Cut Elimination for Monomial MALL Proof Nets

Roberto Maieli

Università degli Studi "Roma Tre"
maieli@uniroma3.it

joint work with

Olivier Laurent
CNRS and Université Paris VII

Twenty-Third Annual IEEE Symposium on
Logic in Computer Science
27th June 2008, Pittsburgh

Proof Theory & Linear Logic

- Since its inception linear logic (LL, Girard 1987) has changed the proof theoretical way of dealing with cut elimination.
- This task was traditionally carried out by means of sequent calculi with the consequence that the most part of these works were engrossed by tedious commutations of rules.

Proof Theory & Linear Logic

- Since its inception linear logic (LL, Girard 1987) has changed the proof theoretical way of dealing with cut elimination.

- This task was traditionally carried out by means of sequent calculi with the consequence that the most part of these works were engrossed by tedious commutations of rules.

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $\parr$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^\perp$ extends to any formula by de Morgan laws:

$$(A \otimes B)^\perp = (B^\perp \parr A^\perp) \qquad (A \parr B)^\perp = (B^\perp \otimes A^\perp)$$
$$(A \& B)^\perp = (B^\perp \oplus A^\perp) \qquad (A \oplus B)^\perp = (B^\perp \& A^\perp)$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n\geq 1}$, proved using the following rules:
  - identity: $$\frac{}{A, A^\perp}\,ax \qquad \frac{\Gamma, A \qquad \Delta, A^\perp}{\Gamma, \Delta}\,cut$$
  - multiplicatives: $$\frac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr$$
  - additives: $$\frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\,\& \qquad \frac{\Gamma, A}{\Gamma, A \oplus B}\,\oplus_1 \qquad \frac{\Gamma, B}{\Gamma, A \oplus B}\,\oplus_2$$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $⅋$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^{\perp}$ extends to any formula by de Morgan laws:

$$(A \otimes B)^{\perp} = (B^{\perp} ⅋ A^{\perp}) \qquad (A ⅋ B)^{\perp} = (B^{\perp} \otimes A^{\perp})$$
$$(A \& B)^{\perp} = (B^{\perp} \oplus A^{\perp}) \qquad (A \oplus B)^{\perp} = (B^{\perp} \& A^{\perp})$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity: $\dfrac{}{A, A^{\perp}}\text{ax} \qquad \dfrac{\Gamma, A \qquad \Delta, A^{\perp}}{\Gamma, \Delta}\text{cut}$
  - multiplicatives: $\dfrac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\otimes \qquad \dfrac{\Gamma, A, B}{\Gamma, A ⅋ B}⅋$
  - additives: $\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\& \quad \dfrac{\Gamma, A}{\Gamma, A \oplus B}\oplus_1 \quad \dfrac{\Gamma, B}{\Gamma, A \oplus B}\oplus_2$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $\parr$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^{\perp}$ extends to any formula by de Morgan laws:

$$(A \otimes B)^{\perp} = (B^{\perp} \parr A^{\perp}) \qquad (A \parr B)^{\perp} = (B^{\perp} \otimes A^{\perp})$$
$$(A \& B)^{\perp} = (B^{\perp} \oplus A^{\perp}) \qquad (A \oplus B)^{\perp} = (B^{\perp} \& A^{\perp})$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity: $\dfrac{}{A, A^{\perp}}\,\text{ax}$ $\qquad$ $\dfrac{\Gamma, A \qquad \Delta, A^{\perp}}{\Gamma, \Delta}\,\text{cut}$
  - multiplicatives: $\dfrac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\,\otimes$ $\qquad$ $\dfrac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr$
  - additives: $\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\,\&$ $\quad$ $\dfrac{\Gamma, A}{\Gamma, A \oplus B}\,\oplus_1$ $\quad$ $\dfrac{\Gamma, B}{\Gamma, A \oplus B}\,\oplus_2$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $\parr$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^\perp$ extends to any formula by de Morgan laws:

$$(A \otimes B)^\perp = (B^\perp \parr A^\perp) \qquad (A \parr B)^\perp = (B^\perp \otimes A^\perp)$$
$$(A \& B)^\perp = (B^\perp \oplus A^\perp) \qquad (A \oplus B)^\perp = (B^\perp \& A^\perp)$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity: $\dfrac{}{A, A^\perp}\,\mathrm{ax} \qquad \dfrac{\Gamma, A \qquad \Delta, A^\perp}{\Gamma, \Delta}\,\mathrm{cut}$
  - multiplicatives: $\dfrac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\,\otimes \qquad \dfrac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr$
  - additives: $\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\,\& \qquad \dfrac{\Gamma, A}{\Gamma, A \oplus B}\,\oplus_1 \qquad \dfrac{\Gamma, B}{\Gamma, A \oplus B}\,\oplus_2$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $⅋$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^\perp$ extends to any formula by de Morgan laws:

$$(A \otimes B)^\perp = (B^\perp ⅋ A^\perp) \qquad (A ⅋ B)^\perp = (B^\perp \otimes A^\perp)$$
$$(A \& B)^\perp = (B^\perp \oplus A^\perp) \qquad (A \oplus B)^\perp = (B^\perp \& A^\perp)$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity:

$$\frac{}{A, A^\perp}\ \text{ax} \qquad \frac{\Gamma, A \qquad \Delta, A^\perp}{\Gamma, \Delta}\ \text{cut}$$

