”?

Linearity in Linear Logic

Factorization of the intuitionistic arrow

$$A \Rightarrow B = !A \multimap B$$

- More on the exponential modality later, now let us consider the linear arrow
- Linearity in R-modules... when $R = \{0, 1\}$ (practically, sets):

$$F(U + V) = F(U) + F(V)$$
$$F(aU) = aF(U)$$

- A linear map F is completely determined by singletons (if also continous wrt directed union)
- In fact for a linear continous function $F : A \multimap B$ atomic bits in the input A contribute each exactly once to the output B

Linearity in Linear Logic

Factorization of the intuitionistic arrow

$$A \Rightarrow B = !A \multimap B$$

- More on the exponential modality later, now let us consider the linear arrow
- Linearity in R-modules... when $R = \{0, 1\}$ (practically, sets):

$$F(U \cup V) = F(U) \cup F(V)$$
$$F(\emptyset) = \emptyset$$

- A linear map F is completely determined by singletons (if also continous wrt directed union)
- In fact for a linear continous function $F : A \multimap B$ atomic bits in the input A contribute each exactly once to the output B

Linearity in Linear Logic

Factorization of the intuitionistic arrow

$$A \Rightarrow B = !A \multimap B$$

- More on the exponential modality later, now let us consider the linear arrow
- Linearity in R-modules... when $R = \{0, 1\}$ (practically, sets):

$$F(U \cup V) = F(U) \cup F(V)$$
$$F(\emptyset) = \emptyset$$

- A linear map F is completely determined by singletons (if also continous wrt directed union)
- In fact for a linear continous function $F : A \multimap B$ atomic bits in the input A contribute each exactly once to the output B

Linearity in Linear Logic

Factorization of the intuitionistic arrow

$$A \Rightarrow B = !A \multimap B$$

- More on the exponential modality later, now let us consider the linear arrow
- Linearity in R-modules... when $R = \{0, 1\}$ (practically, sets):

$$F(U \cup V) = F(U) \cup F(V)$$
$$F(\emptyset) = \emptyset$$

- A linear map F is completely determined by singletons (if also continous wrt directed union)
- In fact for a linear proof $\pi : A \multimap B$ atomic bits in the hypothesis A contribute each exactly once to the thesis B

Linearity at last

in Logic

using a hypothesis exacly once

in Computer Science

using an input/resource exacly once

in Analysis

$F(aU + bV) = aF(U) + bF(V)$ (and continuity)

...though the link with the third is not really convincing for now...

Linearity at last

in Logic

using a hypothesis exacly once

in Computer Science

using an input/resource exacly once

in Analysis

$F(aU + bV) = aF(U) + bF(V)$ (and continuity)

...though the link with the third is not really convincing for now...

A note about duals

- $A^\perp$ in R-modules is the space of linear functions $A \multimap R$
- This models negation: in case of bidual spaces ($A \cong A^{\perp\perp}$), there is an isomorphism between $A \multimap B$ and $B^\perp \multimap A^\perp$
- Moreover, at the core of the cut rule is the interaction between A and $A^\perp$

$$U \in A, V \in A^\perp : \langle U, V \rangle = \sum_{x \in |A|} U_x V_x$$

- The condition defining coherent spaces is that the support of $\langle U, V \rangle$ is at most 1, i.e. $\#U \cap V \leq 1$

Let us do it better than LL

- We relax the condition and just ask $\langle U, V \rangle$ to be finite sums:

  $$\#U \cap V < \omega$$

  (finiteness spaces, Ehrhard 2003)
- $R\langle A \rangle$, a submodule of $R^{|A|}$ (with a condition on supports), for any unitary semiring R yields spaces for which the all the connectives really correspond to the vector space constructs
- For example: $R\langle A \multimap B \rangle$ is isomorphic to $R\langle A \rangle \multimap R\langle B \rangle$, i.e. the linear morphisms continuous in the linear topologies of $R\langle A \rangle$ and $R\langle B \rangle$ (defined independently of the structure of finiteness spaces)

Formal sums – 1

- Condition $\#U \cap V \leq 1$ in coherent spaces can be seen as preventing proofs to be spread in sets of proofs during reduction
- Here the support of an interaction does not need to be a singleton, i.e. the result of an interaction is a generic (though finite) sum with coefficients in R: a proof can reduce to a sum!
- What does it mean?

Formal sums – 2

- The easiest way to regard this is non-deterministic choice. The sum keeps track of the various possible choices (exspecially if $R = \mathbb{N}$):

$$\sigma \begin{array}{c} \nearrow \pi_1 \\ \searrow \pi_2 \end{array} \quad \rightsquigarrow \sigma \to \pi_1 + \pi_2$$

- Other interpretations are possible, for example by taking $R = \mathbb{R}^+$ and interpreting probabilistic choice:

$$\sigma \begin{array}{c} \overset{p}{\nearrow} \pi_1 \\ \underset{1-p}{\searrow} \pi_2 \end{array} \quad \rightsquigarrow \sigma \to p\pi_1 + (1-p)\pi_2$$

The exponential

- A linear world would be boring. . . the exponential modality $!A$ brings back contraction and weakening of hypotheses
- In finiteness spaces, the modality operates a shift from *linear* formal sums to formal power series.

$A \multimap B$ are linear functions

- In fact a power x^n is a multilinear use of multiple (possibly none) occurrences of x

The exponential

- A linear world would be boring. . . the exponential modality $!A$ brings back duplication and erasing of inputs
- In finiteness spaces, the modality operates a shift from *linear* formal sums to formal power series.

$A \multimap B$ are linear functions

- In fact a power x^n is a multilinear use of multiple (possibly none) occurrences of x

The exponential

- A linear world would be boring. . . the exponential modality $!A$ brings back contraction and weakening of hypotheses
- In finiteness spaces, the modality operates a shift from *linear* formal sums to formal power series.

$A \multimap B$ are linear functions

- In fact a power x^n is a multilinear use of multiple (possibly none) occurrences of x

The exponential

- A linear world would be boring. . . the exponential modality $!A$ brings back contraction and weakening of hypotheses
- In finiteness spaces, the modality operates a shift from *linear* formal sums to formal power series.

$!A \multimap B$ are analytical functions

- In fact a power x^n is a multilinear use of multiple (possibly none) occurrences of x

Linearity revisited

in Logic

Linear hypotheses, to be used exacly once

in Computer Science

Linear inputs, to be used exactly once

in Analysis

Linear arguments occurring in linear position

We will see some of the consequences that this approach brings in the top areas

Linearity revisited

in Logic

Non-linear hypotheses, that can be contracted or weakened

in Computer Science

Non-linear inputs, that can be duplicated or erased

in Analysis

Non-linear arguments occurring as powers

We will see some of the consequences that this approach brings in the top areas

Linearity revisited

in Logic

Proofs

in Computer Science

Programs

in Analysis

Analytical functions

We will see some of the consequences that this approach brings in the top areas

Outline

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \Longrightarrow \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \Longrightarrow (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

Lambda-Calculus

- A core functional programming language:

$$M ::= x \mid \lambda x.M \mid (M)N$$

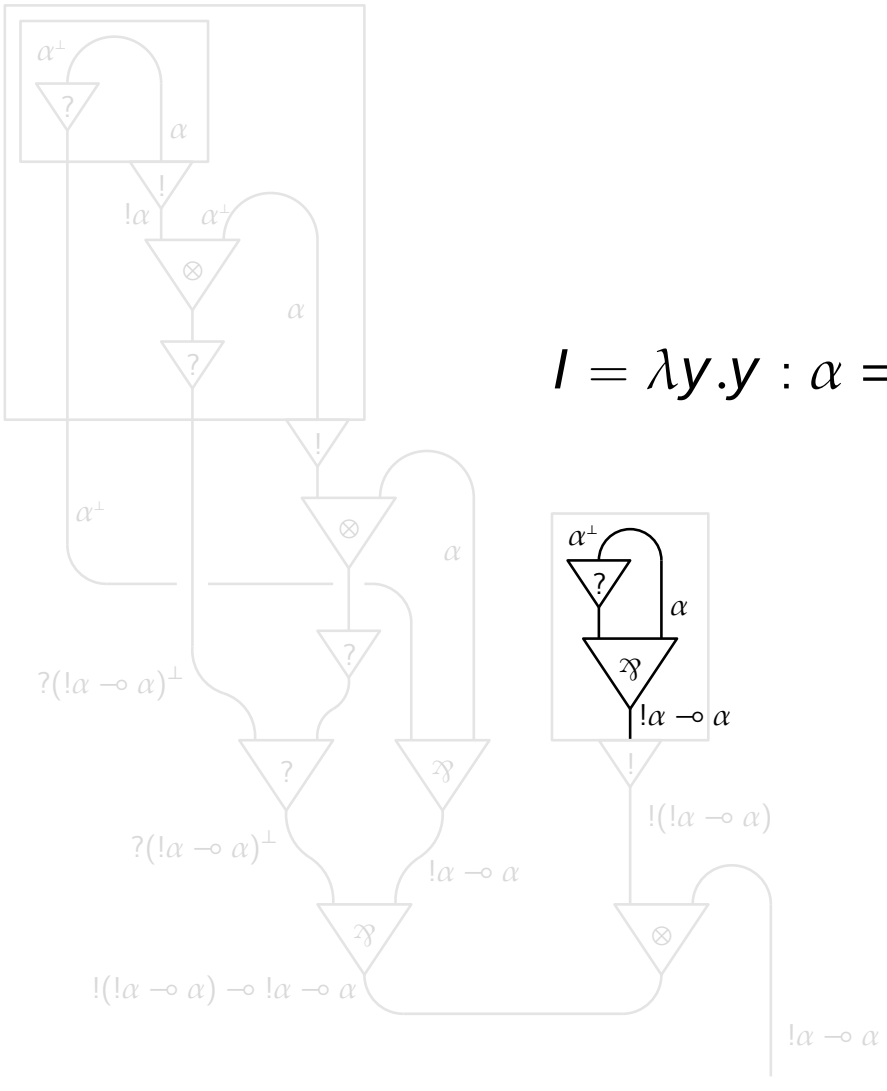
. . . with types:

$$x : A \qquad x : A, M : B \implies \lambda x.M : A \Rightarrow B,$$

$$M : A \Rightarrow B, N : A \implies (M)N : B$$

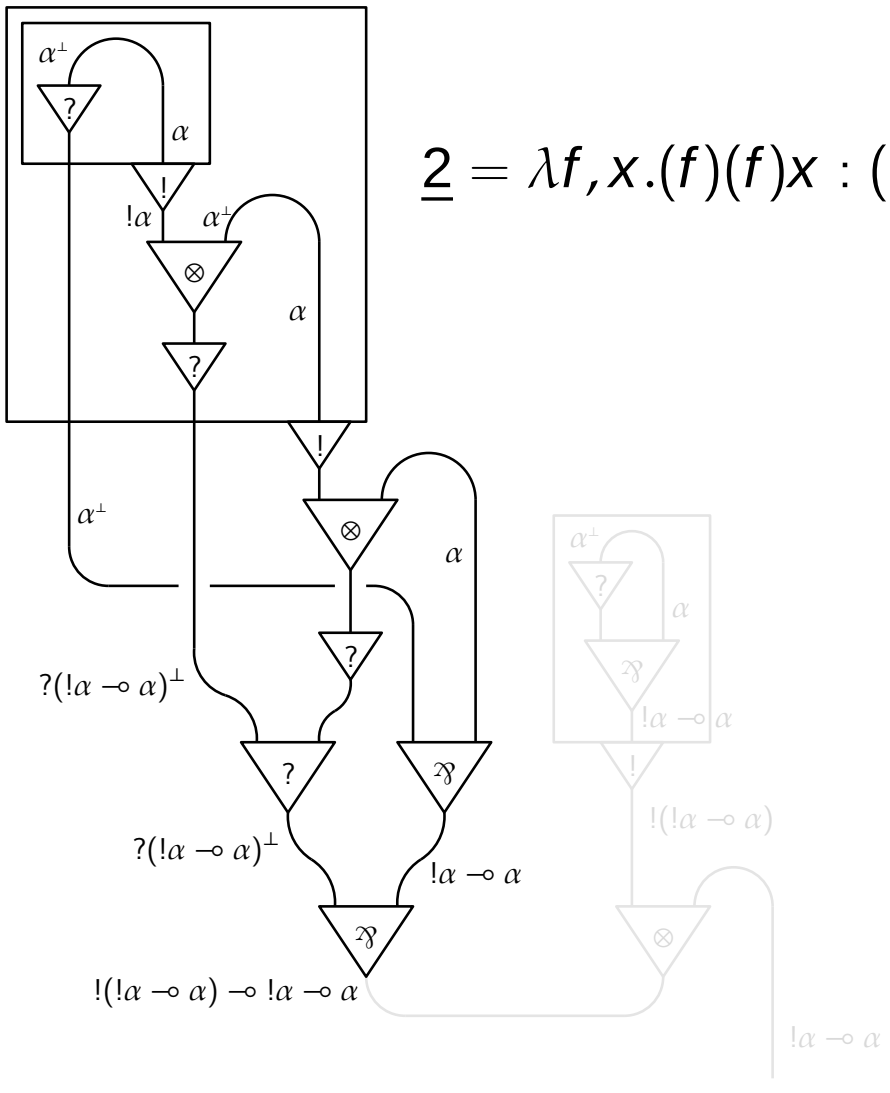
- Linear Logic Proof-Nets provide a geometrical representation of λ-terms

An example



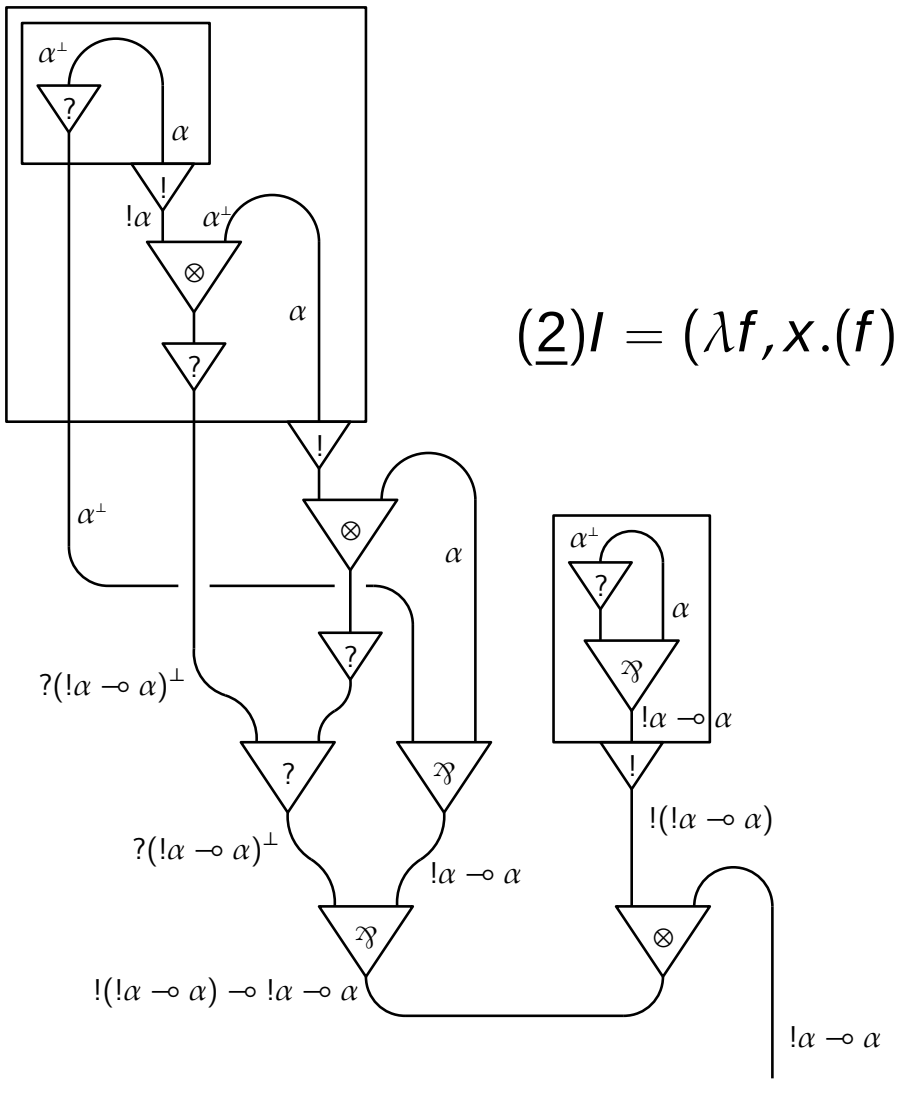
$$I = \lambda y.y : \alpha \Rightarrow \alpha$$

