Proof nets for additive linear logic with units

Willem Heijltjes

LFCS
School of Informatics
University of Edinburgh

LICS, Toronto, 21–24 June 2011

Motivation: proof nets

For a given logic,

- Syntax: proofs, terms
- Semantics: games, sets and relations (complete partial orders, coherence spaces, Kripke frames), categories

But: many proofs may correspond to the same semantic entity
The aim of proof nets is to obtain a 1-1 correspondence between syntax and semantics

Motivation: additive linear logic

- "Simple" fragment of linear logic, but units are hard
  (Girard)
- Categorical semantics: free products and coproducts
  (Joyal)
- Game-semantics: two communicating games of binary choice
  (Cockett, Seely)
- Process semantics: "the logic of message passing"
  (Cockett)

Additive linear logic

Additive linear logic

$$X \;:=\; A \;\mid\; \mathbf{0} \;\mid\; \top \;\mid\; X \oplus X \;\mid\; X \,\&\, X$$

Proofs of $X \vdash Y$ (or $X \multimap Y$, or $X^{\perp} \parr Y$)

Additive linear logic

Additive linear logic

$$X \;:=\; A \;\mid\; \mathbf{0} \;\mid\; \top \;\mid\; X \oplus X \;\mid\; X \,\&\, X$$

Proofs of $X \vdash Y$ (or $X \multimap Y$, or $X^{\perp} \parr Y$)

Categorical (free) finite products and coproducts (over $\mathcal{C}$)

$$X \;:=\; A \in \mathrm{ob}(\mathcal{C}) \;\mid\; \mathbf{0} \;\mid\; \mathbf{1} \;\mid\; X + X \;\mid\; X \times X$$

Morphisms $f : X \to Y$

Properties of free (co)products

Zero and one are units

$$\mathbf{0} + X \cong X \qquad\qquad \mathbf{1} \times X \cong X$$

and products and coproducts are perfectly dual

But there is no distributivity

$$\not\models \quad \mathbf{0} \times X \cong \mathbf{0} \qquad\qquad \not\models \quad \mathbf{1} + X \cong \mathbf{1}$$

$$\not\models \quad X \times (Y + Z) \;\cong\; (X \times Y) + (X \times Z)$$

(there may not even be a single arrow from left to right!)

Sum–product logic

$$\frac{a \in \mathcal{C}(A,B)}{A \xrightarrow{a} B} \qquad \frac{}{\mathbf{0} \xrightarrow{?} X} \qquad \frac{}{X \xrightarrow{!} \mathbf{1}}$$

$$\frac{X \xrightarrow{f} Y_i}{X \xrightarrow{\iota_i \circ f} Y_0 + Y_1} \qquad \frac{X_0 \xrightarrow{f} Y \quad X_1 \xrightarrow{g} Y}{X_0 + X_1 \xrightarrow{[f,g]} Y}$$

$$\frac{X \xrightarrow{f} Y_0 \quad X \xrightarrow{g} Y_1}{X \xrightarrow{\langle f,g \rangle} Y_0 \times Y_1} \qquad \frac{X_i \xrightarrow{f} Y}{X_0 \times X_1 \xrightarrow{f \circ \pi_i} Y}$$

$$\frac{}{X \xrightarrow{id} X} \qquad \frac{X \xrightarrow{f} Y \quad Y \xrightarrow{g} Z}{X \xrightarrow{g \circ f} Z}$$

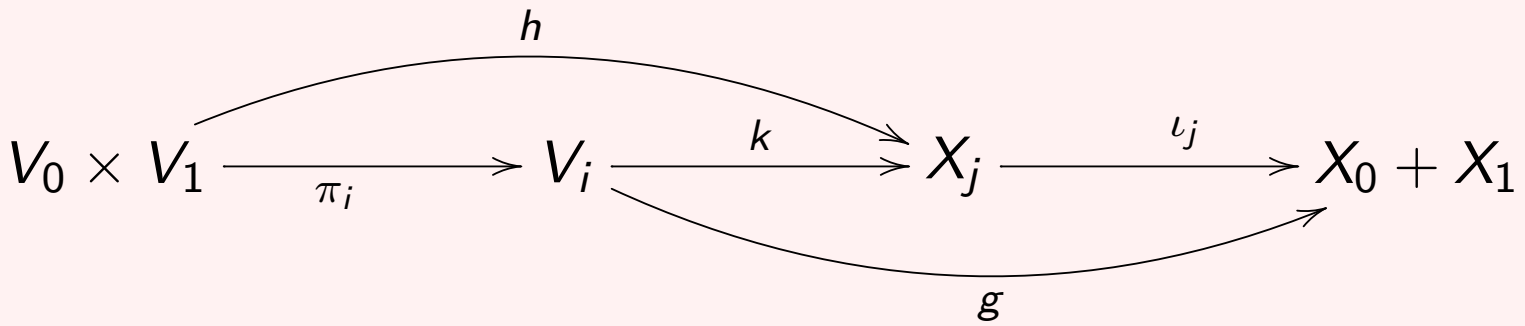
Softness

Joyal: Free Bicompletions of Categories (1995)
a morphism $f : V_0 \times V_1 \to X_0 + X_1$ has one of these forms

$$V_0 \times V_1 \xrightarrow{\pi_i} V_i \xrightarrow{g} X_0 + X_1$$

$$V_0 \times V_1 \xrightarrow{h} X_j \xrightarrow{\iota_j} X_0 + X_1$$

and if it has both, then



Proof identity

Cockett and Seely: Finite Sum–Product Logic (2001)

$$\iota_i \circ (f \circ \pi_j) = (\iota_i \circ f) \circ \pi_j$$

$$[\iota_i \circ f, \iota_i \circ g] = \iota_i \circ [f, g] \qquad \langle f \circ \pi_i, g \circ \pi_i \rangle = \langle f, g \rangle \circ \pi_i$$

$$[\langle f_0, g_0 \rangle, \langle f_1, g_1 \rangle] = \langle [f_0, f_1], [g_0, g_1] \rangle$$

$$?_1 = !_0$$

$$\langle ?, ? \rangle = ? \qquad [!, !] = !$$

$$\pi_i \circ ? = ? \qquad ! \circ \iota_i = !$$

Proof equality is decidable: terms are equal if and only if their normal forms are equated by the above equational theory

Proof identity

Cockett and Santocanale (2009):

Proof equality for sum–product logic is tractable

Equality of $f, g : X \to Y$ can be decided in time

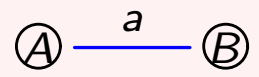
$$\mathcal{O}((hgt(X) + hgt(Y)) \times |X| \times |Y|)$$

(where $hgt(X)$ is the height and $|X|$ the total size of the syntax tree of X)

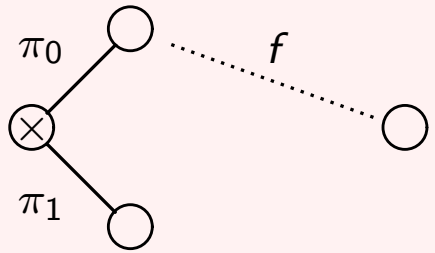
Proof nets (without units)

Hughes (2002), Hughes and Van Glabbeek (2005)

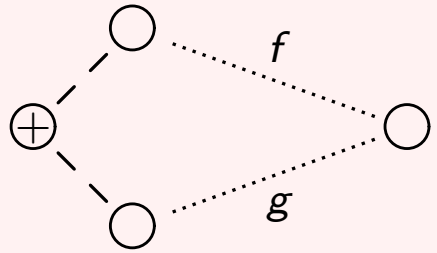
$$\frac{}{A \xrightarrow{a} B}$$



$$\frac{X_i \xrightarrow{f} Y}{X_0 \times X_1 \xrightarrow{f \circ \pi_i} Y}$$



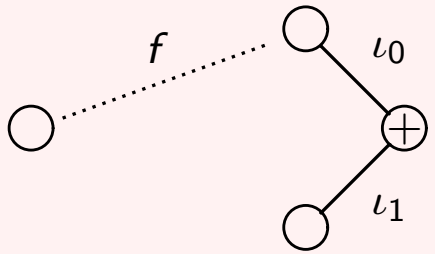
$$\frac{X_0 \xrightarrow{f} Y \quad X_1 \xrightarrow{g} Y}{X_0 + X_1 \xrightarrow{[f,g]} Y}$$



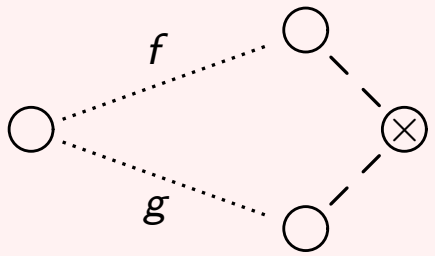
Proof nets (without units)

Hughes (2002), Hughes and Van Glabbeek (2005)

$$\frac{X \xrightarrow{f} Y_i}{X \xrightarrow{\iota_i \circ f} Y_0 + Y_1}$$



$$\frac{X \xrightarrow{f} Y_0 \quad X \xrightarrow{g} Y_1}{X \xrightarrow{\langle f,g \rangle} Y_0 \times Y_1}$$

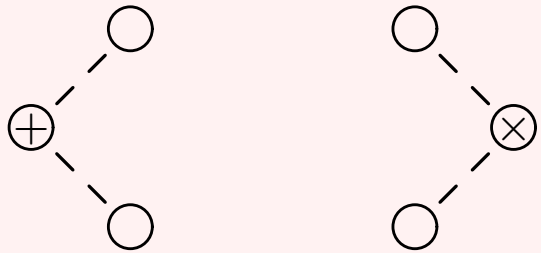


Switching

A net $X \xrightarrow{R} Y$ has

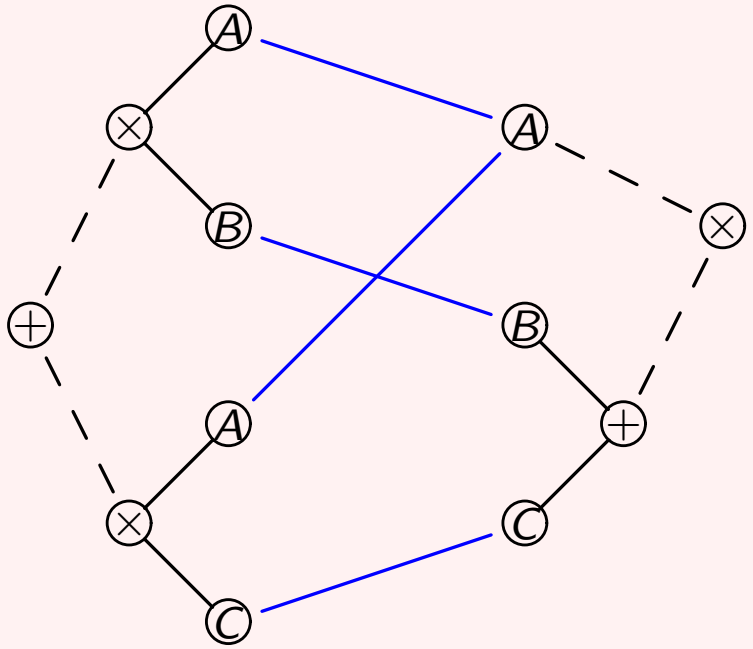
- a source object X
- a target object Y
- a labelled relation R from the leaves in X to the leaves in Y

Any such triple is a net if it satisfies the switching condition:



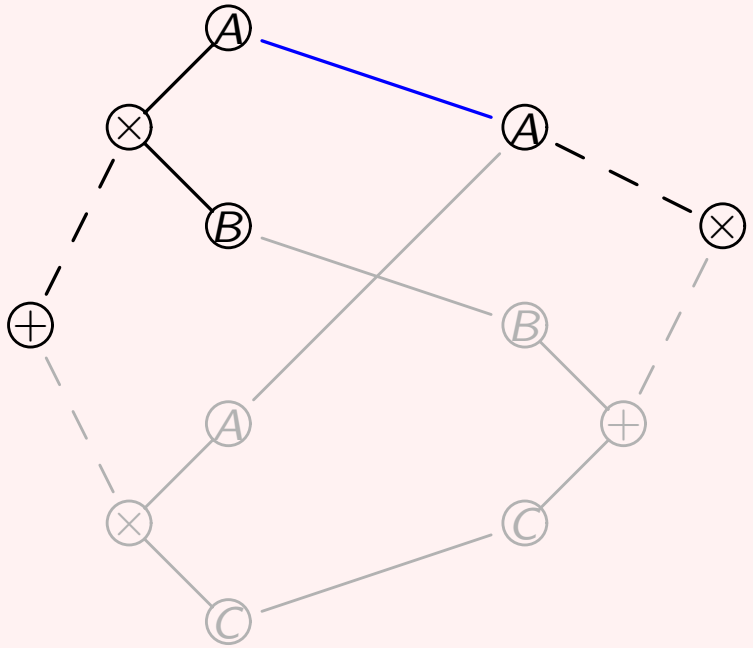
After choosing one branch for each coproduct in X and each product in Y there must be exactly one path from left to right.