  - multiplicatives:

$$\frac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\ \otimes \qquad \frac{\Gamma, A, B}{\Gamma, A ⅋ B}\ ⅋$$

  - additives:

$$\frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\ \& \qquad \frac{\Gamma, A}{\Gamma, A \oplus B}\ \oplus_1 \qquad \frac{\Gamma, B}{\Gamma, A \oplus B}\ \oplus_2$$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $\parr$ (*par*), & (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^\perp$ extends to any formula by de Morgan laws:

$$(A \otimes B)^\perp = (B^\perp \parr A^\perp) \qquad (A \parr B)^\perp = (B^\perp \otimes A^\perp)$$
$$(A \& B)^\perp = (B^\perp \oplus A^\perp) \qquad (A \oplus B)^\perp = (B^\perp \& A^\perp)$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity: $\dfrac{}{A, A^\perp}\,\text{ax} \qquad \dfrac{\Gamma, A \qquad \Delta, A^\perp}{\Gamma, \Delta}\,\text{cut}$
  - multiplicatives: $\dfrac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\,\otimes \qquad \dfrac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr$
  - additives: $\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\,\& \qquad \dfrac{\Gamma, A}{\Gamma, A \oplus B}\,\oplus_1 \qquad \dfrac{\Gamma, B}{\Gamma, A \oplus B}\,\oplus_2$

MALL Sequent Calculus (The Multiplicative-Additive fragment of LL)

- *Formulas* $A, B, \ldots$ are built from *literals* by the binary connectives $\otimes$ (*tensor*), $⅋$ (*par*), $\&$ (*with*) and $\oplus$ (*plus*).
- *Negation* $(.)^\perp$ extends to any formula by de Morgan laws:

$$(A \otimes B)^\perp = (B^\perp ⅋ A^\perp) \qquad (A ⅋ B)^\perp = (B^\perp \otimes A^\perp)$$
$$(A \& B)^\perp = (B^\perp \oplus A^\perp) \qquad (A \oplus B)^\perp = (B^\perp \& A^\perp)$$

- *Sequents* Γ, Δ are sets of formula occurrences $A_1, \ldots, A_{n \geq 1}$, proved using the following rules:
  - identity: $\dfrac{}{A, A^\perp}\ \text{ax} \qquad \dfrac{\Gamma, A \qquad \Delta, A^\perp}{\Gamma, \Delta}\ \text{cut}$
  - multiplicatives: $\dfrac{\Gamma, A \qquad \Delta, B}{\Gamma, \Delta, A \otimes B}\ \otimes \qquad \dfrac{\Gamma, A, B}{\Gamma, A ⅋ B}\ ⅋$
  - additives: $\dfrac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \& B}\ \& \quad \dfrac{\Gamma, A}{\Gamma, A \oplus B}\ \oplus_1 \quad \dfrac{\Gamma, B}{\Gamma, A \oplus B}\ \oplus_2$

Cut-elimination with the SC is problematic

$$\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, A^\perp}\,\& \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A \& A, A^\perp \oplus B^\perp}\,cut$$

it may reduce to:

$$\dfrac{\dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut \qquad \dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut}{A \& A, A^\perp \oplus B^\perp}\,\&$$

or to (according to permutability of rules):

$$\dfrac{\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, A^\perp}\,\& \qquad A, A^\perp}{A \& A, A^\perp}\,cut}{A \& A, A^\perp \oplus B^\perp}\,\oplus_1$$

Cut-elimination with the SC is problematic

$$\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A\&A, A^\perp}\,\& \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A\&A, A^\perp \oplus B^\perp}\,cut$$

it may reduce to:

$$\dfrac{\dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut \qquad \dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut}{A\&A, A^\perp \oplus B^\perp}\,\&$$

or to (according to permutability of rules):

$$\dfrac{\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A\&A, A^\perp}\,\& \qquad A, A^\perp}{A\&A, A^\perp}\,cut}{A\&A, A^\perp \oplus B^\perp}\,\oplus_1$$

Cut-elimination with the SC is problematic

$$\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, A^\perp}\,\& \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A \& A, A^\perp \oplus B^\perp}\,cut$$

it may reduce to:

$$\dfrac{\dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut \qquad \dfrac{A, A^\perp \qquad \dfrac{A, A^\perp}{A, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,cut}{A \& A, A^\perp \oplus B^\perp}\,\&$$

or to (according to permutability of rules):

$$\dfrac{\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, A^\perp}\,\& \qquad A, A^\perp}{A \& A, A^\perp}\,cut}{A \& A, A^\perp \oplus B^\perp}\,\oplus_1$$

Cut-elimination with the SC is problematic

$$\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, \color{red}{A^\perp}}\,\& \qquad \dfrac{A, A^\perp}{\color{red}{A}, A^\perp \oplus B^\perp}\,\oplus_1}{A \& A, A^\perp \oplus B^\perp}\,\color{blue}{\mathit{cut}}$$

it may reduce to:

$$\dfrac{\dfrac{A, \color{red}{A^\perp} \qquad \dfrac{A, A^\perp}{\color{red}{A}, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,\color{blue}{\mathit{cut}} \qquad \dfrac{A, \color{red}{A^\perp} \qquad \dfrac{A, A^\perp}{\color{red}{A}, A^\perp \oplus B^\perp}\,\oplus_1}{A, A^\perp \oplus B^\perp}\,\color{blue}{\mathit{cut}}}{A \& A, A^\perp \oplus B^\perp}\,\&$$

or to (according to permutability of rules):

$$\dfrac{\dfrac{\dfrac{A, A^\perp \qquad A, A^\perp}{A \& A, \color{red}{A^\perp}}\,\& \qquad \color{red}{A}, A^\perp}{A \& A, A^\perp}\,\color{blue}{\mathit{cut}}}{A \& A, A^\perp \oplus B^\perp}\,\oplus_1$$

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets (PNs): a possible solution

- PNs are parallel presentations of sequential proofs
- they quotient classes of equivalent proofs, modulo permutations of derivation rules.