An example



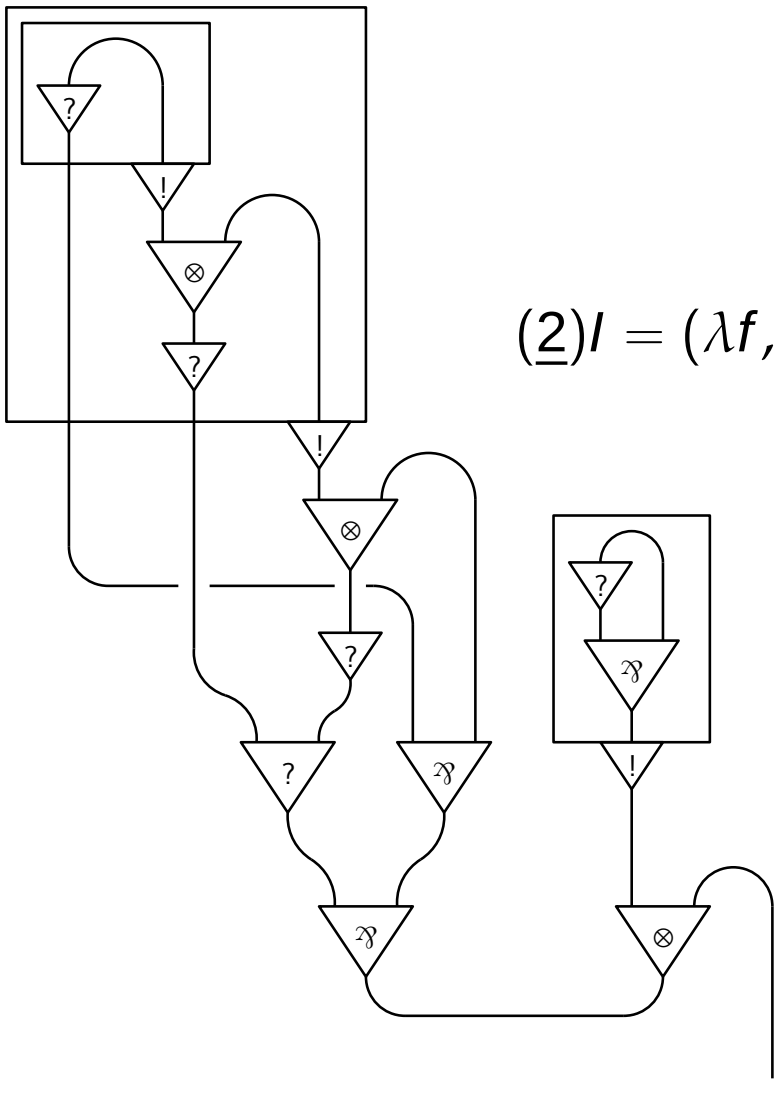
$$\underline{2} = \lambda f, x.(f)(f)x : (\alpha \Rightarrow \alpha) \Rightarrow \alpha \Rightarrow \alpha$$

An example



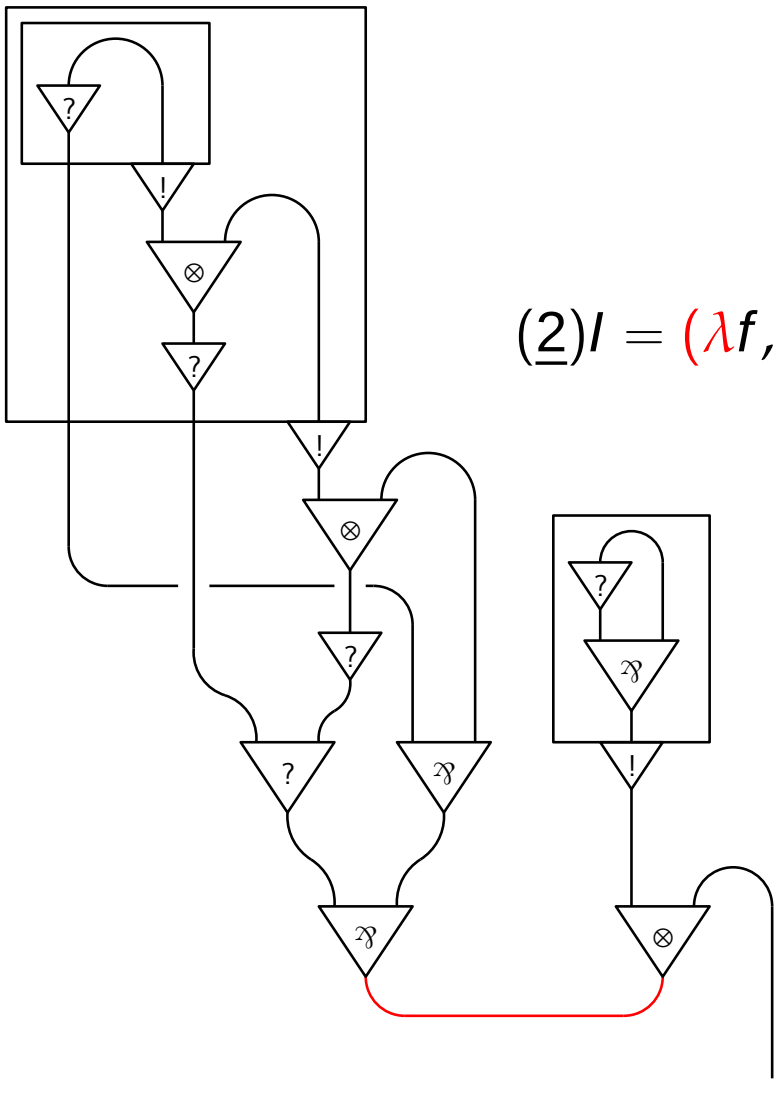
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y : \alpha \Rightarrow \alpha$$

An example



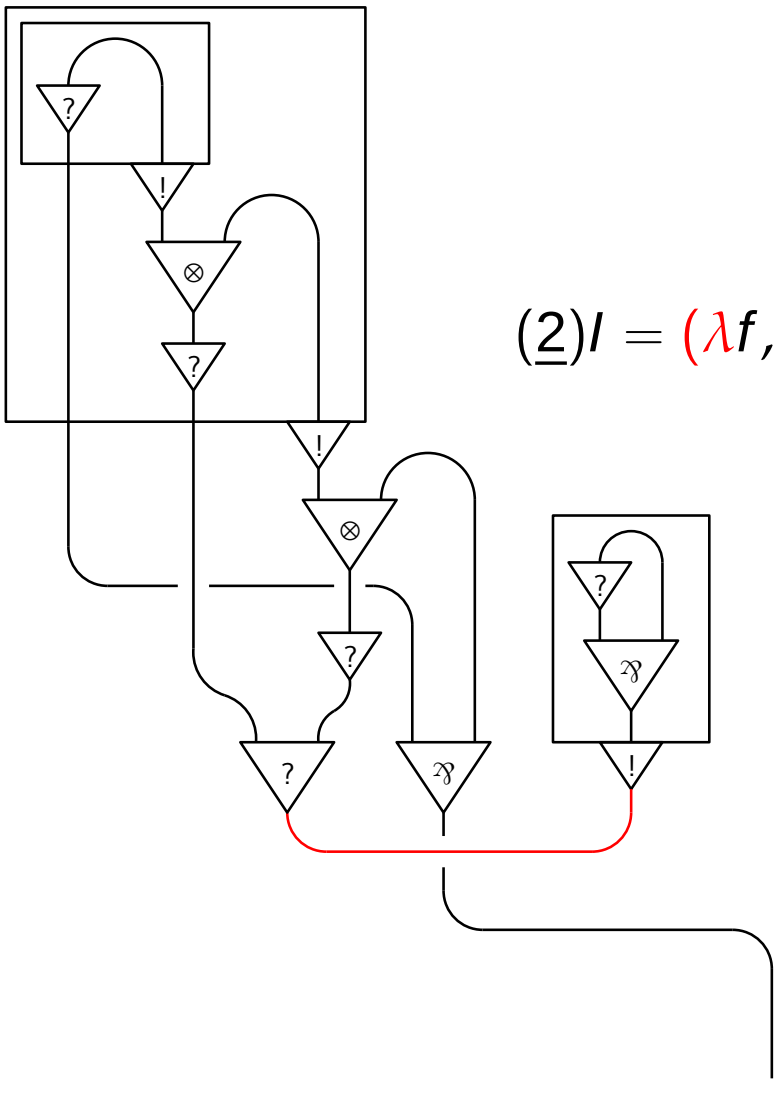
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



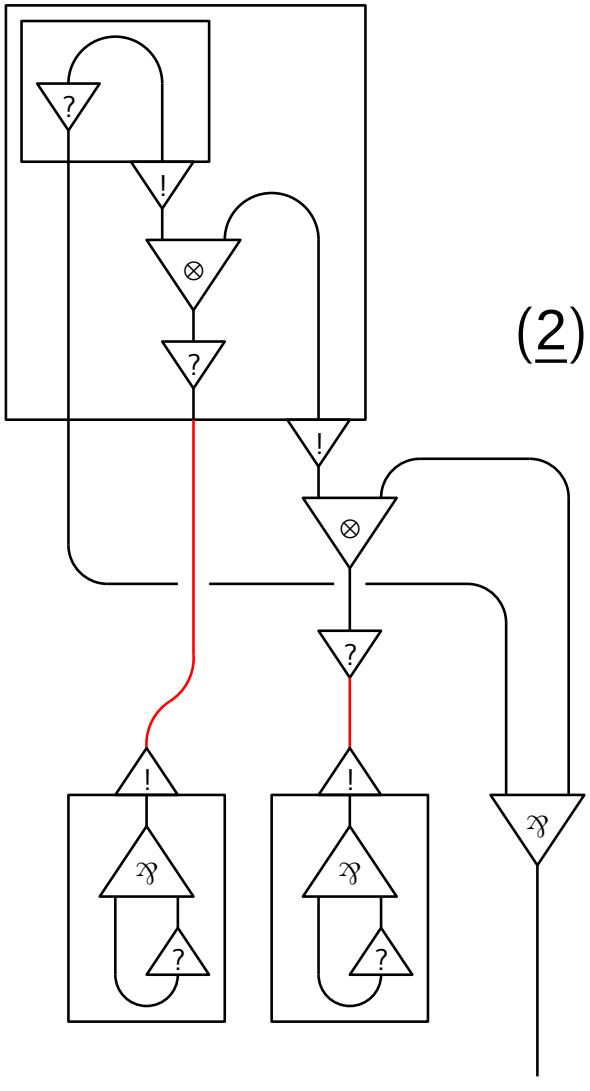
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



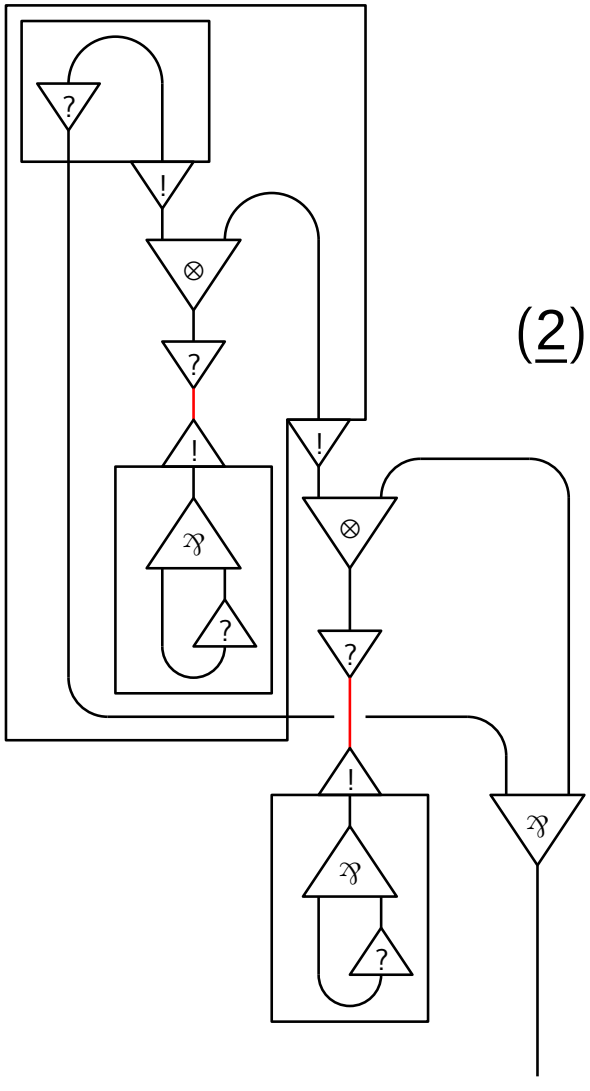
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



