Proof nets for sum–product logic

Willem Heijltjes

LFCS
School of Informatics
University of Edinburgh

Kananaskis, 11-12 June 2011

This talk...

Part 1

- Background
- Sum–product nets without units
- Sum–product nets with units
- Results and future work

Part 2

- Proofs

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

Sum–product logic

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 $f : X \to Y$

Sum–product logic

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 $f : X \to Y$

Additive linear logic

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

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

Free lattice completions of a poset $(P, \leq)$

$$x \;:=\; a \in P \;\mid\; \bot \;\mid\; \top \;\mid\; x \vee x \;\mid\; x \wedge x$$

Justifications that $x \leq y$

Idiosyncrasies 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

$$\frac{}{A \xrightarrow{a} B} \qquad \frac{}{\mathbf{0} \xrightarrow{?} X} \qquad \frac{}{X \xrightarrow{!} \mathbf{1}}$$

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

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

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

Cut elimination / subformula property

Whitman's Theorem for free lattices (1940s)

e.g.: $u \wedge v \;\leq\; x \vee y$ only if

$$\begin{array}{lllll} u \leq x \vee y & \text{or} & v \wedge u \leq x & \text{or} \\ v \leq x \vee y & \text{or} & v \wedge u \leq y & \end{array}$$

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

$$\begin{tikzcd} V_0 \times V_1 \arrow[r, "\pi_i"'] \arrow[rr, bend left, "h"] & V_i \arrow[r, "k"] \arrow[rr, bend right, "g"'] & X_j \arrow[r, "\iota_j"] & X_0 + X_1 \end{tikzcd}$$

Softness

Joyal: Free Bicompletions of Categories (1995)

For any (small) diagrams $D : I \longrightarrow \mathcal{C}$ and $E : J \longrightarrow \mathcal{C}$:

$$
\begin{tikzcd}
\operatorname*{colim}_{I \times J}(\hom(D^{op}, E)) \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner", very near end] & \operatorname*{colim}_{J}(\hom(\lim_{I} D, E)) \arrow[d] \\
\operatorname*{colim}_{I}(\hom(D^{op}, \operatorname*{colim}_{J} E)) \arrow[r] & \hom(\lim_{I} D, \operatorname*{colim}_{J} E)
\end{tikzcd}
$$

Proof identity

Proofs equal up to permutations denote the same morphism

$$\dfrac{\dfrac{X_1 \xrightarrow{f} Y_0}{X_0 \times X_1 \xrightarrow{f \circ \pi_1} Y_0}}{X_0 \times X_1 \xrightarrow{\iota_0 \circ (f \circ \pi_1)} Y_0 + Y_1} \quad = \quad \dfrac{\dfrac{X_1 \xrightarrow{f} Y_i}{X_1 \xrightarrow{\iota_i \circ f} Y_0 + Y_1}}{X_0 \times X_1 \xrightarrow{(\iota_0 \circ f) \circ \pi_1} Y_0 + Y_1}$$

$$\dfrac{\dfrac{}{\mathbf{0} \xrightarrow{?} Y_0} \quad \dfrac{}{\mathbf{0} \xrightarrow{?} Y_1}}{\mathbf{0} \xrightarrow{\langle ?, ? \rangle} Y_0 \times Y_1} \quad = \quad \dfrac{}{\mathbf{0} \xrightarrow{?} Y_0 \times Y_1}$$

Proof identity

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

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

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

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

$$?_{\mathbf{1}} = !_{\mathbf{0}}$$

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

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

Cut-free proofs up to these permutations denote the same categorical morphism—and proof identity is decidable.

Proof identity

Cockett and Santocanale (2009):

Proof identity for sum–product logic is tractable

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

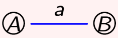
$$\mathcal{O}((hgt(X) + hgt(Y)) \times |X| \times |Y|)$$

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

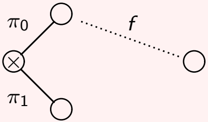
Proof nets (without units)

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

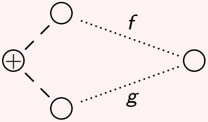
$$\frac{}{A \xrightarrow{a} B}$$



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



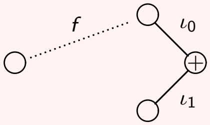
$$\frac{X_0 \xrightarrow{f} Y \quad X_1 \xrightarrow{g} Y}{X_0 + X_1 \xrightarrow{[f,g]} Y}$$



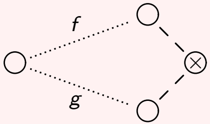
Proof nets (without units)

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

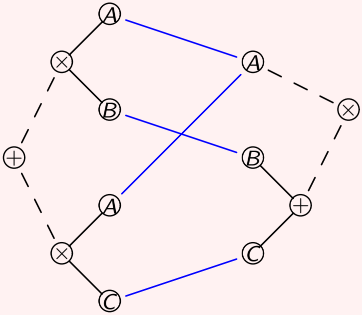
$$\frac{X \xrightarrow{f} Y_i}{X \xrightarrow{\iota_i \circ f} Y_0 + Y_1}$$



