

From Proof Nets to Geometry of Interaction

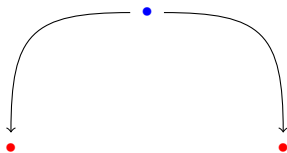
Thomas Seiller

LAMA — Université de Savoie

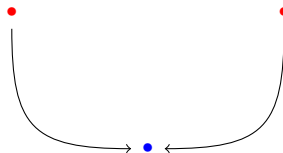
Proof Theory: Linear Logic, Ludics and Geometry of Interaction
August, 29th 2012 — Paraty, Brazil

Remarks on Proof Structures

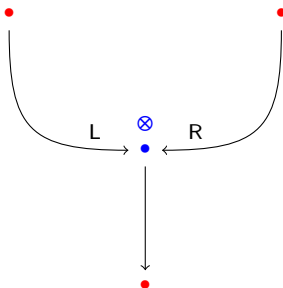
We begin with some remarks on proof structures. We will consider only atomic axiom links and we usually won't write the formulas explicitly in the figures.



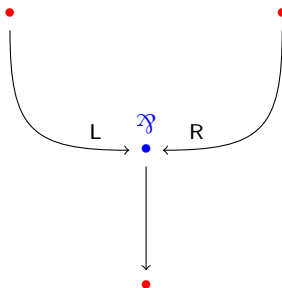
(a) Axiom Brick



(b) Cut Brick



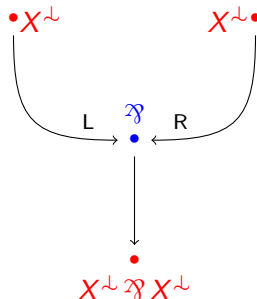
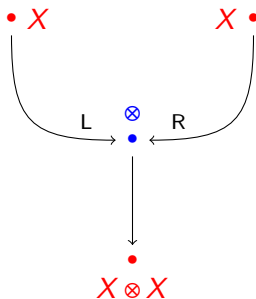
(c) Tensor Brick



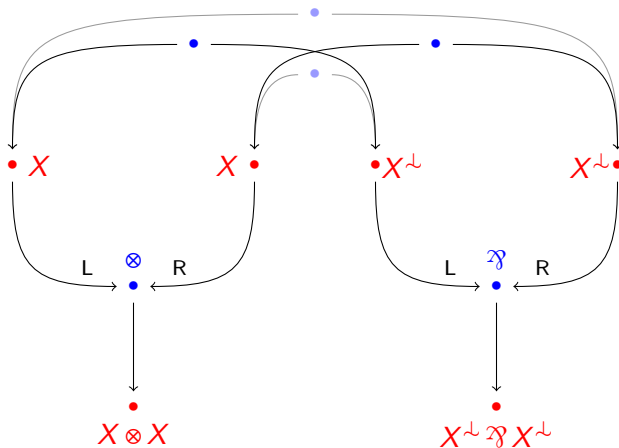
(d) Parr Brick

Figure : Basic bricks

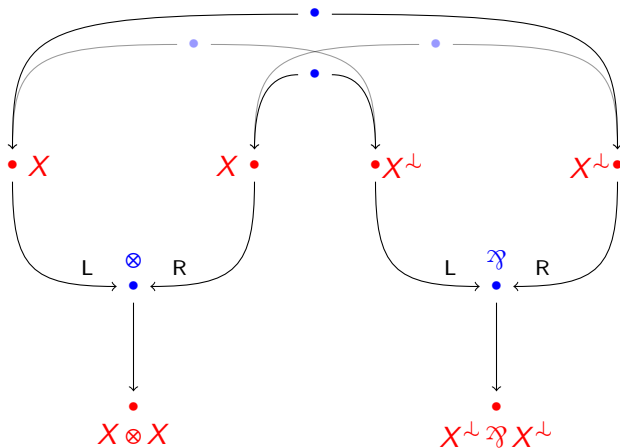
Proofs as sets of axioms



Proofs as sets of axioms



Proofs as sets of axioms



Given a proof structure $\mathcal{R}$, we therefore distinguish between:

- the axiomatic part of $\mathcal{R}$ — denoted by $\mathcal{R}_{ax}$ — which is the set of axiom bricks of $\mathcal{R}$;
- the typing part of $\mathcal{R}$ — denoted by $\mathcal{R}_{type}$ — which is the set of $\otimes$, $\wp$ and cut bricks of $\mathcal{R}$;

We will therefore consider a proof structure as a couple of:

- $\mathcal{R}_{ax}$: an "untyped proof" (like a pure lambda-term);
- $\mathcal{R}_{type}$: a type (it is a formula tree);

Given a proof structure $\mathcal{R}$:

- 1 we define a finite set S of objects depending on the typing part $\mathcal{R}_{type}$ of $\mathcal{R}$: trips (Longtrips criterion), graphs (DR);
- 2 we show that $\mathcal{R}$ is sequentializable if and only if all elements of S interact properly with the axiomatic part $\mathcal{R}_{ax}$ of $\mathcal{R}$: the resulting trip is long (Longtrips), the resulting graph is connected and acyclic (DR).

The correctness criteria can therefore be seen as a way of checking if the untyped proof $\mathcal{R}_{ax}$ can be given the type corresponding to $\mathcal{R}_{type}$.