Example: switching



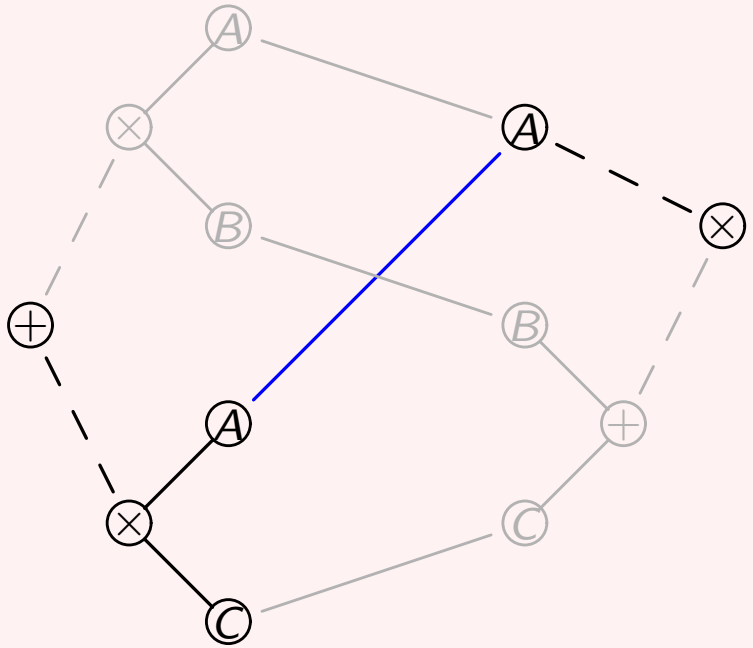
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



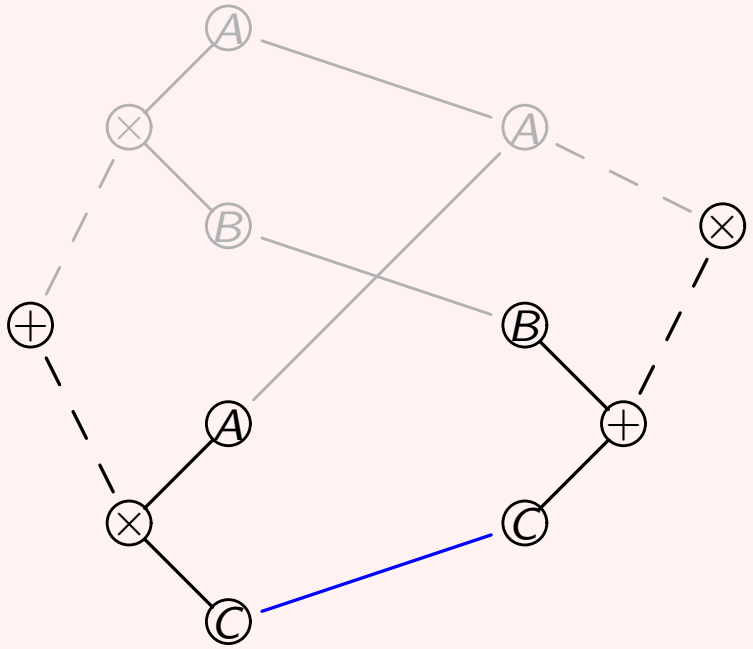
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



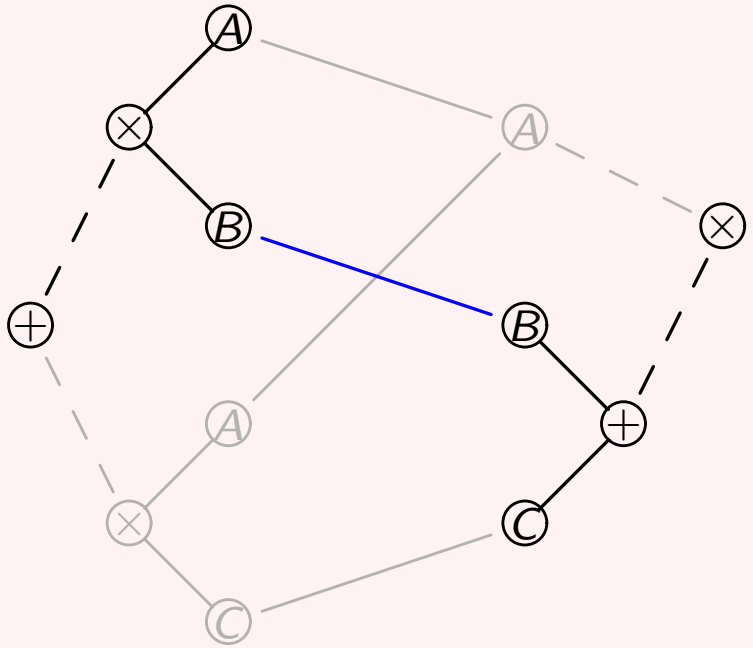
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



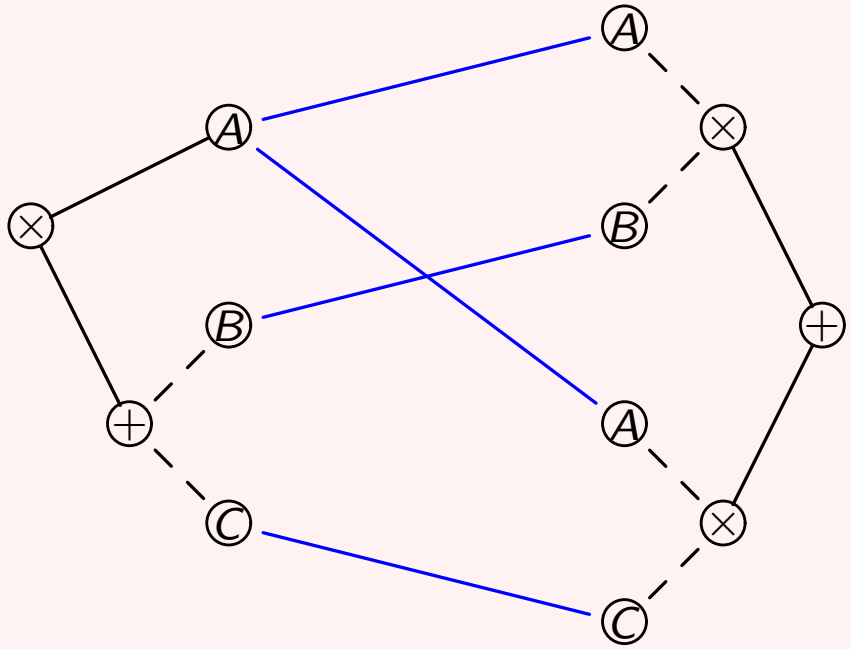
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



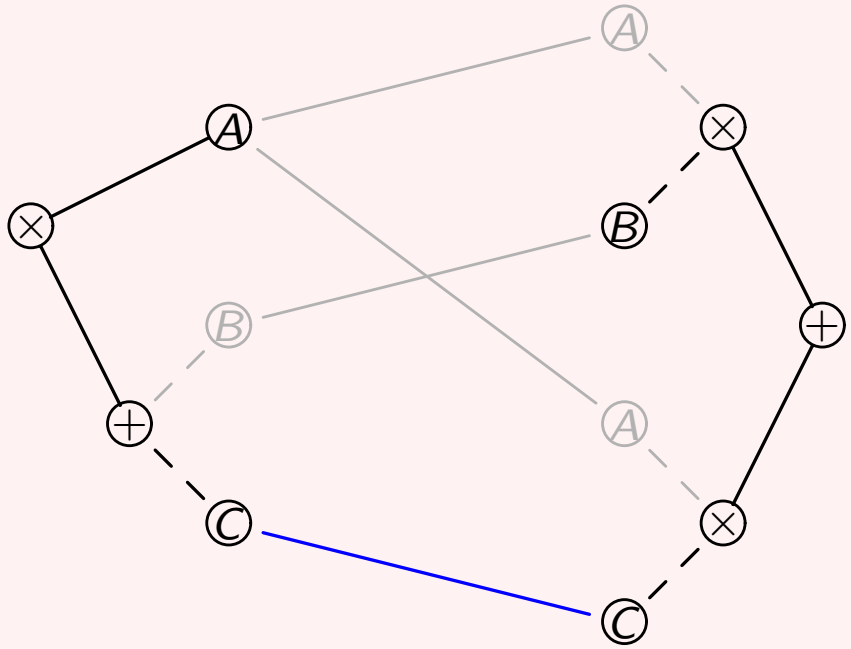
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Non-example: switching



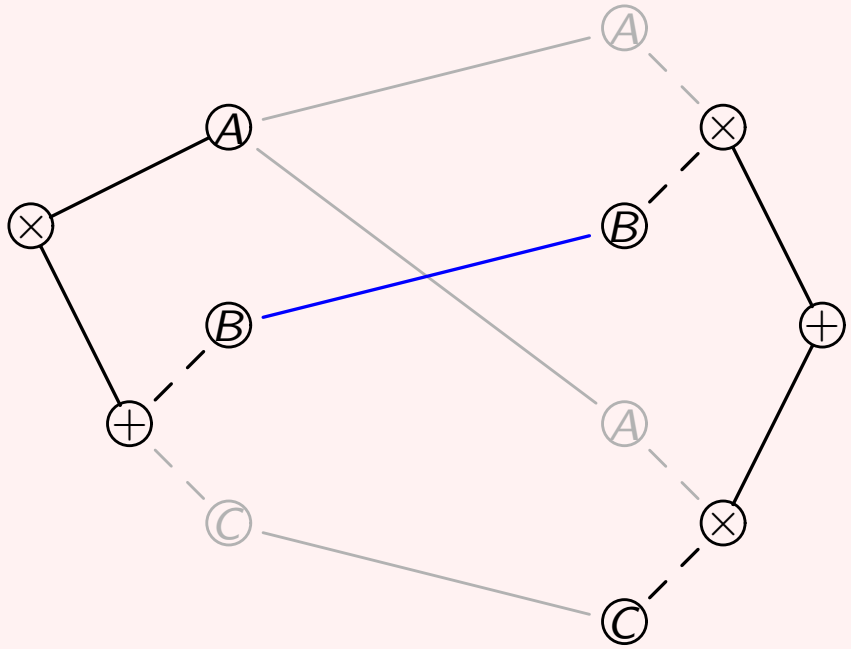
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



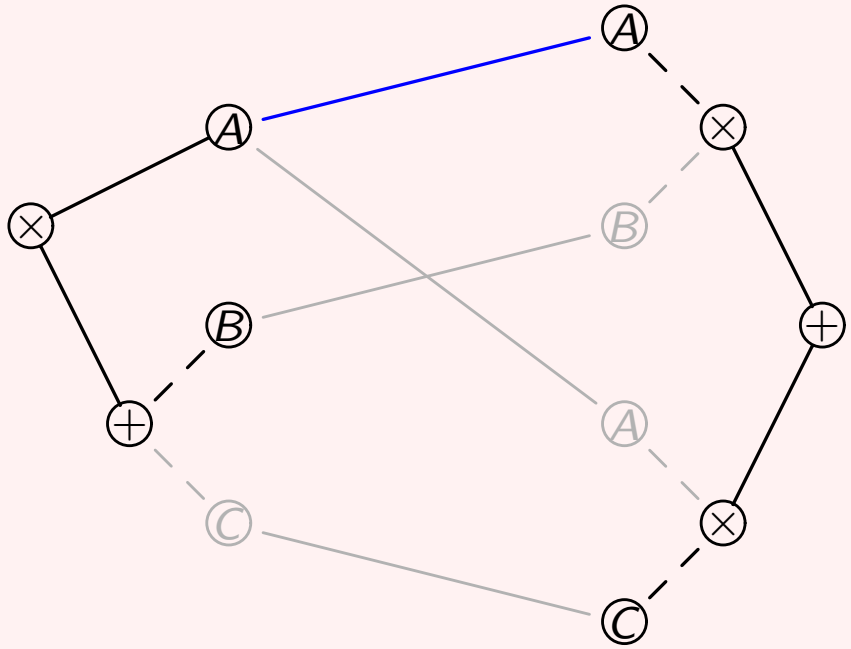
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



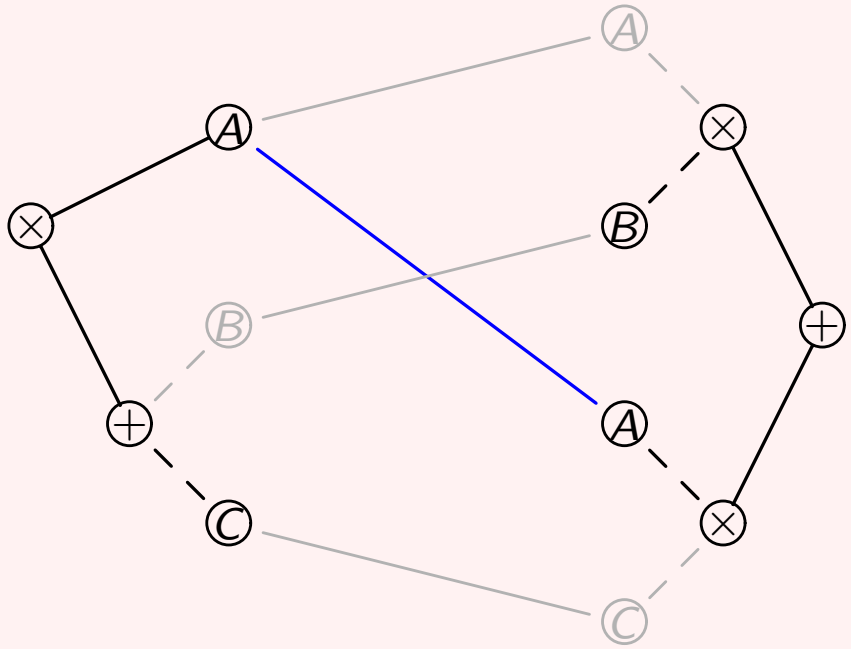
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