MLL: The Multiplicative Fragment of LL is the perfect setting:

1. a PN is a canonical representative of a proof of the SC;
2. the (strong) cut elimination procedure is purely **local**: reducing a cut consists in to modifying only the nodes connected to it.

MALL: A lot of work has been done in order to extend (1) and (2) [Girard'96, Hughes-Van Glabbeek'03...]

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Proof Nets of MALL

In 1996, Girard proposed a new syntax for MALL PNs:

- without additive boxes (sequentiality)
- allowing graph super-positions (weights, slices)

But Girard's proposal was not as good as for MLL:

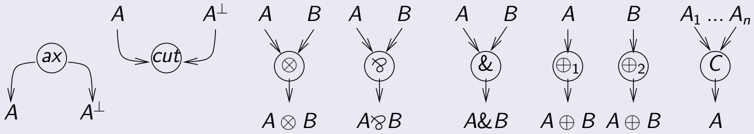
1. no canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.
   This problem has been solved by Hughes-Van Glabbeek (LICS2003)
2. no full cut elimination: only the *logical* (ready) cuts are reduced in a non-local way

Our Goal

To provide a **New Syntax for Monomial MALL PNs** *with a* **(local) full strong cut elimination**.

Definition (MALL Pre-Proof Structure)

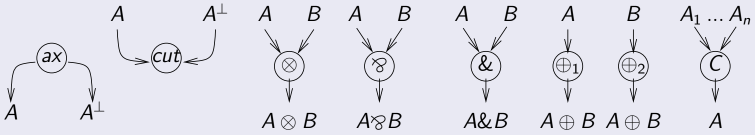
A **PPS** π is an oriented graph built on the following *links*:



- entering (*premisses*) and exiting (*conclusions*) edges are labelled by MALL formulas;
- a *contraction node* C has $A = A_1 = \ldots = A_{n \geq 1}$
- two C nodes have no common edges (they are *maximal*).

Definition (MALL Pre-Proof Structure)

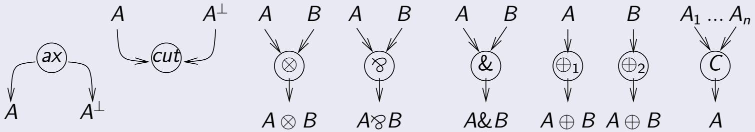
A **PPS** π is an oriented graph built on the following *links*:



- entering (*premisses*) and exiting (*conclusions*) edges are labelled by MALL formulas;
- a *contraction node* C has $A = A_1 = \ldots = A_{n \geq 1}$
- two C nodes have no common edges (they are *maximal*).

Definition (MALL Pre-Proof Structure)

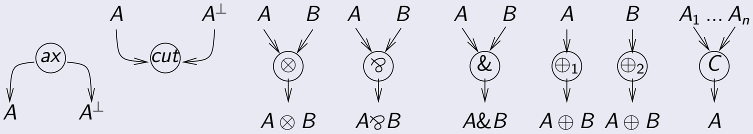
A **PPS** π is an oriented graph built on the following *links*:



- entering (*premisses*) and exiting (*conclusions*) edges are labelled by MALL formulas;
- a *contraction node* C has $A = A_1 = \ldots = A_{n \geq 1}$
- two C nodes have no common edges (they are *maximal*).

Definition (MALL Pre-Proof Structure)

A **PPS** π is an oriented graph built on the following *links*:



- entering (*premisses*) and exiting (*conclusions*) edges are labelled by MALL formulas;
- a *contraction node* C has $A = A_1 = \ldots = A_{n \geq 1}$
- two C nodes have no common edges (they are *maximal*).

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Weights)

a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of Boolean variables or negations of Boolean variables $p, \bar{p}, q, \bar{q}, \ldots$

- ϵ_p, for a variable p or its negation $\overline{p}$;
- 1, for the empty product;
- 0, for a product where both p and $\bar{p}$ appear;
- two weights, v and w, are **disjoint** when $v.w = 0$.
- a weight w **depends on a variable** p when ϵ_p appears in w;

Definition (Girard's MALL Proof Structure)

A MALL **GPS** π is a PPS with weights associated as follows:

1.

Definition (Girard's MALL Proof Structure)

A MALL **GPS** π is a PPS with weights associated as follows:

1. a & node is equipped with a (different) *eigen weight* p;
2.

Definition (Girard's MALL Proof Structure)

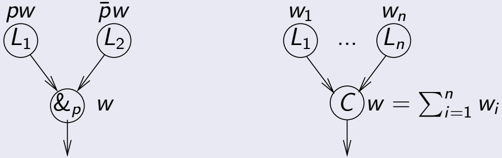
A MALL **GPS** π is a PPS with weights associated as follows:

1. a & node is equipped with a (different) *eigen weight* p;
2. a conclusion node has weight 1;
3.