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

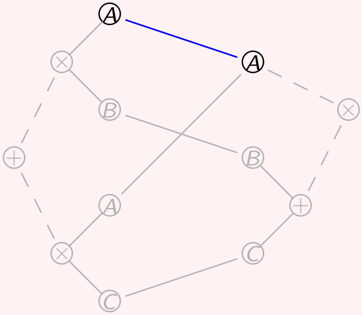


Example: construction



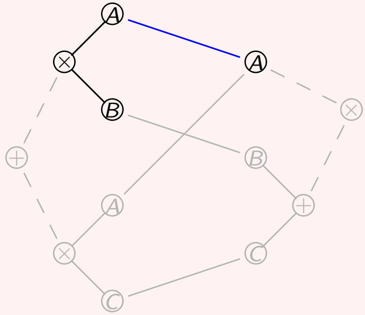
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



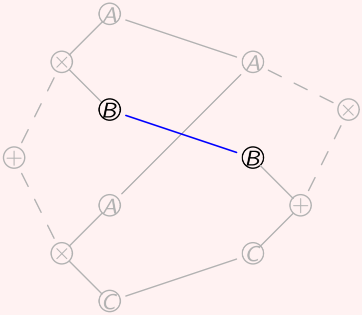
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



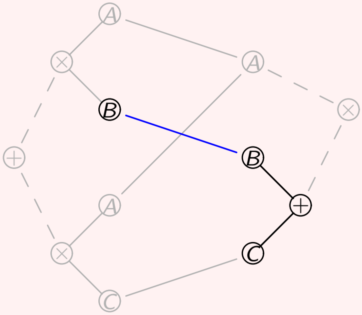
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



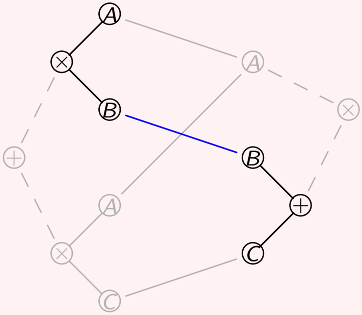
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



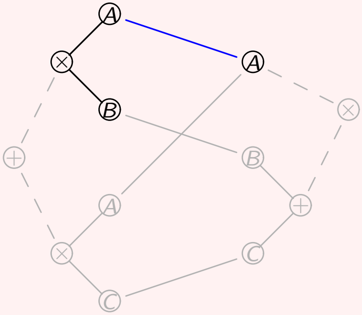
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



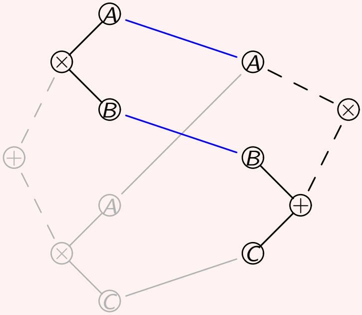
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



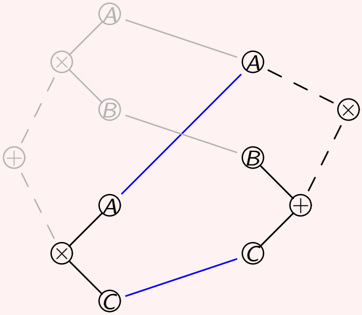
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



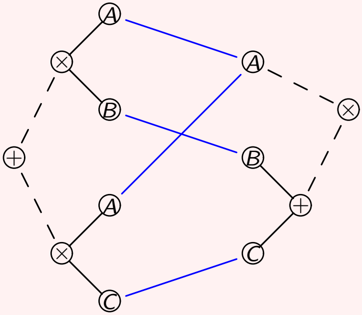
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: construction



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

Example: construction



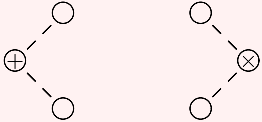
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Switching

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

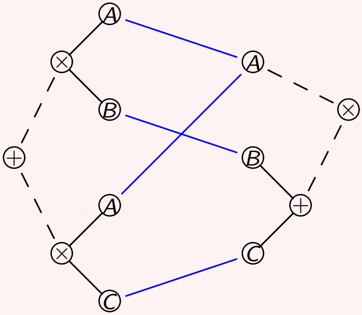
- a source object X
- a target object Y
- a labelled relation R from the leaves in X to the leaves in Y

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



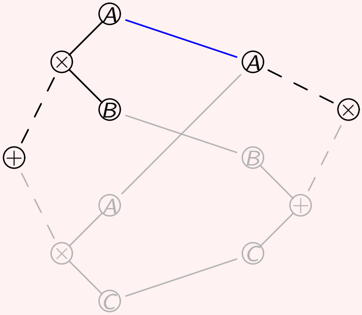
After choosing one branch for each coproduct in X and each product in Y there must be exactly one path from left to right.