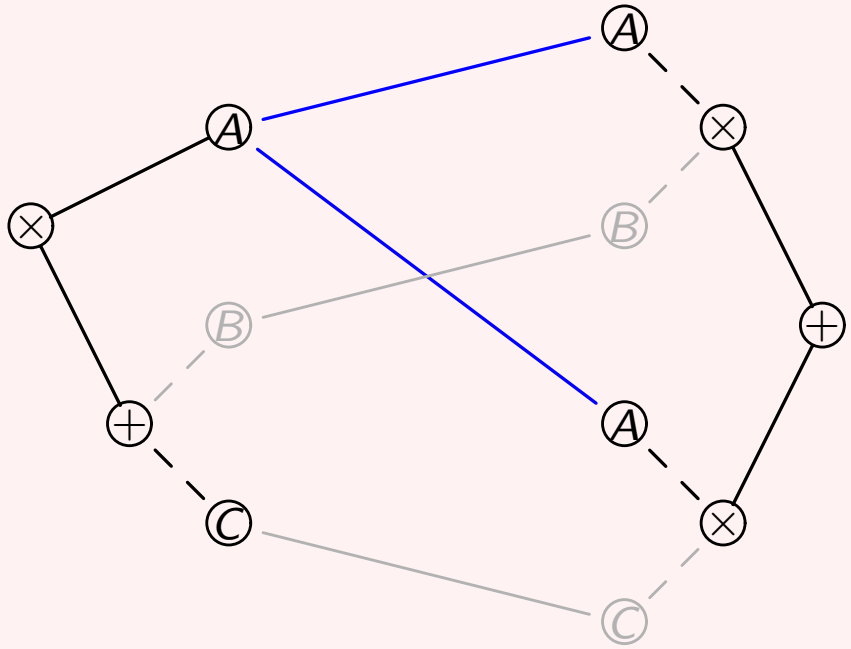
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



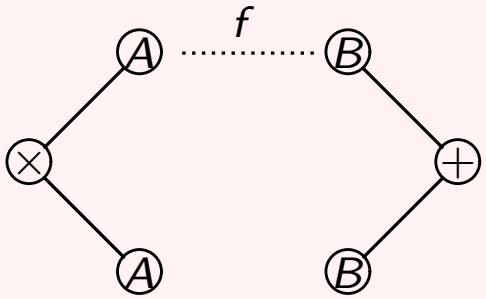
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching

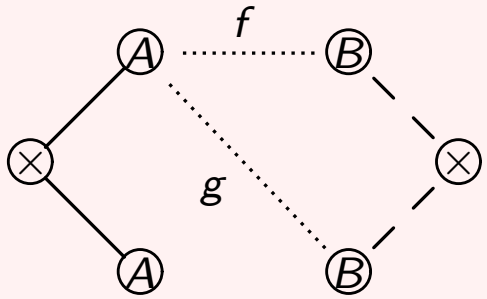


$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

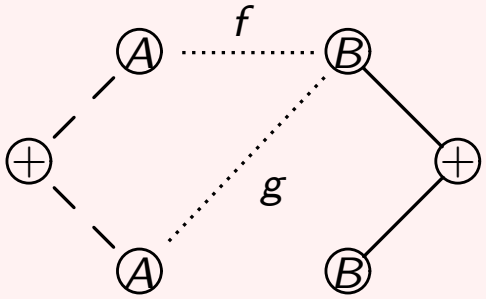
Equalities factored out



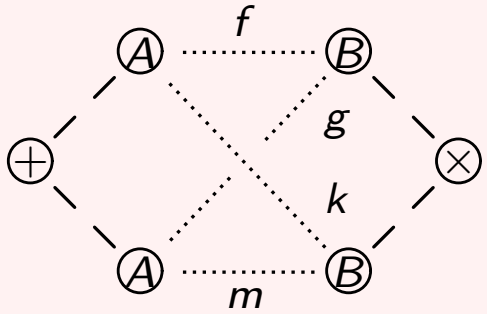
$$\iota_0 \circ (f \circ \pi_0) = (\iota_0 \circ f) \circ \pi_0$$



$$\langle f \circ \pi_0, g \circ \pi_0 \rangle = \langle f, g \rangle \circ \pi_0$$



$$[\iota_0 \circ f, \iota_0 \circ g] = \iota_0 \circ [f, g]$$

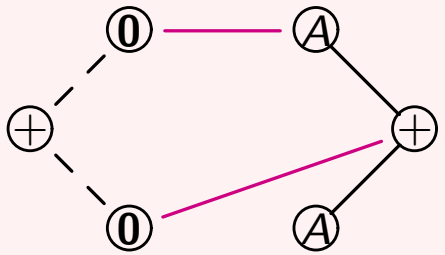


$$\langle [f, g], [k, m] \rangle = [\langle f, k \rangle, \langle g, m \rangle]$$

The units

For initial and terminal maps $? : \mathbf{0} \to Y$ or $! : X \to \mathbf{1}$ the objects X and Y may be a product or coproduct.

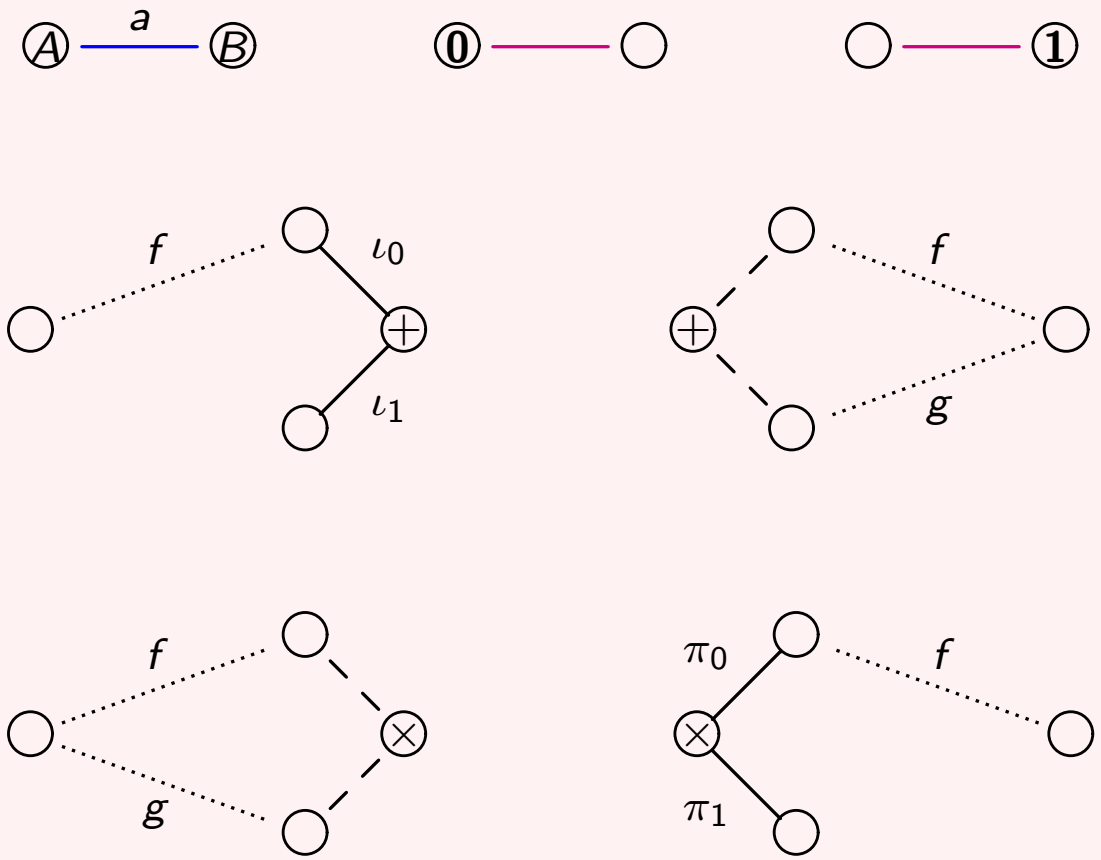
The above links are added, which are not restricted to the leaves.



The switching condition is unaffected.
Omitting the label factors out an additional equality:

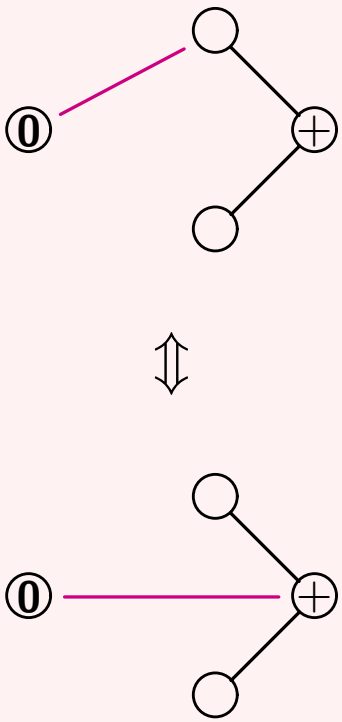
$$\mathbf{0} \underset{!}{\overset{?}{\longrightarrow}} \mathbf{1} \qquad \textcircled{0} \text{——} \textcircled{1}$$

The full net calculus



The unit equations

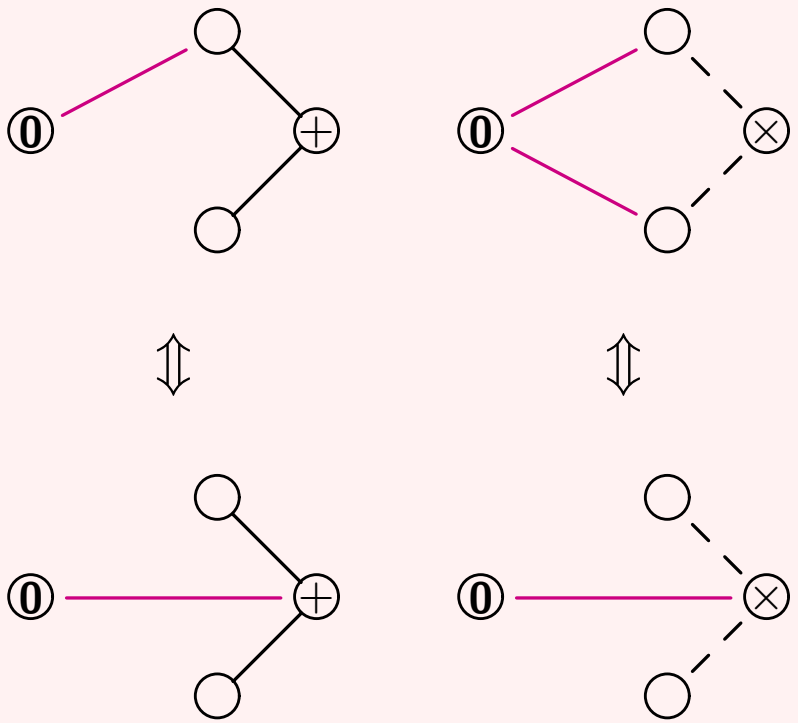
$$\iota_i \circ ? = ?$$



. . . define an equational theory ($\Leftrightarrow$) over nets, via graph rewriting