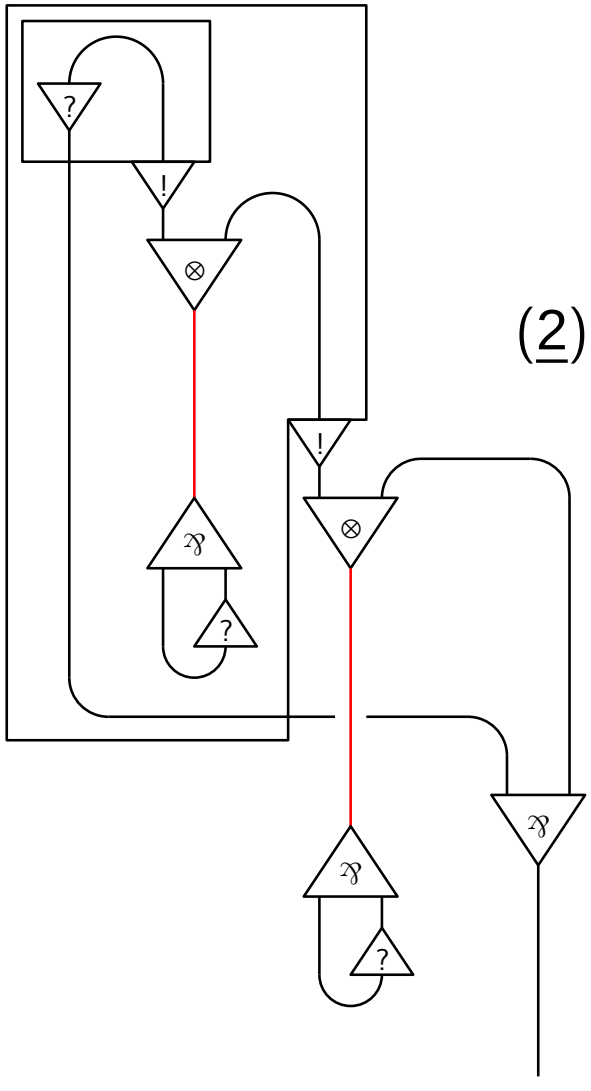
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



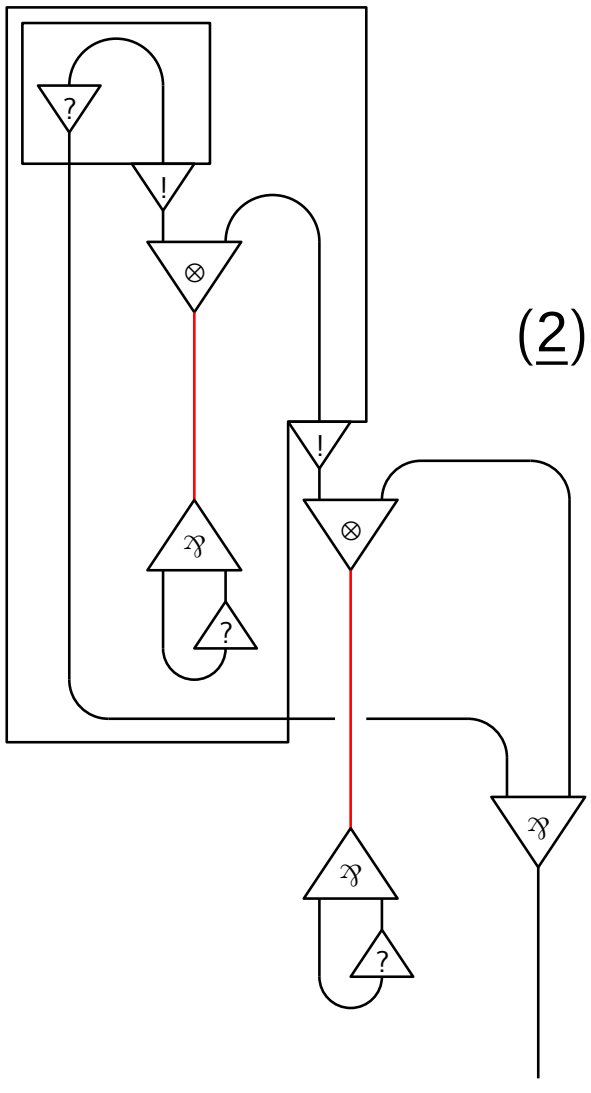
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



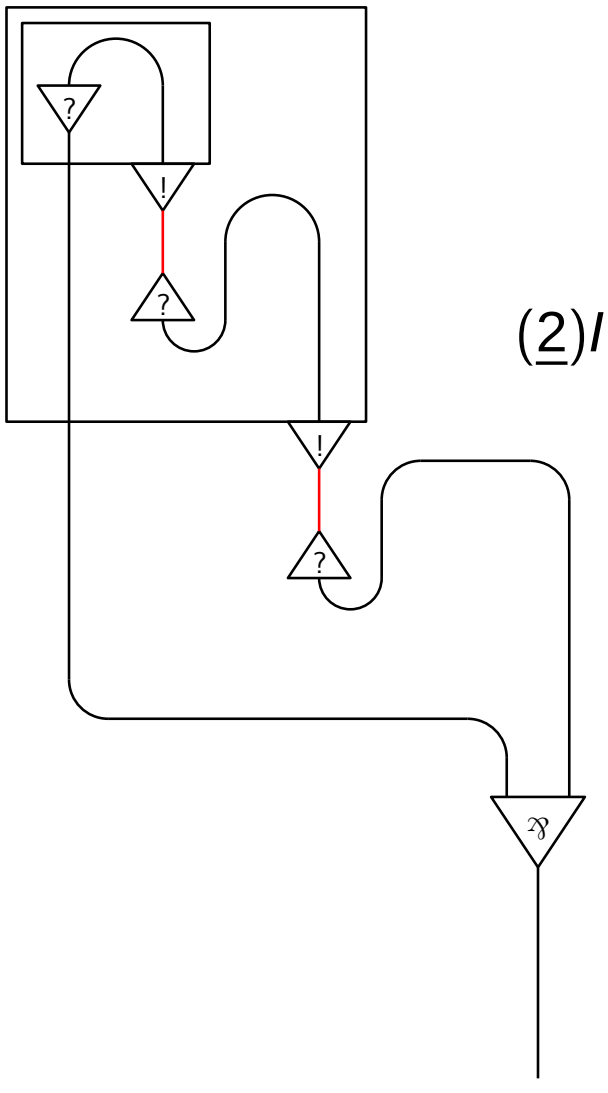
$$(\underline{2})I = (\lambda f, x.(f)(f)x)\lambda y.y$$

An example



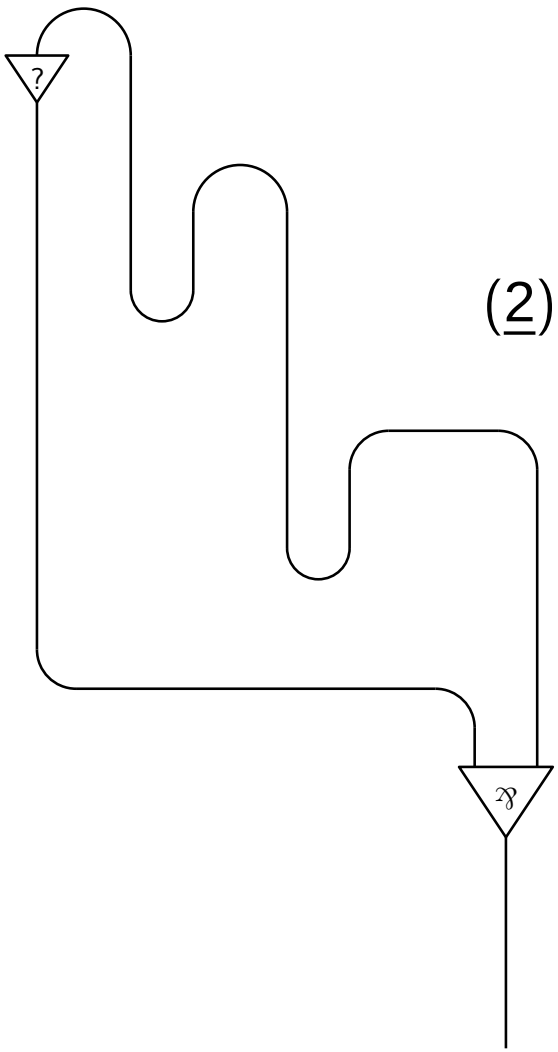
$$(\underline{2})I \xrightarrow{\beta} \lambda x.(\lambda y.y)(\lambda y.y)x$$

An example



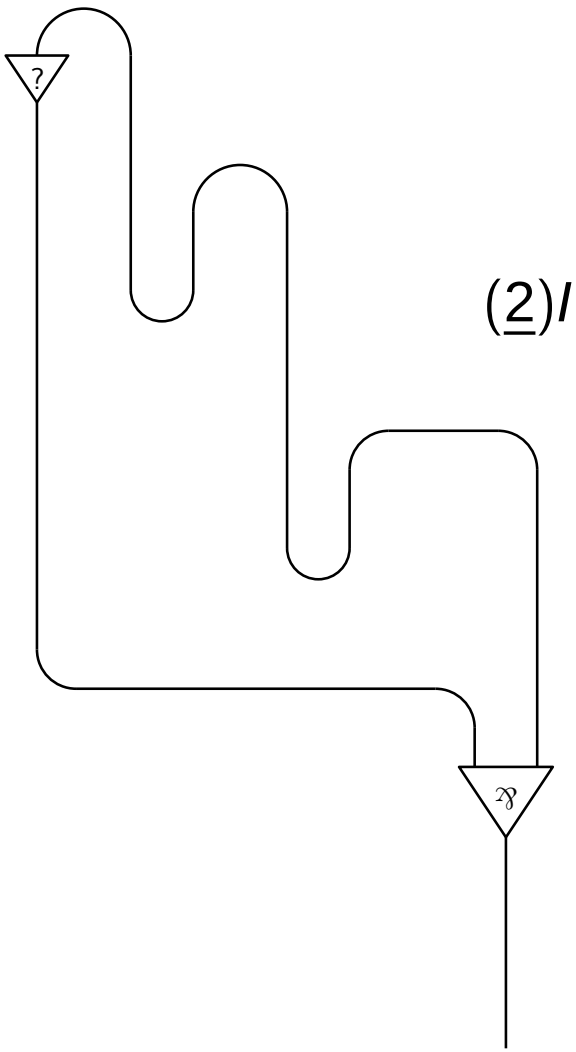
$$(\underline{2})I \xrightarrow{\beta} \lambda x.(\lambda y.y)(\lambda y.y)x$$

An example



$$(\underline{2})I \xrightarrow{\beta} \lambda x.(\lambda y.y)(\lambda y.y)x$$

An example



$$(\underline{2})I \xrightarrow{\beta*} \lambda x.x$$

The results

What we have is a translation M° from terms to proof nets.

Theorem (Girard, Danos-Regnier)

- *each redex* $(\lambda x.S)T$ *in* M *corresponds bijectively to multiplicative cuts in* M°
- $M \xrightarrow{\beta} N \iff M^\circ \xrightarrow{m} \xrightarrow{e*} \!\!\!\!\rightarrow N^\circ$
- $M \xrightarrow{\beta*} N \iff M^\circ \xrightarrow{*} N^\circ$

Inputs are packages

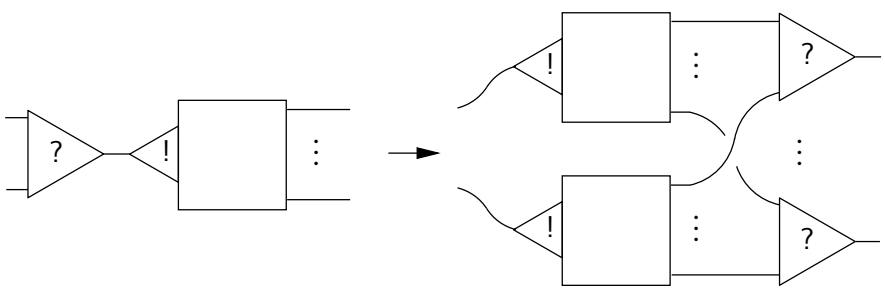
Inputs are packages that are

- duplicable
- erasable
- openable

Inputs are packages

Inputs are packages that are

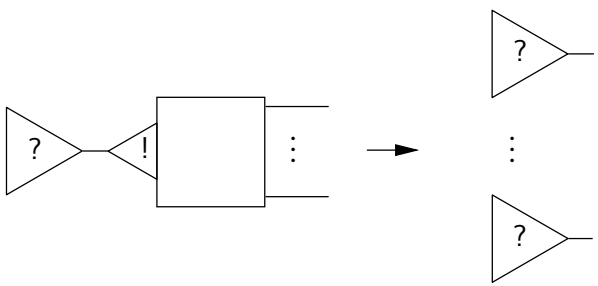
- duplicable
- erasable
- openable



Inputs are packages

Inputs are packages that are

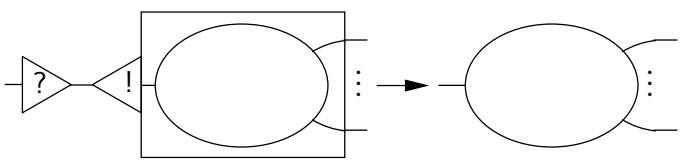
- duplicable
- erasable
- openable



Inputs are packages

Inputs are packages that are

- duplicable
- erasable
- openable



Outline

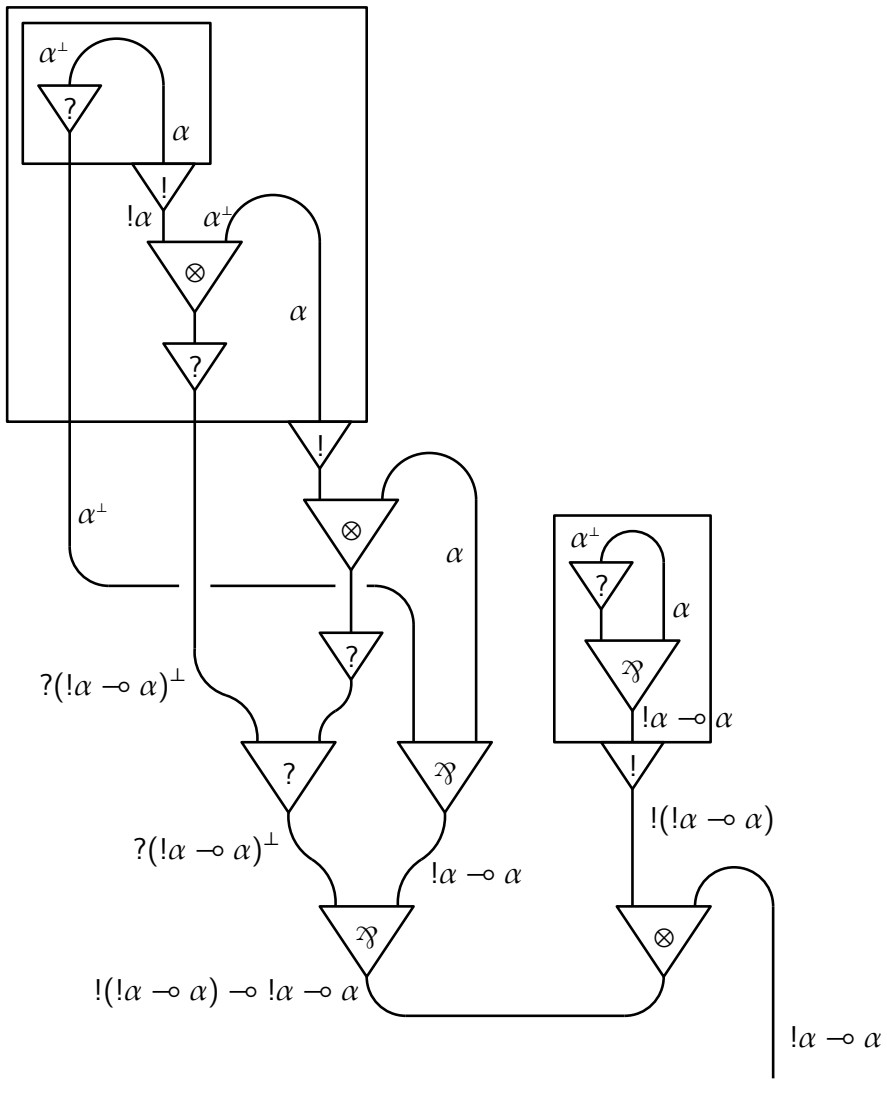
Single-use hypotheses/inputs

- There is a situation not covered by the properties of boxes
- What if a hypothesis/input is available just once (after which it “expires”), but is asked for more than once?
- We can imagine that it can be assigned non-deterministically (formal sums involved)
- We are able to see this in differential nets, the geometrical representation of Differential Linear Logic (Ehrhard and Regnier, 2004)