Example: switching



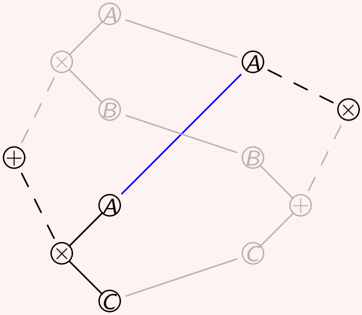
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



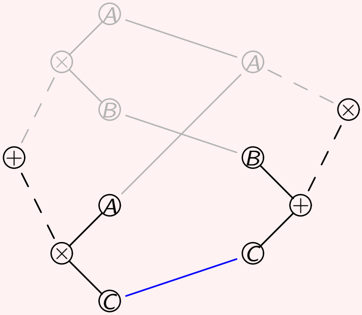
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



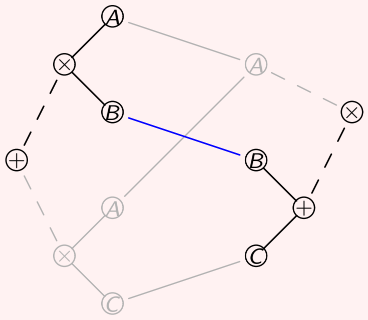
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Example: switching



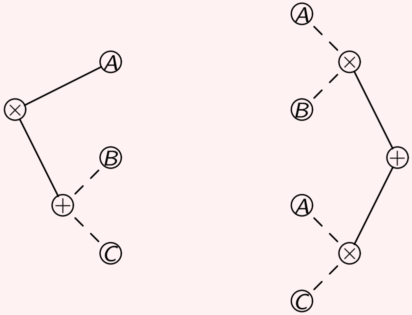
$$(A \times B) + (A \times C) \;\longrightarrow\; A \times (B + C)$$

Example: switching



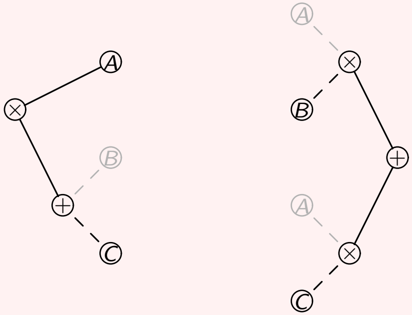
$$(A \times B) + (A \times C) \longrightarrow A \times (B + C)$$

Non-example: switching



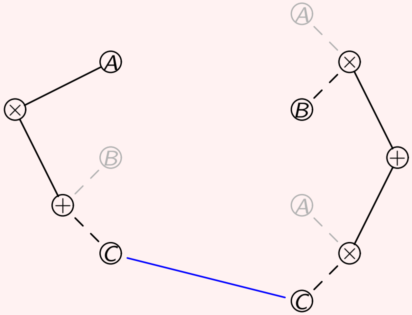
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



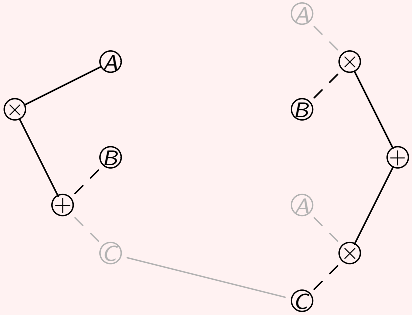
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



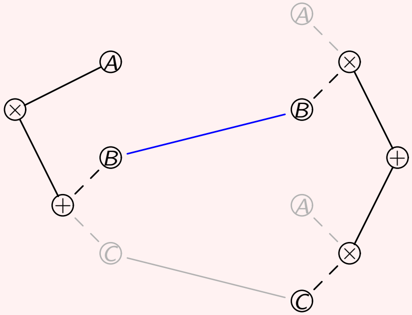
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



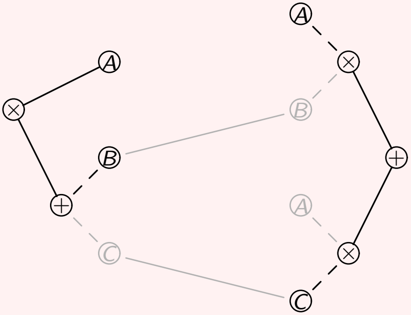
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



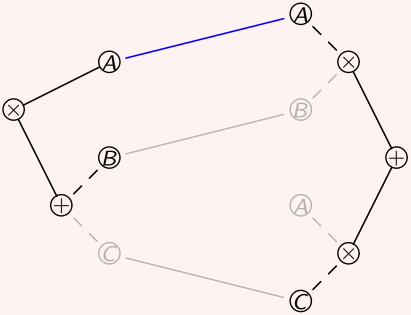
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



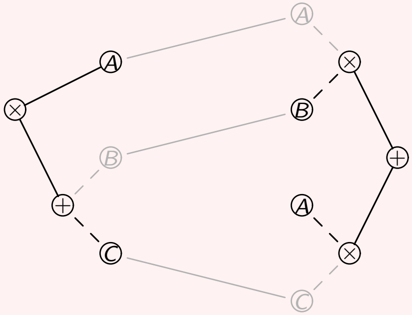
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