The unit equations

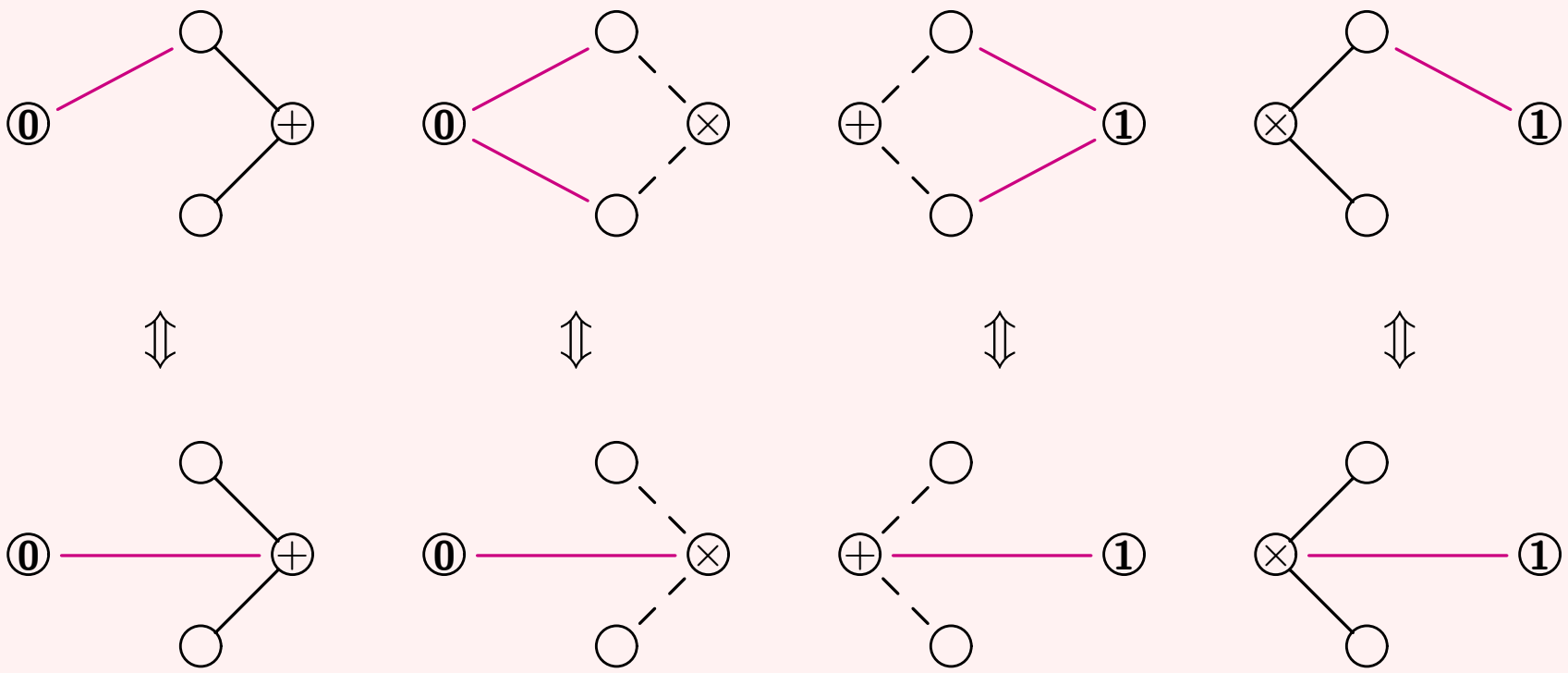
$\iota_i \circ ? = ?$ $\qquad$ $\langle ?, ? \rangle = ?$



. . . define an equational theory ($\Leftrightarrow$) over nets, via graph rewriting

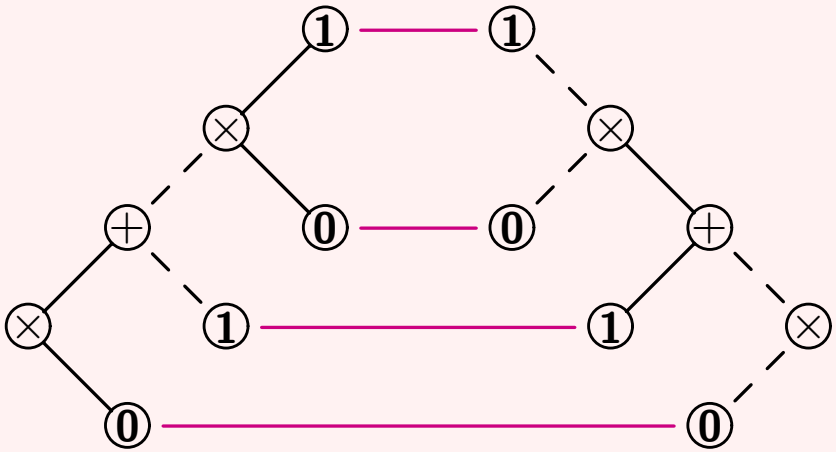
The unit equations

$\iota_i \circ ? = ?$ $\quad \langle ?, ? \rangle = ?$ $\quad [\,!, !\,] = !$ $\quad ! \circ \pi_i = !$

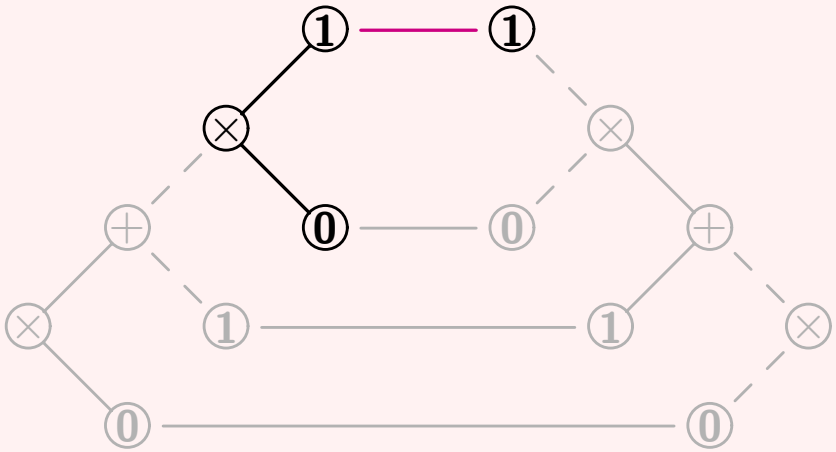


. . . define an equational theory ($\Leftrightarrow$) over nets, via graph rewriting