Definition (Girard's MALL Proof Structure)

A MALL **GPS** π is a PPS with weights associated as follows:

1. a & node is equipped with a (different) *eigen weight* p;
2. a conclusion node has weight 1;
3. a node is equipped with a weight $w \neq 0$: two nodes have the same weight if they have a common edge, except when



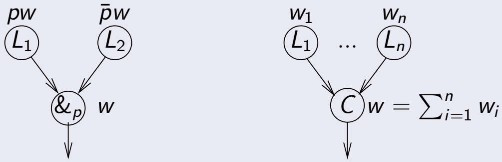
ϵ_p does not occur in w $\qquad$ $\forall i \forall j, w_i w_j = 0 \; (1 \leq i, j \leq n)$

4.

Definition (Girard's MALL Proof Structure)

A MALL **GPS** π is a PPS with weights associated as follows:

1. a & node is equipped with a (different) *eigen weight* p;
2. a conclusion node has weight 1;
3. a node is equipped with a weight $w \neq 0$: two nodes have the same weight if they have a common edge, except when

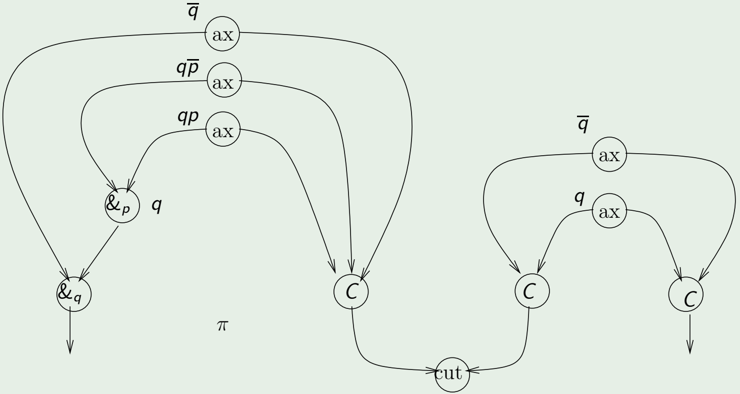


ϵ_p does not occur in w $\quad$ $\forall i \forall j, w_i w_j = 0$ $(1 \leq i, j \leq n)$

4. *dependency condition*: if v depends on p and w is the weight of the $\&_p$ node, then $v \leq w$.

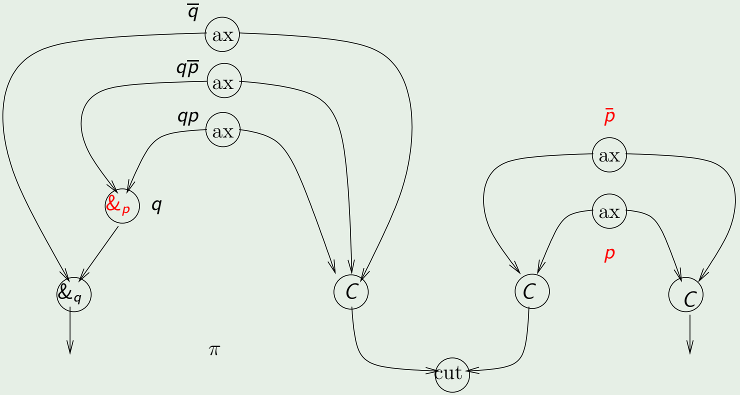
Example (1)

This is a GPS:



Example (2)

This is not a GPS: it violates the *dependency condition*, $p, \bar{p} \not\leq q$

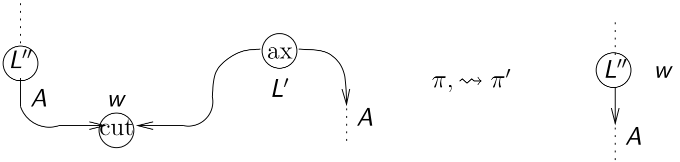


Cut Elimination

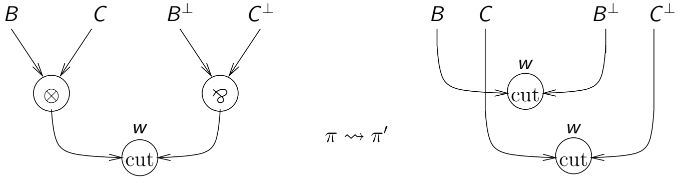
The original Girard's cut elimination is only lazy !

i.e., it only reduces the logical (or ready) cuts

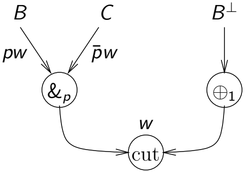
Ready Cut Elimination: *ax*-step



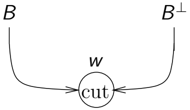
Ready Cut Elimination: $(\otimes/\parr)$-step



Ready Cut Elimination: $(\oplus_i/\&)$-step



$\pi \rightsquigarrow \pi'[p/1]$

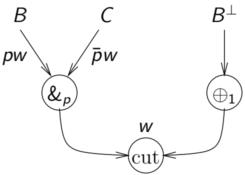


π' is obtained by **erasing the $\bar{p}$ slice** in π (i.e., $p = 1$ resp., $\bar{p} = 0$).

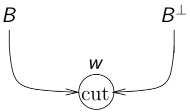
Girard's cut elimination stops here!