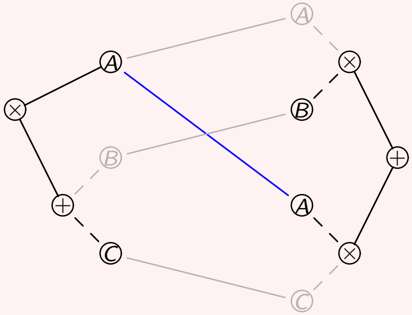
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



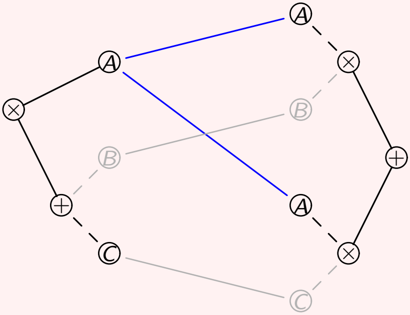
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching



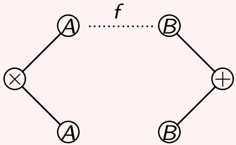
$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

Non-example: switching

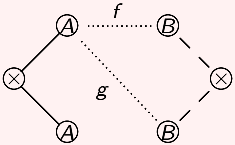


$$A \times (B + C) \longrightarrow (A \times B) + (A \times C)$$

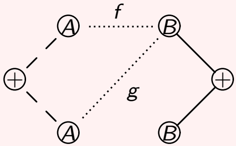
Equalities factored out



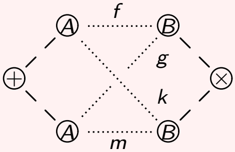
$$\iota_0 \circ (f \circ \pi_0) = (\iota_0 \circ f) \circ \pi_0$$



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



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

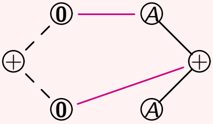


$$\langle [f, g], [k, m] \rangle = [\langle f, k \rangle, \langle g, m \rangle]$$

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.
These (unlabelled) links are added:

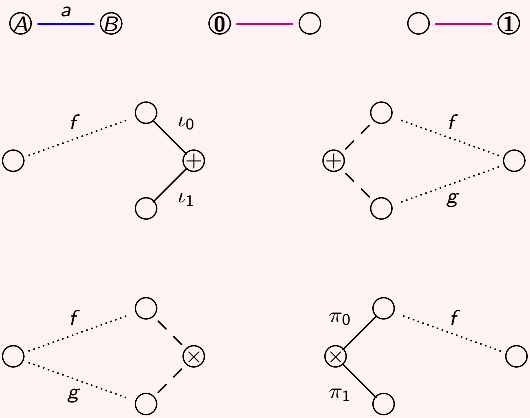
Links are no longer restricted to the leaves. For example:



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

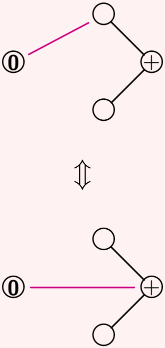
$$\mathbf{0} \underset{!}{\overset{?}{\longrightarrow}} \mathbf{1}$$

The full net calculus



The unit equations

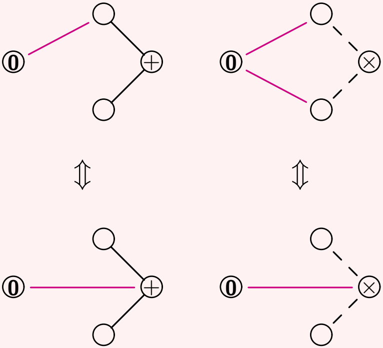
$\iota_i \circ ? = ?$



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

The unit equations

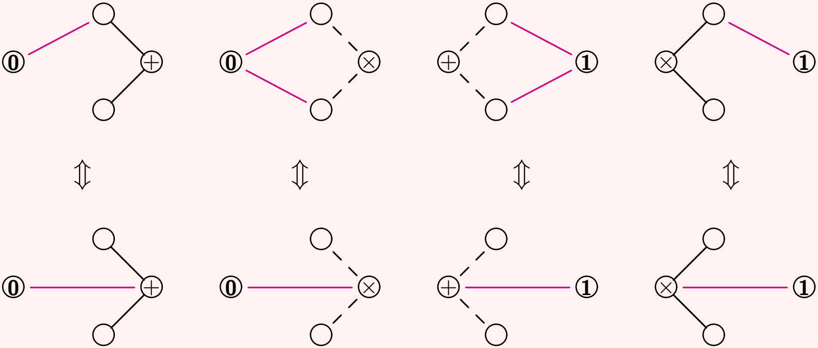
$\iota_i \circ ? = ?$ $\qquad$ $\langle ?, ? \rangle = ?$