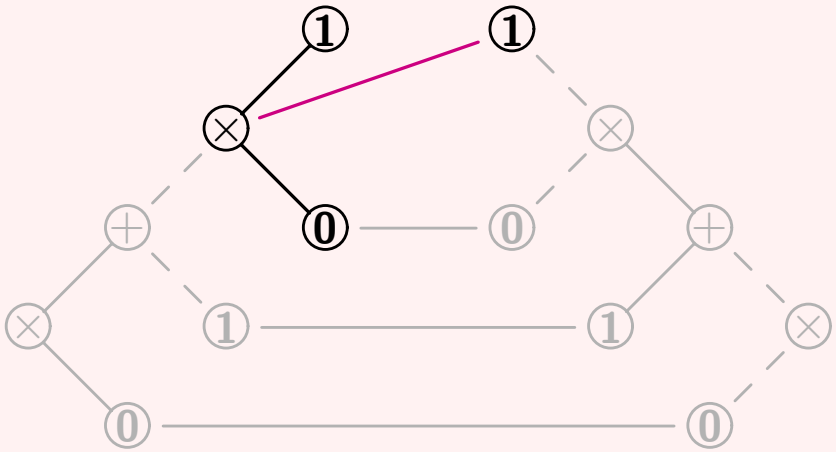
Example



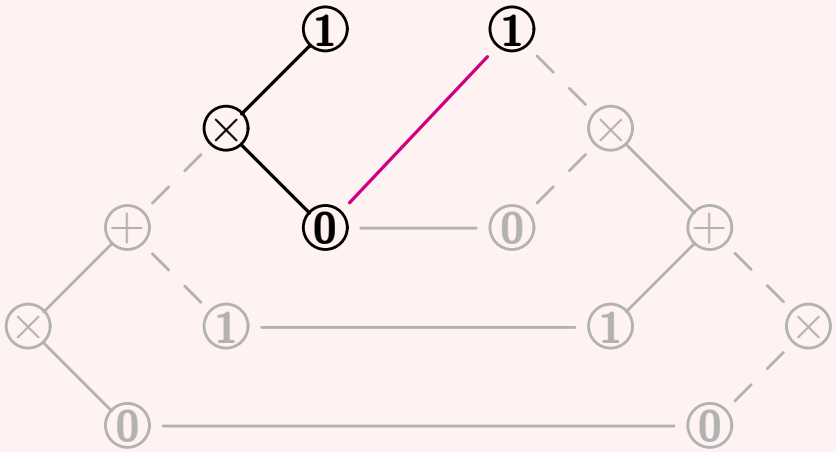
Example



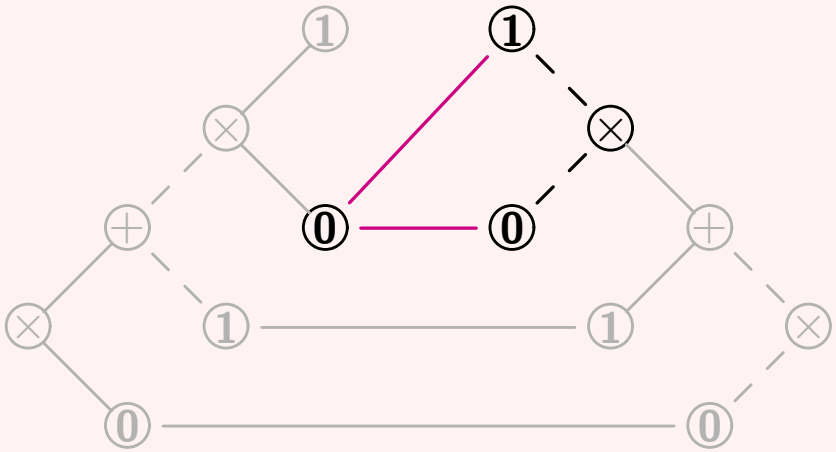
Example



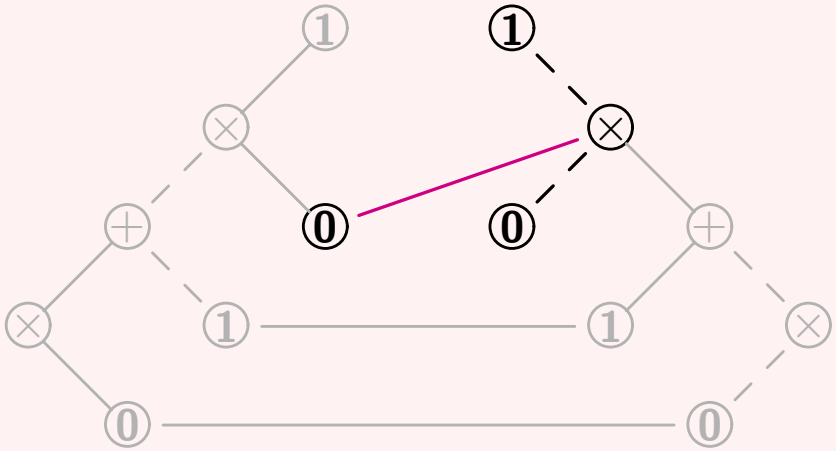
Example



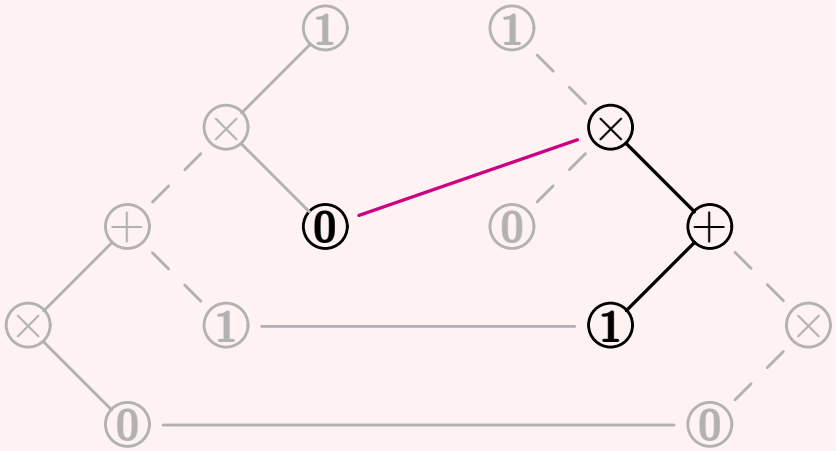
Example



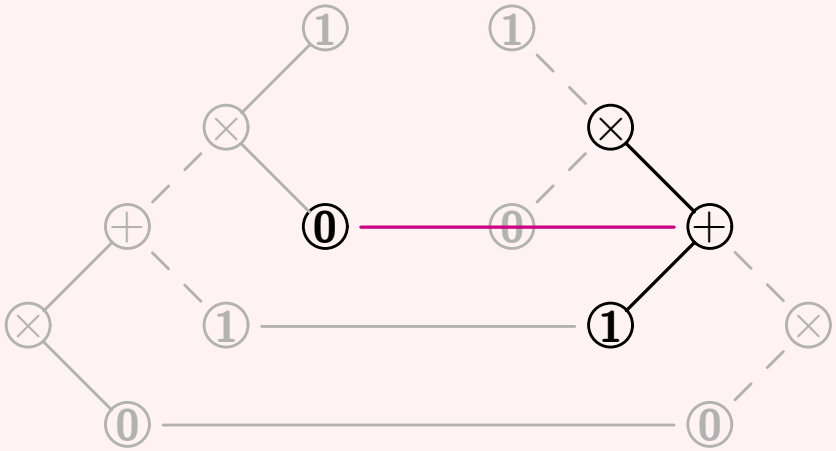
Example



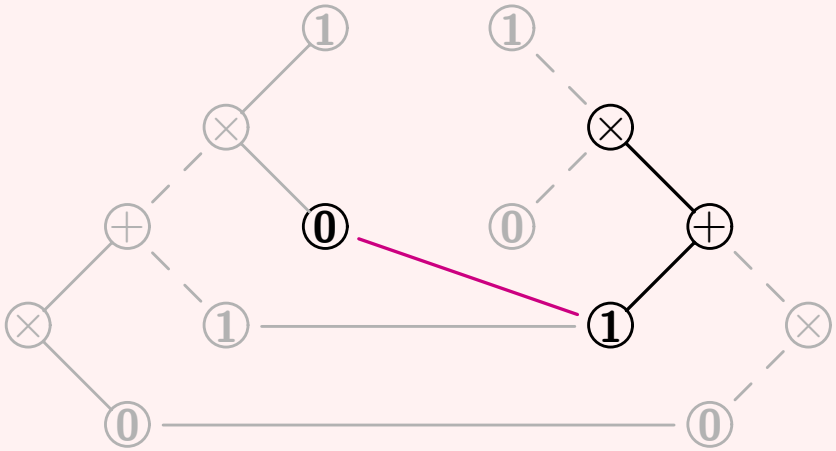
Example



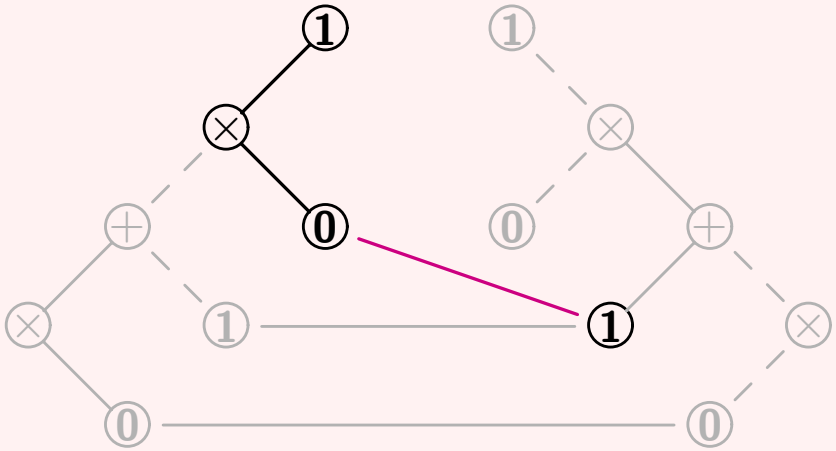
Example



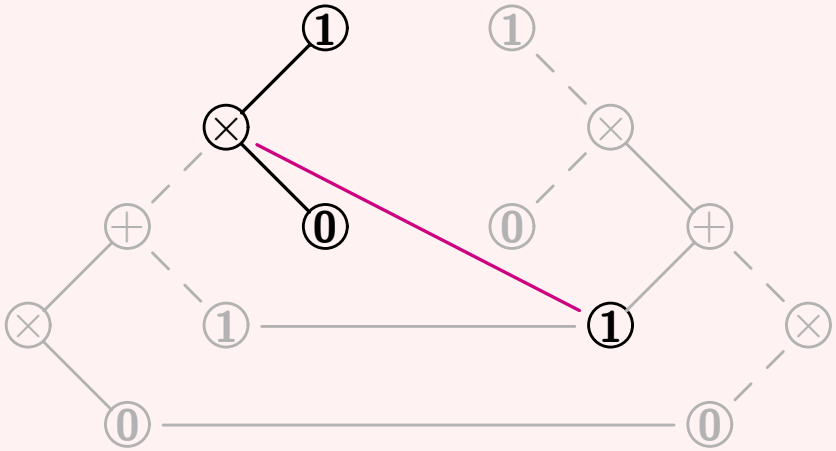
Example



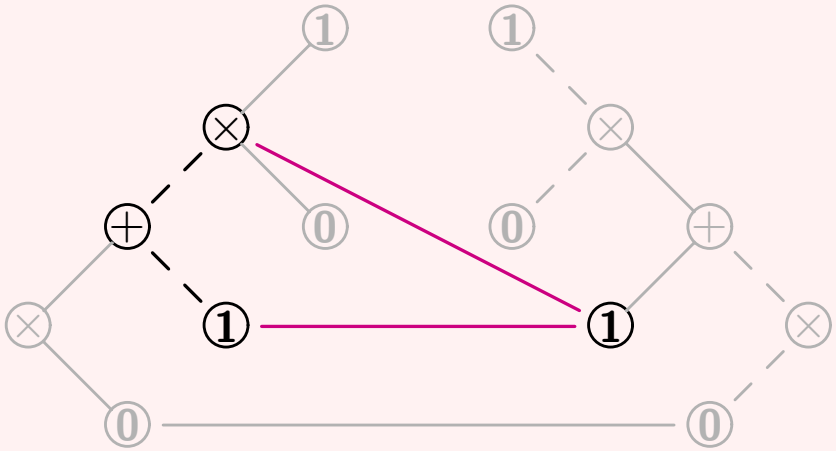
Example



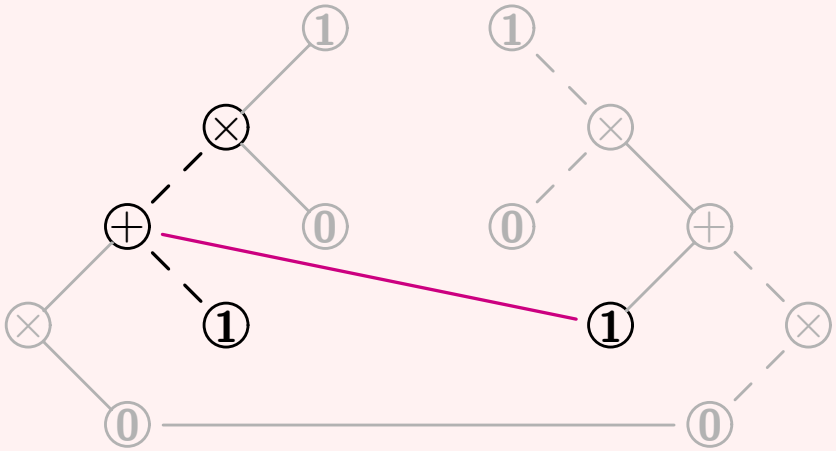
Example



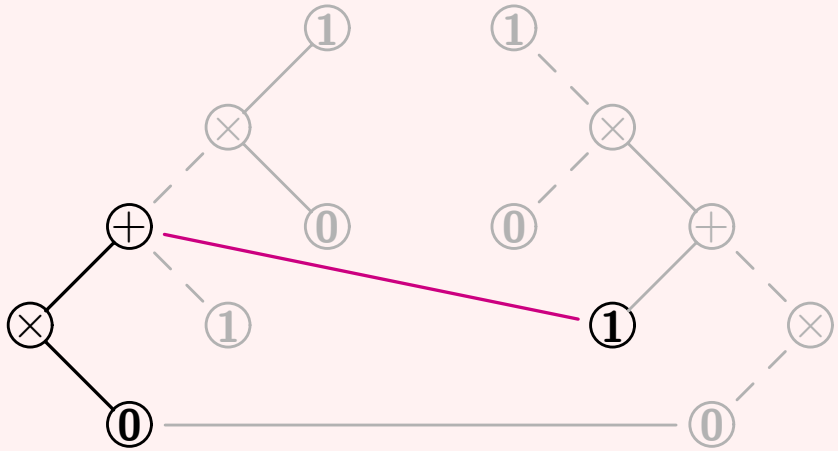
Example



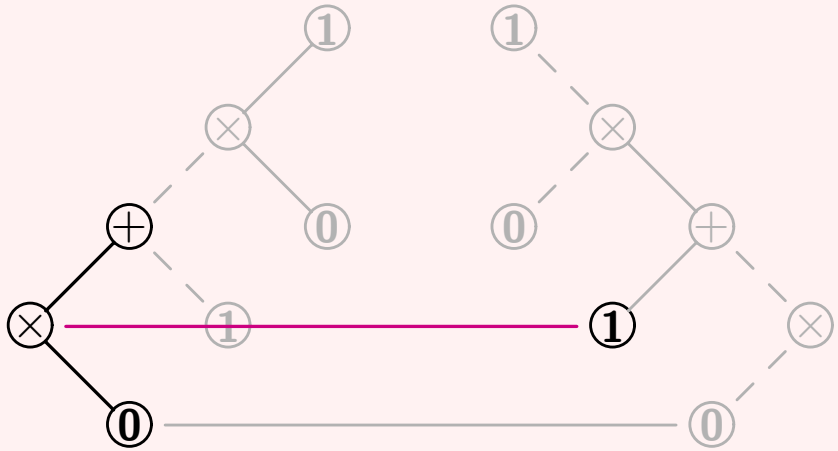
Example



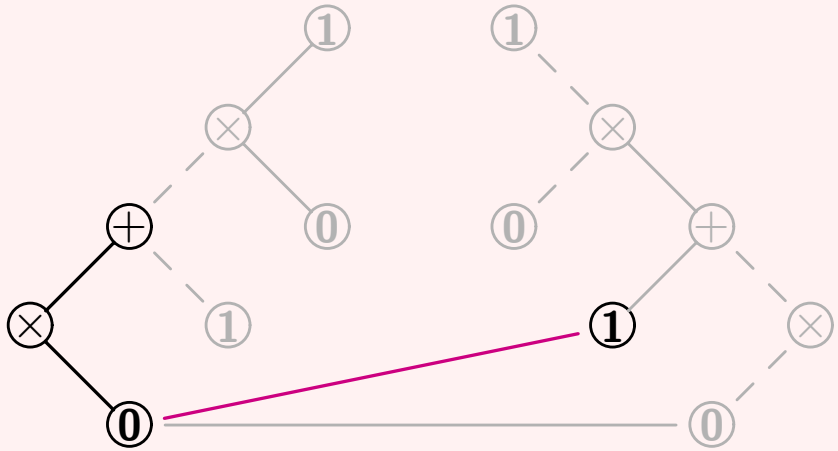
Example



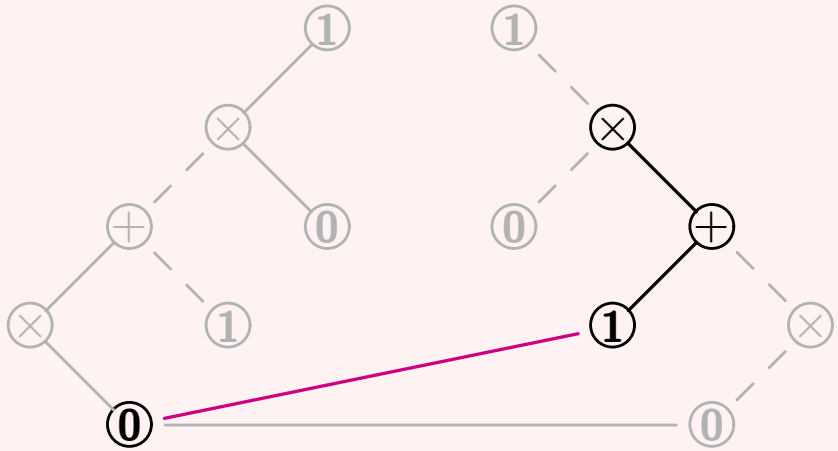
Example



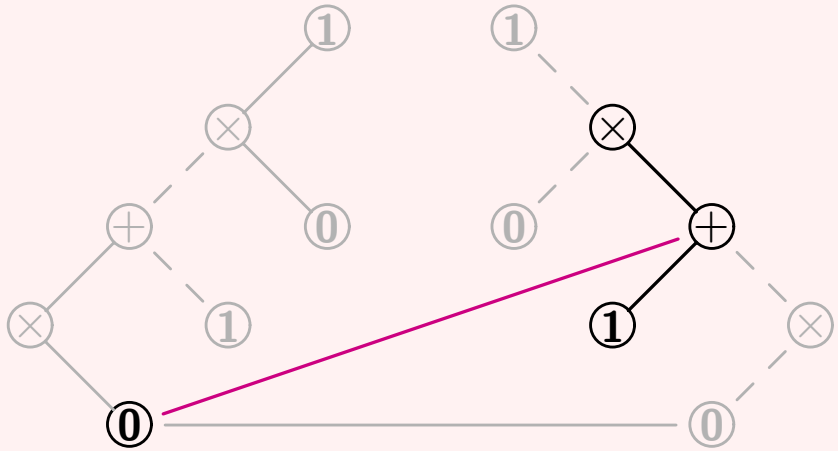
Example



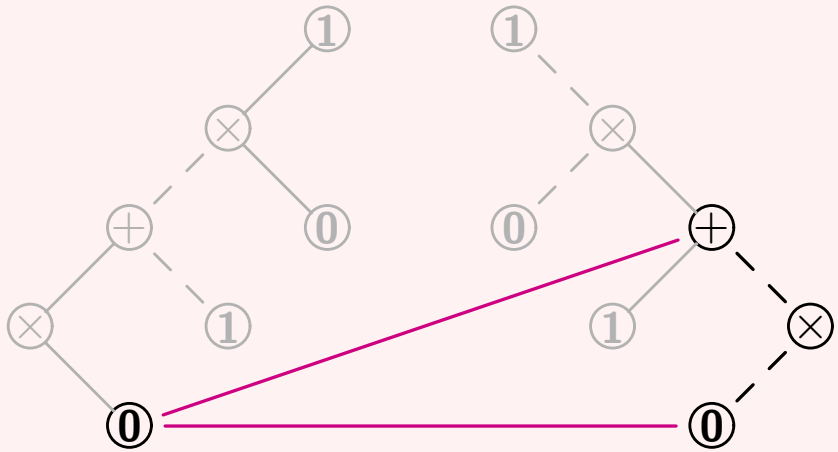
Example



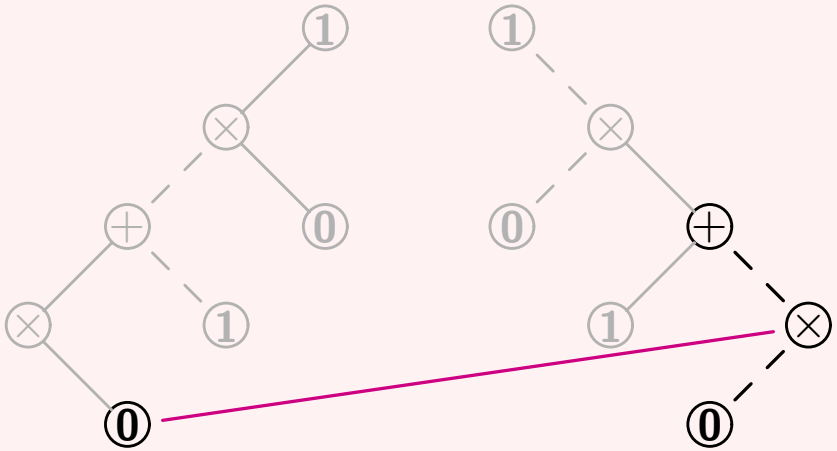
Example



Example



Example



The problem

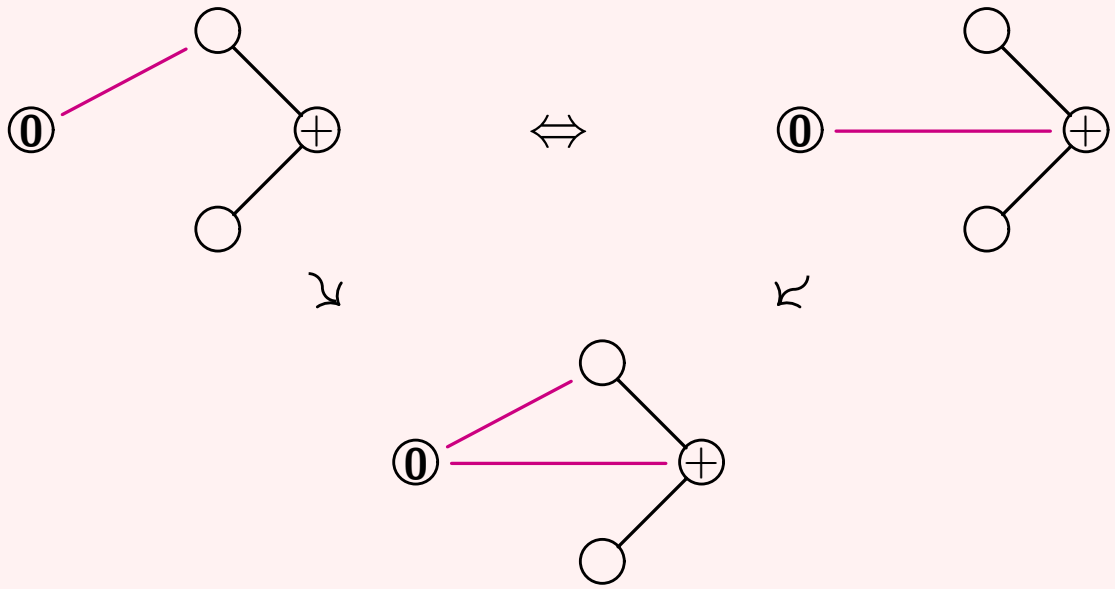
We would like canonical representations for the equivalence classes of proof nets generated by ($\Leftrightarrow$).

A standard approach is to rewrite towards a normal form, using a confluent and terminating rewrite relation.