...in the following we fix this problem

Ready Cut Elimination: $(\oplus_i/\&)$-step



$\pi \rightsquigarrow \pi'[p/1]$

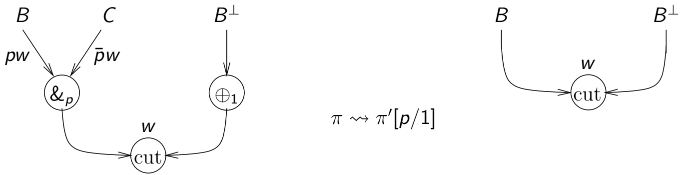


π' is obtained by **erasing the $\bar{p}$ slice** in π (i.e., $p = 1$ resp., $\bar{p} = 0$).

Girard's cut elimination stops here!

...in the following we fix this problem

Ready Cut Elimination: $(\oplus_i/\&)$-step

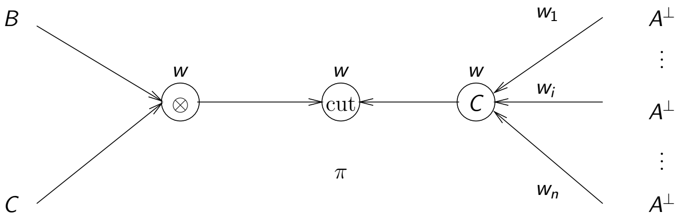


π' is obtained by **erasing the $\bar{p}$ slice** in π (i.e., $p = 1$ resp., $\bar{p} = 0$).

Girard's cut elimination stops here!

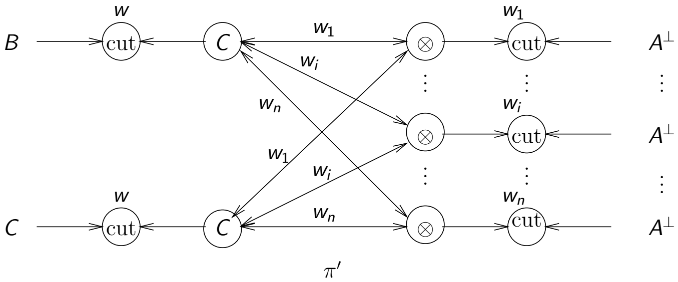
...in the following we fix this problem

Commutative Cut Elimination: $(\otimes/C)$-step



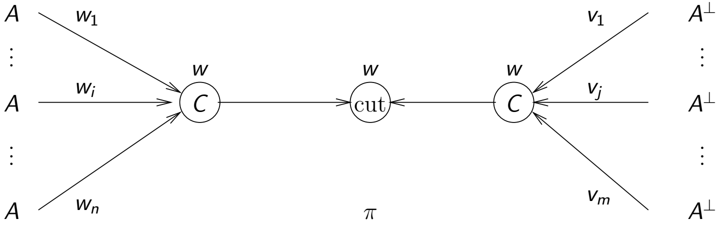
the $(\wp/C)$-step is similar

Commutative Cut Elimination: $(\otimes/C)$-step

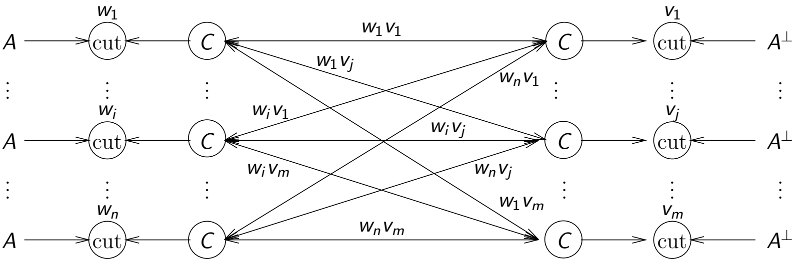


the "↔" edges are axiom links

Commutative Cut Elimination: (C/C)-step

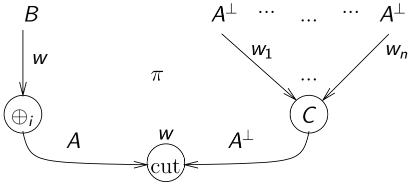


Commutative Cut Elimination: (C/C)-step

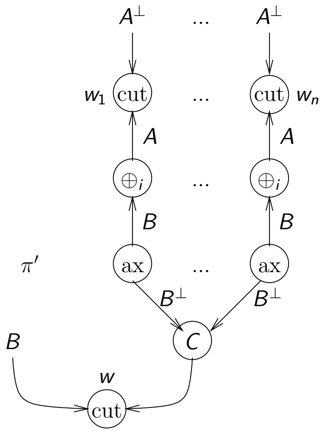


π'

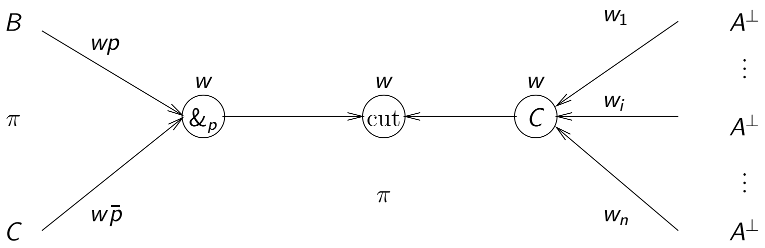
Commutative Cut Elimination: $(\oplus_i / C)$-step