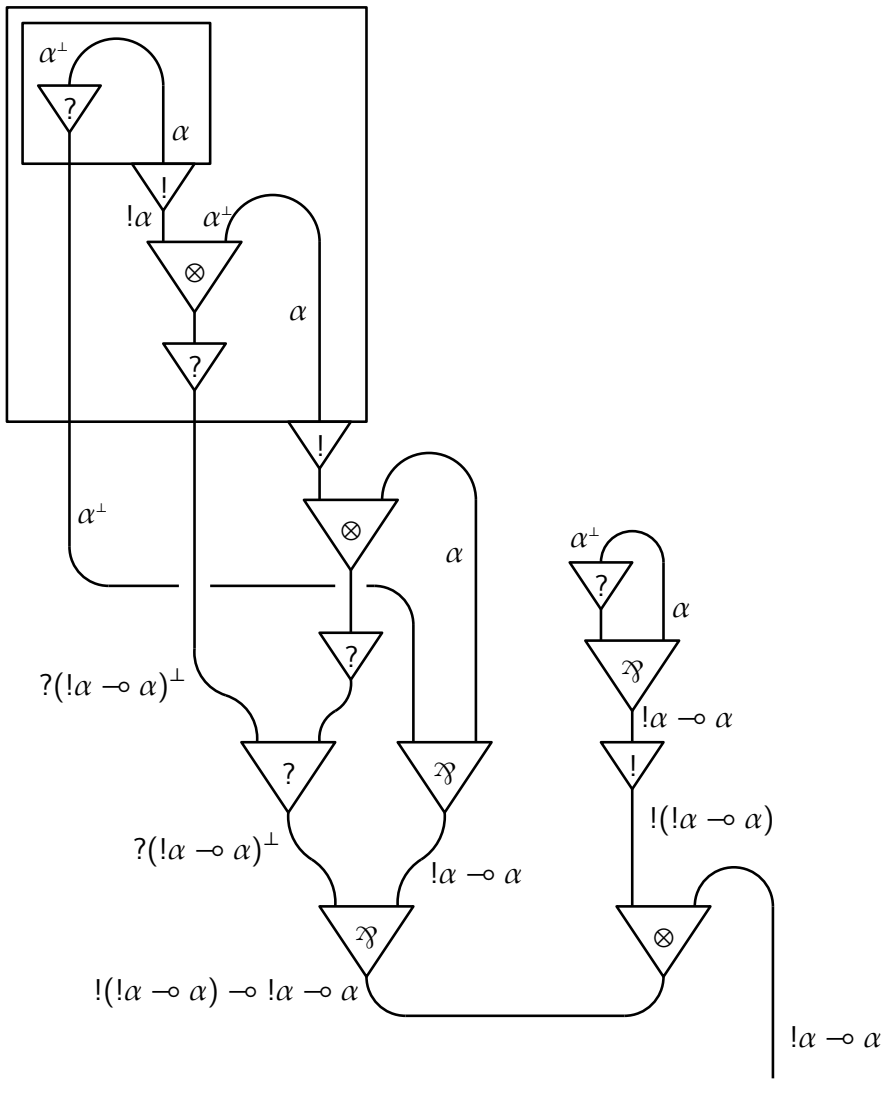
Single-use hypotheses/inputs

- There is a situation not covered by the properties of boxes
- What if a hypothesis/input is available just once (after which it “expires”), but is asked for more than once?
- We can imagine that it can be assigned non-deterministically (formal sums involved)
- We are able to see this in differential nets, the geometrical representation of Differential Linear Logic (Ehrhard and Regnier, 2004)