As the previous example illustrated, showing equivalence in ($\Leftrightarrow$) requires rewrites in all directions—simply directing it will not work.

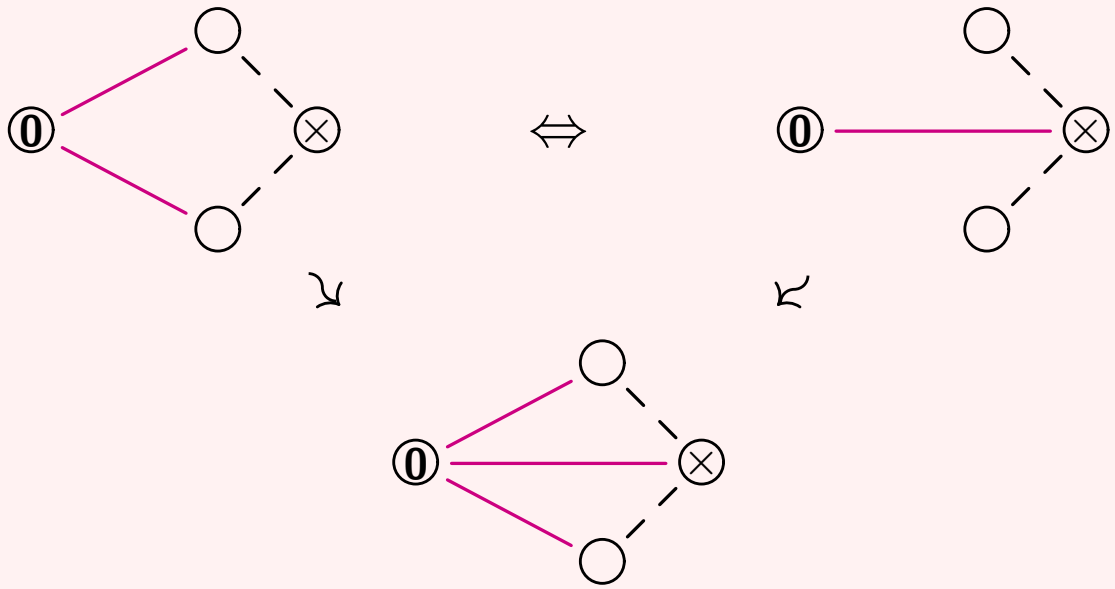
Saturation

Confluent rewriting seems impossible without breaking the switching condition. So: break it. Then there is a simple confluent and normalising rewrite relation: saturation ($\rightsquigarrow$).



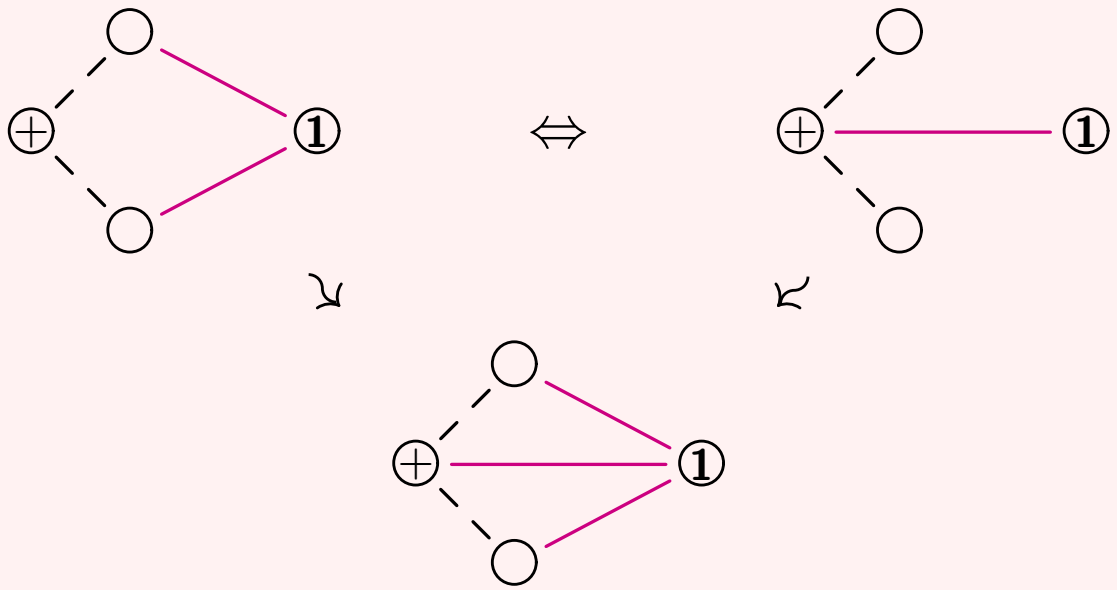
Saturation

Confluent rewriting seems impossible without breaking the switching condition. So: break it. Then there is a simple confluent and normalising rewrite relation: saturation ($\rightsquigarrow$).



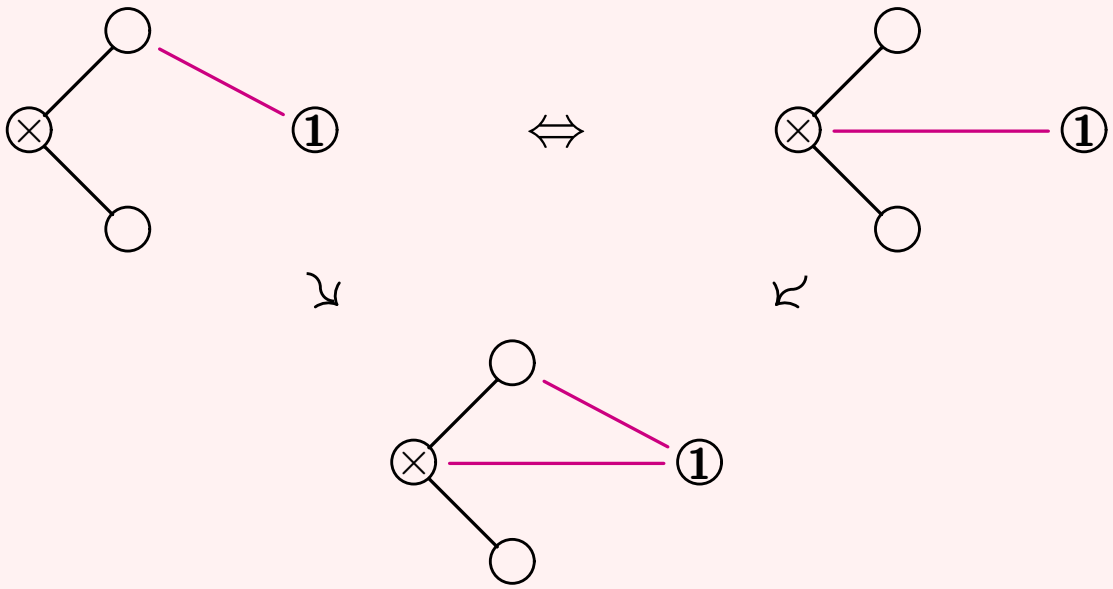
Saturation

Confluent rewriting seems impossible without breaking the switching condition. So: break it. Then there is a simple confluent and normalising rewrite relation: saturation ($\rightsquigarrow$).

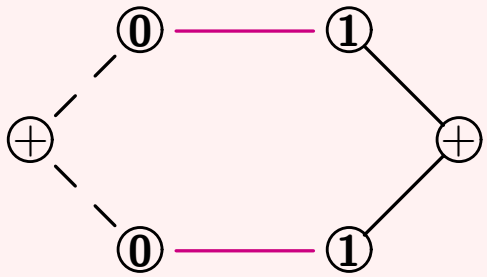


Saturation

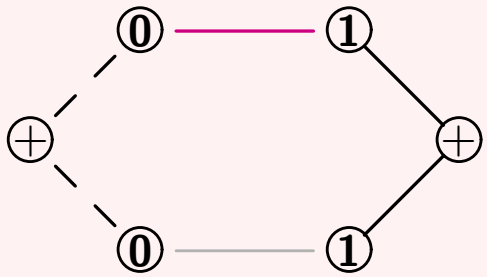
Confluent rewriting seems impossible without breaking the switching condition. So: break it. Then there is a simple confluent and normalising rewrite relation: saturation ($\rightsquigarrow$).