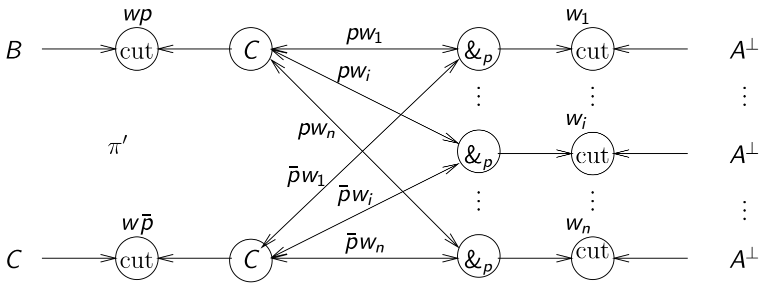
Commutative Cut Elimination: $(\oplus_i/C)$-step



Commutative Cut Elimination: the local ($\&/C$)-step

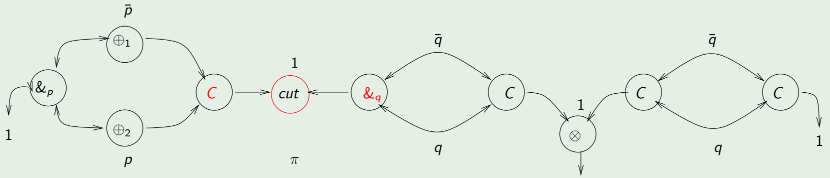


Commutative Cut Elimination: the local $(\&/C)$-step



Problems with the local $(\&/C)$ reduction step

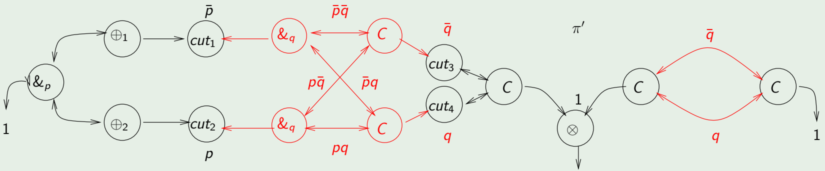
Example (4)



Let us reduce the $(C/\&_q)$ cut of the above GPS π

Problems with the local $(\&/C)$ reduction step

Example (4)

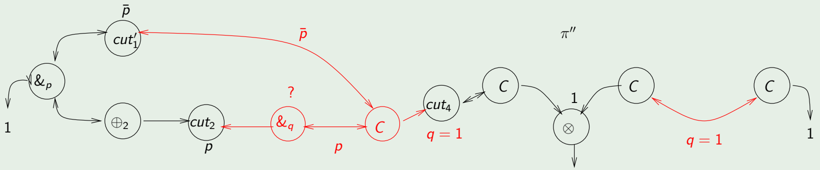


(1) We get a π' that **is not a PS**!

e.g. two axioms, with weights q and $\bar{q}$, do not satisfy the GPS *dependency condition* $(q, \bar{q} \not\leq p$, resp., $q, \bar{q} \not\leq \bar{p})$

Problems with the local $(\&/C)$ reduction step

Example (4)



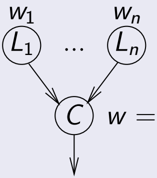
(2) π' may reduce the $(\oplus_1/\&_q)$ cut to π'' that **is not** even **a PPS**!

e.g., erasing the $\bar{q}$ slice induces a "degenerated" $\&_q$-link

Definition (New Monomial MALL Proof Structure)

A MALL **proof structure** (PS), is a pair $\langle \pi, E \rangle$ s.t.:

- $E = \{\epsilon_p.w = 0 \mid w \text{ is a weight } \epsilon_p\text{-free}\}$;
- π is a GPS with these modifications:
  - two & nodes may have the **same** eigen weight p;
  - all weights $v_1(\&_p), ..., v_n(\&_p)$ are **pairwise disjoint** ($v_i.v_j = 0$);
  - the weight of each contraction (C) is taken **modulo** E:



$w = \sum_{i=1}^{n} w_i \mod E$

$\forall i, j,\ w_i w_j = 0 \text{ and } w_i \leq w \ (1 \leq i, j \leq n)$

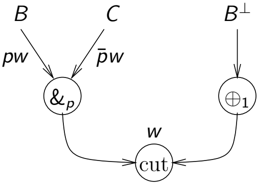
Definition (...continues)

new dependency condition: if w occurs in $\langle \pi, E \rangle$, then

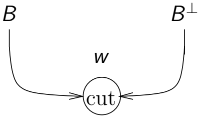
$$w \leq (\sum_{i=1}^{n} v_i) \mod E$$

- each v_i is :
  - either the weight of a node $\&_p$
  - or the suffix of an equation $\epsilon_p.v_i = 0$ of E;
- $\sum_{i=1}^{n} v_i$ is a monomial weight (modulo E);
- all weights $v_1, \dots v_n$ are pairwise disjoint.