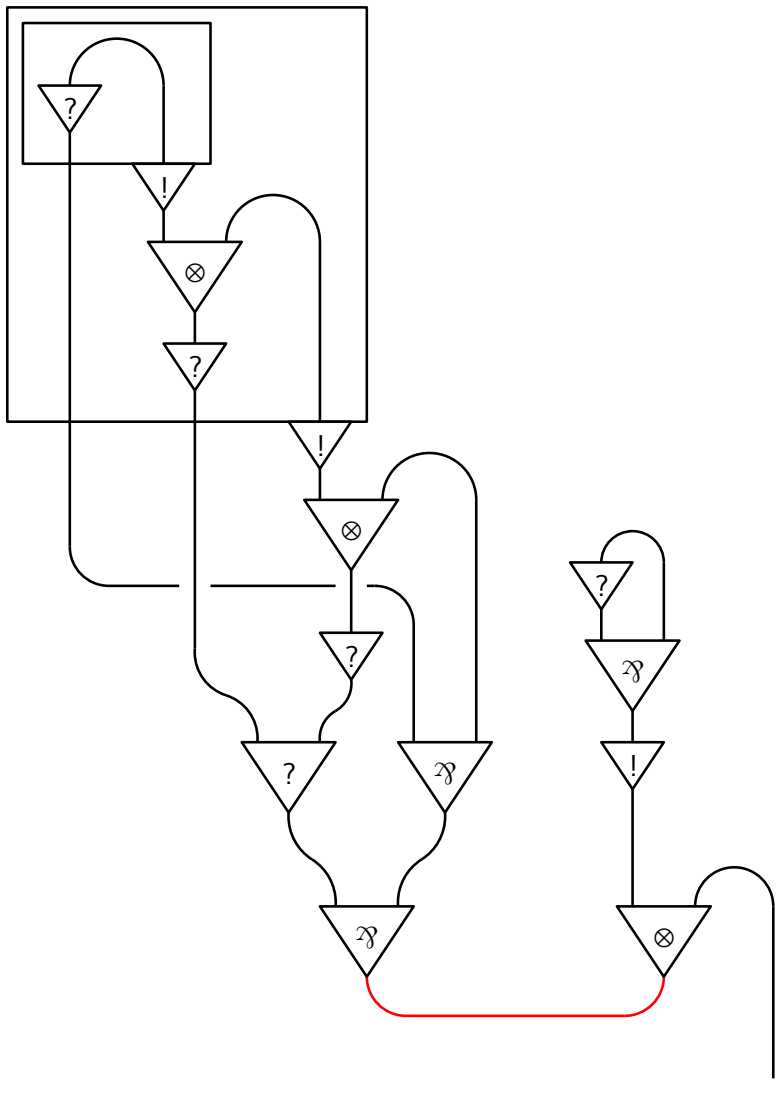
Back to an example



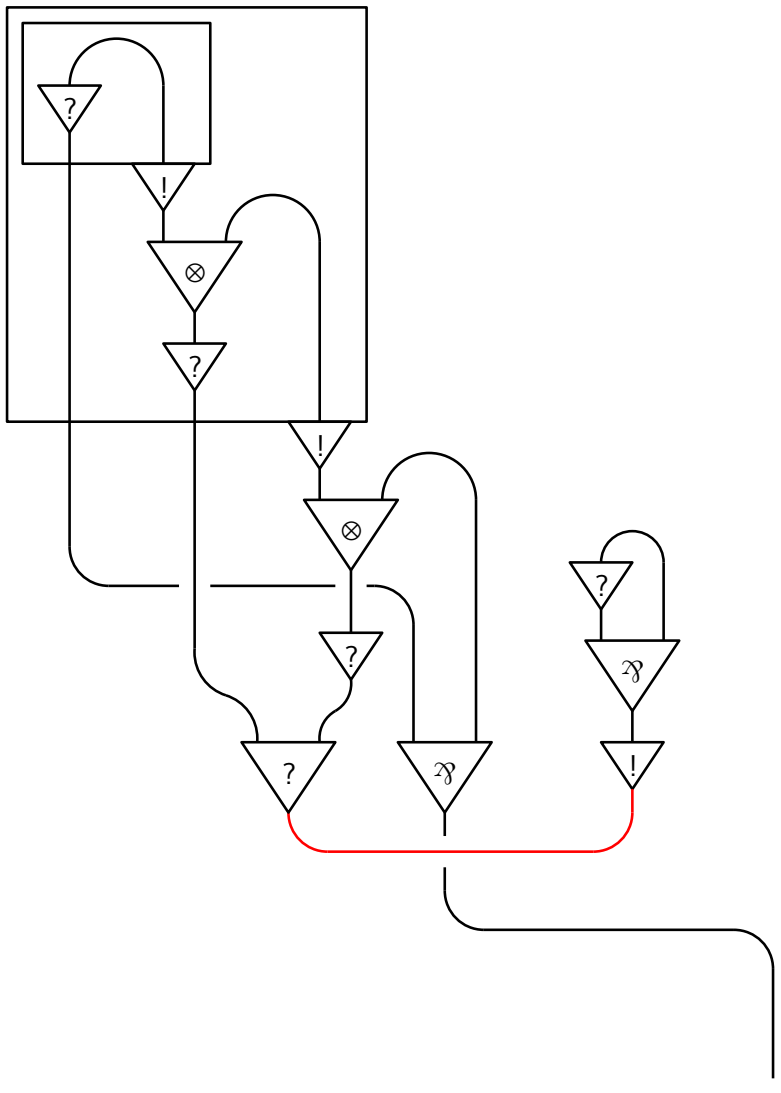
Back to an example



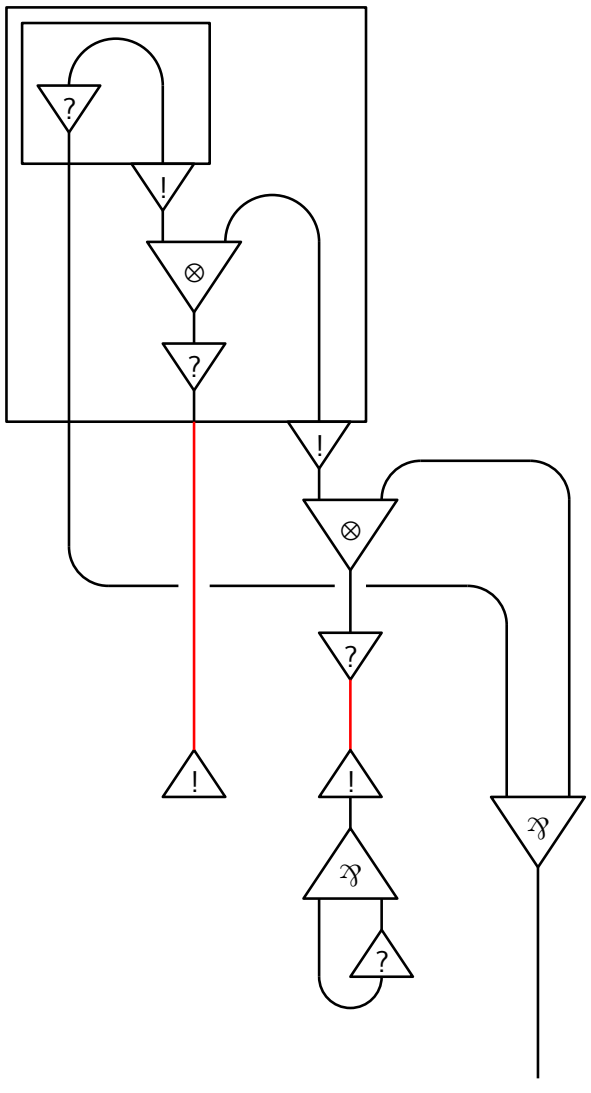
Back to an example



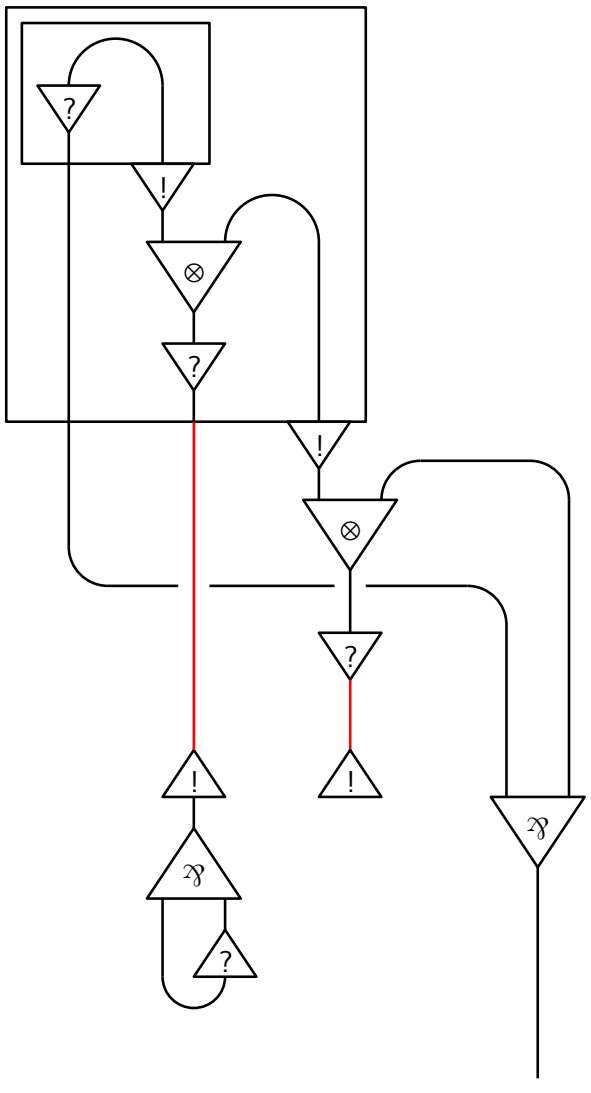
Back to an example



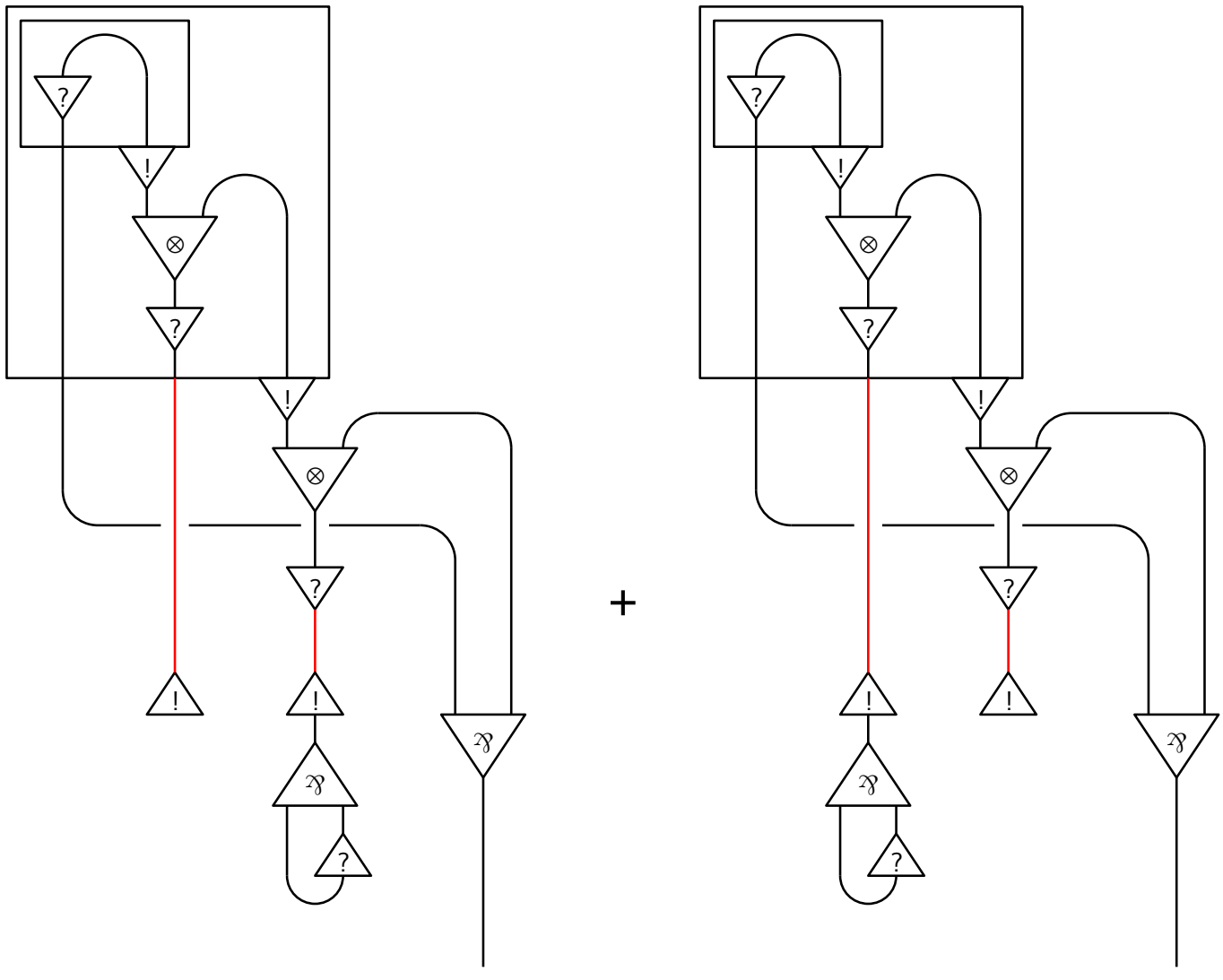
Back to an example



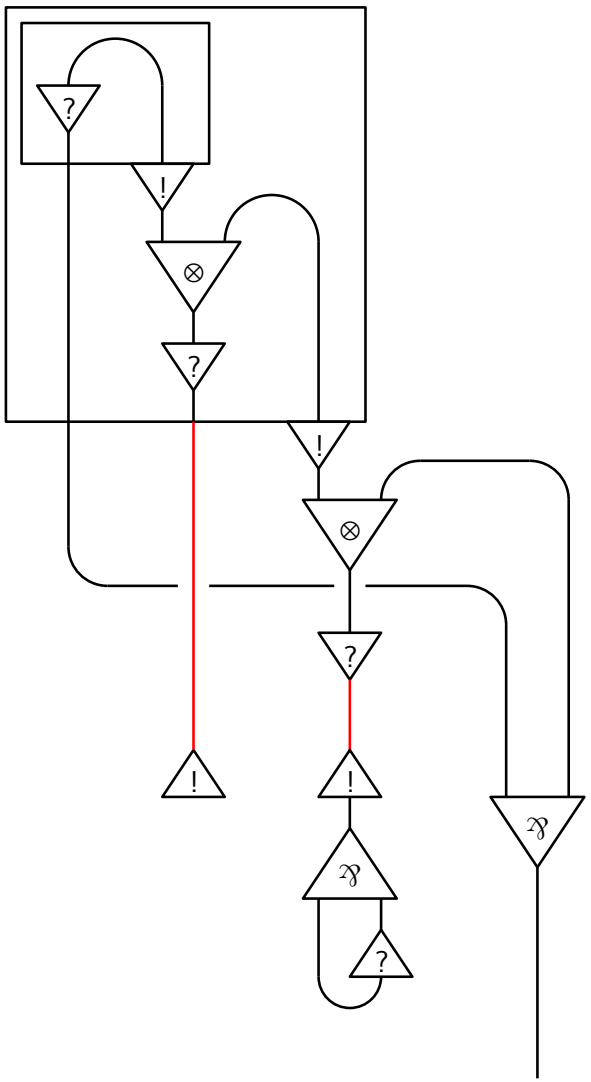
Back to an example



Back to an example

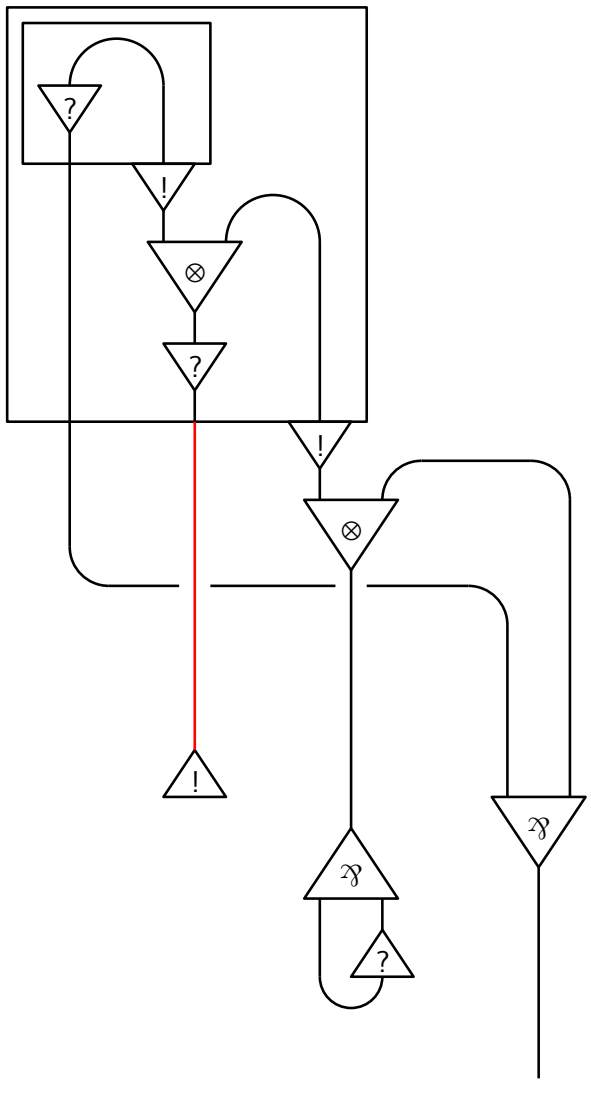


Back to an example

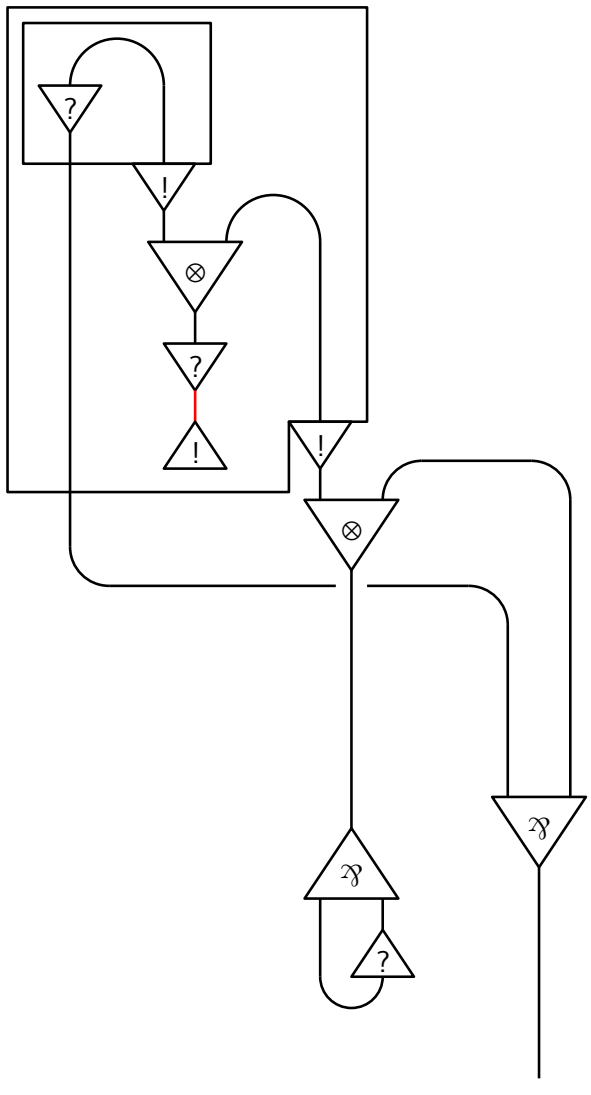


+ 0

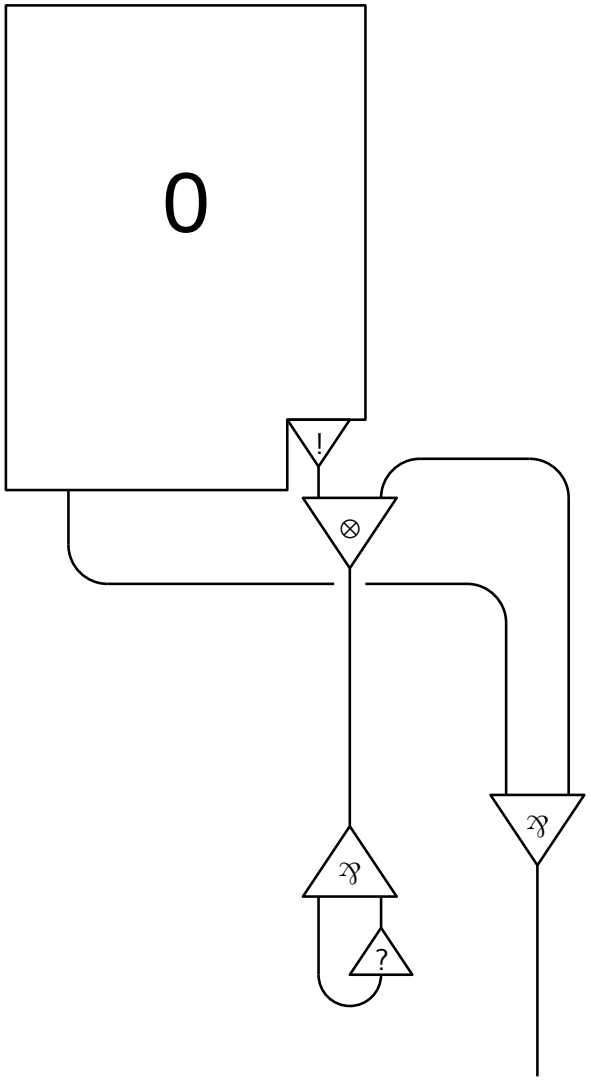
Back to an example



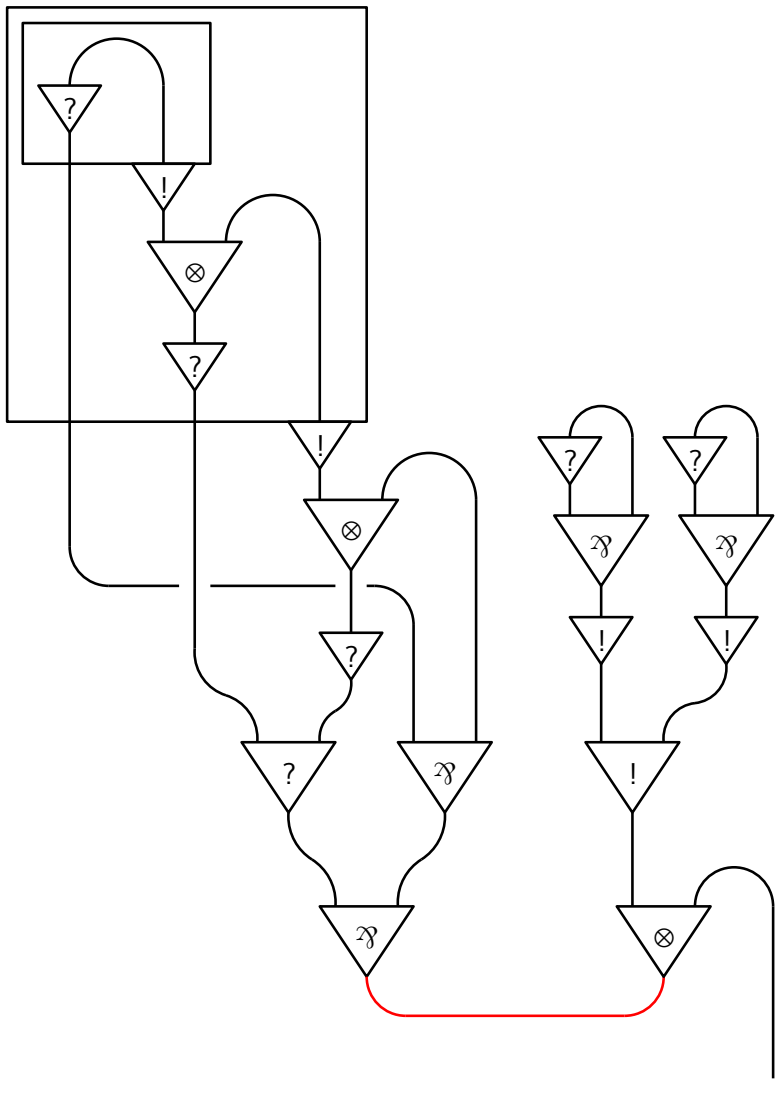
Back to an example



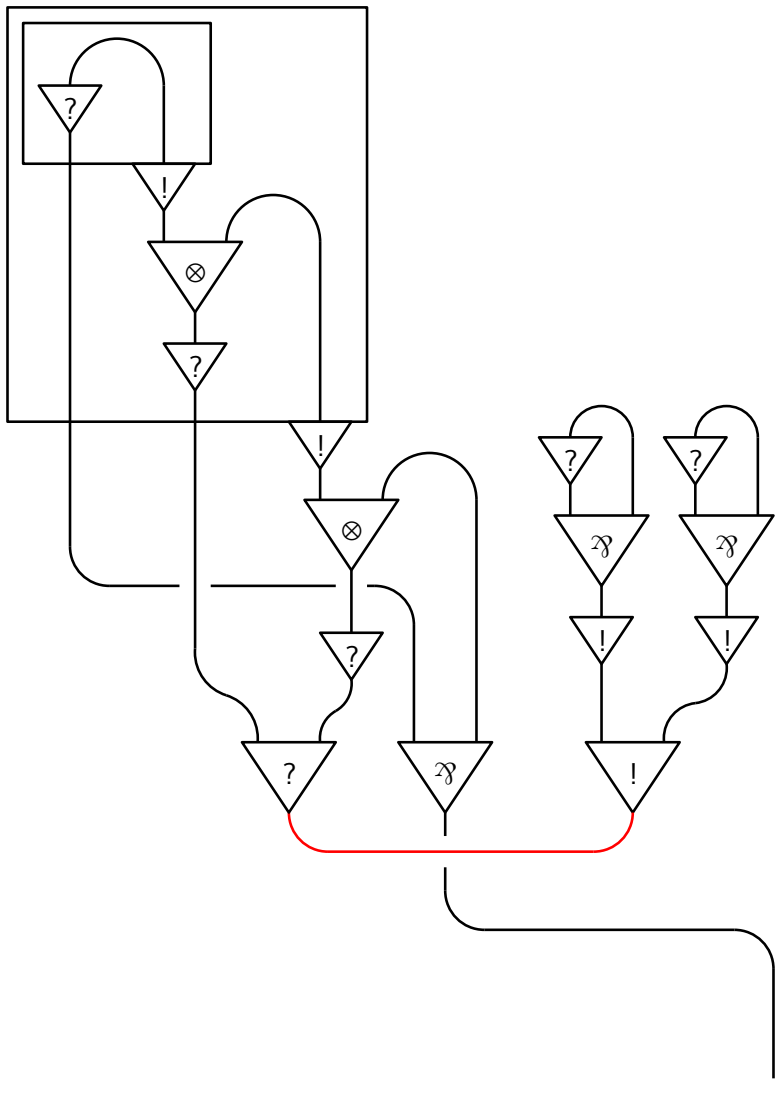
Back to an example



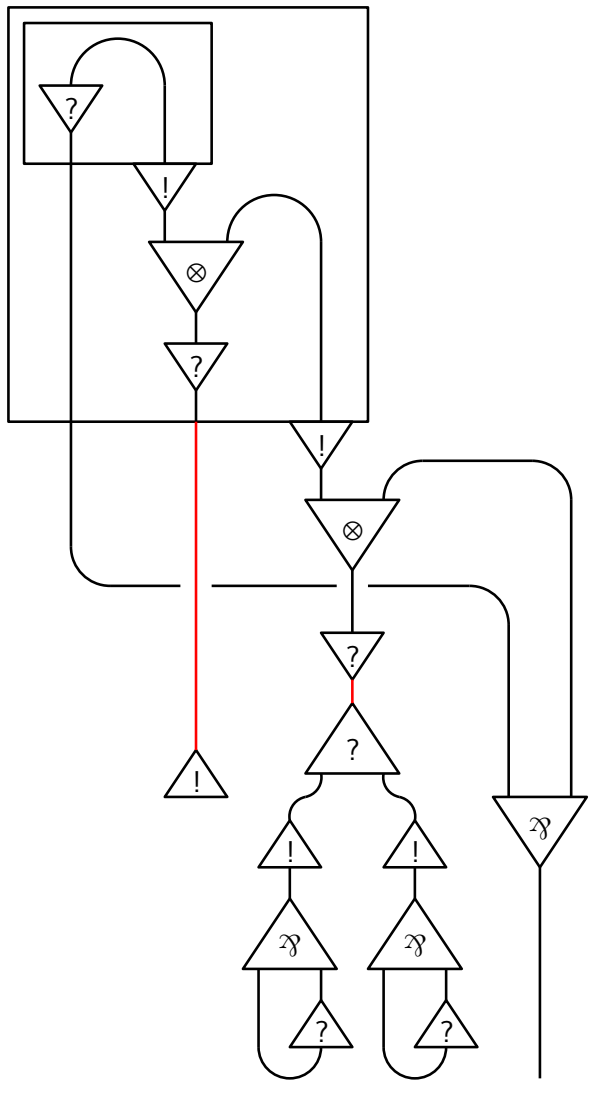
Yet another example



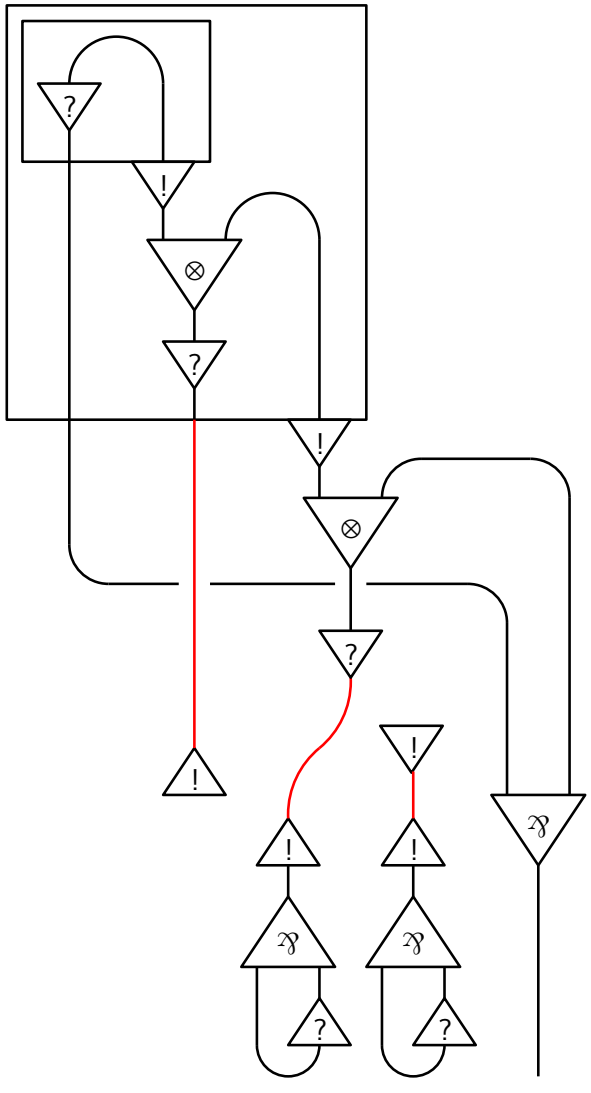
Yet another example