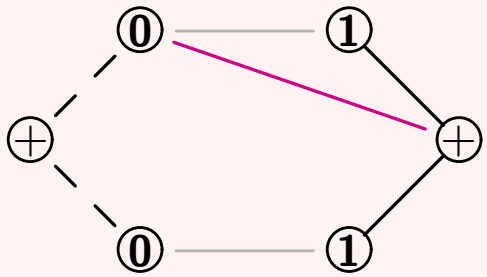
Example



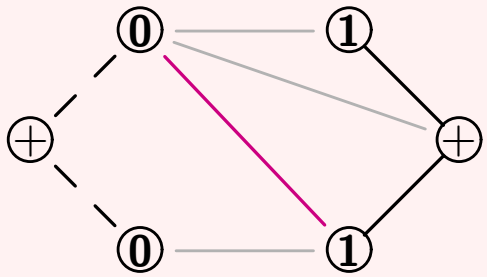
Example



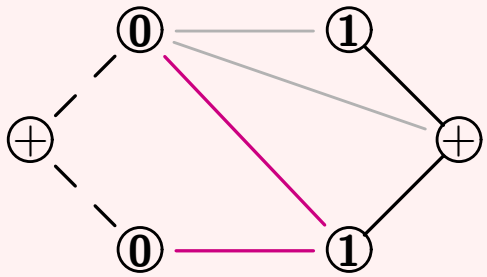
Example



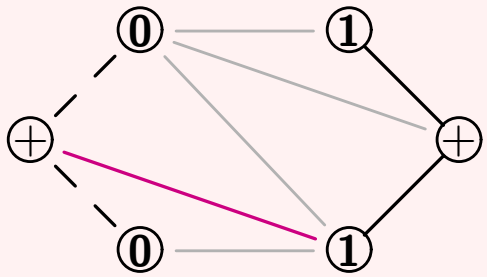
Example



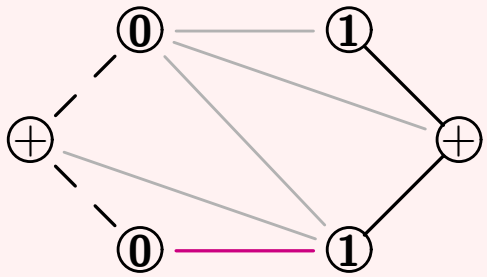
Example



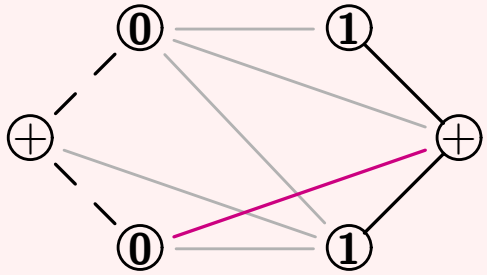
Example



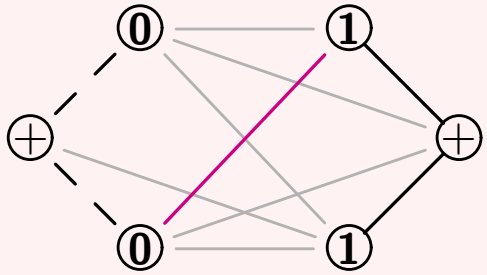
Example



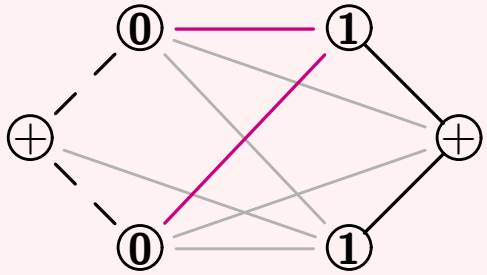
Example



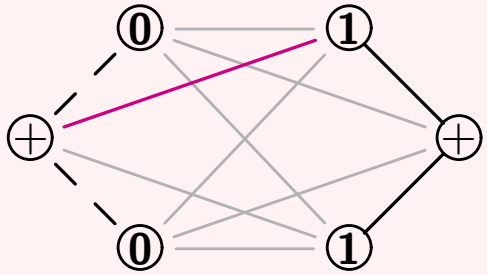
Example



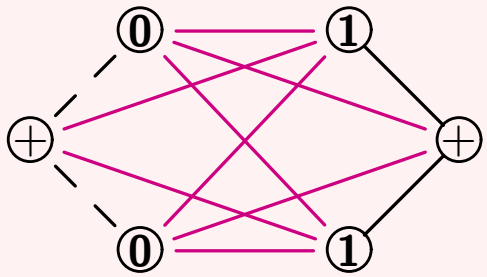
Example



Example



Example



Immediate results

The saturation relation ($\rightsquigarrow$) is

confluent	rewrite steps add links, depending on the presence of other links
strongly normalising	bounded by the number of possible links ($\lvert X\rvert \times \lvert Y\rvert$ for $X \xrightarrow{R} Y$)
linear-time	(in $\lvert X\rvert \times \lvert Y\rvert$); saturation steps are constant-time

Immediate results

The saturation relation ($\rightsquigarrow$) is

confluent	rewrite steps add links, depending on the presence of other links
strongly normalising	bounded by the number of possible links ($\lvert X\rvert \times \lvert Y\rvert$ for $X \xrightarrow{R} Y$)
linear-time	(in $\lvert X\rvert \times \lvert Y\rvert$); saturation steps are constant-time

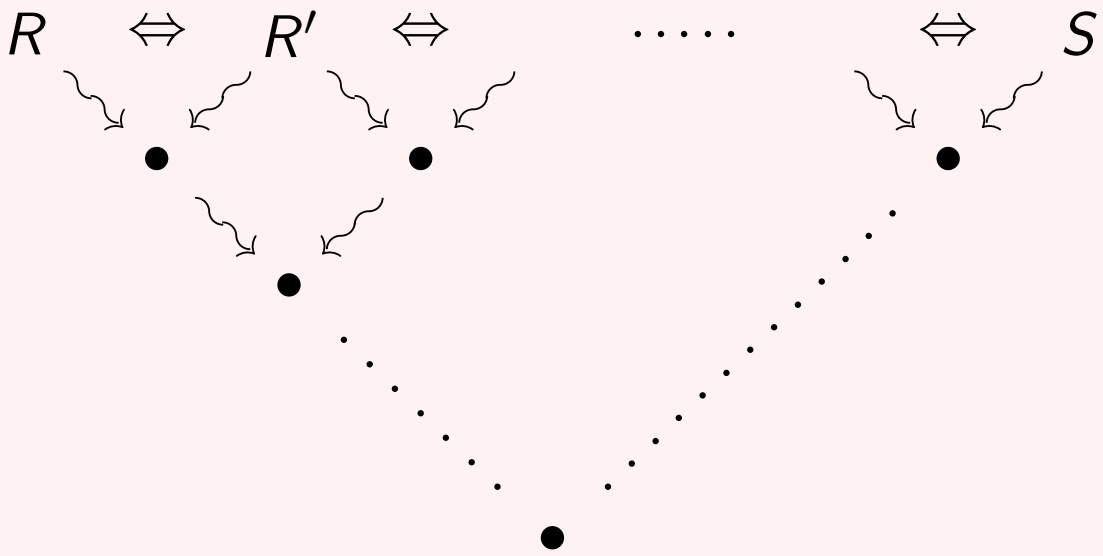
Write $X \xrightarrow{\sigma R} Y$ for the normal form (the saturation) of a net $X \xrightarrow{R} Y$ and call it a saturated net

Main results

Thm. Saturation gives a decision procedure for term equality in sum–product logic:

$$X \xrightarrow{R} Y \;\Leftrightarrow\; X \xrightarrow{S} Y \qquad \Longleftrightarrow \qquad X \xrightarrow{\sigma R} Y \;=\; X \xrightarrow{\sigma S} Y$$

Completeness ($\Rightarrow$)



Soundness ($\Leftarrow$) is the difficult (and important) part

Four obstacles in the soundness proof

- Saturation paths don't give much

$$R \rightsquigarrow R' \rightsquigarrow R'' \rightsquigarrow \ldots \rightsquigarrow \sigma R = \sigma S \leftsquigarrow \ldots \leftsquigarrow S'' \leftsquigarrow S' \leftsquigarrow S$$

Four obstacles in the soundness proof

- Saturation paths don't give much

$$R \rightsquigarrow R' \rightsquigarrow R'' \rightsquigarrow \ldots \rightsquigarrow \sigma R = \sigma S \leftsquigarrow \ldots \leftsquigarrow S'' \leftsquigarrow S' \leftsquigarrow S$$

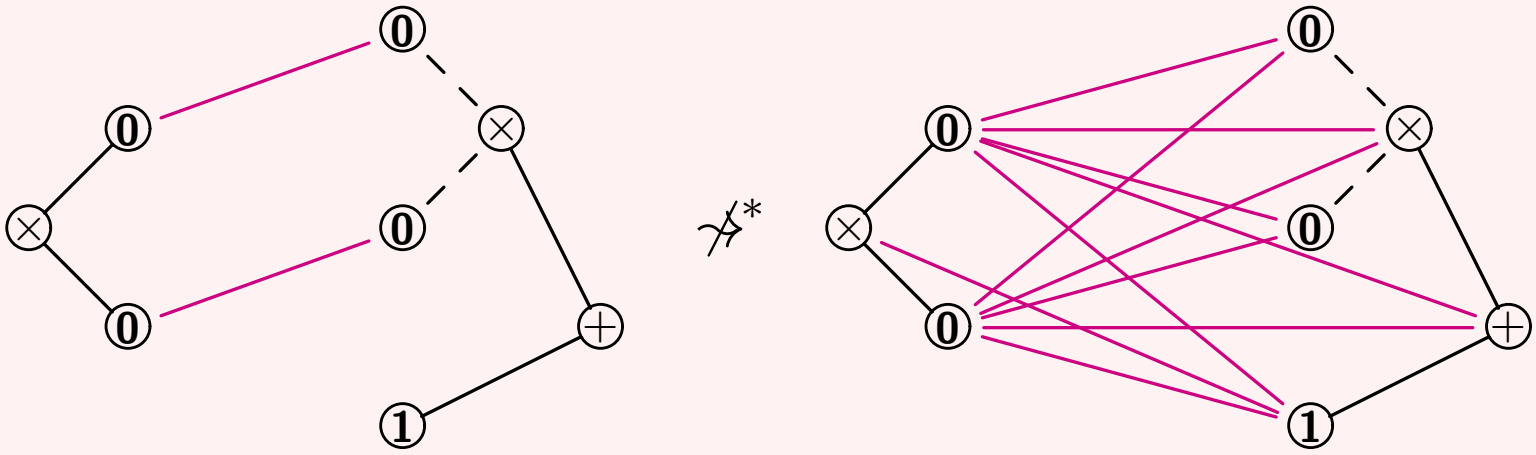
Does this give a corresponding path of equivalences?

$$R \Leftrightarrow R_0 \Leftrightarrow R_1 \Leftrightarrow \ldots \Leftrightarrow R_m \quad ?? \quad S_n \Leftrightarrow \ldots \Leftrightarrow S_1 \Leftrightarrow S_0 \Leftrightarrow S$$

How to show that $\sigma R = \sigma S$ gives $R_m \Leftrightarrow S_n$?

Four obstacles in the soundness proof

- Saturation paths don't give much
- De-saturation is non-trivial



Four obstacles in the soundness proof

- Saturation paths don't give much
- De-saturation is non-trivial

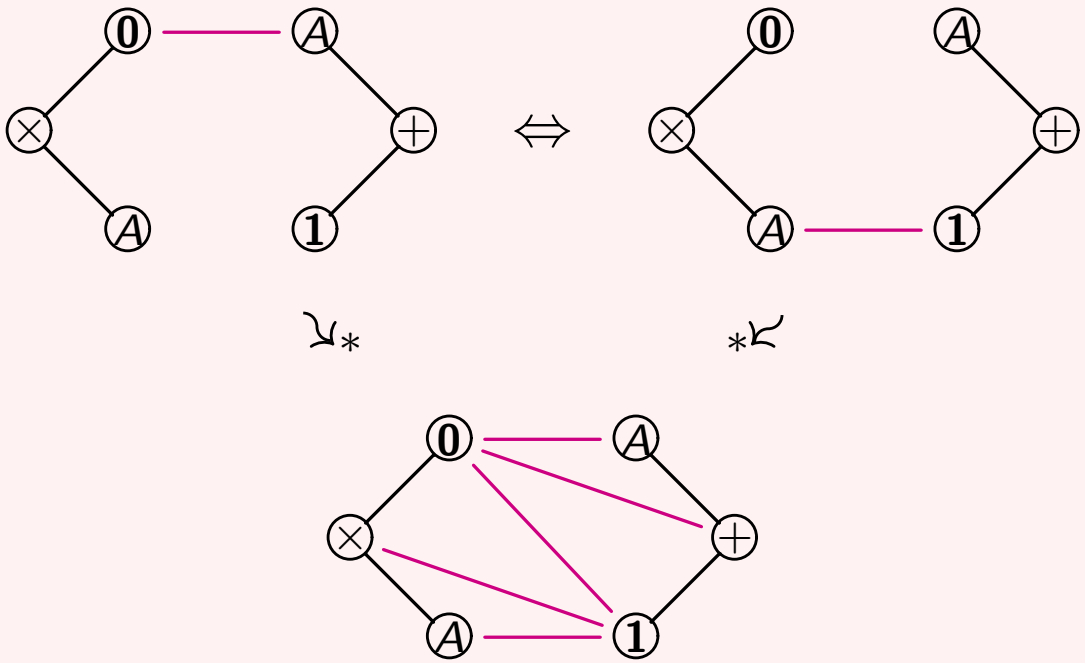
Approach: induction on source and target object