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

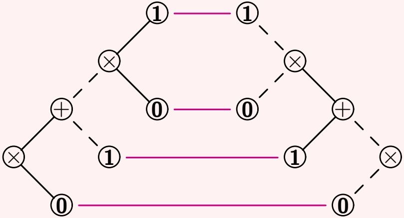
The unit equations

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

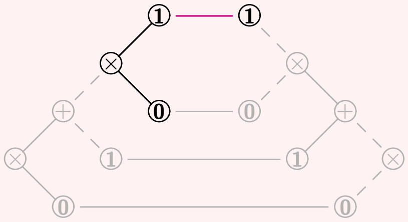


. . . define an equational theory ($\Leftrightarrow$) over nets, via graph rewriting

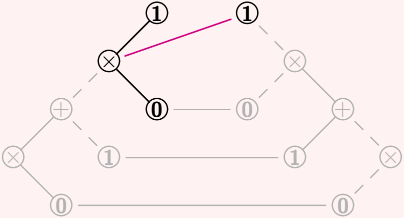
Example



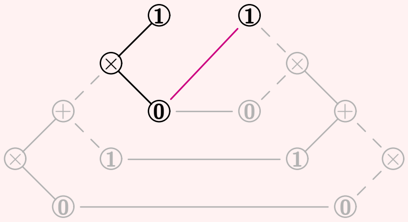
Example



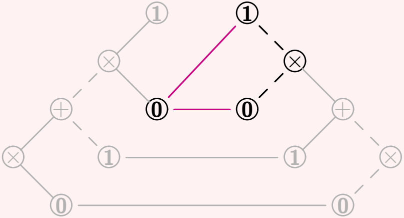
Example



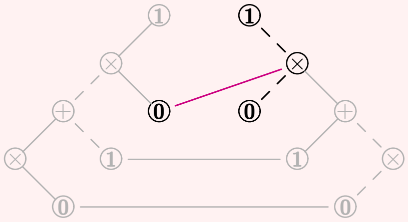
Example



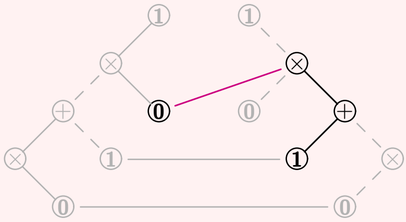
Example



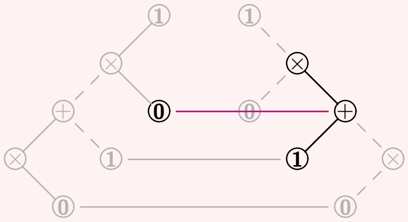
Example



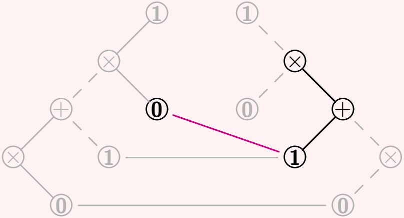
Example



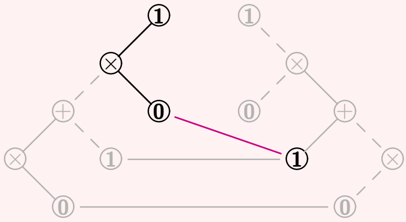
Example



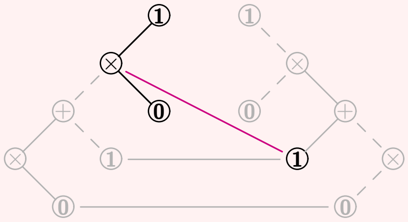
Example



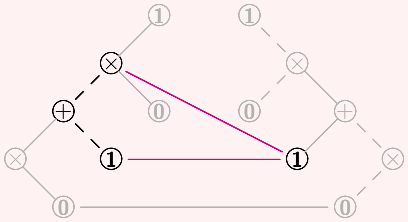
Example



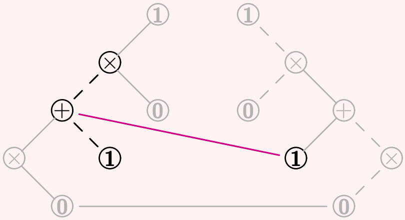
Example



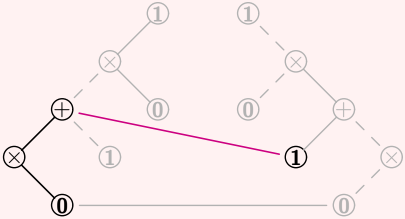
Example



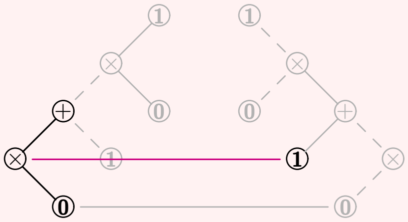
Example