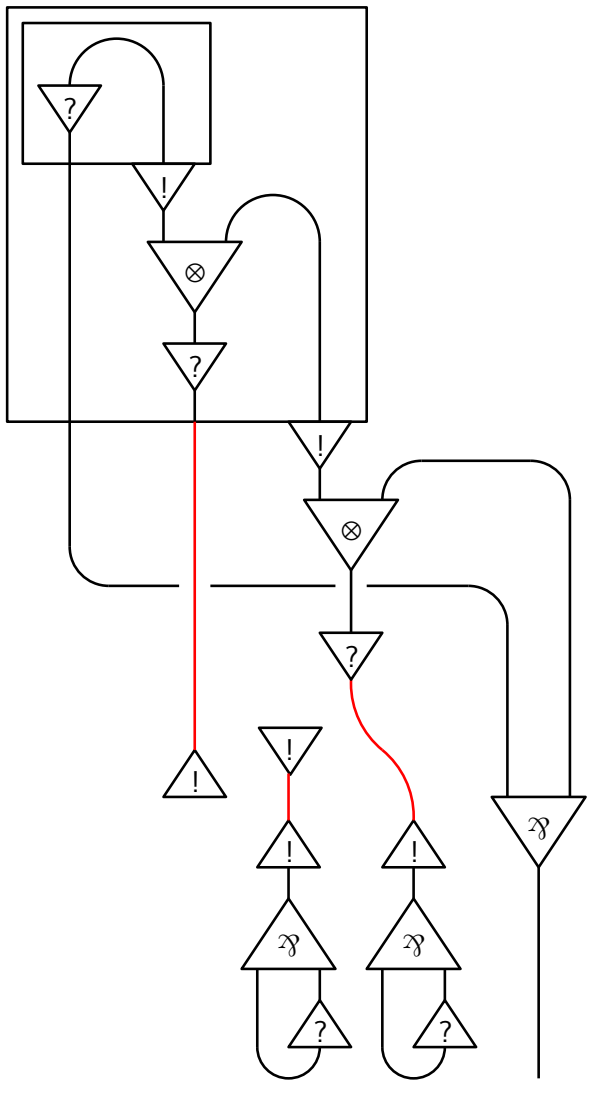
Yet another example



Yet another example



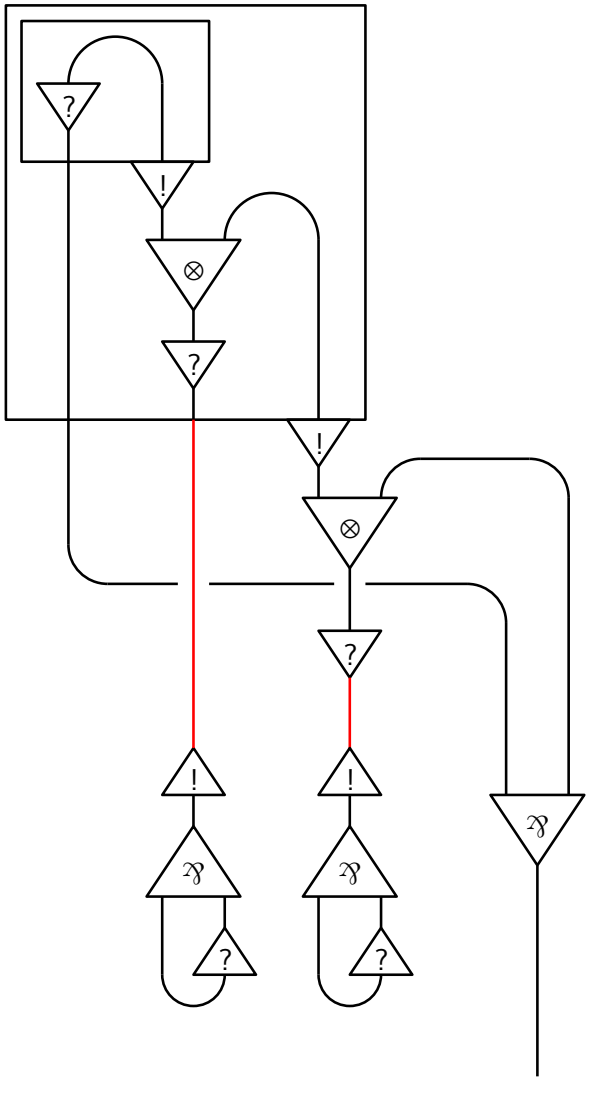
Yet another example



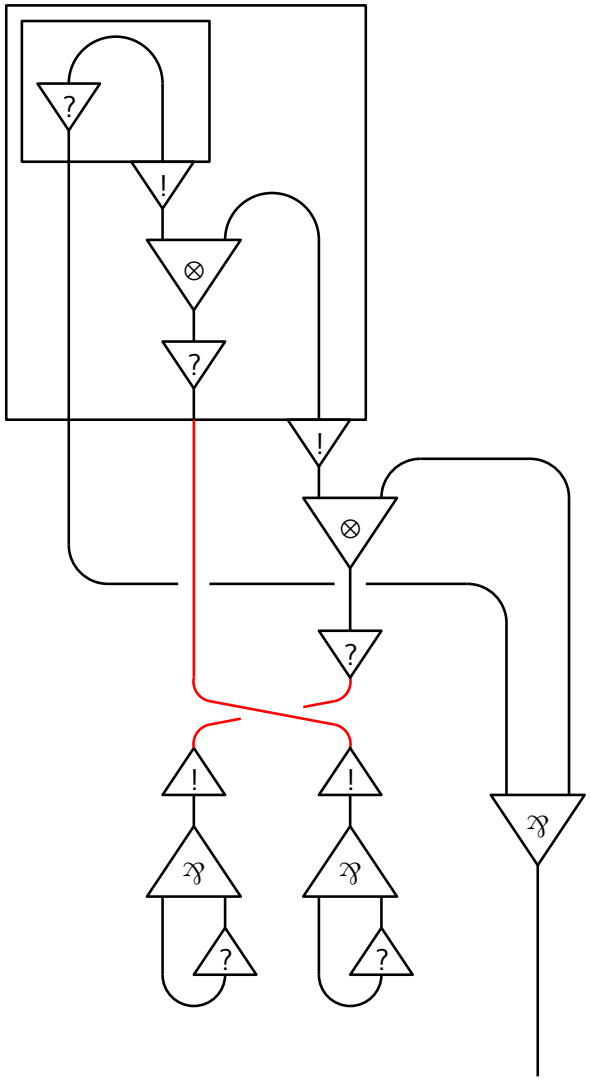
Yet another example

0

Yet another example

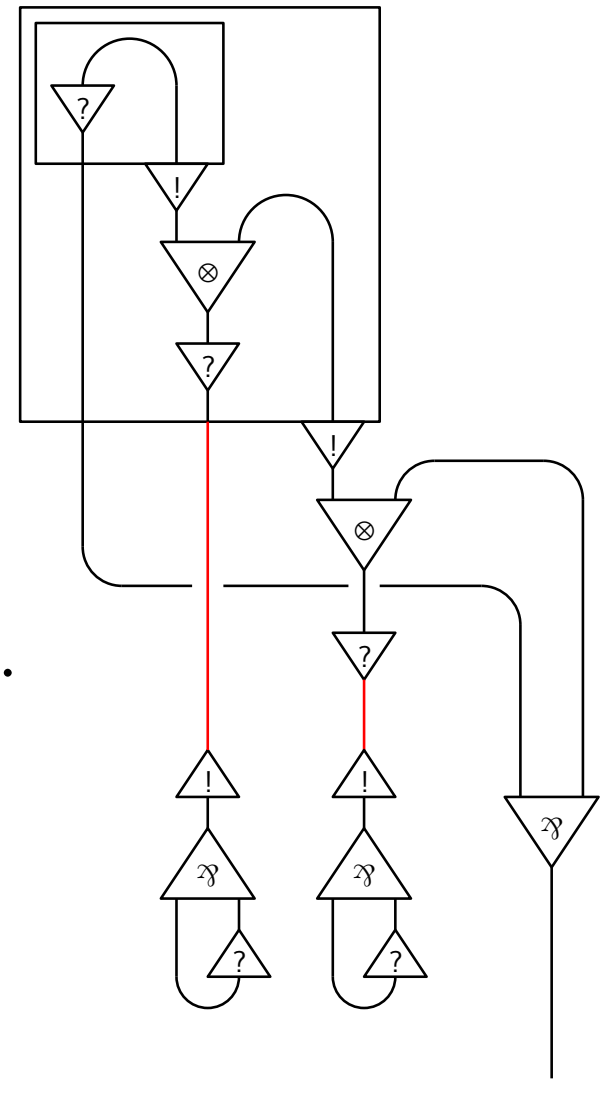


Yet another example



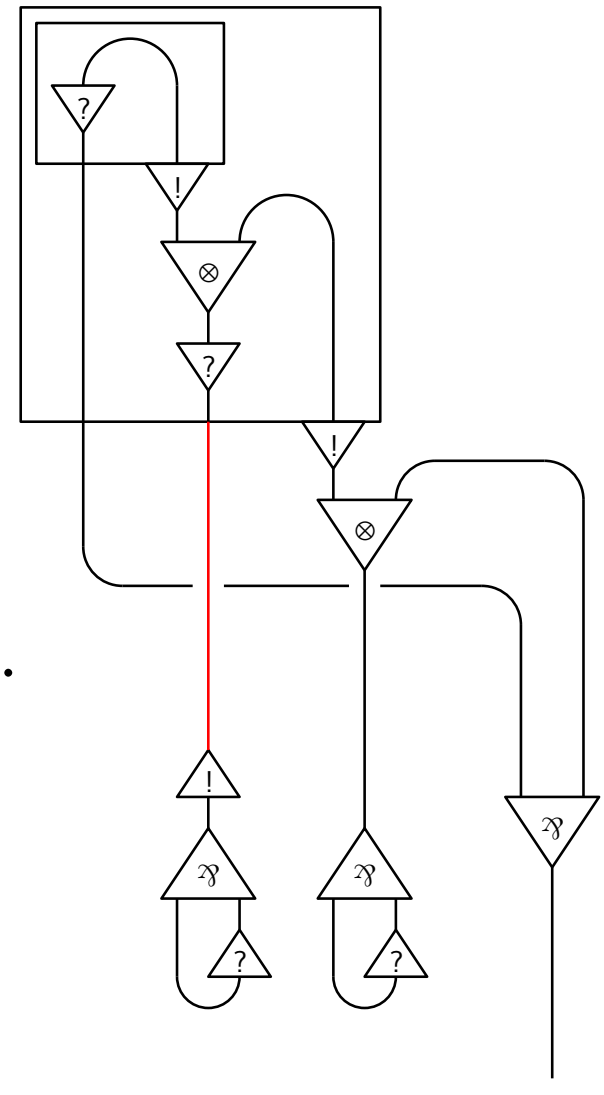
Yet another example

2·



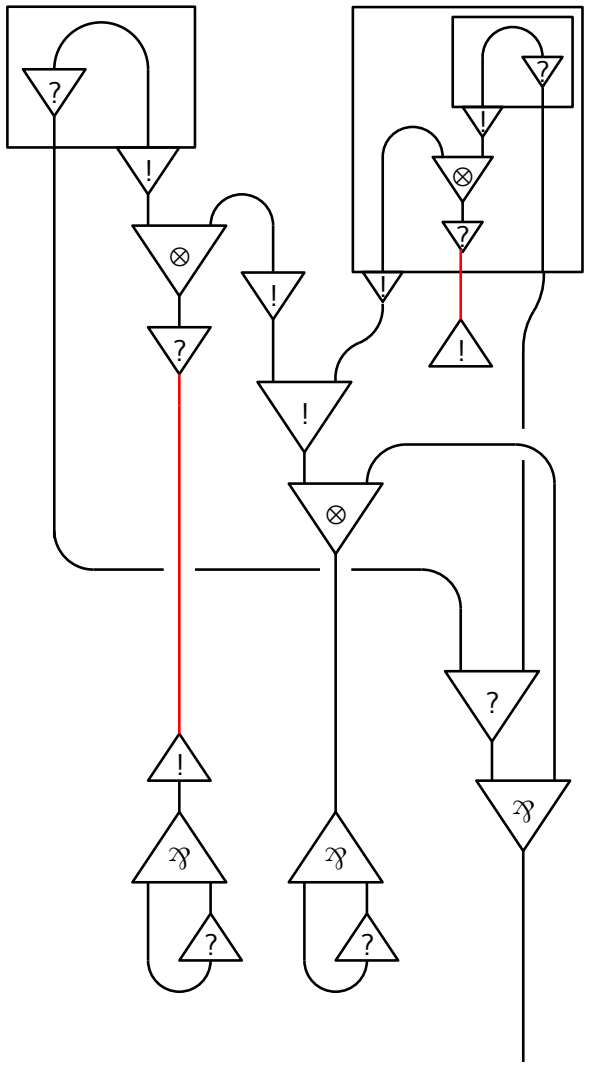
Yet another example

2·



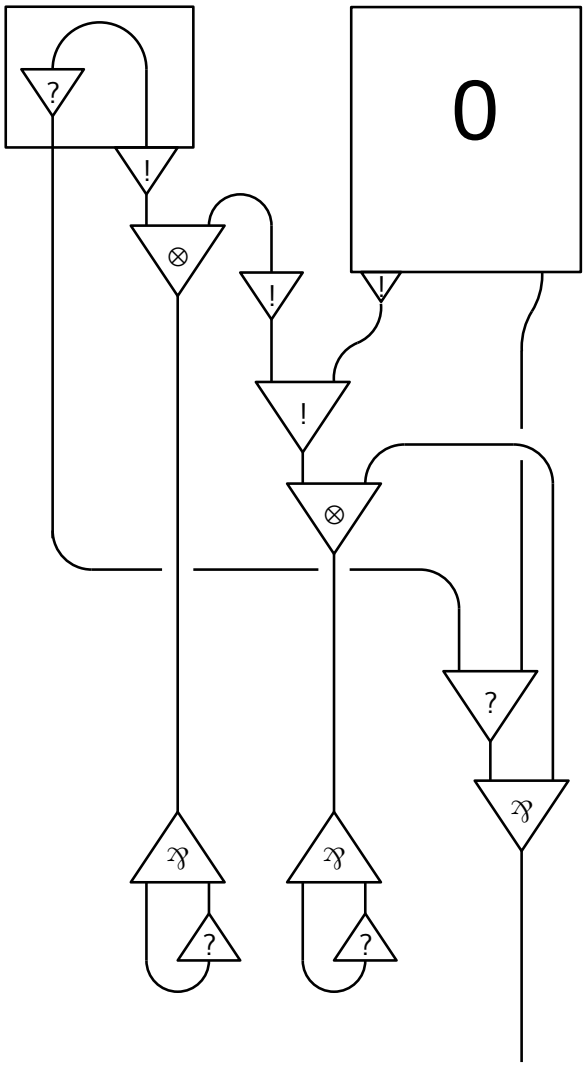
Yet another example

2·



Yet another example

2·



Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?

Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= [M_1^{p_1}, \ldots, M_n^{p_n}] \text{ with } p_i \in \mathbb{N}$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

Resource Calculus