Cut-Elimination

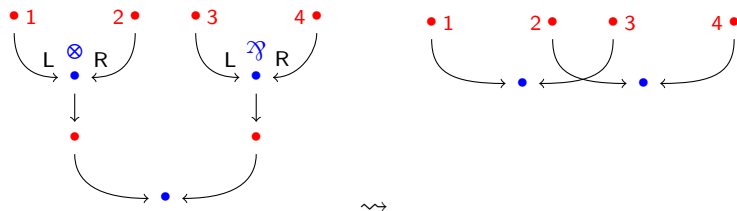


Figure : Cut-Elimination: $\otimes - \mathcal{Y}$ case

Cut Elimination

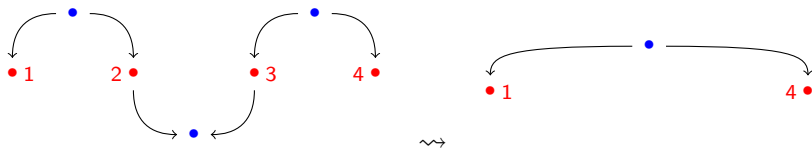


Figure : Cut-Elimination: $ax - ax$ case

We can choose the following reduction "strategy" (it is in fact a family of strategies):

- First eliminate all $\otimes$ - $\wp$ cuts: this step does nothing more than associating (identifying) atoms of the cut formulas;
- Then eliminate ax - ax cuts: this is where the real computation takes place, the atoms of the cut formulas are erased and sequences of axioms and cuts are replaced by axioms.