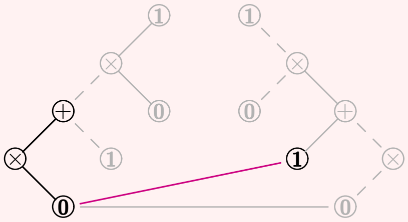
Example



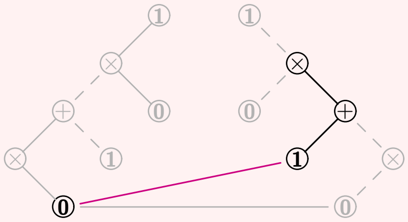
Example



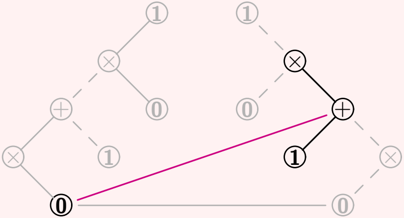
Example



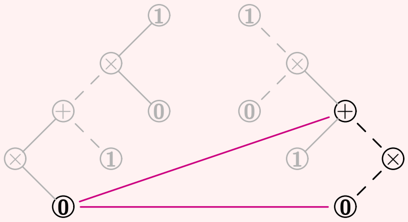
Example



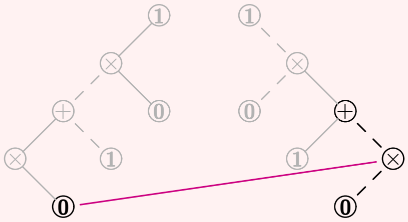
Example



Example



Example



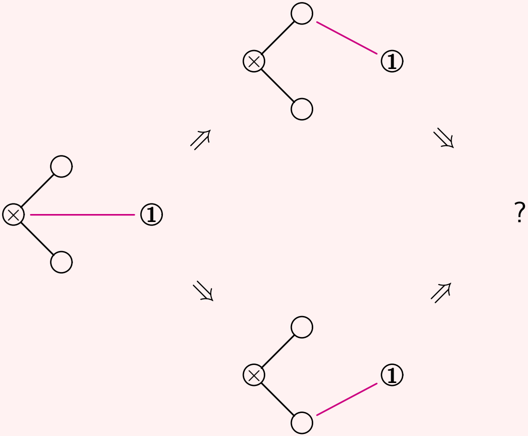
The problem

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

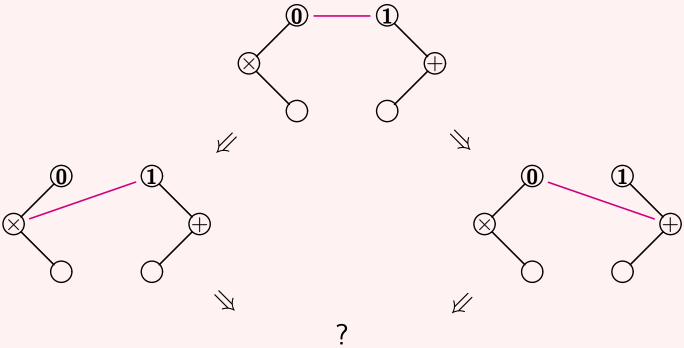
A standard approach is to rewrite towards a normal form, using a confluent and terminating rewrite relation.

The first question is then whether restricting the equivalences of ($\Leftrightarrow$) to a single direction can provide a suitable rewrite relation.

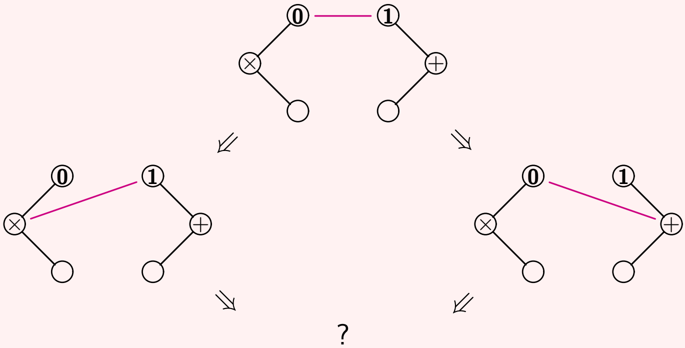
Rewriting towards the leaves



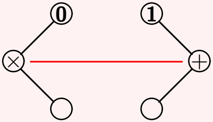
Rewriting towards the roots



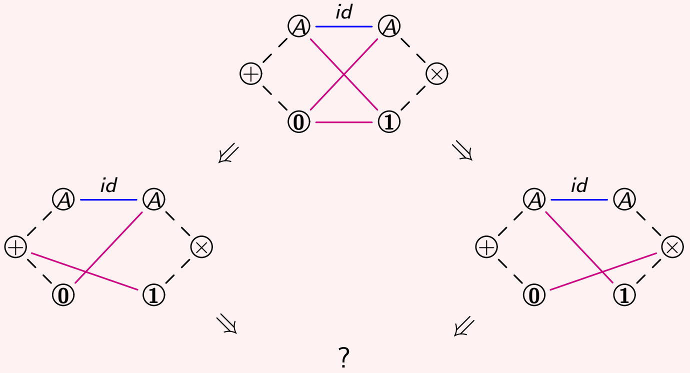
Rewriting towards the roots



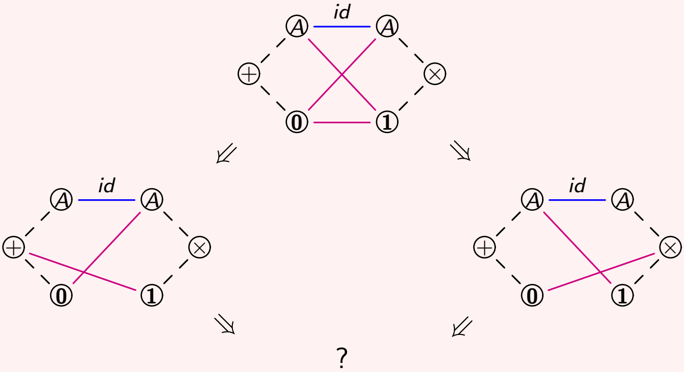
A first attempt at a solution: a new type of link