Full Cut Elimination: the $(\oplus_i/\&)$-step

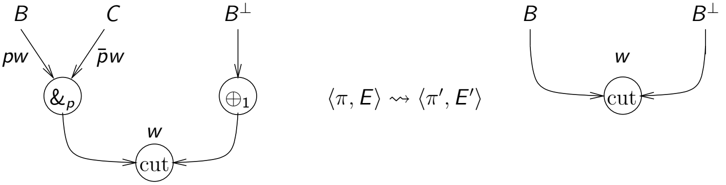


$\langle\pi, E\rangle \rightsquigarrow \langle\pi', E'\rangle$



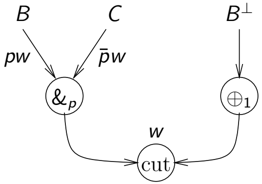
- $E' = E \cup \{\bar{p}.w = 0\}$;
- π' is obtained from π by :
  - erasing the slice $\bar{p}$ rooted at w
  - replacing weight pw with w (resp., $\bar{p}w$ with 0)

Full Cut Elimination: the $(\oplus_i/\&)$-step

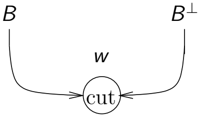


- $E' = E \cup \{\bar{p}.w = 0\}$;
- π' is obtained from π by :
  - erasing the slice $\bar{p}$ rooted at w
  - replacing weight pw with w (resp., $\bar{p}w$ with 0)

Full Cut Elimination: the $(\oplus_i/\&)$-step

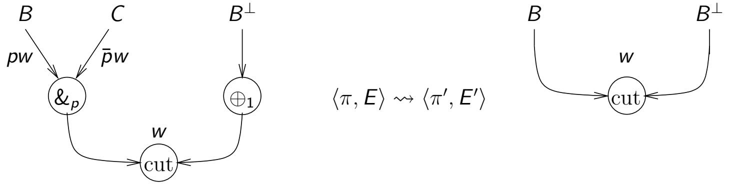


$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$



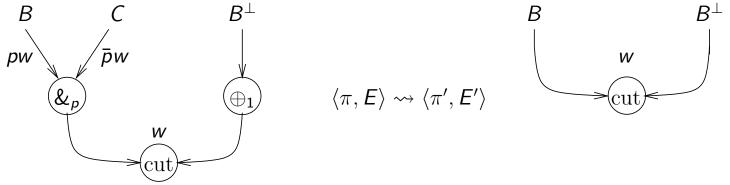
- $E' = E \cup \{\bar{p}.w = 0\}$;
- π' is obtained from π by :
  - erasing the slice $\bar{p}$ rooted at w
  - replacing weight pw with w (resp., $\bar{p}w$ with 0)

Full Cut Elimination: the $(\oplus_i/\&)$-step



- $E' = E \cup \{\bar{p}.w = 0\}$;
- π' is obtained from π by :
  - erasing the slice $\bar{p}$ rooted at w
  - replacing weight pw with w (resp., $\bar{p}w$ with 0)

Full Cut Elimination: the $(\oplus_i/\&)$-step



- $E' = E \cup \{\bar{p}.w = 0\}$;
- π' is obtained from π by :
  - erasing the slice $\bar{p}$ rooted at w
  - replacing weight pw with w (resp., $\bar{p}w$ with 0)

Full Cut Elimination

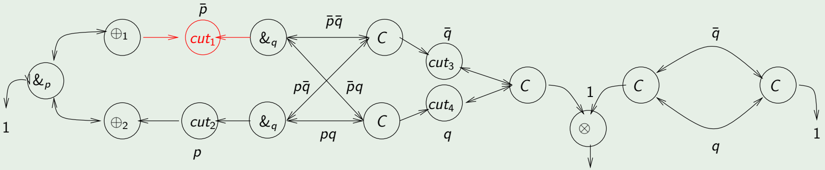
The below reduction steps R are performed like before with GPS

- *axiom*-step
- $(\otimes/\invamp)$-step
- $(\otimes/C)$-step
- $(\invamp/C)$-step
- $(\oplus_i/C)$-step
- $(\&/C)$-step
- (C/C)-step

$$\langle \pi, E \rangle \rightsquigarrow_R \langle \pi', E \rangle$$

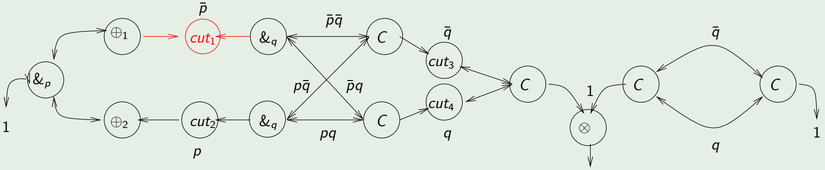
– $\pi \rightsquigarrow_R \pi'$ like before with GPS
– E remains unchanged.

Example (5)



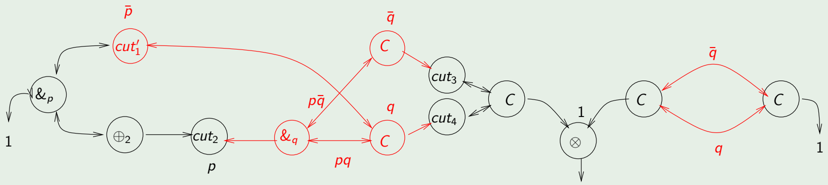
Observe: the $\langle \pi, \emptyset \rangle$ above is (now) a PS
(it satisfies the new dependency condition: $q, \bar{q} \leq p + \bar{p}$)

Example (5)



Let us reduce the cut_1 $(\oplus_1/\&_q)$.

Example (5)



We get the PS $\langle \pi', \{\bar{p}\bar{q} = 0\} \rangle$ above.

Observe: modulo $\{\bar{p}\bar{q} = 0\}$, $\bar{q} = p\bar{q}$ and $q = (\bar{p} + pq)$

Definition (Proof Nets)

A PS is correct (it is a **Proof Net**) if all local switchings are ACC.