Does this behaviour correspond to a calculus like Proof Nets do for λ-calculus?
Yes, λ-calculus with multiplicities (Boudol, 1993), exactly presents this behaviour.

Terms of the resource λ-calculus

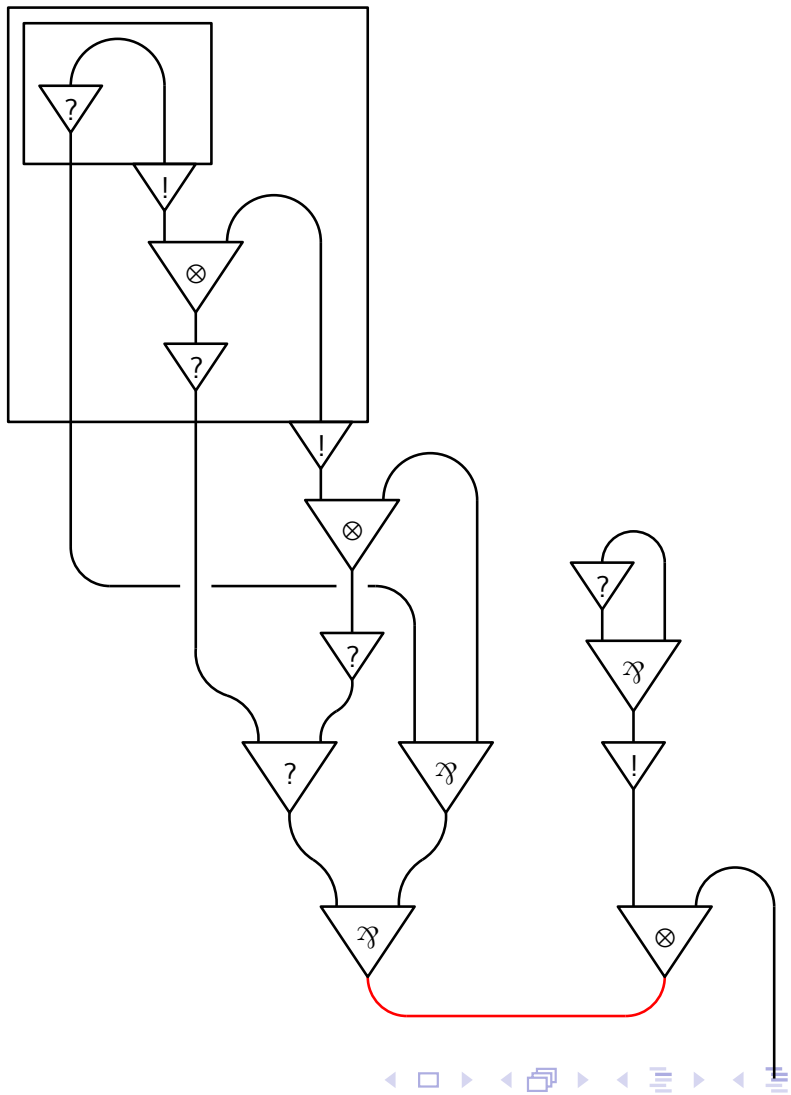
$$M ::= x \mid \lambda x.M \mid \langle M \rangle A$$

$$A ::= M_1^{e_1} \cdots M_k^{e_k} \quad \text{with } e_i \leq \infty$$

Boudol described a non-deterministic reduction. We, as already explained, employ formal sums. The ground for this is *regular substitution* for M^∞ arguments, and *linear substitution* for finitely available arguments

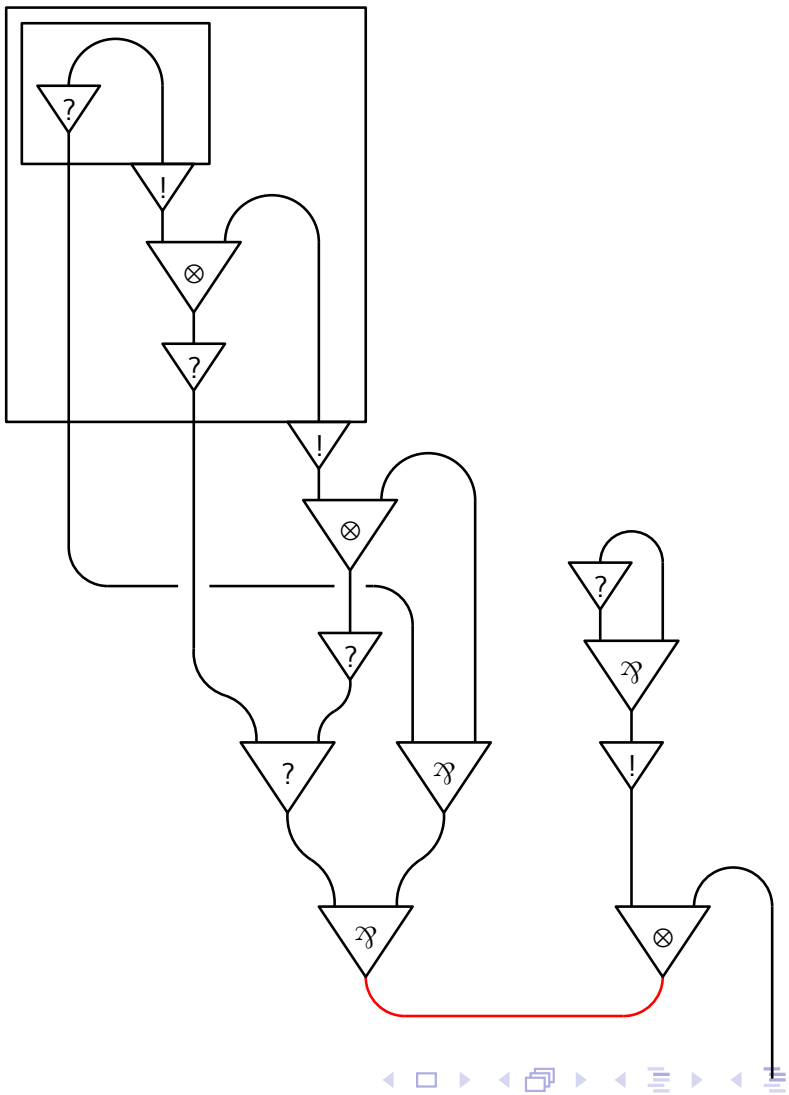
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle \lambda y. y \rightsquigarrow$$



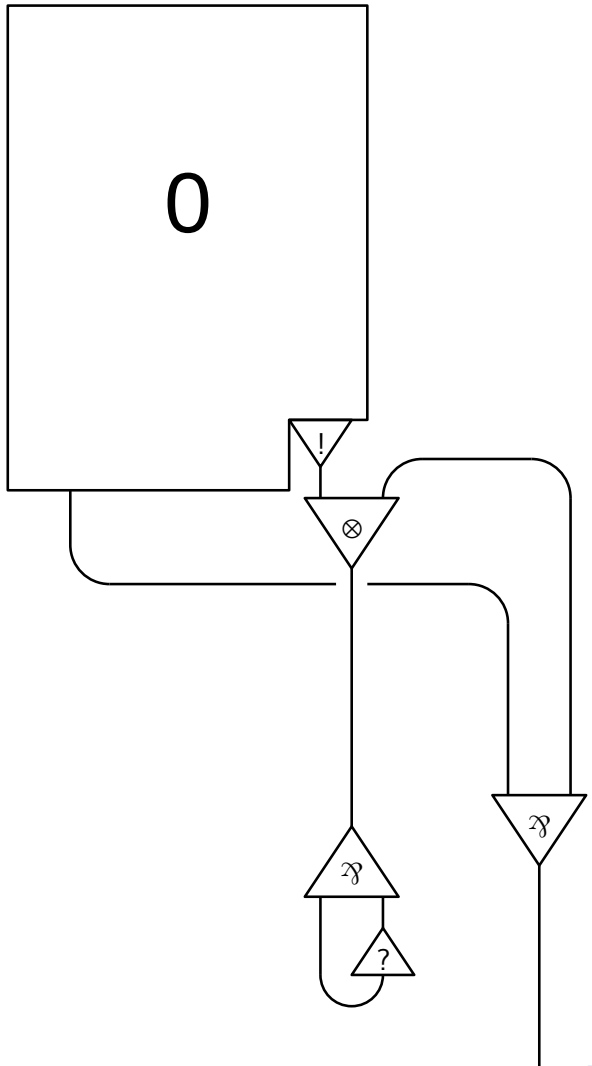
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle \lambda y. y \rightsquigarrow$$