Example

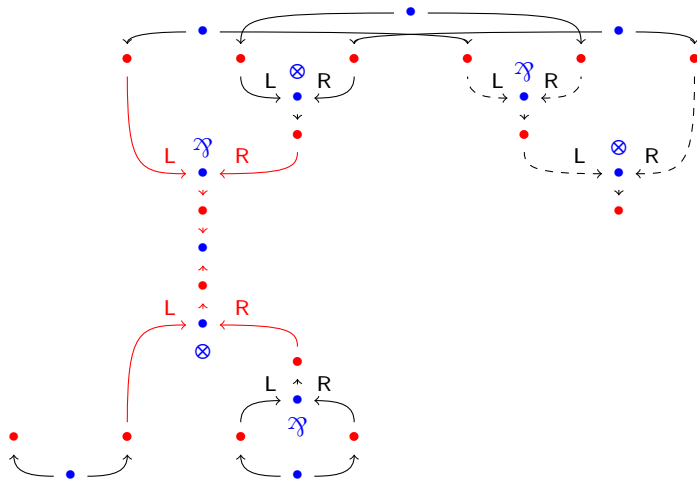


Figure : Example

Example

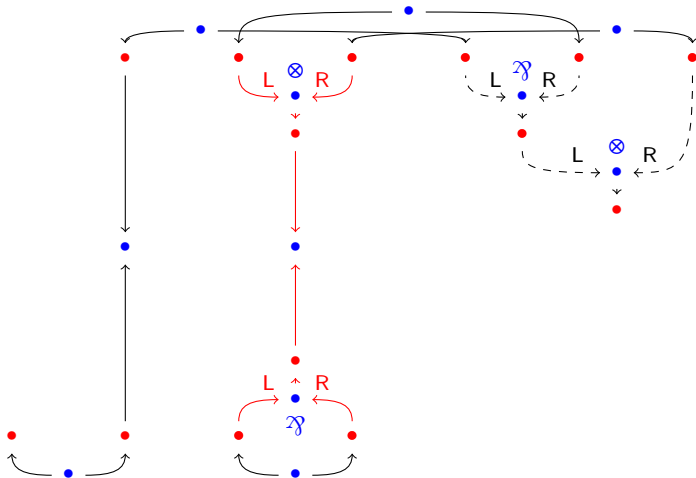


Figure : Example

Example

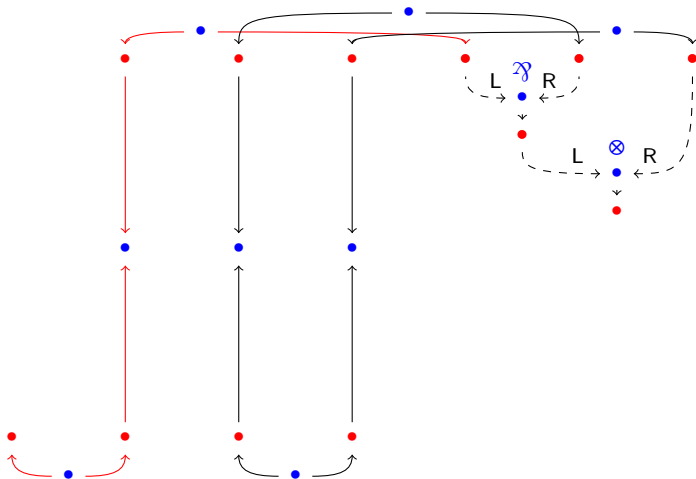


Figure : Example

Example

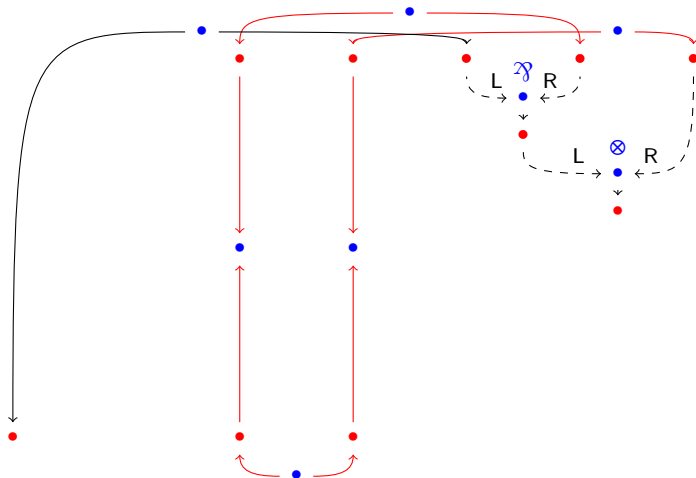


Figure : Example

Example

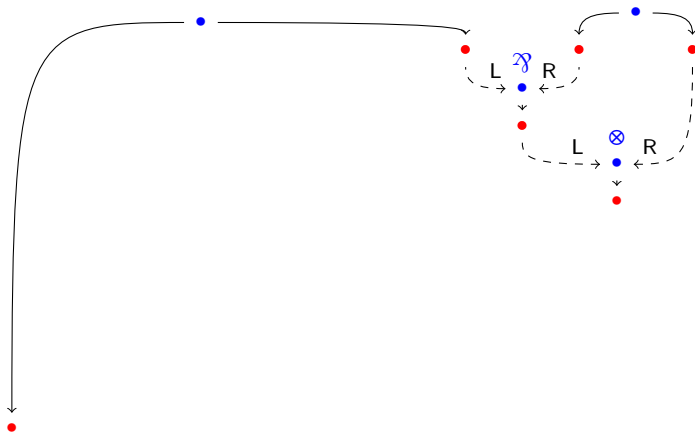


Figure : Example

LongTrips and Permutations

LongTrips criterion and permutations

We briefly recall the longtrips criterion.

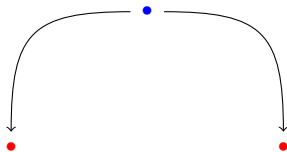
Definition

A switching is a map from the set of $\otimes$ and $\wp$ vertices to the set $\{1,2\}$.

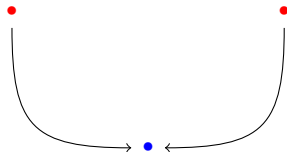
Define $\mathcal{R}_\bullet$ to be the set of couples (v,w) where v is a formula vertex (a red vertex) and w is an element in $\{\uparrow, \downarrow\}$.

Definition

A switching s defines a graph $\mathcal{R}_s$ whose vertices is $\mathcal{R}_\bullet$ (defined in the next slides).

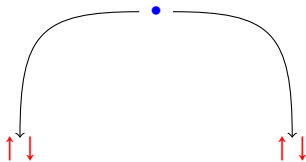


(a) Axiom Brick

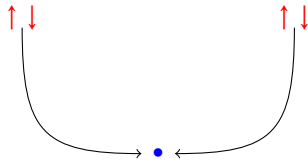


(b) Cut Brick

Figure : Axiom and Cut bricks

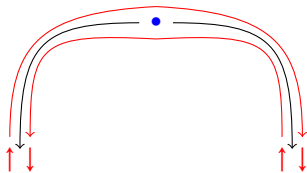


(a) Axiom Brick

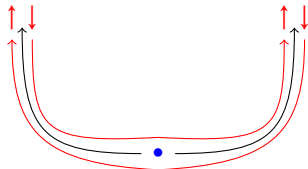


(b) Cut Brick

Figure : Axiom and Cut bricks



(a) Axiom Brick



(b) Cut Brick

Figure : Directions for Axiom and Cut bricks

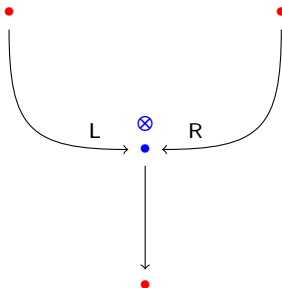


Figure : Tensor bricks

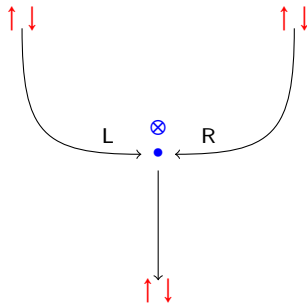
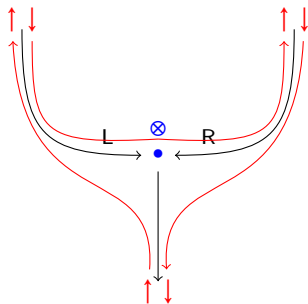
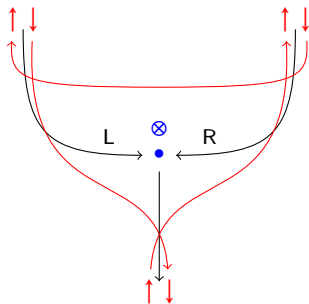


Figure : Tensor bricks

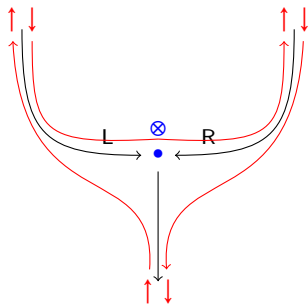


(a) First switching

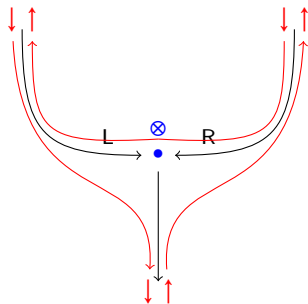


(b) Second switching

Figure : Directions for Tensor bricks



(a) First switching



(b) Second switching

Figure : Directions for Tensor bricks

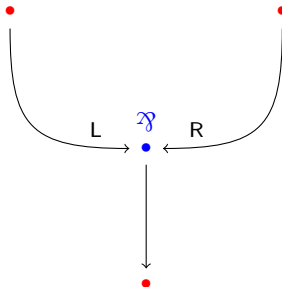
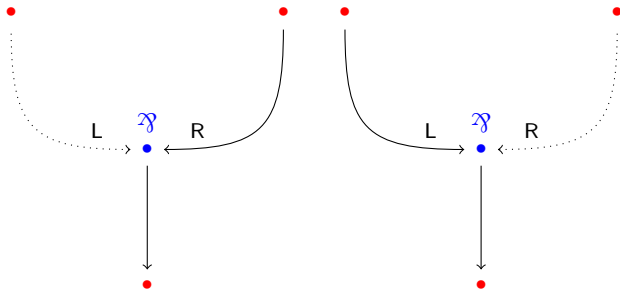