(the notion of *local switching* is a variant of the Girard's switching)

Theorem (Sequentialization)

A PN with conclusion Γ *can be sequentialized into a sequent proof with same conclusion* Γ *and vice-versa.*

Proof.

- we exploit an expansion procedure which allows us to unfold each PN into a GPN;
- it can be shown that each expansion step preserves the *Girard's sequentialization.*

□

Definition (Proof Nets)

A PS is correct (it is a **Proof Net**) if all local switchings are ACC.

(the notion of *local switching* is a variant of the Girard's switching)

Theorem (Sequentialization)

A PN with conclusion Γ can be sequentialized into a sequent proof with same conclusion Γ and vice-versa.

Proof.

- we exploit an expansion procedure which allows us to unfold each PN into a GPN;
- it can be shown that each expansion step preserves the *Girard's sequentialization.*

□

Definition (Proof Nets)

A PS is correct (it is a **Proof Net**) if all local switchings are ACC.

(the notion of *local switching* is a variant of the Girard's switching)

Theorem (Sequentialization)

A PN with conclusion Γ can be sequentialized into a sequent proof with same conclusion Γ and vice-versa.

Proof.

- we exploit an expansion procedure which allows us to unfold each PN into a GPN;
- it can be shown that each expansion step preserves the *Girard's sequentialization.*

□

Definition (Proof Nets)

A PS is correct (it is a **Proof Net**) if all local switchings are ACC.

(the notion of *local switching* is a variant of the Girard's switching)

Theorem (Sequentialization)

A PN with conclusion Γ can be sequentialized into a sequent proof with same conclusion Γ and vice-versa.

Proof.

- we exploit an expansion procedure which allows us to unfold each PN into a GPN;
- it can be shown that each expansion step preserves the *Girard's sequentialization*.

□

Stability under the Cut Elimination

Theorem (Stability of proof structures)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ and $\langle \pi, E \rangle$ is a PS, then $\langle \pi', E' \rangle$ is a PS too.

Theorem (Stability of the correctness criterion)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ and $\langle \pi, E \rangle$ is a PN, then $\langle \pi', E' \rangle$ is a PN too.

Stability under the Cut Elimination

Theorem (Stability of proof structures)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ and $\langle \pi, E \rangle$ is a PS, then $\langle \pi', E' \rangle$ is a PS too.

Theorem (Stability of the correctness criterion)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ and $\langle \pi, E \rangle$ is a PN, then $\langle \pi', E' \rangle$ is a PN too.

Stability under the Cut Elimination

Theorem (Stability of proof structures)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ *and* $\langle \pi, E \rangle$ *is a PS, then* $\langle \pi', E' \rangle$ *is a PS too.*

Theorem (Stability of the correctness criterion)

$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$ *and* $\langle \pi, E \rangle$ *is a PN, then* $\langle \pi', E' \rangle$ *is a PN too.*

Strong Cut Elimination and Confluence

Theorem (Strong Cut Elimination)

We can always reduce a PN $\langle \pi, E \rangle$ into a PN $\langle \pi', E' \rangle$ that is cut-free; this reduction is strongly terminating.

Theorem (local confluence)

Assume a PN $\langle \pi, E \rangle$ s.t.

- $\langle \pi, E \rangle \rightsquigarrow_{cut_1} \langle \pi_1, E_1 \rangle$
- $\langle \pi, E \rangle \rightsquigarrow_{cut_2} \langle \pi_2, E_2 \rangle$,

with $cut_1 \neq cut_2$, then there exists PN $\langle \pi^*, E^* \rangle$ to which $\langle \pi_i, E_i \rangle$, for $1 \leq i \leq 2$, reduces in **at most one cut reduction step**.

Strong Cut Elimination and Confluence

Theorem (Strong Cut Elimination)

We can always reduce a PN $\langle \pi, E \rangle$ into a PN $\langle \pi', E' \rangle$ that is cut-free; this reduction is strongly terminating.

Theorem (local confluence)

Assume a PN $\langle \pi, E \rangle$ s.t.

- $\langle \pi, E \rangle \rightsquigarrow_{cut_1} \langle \pi_1, E_1 \rangle$
- $\langle \pi, E \rangle \rightsquigarrow_{cut_2} \langle \pi_2, E_2 \rangle$,

with $cut_1 \neq cut_2$, then there exists PN $\langle \pi^*, E^* \rangle$ to which $\langle \pi_i, E_i \rangle$, for $1 \leq i \leq 2$, reduces in **at most one cut reduction step**.

Strong Cut Elimination and Confluence

Theorem (Strong Cut Elimination)

We can always reduce a PN $\langle \pi, E \rangle$ into a PN $\langle \pi', E' \rangle$ that is cut-free; this reduction is strongly terminating.

Theorem (local confluence)

Assume a PN $\langle \pi, E \rangle$ s.t.

- $\langle \pi, E \rangle \rightsquigarrow_{cut_1} \langle \pi_1, E_1 \rangle$
- $\langle \pi, E \rangle \rightsquigarrow_{cut_2} \langle \pi_2, E_2 \rangle$,

with $cut_1 \neq cut_2$, then there exists PN $\langle \pi^*, E^* \rangle$ to which $\langle \pi_i, E_i \rangle$, for $1 \leq i \leq 2$, reduces in **at most one cut reduction step**.

questions ?