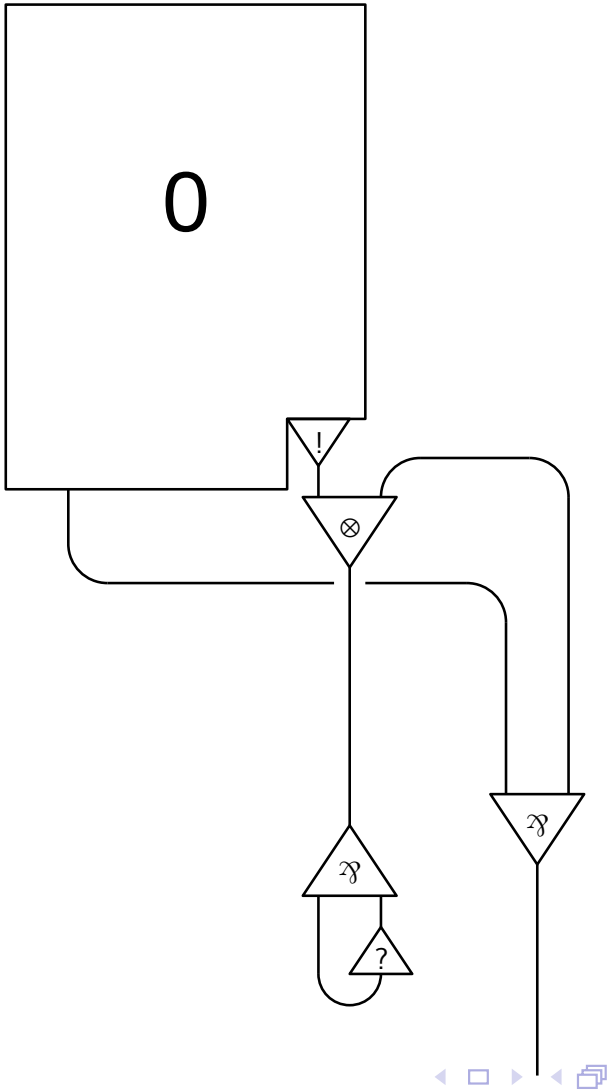
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle \lambda y. y$$
$$\downarrow \beta$$
$$\lambda x. \langle \lambda y. y \rangle 0^{\infty} \quad \rightsquigarrow$$



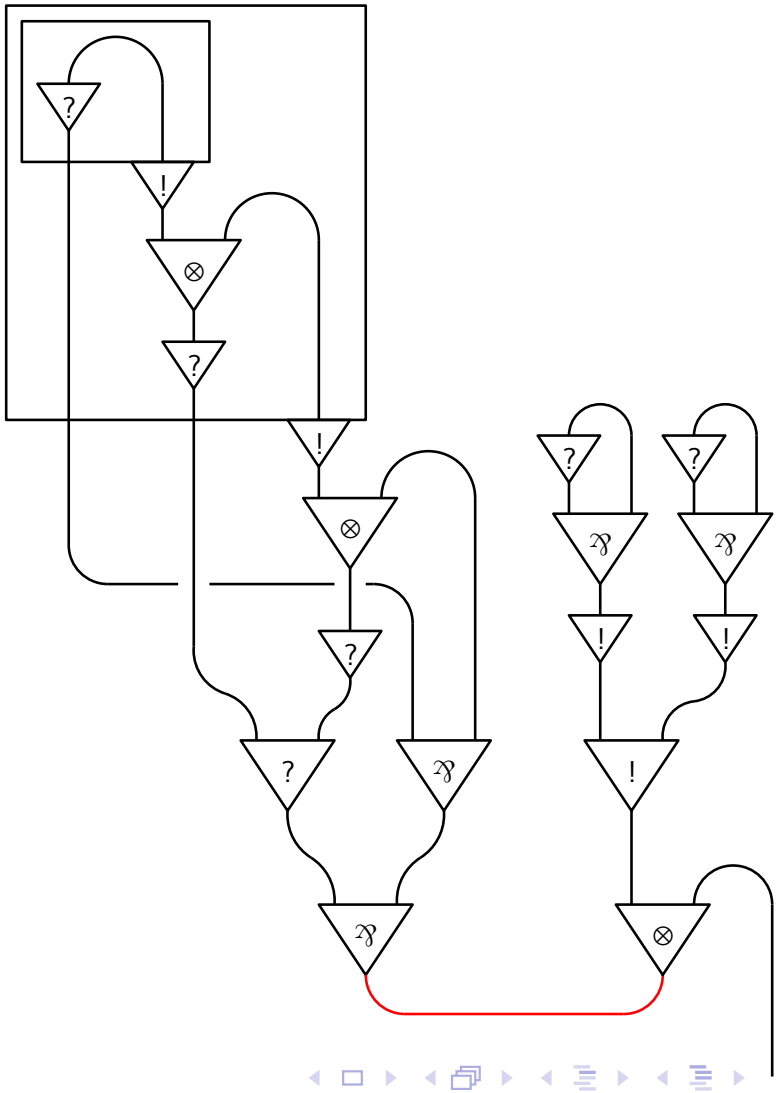
Back to the examples

$$\langle \lambda f, x.\, \langle f \rangle\, (\langle f \rangle\, x^{\infty})^{\infty} \rangle\, \lambda y.y$$
$$\downarrow \beta$$
$$\lambda x.\, \langle \lambda y.y \rangle\, 0^{\infty} \quad \rightsquigarrow$$



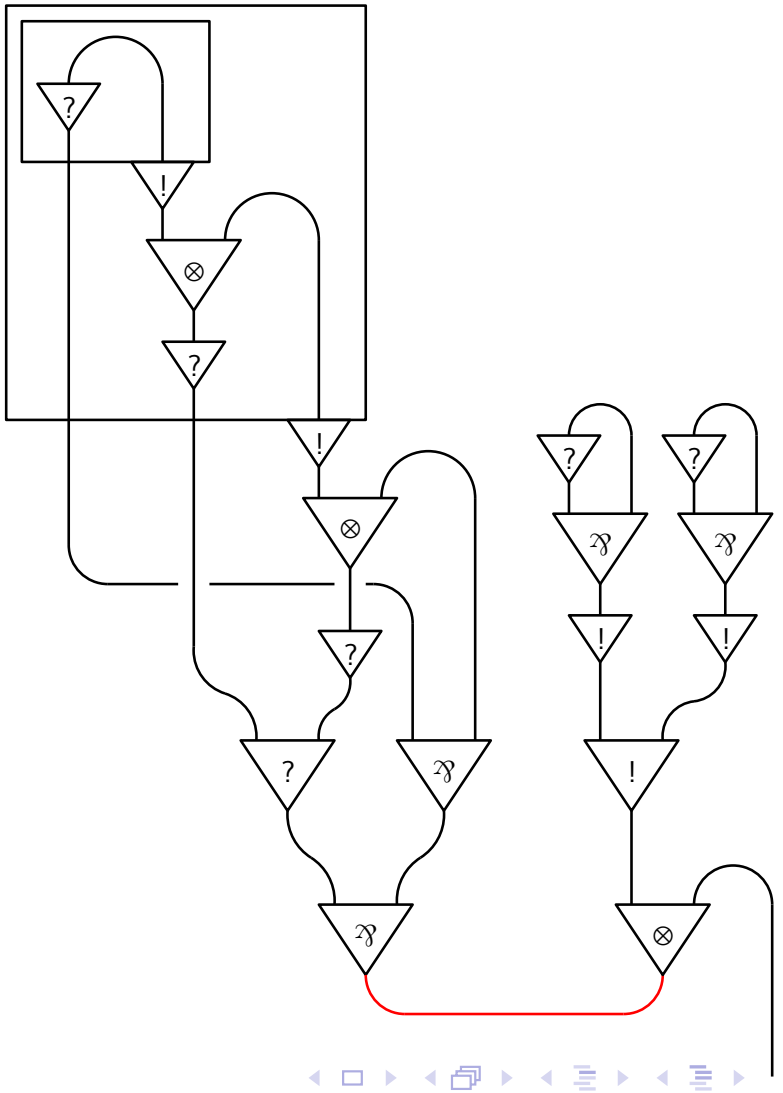
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle (\lambda y. y)^2 \rightsquigarrow$$



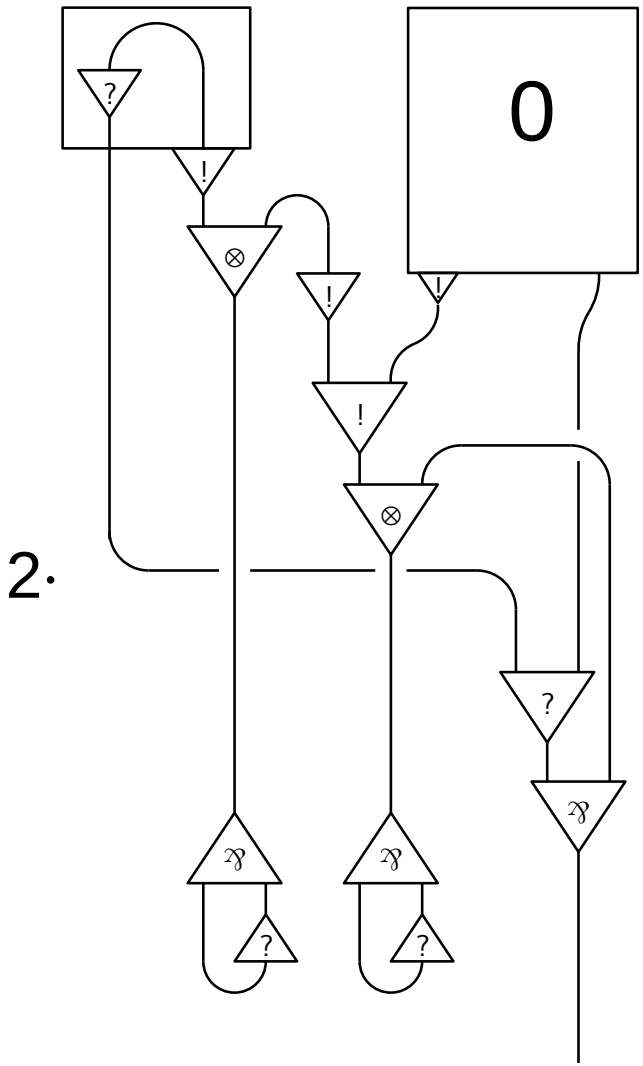
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle (\lambda y. y)^2 \rightsquigarrow$$



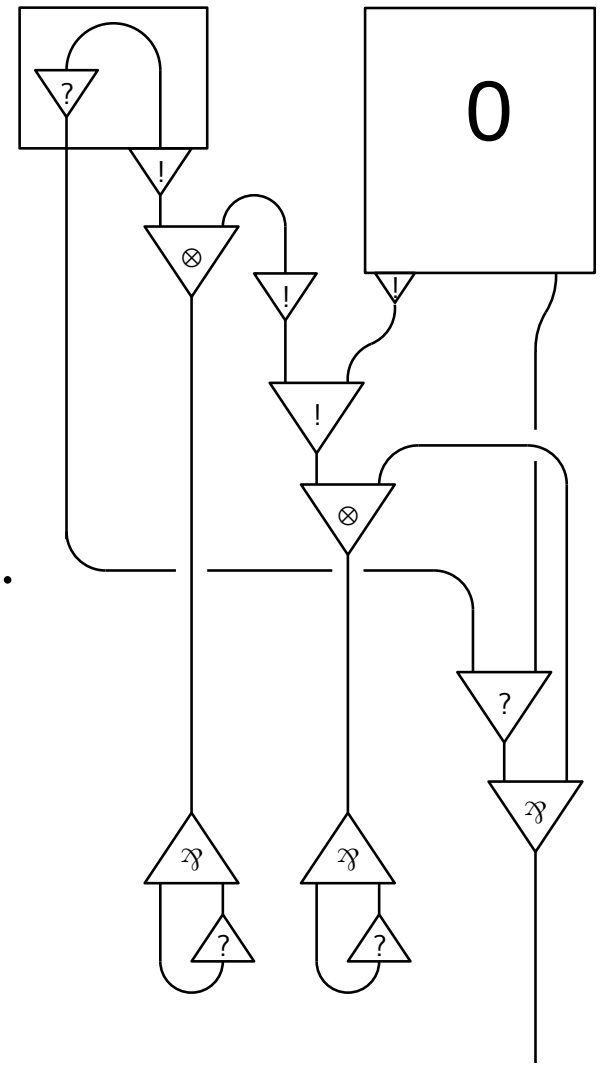
Back to the examples

$$\langle \lambda f, x. \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle (\lambda y.y)^2$$
$$\downarrow \beta$$
$$2 \cdot \lambda x. \langle \lambda y.y \rangle 0^{\infty} \cdot \langle \lambda y.y \rangle x^{\infty} \rightsquigarrow$$



Back to the examples

$$\langle \lambda f, x.\, \langle f \rangle (\langle f \rangle x^{\infty})^{\infty} \rangle (\lambda y.y)^2$$
$$\downarrow \beta$$
$$2 \cdot \lambda x.\, \langle \lambda y.y \rangle 0^{\infty} \cdot \langle \lambda y.y \rangle x^{\infty} \rightsquigarrow 2 \cdot$$



The results

What we have is a translation M° from terms to proof nets.

Theorem (Girard, Danos-Regnier)

- *each redex* $(\lambda x.S)T$ *in* M *corresponds bijectively to multiplicative cuts in* M°
- $M \xrightarrow{\beta} N \iff M^\circ \xrightarrow{m} \xrightarrow{e*} \twoheadrightarrow N^\circ$
- $M \xrightarrow{\beta*} N \iff M^\circ \xrightarrow{*} N^\circ$

The results

What we have is a translation M° from resource terms to differential nets.

Theorem

- *each redex $\langle \lambda x.S \rangle A$ in M corresponds bijectively to multiplicative cuts in M°*
- $M \xrightarrow{\beta} N \iff M^\circ \xrightarrow{m} \xrightarrow{e*} N^\circ$
- $M \xrightarrow{\beta*} N \iff M^\circ \xrightarrow{*} N^\circ$

Concluding remarks

- Linear substitution is differentiation:

$$\frac{\partial x^2 y}{\partial x} \cdot a = \frac{\partial x \cdot x \cdot y}{\partial x} \cdot a = a \cdot x \cdot y + x \cdot a \cdot y = 2xy \cdot a$$

  idea at the basis of Differential λ-Calculus (Ehrhard and Regnier, 2003)
- In fact analytical functions are smooth: Differential Nets are the lifting to syntax of these properties
- Differential Nets are strongly normalizing: ongoing work with Michele Pagani

Concluding remarks

- Linear substitution is differentiation:

$$\frac{\partial x^2 y}{\partial x} \cdot a = \frac{\partial x \cdot x \cdot y}{\partial x} \cdot a = a \cdot x \cdot y + x \cdot a \cdot y = 2xy \cdot a$$

  idea at the basis of Differential λ-Calculus (Ehrhard and Regnier, 2003)
- In fact analytical functions are smooth: Differential Nets are the lifting to syntax of these properties
- Differential Nets are strongly normalizing: ongoing work with Michele Pagani

Concluding remarks

- Linear substitution is differentiation:

$$\frac{\partial x^2 y}{\partial x} \cdot a = \frac{\partial x \cdot x \cdot y}{\partial x} \cdot a = a \cdot x \cdot y + x \cdot a \cdot y = 2xy \cdot a$$

  idea at the basis of Differential λ-Calculus (Ehrhard and Regnier, 2003)
- In fact analytical functions are smooth: Differential Nets are the lifting to syntax of these properties
- Differential Nets are strongly normalizing: ongoing work with Michele Pagani

Concluding remarks

- Linear substitution is differentiation:

$$\frac{\partial x^2 y}{\partial x} \cdot a = \frac{\partial x \cdot x \cdot y}{\partial x} \cdot a = a \cdot x \cdot y + x \cdot a \cdot y = 2xy \cdot a$$

  idea at the basis of Differential λ-Calculus (Ehrhard and Regnier, 2003)
- In fact analytical functions are smooth: Differential Nets are the lifting to syntax of these properties
- Differential Nets are strongly normalizing: ongoing work with Michele Pagani