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

\[ \vdash X \otimes X \]

\[ \vdash X \Downarrow \overline{\Downarrow} X \Downarrow \]

\[ \vdash X \Downarrow \overline{\Downarrow} X \Downarrow \]
Proofs as sets of axioms
Proofs as sets of axioms

X \otimes X

\llcorner X \llcorner

\llcorner X \llcorner

L \otimes R

L \otimes R

\llcorner X \llcorner
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 $\otimes$, $\forall$ 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);
Correctness Criterions

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

1. we define a finite set $S$ of objects depending on the typing part $\mathcal{R}_{\text{type}}$ 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 $\mathcal{R}_{\text{ax}}$ of $\mathcal{R}$: the resulting trip is long (Longtrips), the resulting graph is connected and acyclic (DR).

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

Figure: Cut-Elimination: $\otimes - \otimes$ 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 $\otimes$-$\supset$ 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
Example

Figure: Example
Example

Figure: Example
Example

Figure: Example
LongTrips and Permutations
We briefly recall the longtrips criterion.

**Definition**

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

Define $R_\cdot$ 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 $R_s$ whose vertices is $R_\cdot$ (defined in the next slides).
(a) Axiom Brick

(b) Cut Brick

Figure: Axiom and Cut bricks
Figure: Axiom and Cut bricks
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

(a) First switching

(b) Second switching
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 $\tau_s$ 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 $\sigma$ (which is a disjoint union of transpositions).
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
Figure: Associativity: Towards Permutations
Figure: Associativity: Toward Permutations
Example

Figure: Associativity: Permutations
Example

\[ \sigma \tau \delta \]

**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 $A = \text{Proofs of } A^\perp$
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 $S(A)$ to be the set of permutations induced by the switchings of $A$.

**Definition**

Two permutations $\sigma, \tau$ 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 $P(A) = (S(A))^{\perp}$.

**Remark**

Both $P(A)$ and $P(A^{\perp})$ are non-empty.
Switchings of $A \otimes B$

$\sigma \in \mathcal{I}(A)$

$\tau \in \mathcal{I}(B)$
Switchings of $A \otimes B$

$\sigma \in \mathcal{S}(A)$

$\tau \in \mathcal{S}(B)$
We denote by $\sigma \otimes^j_i \tau$ 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 $A^\perp \otimes B^\perp$ is the union of a switching of $A^\perp$ and a switching of $B^\perp$, we obtain that the permutations $\sigma \otimes_i \tau$ are orthogonal to all the elements in $\mathcal{S}(A^\perp \otimes B^\perp)$.

**Proposition**

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

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 $\sigma \otimes_i \tau$ such that $\sigma \otimes_i \tau(a) = b$. Thus:

**Proposition**

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

$\sigma \in \mathcal{I}(A)$

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

**Proposition**

$$\mathcal{I}(A \otimes B) \subset \mathcal{P}(A^\perp \otimes B^\perp)$$
Switchings of $A \bowtie B$

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**

$$\mathcal{I}(A \bowtie B) = \mathcal{P}(A^\perp \otimes B^\perp)$$

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

$$\mathcal{P}(A^\perp \bowtie B^\perp) = (\mathcal{I}(A^\perp \bowtie B^\perp))^\perp = (\mathcal{P}(A^\perp \otimes B^\perp))^\perp = (\mathcal{I}(A \otimes B))^{\perp \perp}$$
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 \((X, \sigma)\) of a finite set \(X\) and a permutation \(\sigma \in \mathcal{S}(X)\).
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 \((X, \tau)\) is equal to \((Y, \text{Ex}(\sigma, \tau))\), where:

\[
\text{Ex}(\sigma, \tau) = \bigcup_{i=0}^{\infty} [\sigma(\tau \sigma)^i]_{\upharpoonright Y}
\]

\[
= \sigma_{\upharpoonright Y} \cup (\sigma \tau \sigma)_{\upharpoonright Y} \cup (\sigma \tau \sigma \tau \sigma)_{\upharpoonright Y} \cup \ldots
\]

**Exercise**

*Show Ex(\(\sigma, \tau\)) is a permutation.*
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 GoI;
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 $a \in \mathbb{N}$ and $p_a = (X_a, \sigma_a)$ is a paraproof.
Multiplicatives: execution

Definition
Let \((X, \sigma)\) and \((Y, \tau)\) be paraproofs. 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 \(X_a \cap X_b \cap X_c = \emptyset\),

\[
\text{Ex}(a, \text{Ex}(b, c)) = \text{Ex}(\text{Ex}(a, b), c)
\]
Definition

Two designs $a$ and $b$ are orthogonal — written $a \perp b$ — when $X_a = X_b$ and:

$$a + b + \ll (X_a, \sigma_a), (X_b, \sigma_b) \gg = 1$$

Proposition

Two designs $a, b$ are orthogonal if and only if $Ex(a, b) = (1, (\emptyset, \emptyset))$. 
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** $A$ is a non-empty set of designs which is equal to its bi-orthogonal, i.e. $A = A^{\perp\perp}$.

**Exercise**

*Show that $A$ is a type if and only if there exists $B$ such that $A = B^\perp$.*
Multiplicatives: connectives

**Definition**

Let \( a \) and \( b \) be designs such that \( X_a \cap X_b = \emptyset \). We define:

\[
    a \otimes b = (a + b, (X_a \cup X_b, \sigma_a \cup \sigma_b))
\]

Notice that if \( A \) is a type, \( a \in A \) and \( b \in A \), then \( X_a = X_b \). We therefore define \( X_A = X_a \) for \( a \in A \).

**Definition**

Let \( A \) and \( B \) be types such that \( X_A \cap X_B = \emptyset \). We define:

\[
    A \otimes B = \{ a \otimes b \mid a \in A, b \in B \}
\]

**Proposition**

The connective \( \otimes \) is associative, commutative and has a neutral element, namely \( 1 = \{(0, (\emptyset, \emptyset))\} \downarrow \downarrow = \{(0, (\emptyset, \emptyset))\} \).
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:

\[
\ll (X_c, \sigma_c), (X_a \cup X_b, \sigma_a \cup \sigma_b) \gg
\]

\[
= \ll (X_c, \sigma_c), (X_a, \sigma_a) \gg + \ll \text{Ex}((X_c, \sigma_c), (X_a, \sigma_a), (X_b, \sigma_b)) \gg
\]

**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 \(A, B\) be types with \(X_A \cap X_B = \emptyset\). Then:

\[
(A \otimes B^\perp)^\perp = \{f \mid \forall a \in A, \text{Ex}(f, a) \in B\} = A \multimap B
\]
 multiplicatives: truth

**Definition**

A design $a$ is *winning* when $a = 0$ and $(X_a, \sigma_a)$ is a disjoint union of transpositions. A type $A$ is *true* when it contains a winning design.

**Proposition**

*The types $A, \lnot A$ cannot be simultaneously true.*

**Proposition**

*If $A \rightarrow B$ and $B \rightarrow C$ are true, then $A \rightarrow C$ is true.*
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 $\pi$ 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;