Figure : Parr bricks



(a) First switching

(b) Second switching

Figure : Parr bricks

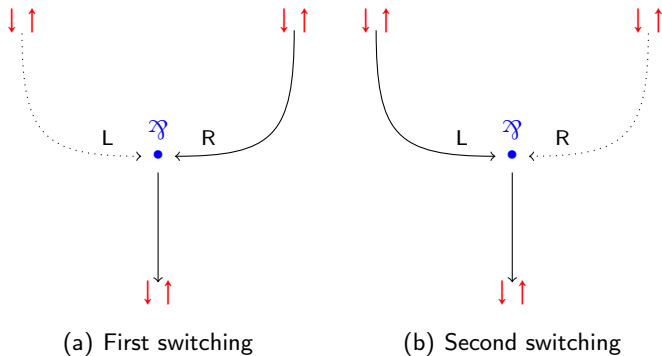


Figure : Directions for Parr bricks

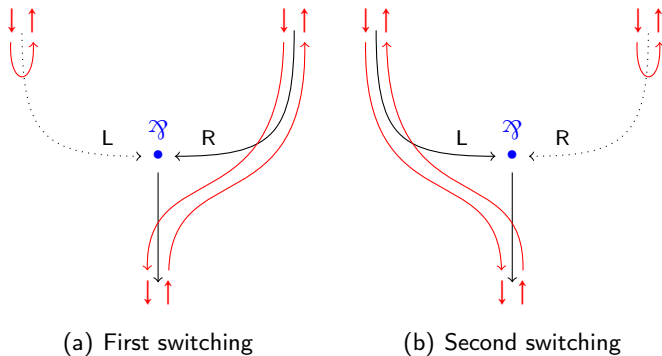


Figure : Directions for Parr bricks

Theorem

A proof structure $\mathcal{R}$ is sequentializable if and only if for all switching s the graph $\mathcal{R}_s$ defines a long trip: if v is a vertex of $\mathcal{R}_s$, the paths from v to v (by following the edges) go through all the edges.

Let $\mathcal{R}$ be a proof structure. A switching s defines a permutation τ_s over the axioms of the atoms in $\mathcal{R}$ as follows:

- chose an atom a , and consider $a_{\downarrow} = (a, \downarrow)$;
- follow the edges of $\mathcal{R}_s$ until you arrive at a vertex $(b, \uparrow)$;
- define $\tau_s(a) = b$.

Similarly, the axiomatic part $\mathcal{R}_{ax}$ of $\mathcal{R}$ defines a permutation σ (which is a disjoint union of transpositions).