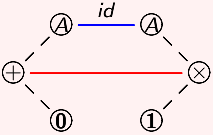
Rewriting towards the roots



Rewriting towards the roots

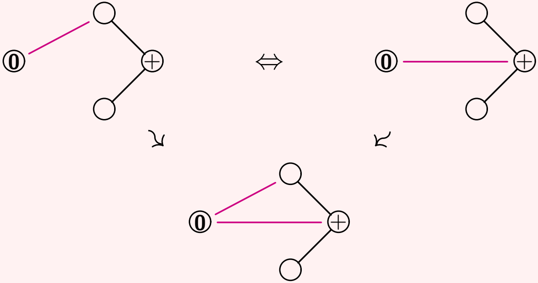


The following breaks the switching condition (and makes no sense)



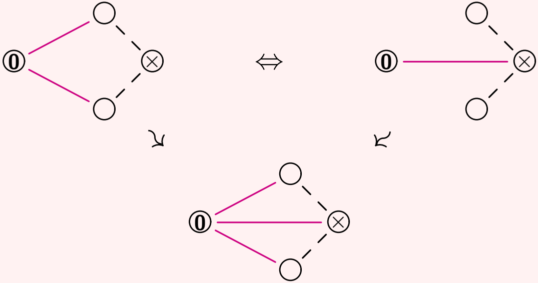
The solution

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



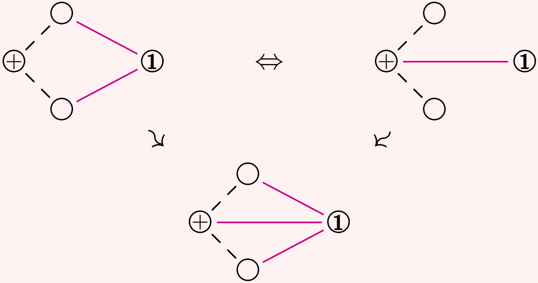
The solution

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



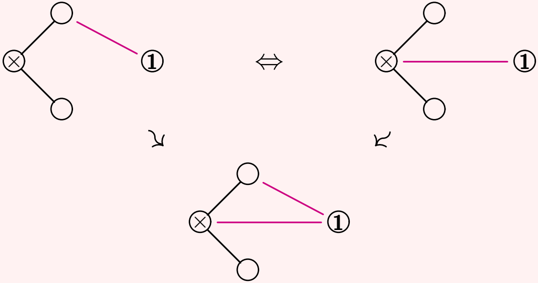
The solution

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

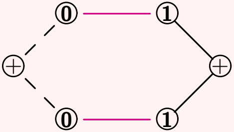


The solution

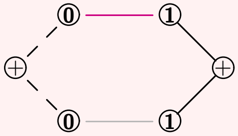
Confluent rewriting seems impossible without breaking the switching condition. So: break it. Then there is a simple confluent and normalising rewrite relation: saturation ($\rightsquigarrow$).



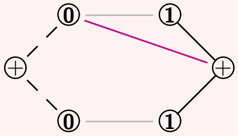
Example



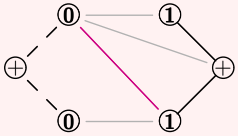
Example



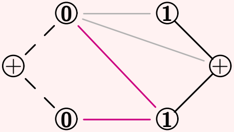
Example



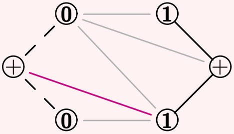
Example



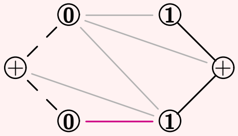
Example



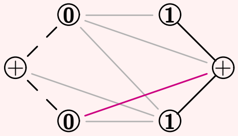
Example



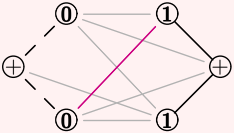
Example



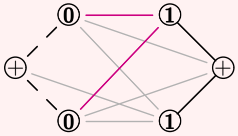
Example



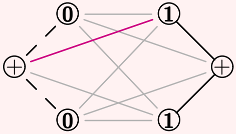
Example



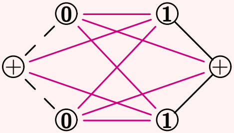
Example



Example



Example



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

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

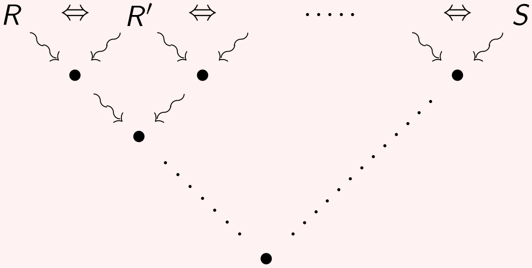
Write $X \xrightarrow{\sigma R} Y$ for the normal form (the saturation) of a net $X \xrightarrow{R} Y$ and call it a saturated net