Four obstacles in the soundness proof

- Saturation paths don't give much
- De-saturation is non-trivial

Approach: induction on source and target object

- Differently constructed nets may be equivalent

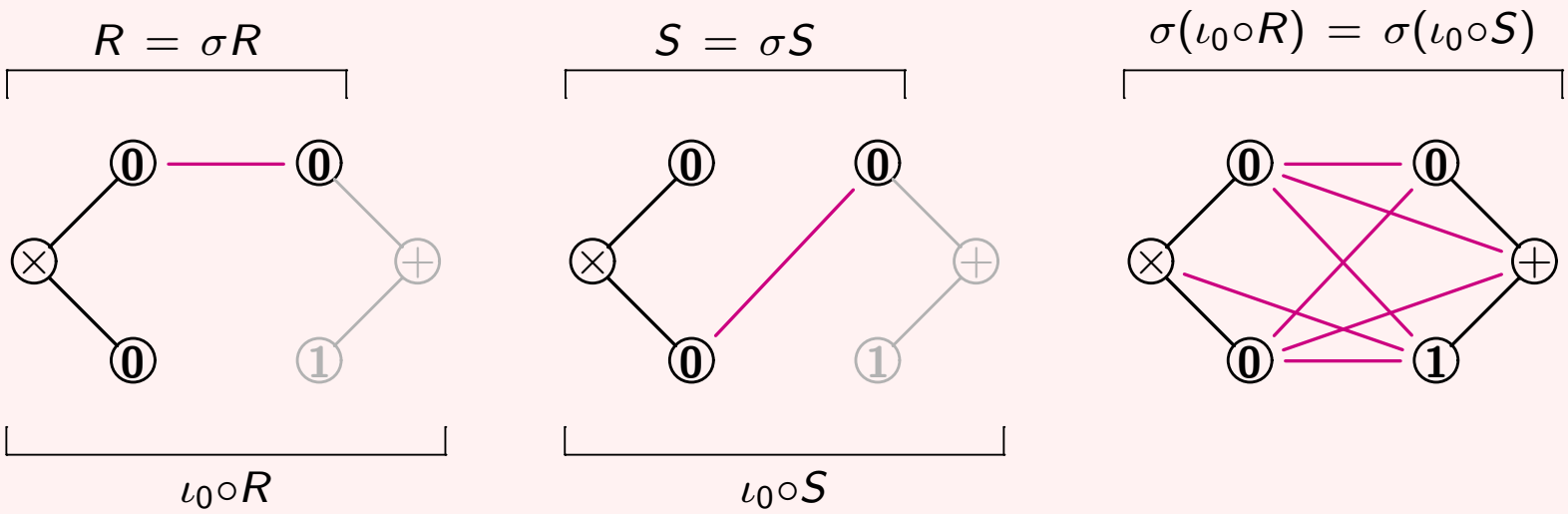


Four obstacles in the soundness proof

- Saturation paths don't give much
- De-saturation is non-trivial

Approach: induction on source and target object

- Differently constructed nets may be equivalent
- Injecting into $X + \mathbf{1}$ / projecting from $X \times \mathbf{0}$ adds equivalences

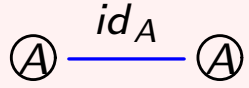


The category of saturated nets

The category of saturated nets is the free completion with finite (nullary and binary) products and coproducts of a base category $\mathcal{C}$.

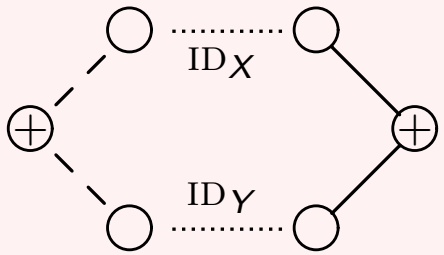
The category of saturated nets

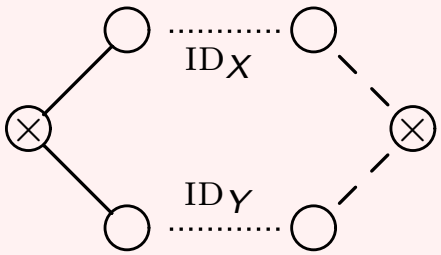
Identities are nets $X \xrightarrow{\sigma \mathrm{ID}_X} X$ where ID_X is the identity relation on the leaves of X.



$id_{\mathbf{0}} = ?_{\mathbf{0}}$

$id_{\mathbf{1}} = !_{\mathbf{1}}$



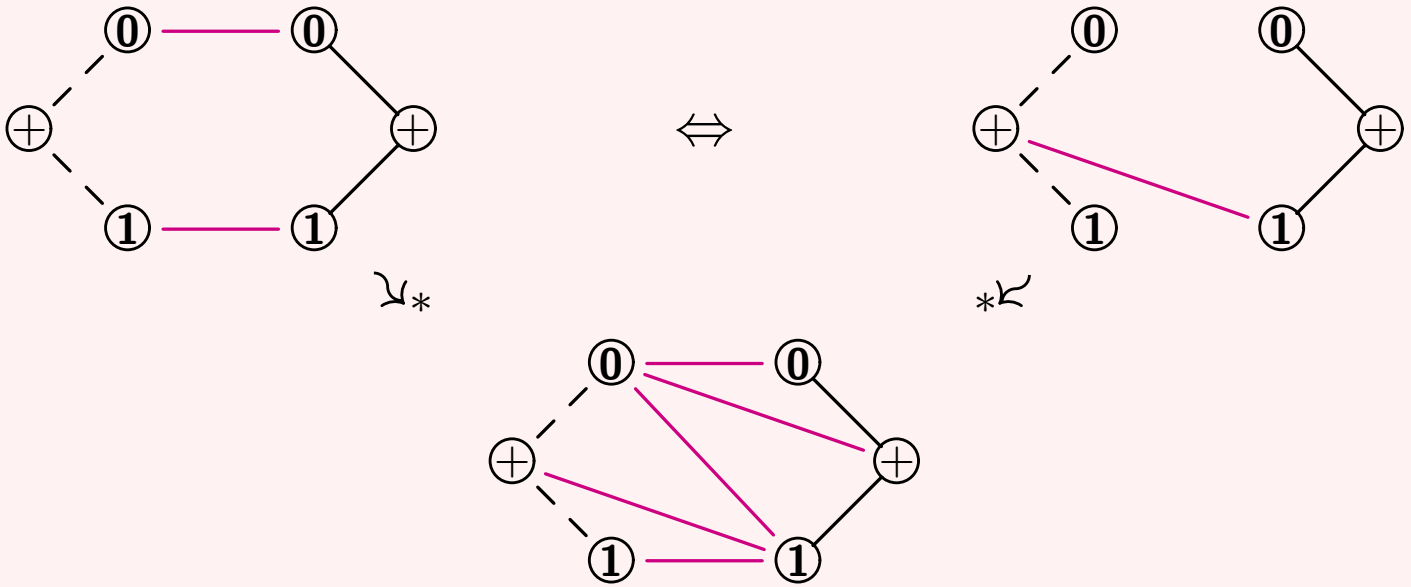


$id_{X+Y} = [\iota_0 \circ id_X, \iota_1 \circ id_Y]$

$id_{X \times Y} = \langle id_X \circ \pi_0, id_Y \circ \pi_1 \rangle$

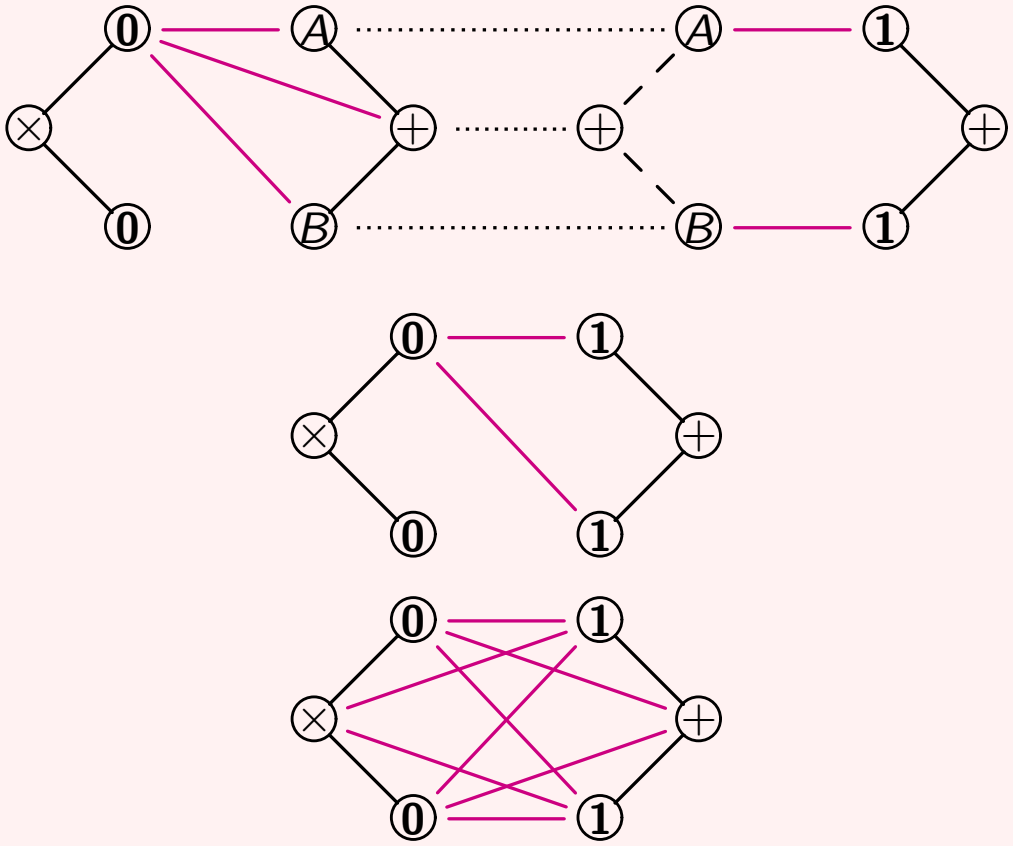
The category of saturated nets

Saturation is necessary: nets ID_X are equivalent to other nets.



The category of saturated nets

Composition is relational composition followed by (re-)saturation.

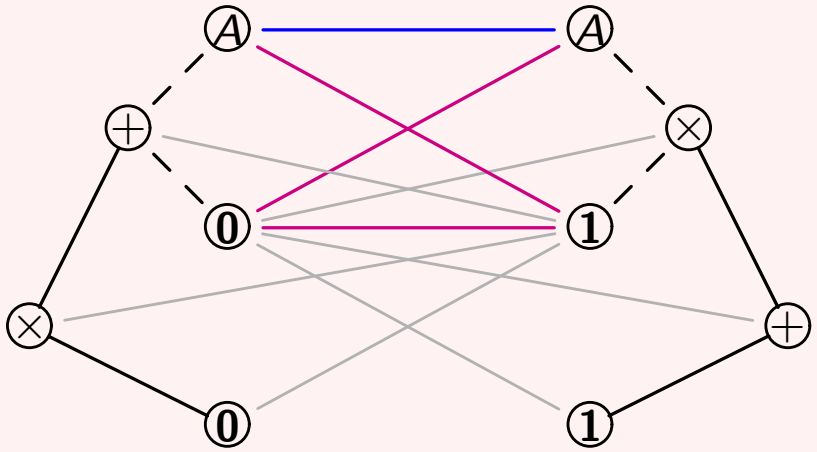


Conclusion

Saturated nets are canonical proof nets for additive linear logic and give a combinatorial description of free sum–product categories

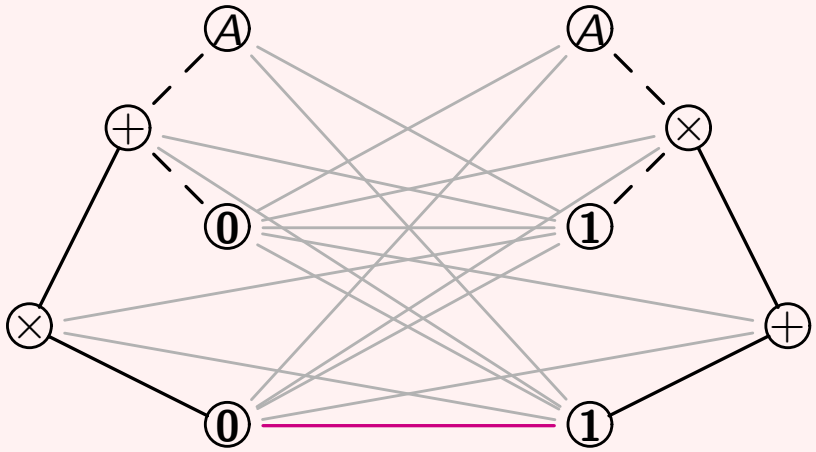
- Based on a simple rewriting algorithm
- Complicated correctness proof
- Work in progress: a correctness condition for saturated nets
- Relevant to concurrent games and communication by message passing

Example



$$(A + \mathbf{0}) \times \mathbf{0} \longrightarrow (A \times \mathbf{1}) + 1$$

Example



$$(A + \mathbf{0}) \times \mathbf{0} \longrightarrow (A \times \mathbf{1}) + 1$$