Example

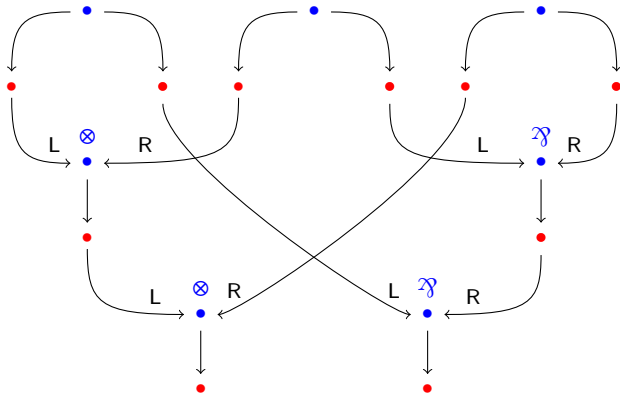


Figure : Associativity: Proof Net

Example

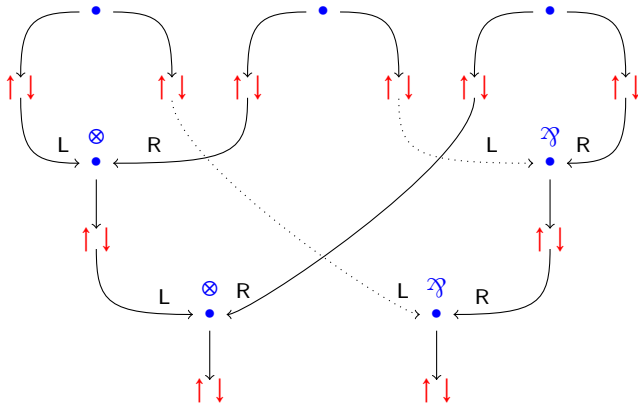


Figure : Associativity: Proof Net

Example

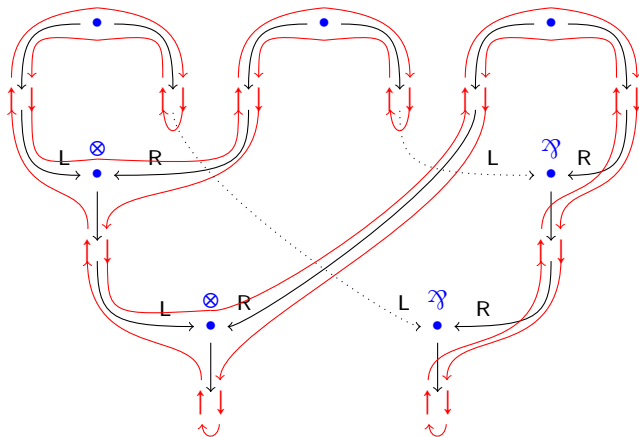


Figure : Associativity: A (Long) Trip

Example

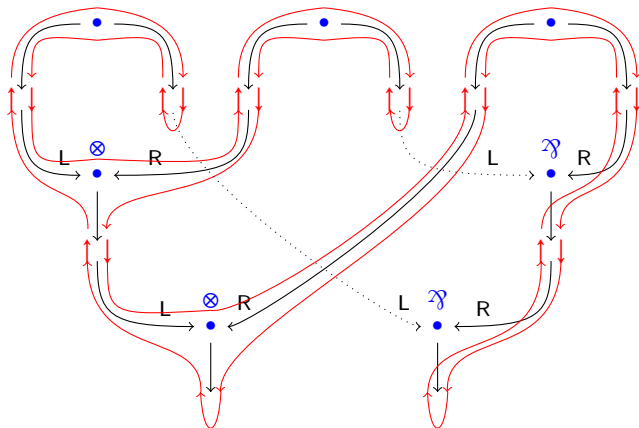


Figure : Associativity: A (Long) Trip

Example

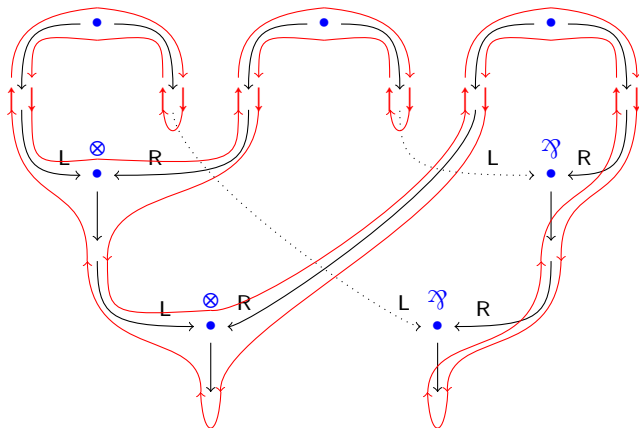


Figure : Associativity: A (Long) Trip

Example

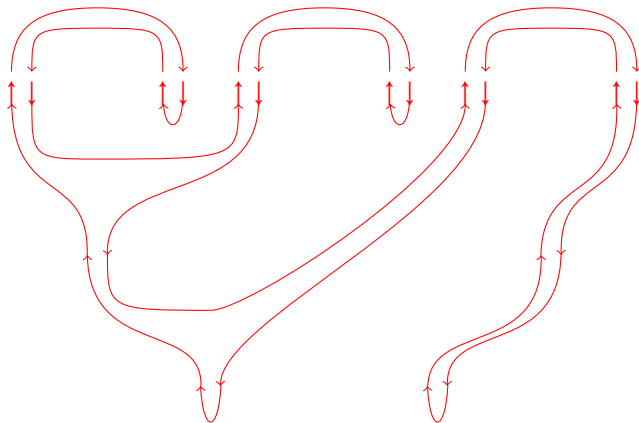


Figure : Associativity: Towards Permutations

Example

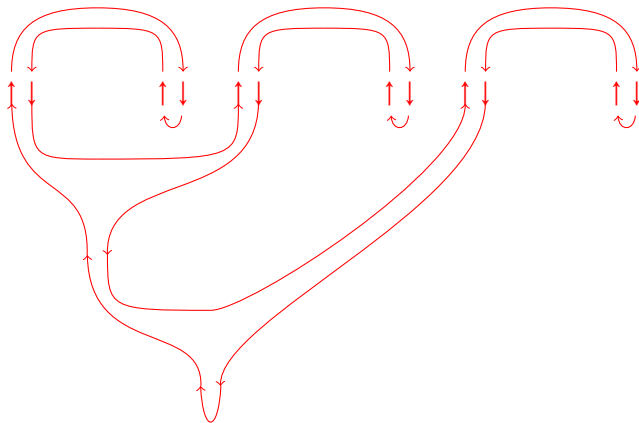


Figure : Associativity: Towards Permutations

Example

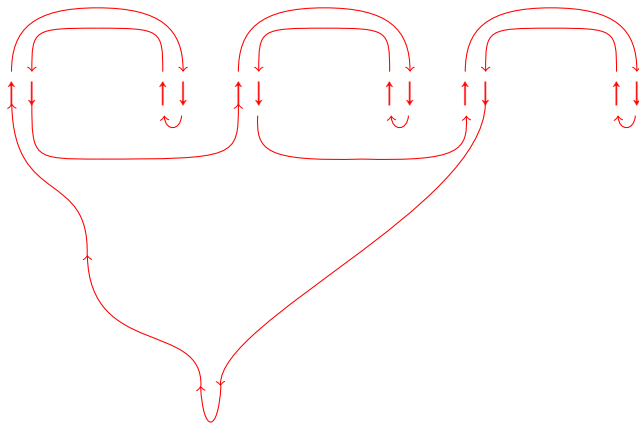


Figure : Associativity: Toward Permutations

Example

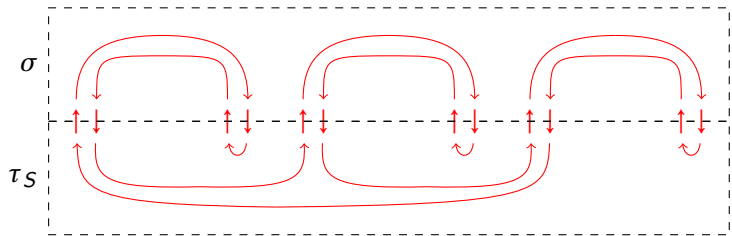


Figure : Associativity: Permutations

Example

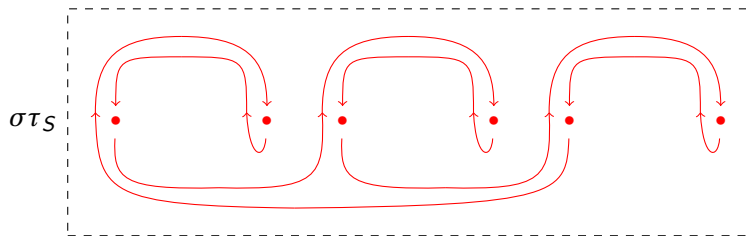


Figure : Associativity: Permutations

We get a reformulation of the correctness criterion.