Results

Saturation gives a decision procedure for 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 hard part

Saturated nets

A saturated net $X \xrightarrow{\sigma R} Y$ combines the links of all equivalent nets

$$\sigma R \;=\; \bigcup \{\, S \;|\; X \xrightarrow{S} Y \;\Leftrightarrow\; X \xrightarrow{R} Y \}$$

Saturated nets

A saturated net $X \xrightarrow{\sigma R} Y$ combines the links of all equivalent nets

$$\sigma R \;=\; \bigcup \{\, S \mid X \xrightarrow{S} Y \;\Leftrightarrow\; X \xrightarrow{R} Y \}$$

call links occurring in the same saturation step neighbours, and an equivalence class of neighbouring links a neighbourhood

Correctness: (tentative) relation of links $R \subseteq X \times Y$ is a saturated net if and only if it is saturated, and for every switching the links switched on form a non-empty neighbourhood.

Saturated nets

A saturated net $X \xrightarrow{\sigma R} Y$ combines the links of all equivalent nets

$$\sigma R \;=\; \bigcup \{\, S \;|\; X \xrightarrow{S} Y \;\Leftrightarrow\; X \xrightarrow{R} Y \}$$

call links occurring in the same saturation step neighbours, and an equivalence class of neighbouring links a neighbourhood

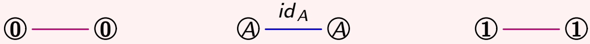
Correctness: (tentative) relation of links $R \subseteq X \times Y$ is a saturated net if and only if it is saturated, and for every switching the links switched on form a non-empty neighbourhood.

Morally, this is a requirement for evidence that all maps expressed in a net commute.

The category of saturated nets

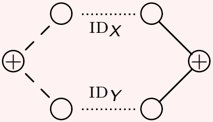
The category of saturated nets is the free completion with finite (nullary and binary) products and coproducts of a base category $\mathcal{C}$.

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

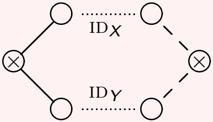


$id_{\mathbf{0}} = ?_{\mathbf{0}}$

$id_1 = !_{\mathbf{1}}$



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



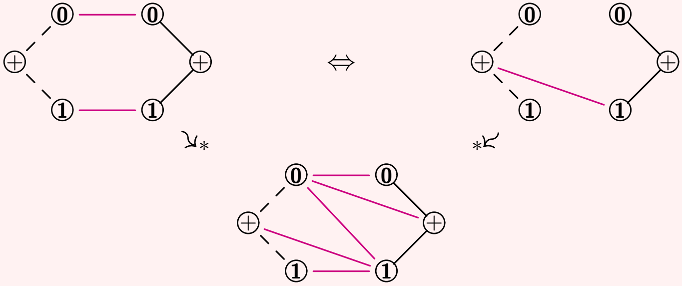
$id_{X \times Y} = \langle id_X \circ \pi_0, id_Y \circ \pi_1 \rangle$

The category of saturated nets

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

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

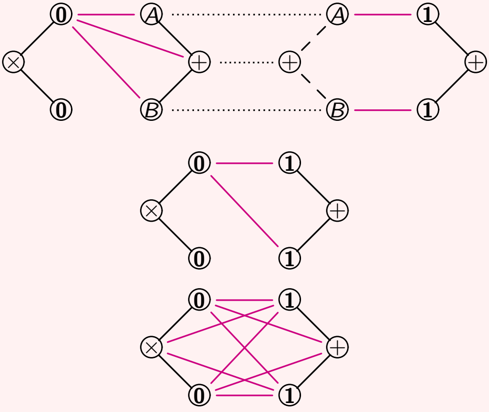
Saturation is necessary: nets ID_X are equivalent to other nets.



The category of saturated nets

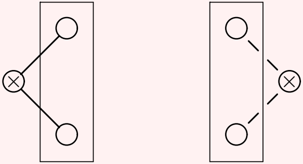
The category of saturated nets is the free completion with finite (nullary and binary) products and coproducts of a base category $\mathcal{C}$.

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



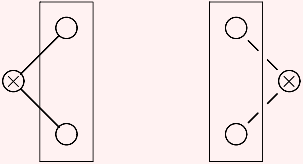
Future work: bicompletions

For products, these are the diagrams

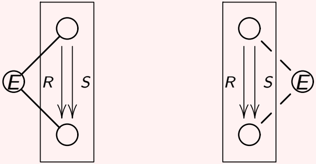


Future work: bicompletions

For products, these are the diagrams



Possibly, equalisers can be added in the following way



Questions?