Theorem

A proof structure $\mathcal{R}$ is sequentializable if and only if for all switching s , the permutation $\sigma\tau_s$ is cyclic.

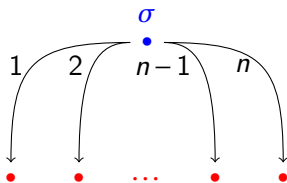
Switchings of $A = \text{Proofs of } A^\downarrow$

Extending the syntax

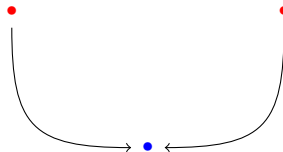
In order to understand better this point of view, we extend the syntax of proof structures. We replace the axiom brick by a *daimon brick*, which allows the introduction of an arbitrary (finite) number of atoms. The daimon brick is given with a cyclic permutation of the atoms it introduces.

Remark

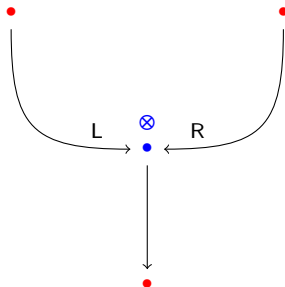
Since permutations can always be written as a product of cycles, the daimon brick allows us to write any permutation. The set of untyped proofs is therefore extended to all permutations (instead of only disjoint unions of transpositions).



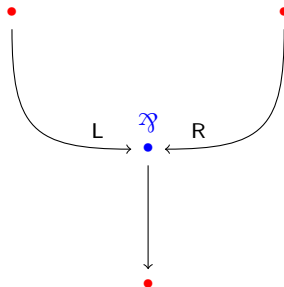
(a) Daimon Brick



(b) Cut Brick



(c) Tensor Brick



(d) Parr Brick

Figure : Basic bricks

Proofs as orthogonals

Given a formula A we define the set $\mathcal{S}(A)$ to be the set of permutations induced by the switchings of A .

Definition

Two permutations σ, τ are said to be *orthogonal* — denoted $\sigma \perp \tau$ — when $\sigma\tau$ is cyclic.

If A is a set of permutations, we write $A^\perp$ the set of permutations that are orthogonal to every element of A .

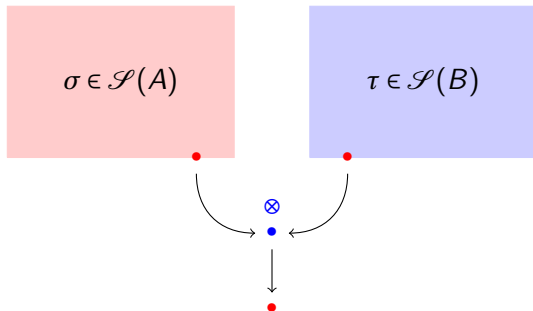
Definition

The set of *paraproofs* of A is the set $\mathcal{P}(A) = (\mathcal{S}(A))^\perp$.

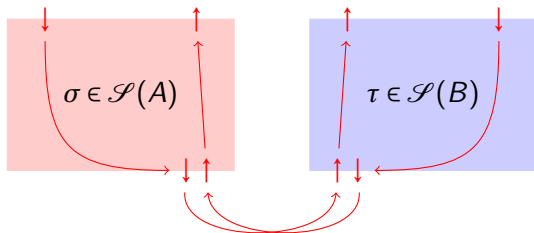
Remark

Both $\mathcal{P}(A)$ and $\mathcal{P}(A^\perp)$ are non-empty.

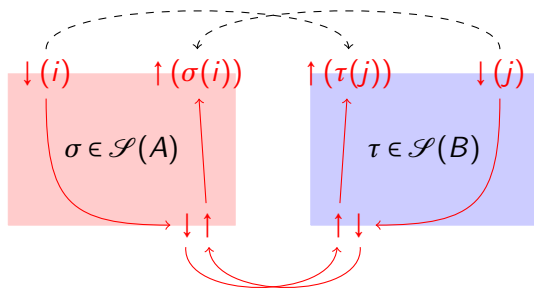
Switchings of $A \otimes B$



Switchings of $A \otimes B$



Switchings of $A \otimes B$



We denote by $\sigma \mathcal{N}_i^j \tau$ the resulting permutation. It can be defined as $(\sigma(i) \tau(j)) \circ (\sigma \cup \tau)$.

Switchings of $A \otimes B$

Since a switching of $A^\perp \wp B^\perp$ is the union of a switching of $A^\perp$ and a switching of $B^\perp$, we obtain that the permutations $\sigma \wp_i^j \tau$ are orthogonal to all the elements in $\mathcal{S}(A^\perp \wp B^\perp)$.

Proposition

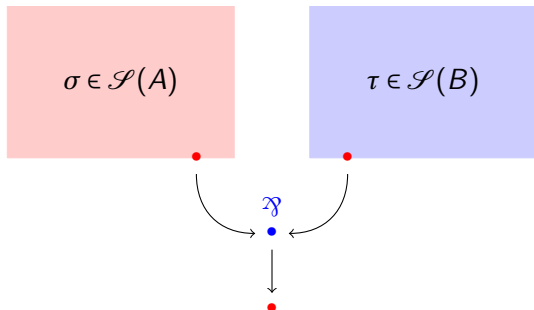
We have the inclusion $\mathcal{S}(A \otimes B) \subset \mathcal{P}(A^\perp \wp B^\perp)$.

Since for all atom a there exists a switching that links a with the conclusion of the proof structure, given a and b two atoms of A and B , there exists an element $\sigma \wp_i^j \tau$ such that $\sigma \wp_i^j \tau(a) = b$. Thus:

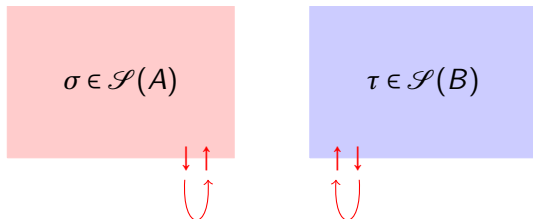
Proposition

$$\mathcal{P}(A \otimes B) = \{\sigma \cup \tau \mid \sigma \in \mathcal{P}(A), \tau \in \mathcal{P}(B)\}$$

Switchings of $A \wp B$



Switchings of $A \wp B$



So switchings of $A \wp B$ are of the form $\sigma \cup \tau$ for $\sigma \in \mathcal{S}(A)$ and $\tau \in \mathcal{S}(B)$.
As a consequence:

Proposition

$$\mathcal{S}(A \wp B) \subset \mathcal{P}(A^\perp \otimes B^\perp)$$

Switchings of $A \wp B$

Using the fact that $\mathcal{P}(A^\perp \otimes B^\perp) = \{\sigma \cup \tau \mid \sigma \in \mathcal{P}(A^\perp), \tau \in \mathcal{P}(B^\perp)\}$, we even have:

Proposition

$$\mathcal{S}(A \wp B) = \mathcal{P}(A^\perp \otimes B^\perp)$$

From the preceding results and the definition of $\mathcal{P}$, we obtain:

$$\begin{aligned}\mathcal{P}(A^\perp \wp B^\perp) &= (\mathcal{S}(A^\perp \wp B^\perp))^\perp \\ &= (\mathcal{P}(A^\perp \otimes B^\perp))^\perp \\ &= (\mathcal{S}(A \otimes B))^{\perp\perp}\end{aligned}$$

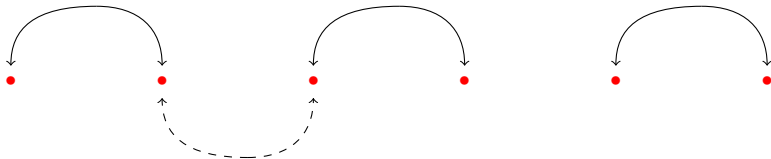
Multiplicatives

Based on all these remarks, we will now build an interpretation of multiplicative linear logic where proofs are represented by permutations. The paraproofs (generalized untyped proofs) will be given by permutations of a finite set.

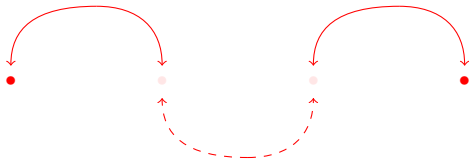
Definition

A *paraproof* is a couple (X, σ) of a finite set X and a permutation $\sigma \in \mathfrak{S}(X)$.

Multiplicatives: execution



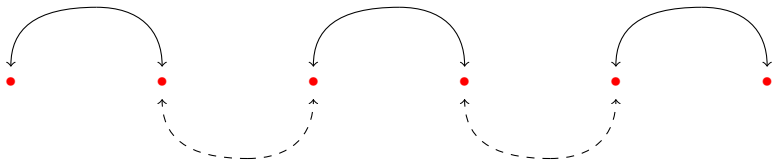
Multiplicatives: execution



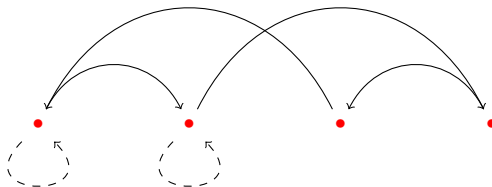
Multiplicatives: execution



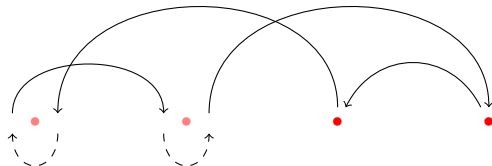
Multiplicatives: execution



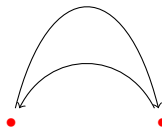
Multiplicatives: execution



Multiplicatives: execution



Multiplicatives: execution



Multiplicatives: execution

Definition

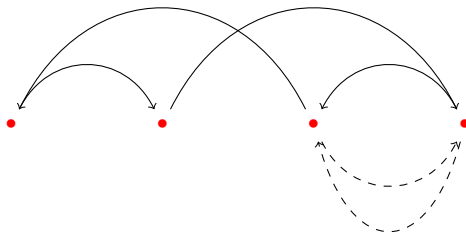
The *execution* of $(X \cup Y, \sigma)$ and (X, τ) is equal to $(Y, \text{Ex}(\sigma, \tau))$, where:

$$\begin{aligned}\text{Ex}(\sigma, \tau) &= \bigcup_{i=0}^{\infty} [\sigma(\tau\sigma)^i]_{|Y} \\ &= \sigma_{|Y} \cup (\sigma\tau\sigma)_{|Y} \cup (\sigma\tau\sigma\tau\sigma)_{|Y} \cup \dots\end{aligned}$$

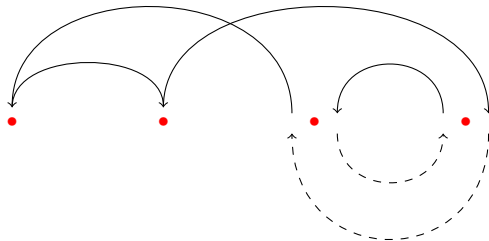
Exercise

Show $\text{Ex}(\sigma, \tau)$ is a permutation.

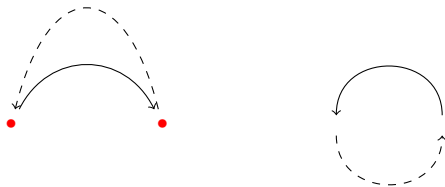
Multiplicatives: execution



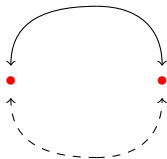
Multiplicatives: execution



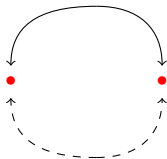
Multiplicatives: execution



Multiplicatives: execution



Multiplicatives: execution



There are two ways of dealing with this:

- Define execution only when no such *internal cycles* appear: this is what is done in *Multiplicatives*;
- Define execution in all cases and remember that such cycles appeared: this is what is done in the last version of Gol;

Multiplicatives: execution

We chose to work in the second case (execution defined in all cases) because:

- the only thing needed to work in this case is to add a *cycle counter* to paraproofs;
- the resulting framework is nicer: it is possible to deal with units and the orthogonality can be defined through the execution;

Definition

A *design* is a couple $\alpha = (a, p_a)$ where $a \in \mathbf{N}$ and $p_a = (X_a, \sigma_a)$ is a paraproof.

Multiplicatives: execution

Definition

Let (X, σ) and (Y, τ) be paraproof. We define:

$$\ll(X, \sigma), (Y, \tau)\gg = \text{Card}\{\text{cycles between } \sigma \text{ and } \tau\}$$

Definition

The execution $\text{Ex}(a, b)$ between two designs is equal to

$$(a + b + \ll p_a, p_b \gg, \text{Ex}(p_a, p_b))$$

Exercise

Show that execution is associative: if a, b, c are designs such that $X_a \cap X_b \cap X_c = \emptyset$,

$$\text{Ex}(a, \text{Ex}(b, c)) = \text{Ex}(\text{Ex}(a, b), c)$$

Multiplicatives: orthogonality

Definition

Two designs α and $\flat$ are orthogonal — written $\alpha \perp \flat$ — when $X_\alpha = X_\flat$ and:

$$a + b + \ll(X_\alpha, \sigma_\alpha), (X_\flat, \sigma_\flat)\gg = 1$$

Proposition

Two designs $\alpha, \flat$ are orthogonal if and only if $Ex(\alpha, \flat) = (1, (\emptyset, \emptyset))$.

Multiplicatives: Types

If A is a set of designs, we write $A^\perp$ the set:

$$\{b \mid \forall a \in A, a \perp b\}$$

Definition

A *type* $\mathbf{A}$ is a non-empty set of designs which is equal to its bi-orthogonal, i.e. $\mathbf{A} = \mathbf{A}^{\perp\perp}$.

Exercise

Show that $\mathbf{A}$ is a type if and only if there exists $\mathbf{B}$ such that $\mathbf{A} = \mathbf{B}^\perp$.

Multiplicatives: connectives

Definition

Let α and $\flat$ be designs such that $X_\alpha \cap X_\flat = \emptyset$. We define:

$$\alpha \otimes \flat = (a + b, (X_\alpha \cup X_\flat, \sigma_a \cup \sigma_b))$$

Notice that if $\mathbf{A}$ is a type, $\alpha \in \mathbf{A}$ and $\flat \in \mathbf{A}$, then $X_\alpha = X_\flat$. We therefore define $X_{\mathbf{A}} = X_\alpha$ for $\alpha \in \mathbf{A}$.

Definition

Let $\mathbf{A}$ and $\mathbf{B}$ be types such that $X_{\mathbf{A}} \cap X_{\mathbf{B}} = \emptyset$. We define:

$$\mathbf{A} \otimes \mathbf{B} = \{\alpha \otimes \flat \mid \alpha \in \mathbf{A}, \flat \in \mathbf{B}\}^{\perp\perp}$$

Proposition

The connective $\otimes$ is associative, commutative and has a neutral element, namely $\mathbf{1} = \{(0, (\emptyset, \emptyset))\}^{\perp\perp} = \{(0, (\emptyset, \emptyset))\}$.

Multiplicatives: Linear Implication

Proposition

Let $(X_a, \sigma_a), (X_b, \sigma_b), (X_c, \sigma_c)$ be three paraproofs with $X_a \cap X_b = \emptyset$. Then:

$$\begin{aligned} & \ll (X_c, \sigma_c), (X_a \cup X_b, \sigma_a \cup \sigma_b) \gg \\ &= \ll (X_c, \sigma_c), (X_a, \sigma_a) \gg + \ll \text{Ex}((X_c, \sigma_c), (X_a, \sigma_a), (X_b, \sigma_b)) \gg \end{aligned}$$

Proposition (Adjunction)

Let a, b, c be designs with $X_a \cap X_b = \emptyset$. Then:

$$\ll c, a \otimes b \gg = \ll \text{Ex}(c, a), b \gg$$

Proposition

Let $\mathbf{A}, \mathbf{B}$ be types with $X_{\mathbf{A}} \cap X_{\mathbf{B}} = \emptyset$. Then:

$$(\mathbf{A} \otimes \mathbf{B}^\perp)^\perp = \{f \mid \forall a \in \mathbf{A}, \text{Ex}(f, a) \in \mathbf{B}\} = \mathbf{A} \multimap \mathbf{B}$$

Multiplicatives: truth

Definition

A design α is *winning* when $a = 0$ and $(X_\alpha, \sigma_\alpha)$ is a disjoint union of transpositions. A type $\mathbf{A}$ is *true* when it contains a winning design.

Proposition

The types $\mathbf{A}, \mathbf{A}^\perp$ cannot be simultaneously true.

Proposition

If $\mathbf{A} \multimap \mathbf{B}$ and $\mathbf{B} \multimap \mathbf{C}$ are true, then $\mathbf{A} \multimap \mathbf{C}$ is true.

Multiplicatives: denotational semantics

In this setting, it is possible to interpret the proofs and formulas of Multiplicative Linear Logic with units (MLL) and this interpretation satisfies the following properties:

- A proof π of $\vdash A$ is interpreted as a winning design $\pi^\bullet$ in the interpretation $A^\bullet$ of A ;
- The interpretation is invariant under cut-elimination;