# Cut Elimination for Monomial MALL Proof Nets

Roberto Maieli

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

Rencontre *Journée Additifs*
Paris, March, 17th, 2008

# Proof Nets: state of the art

MLL PNs are the perfect setting:

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,
2. the (strong) cut elimination procedure is purely local: the reduction of a cut is given by only modifying the nodes connected to it.

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,
2. the (strong) cut elimination procedure is purely local: the reduction of a cut is given by only modifying the nodes connected to it.

A lot of work has been done to extend (1) and (2) to MALL.

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,
2. the (strong) cut elimination procedure is purely local: the reduction of a cut is given by only modifying the nodes connected to it.

A lot of work has been done to extend (1) and (2) to MALL.

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

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,
2. the (strong) cut elimination procedure is purely local: the reduction of a cut is given by only modifying the nodes connected to it.

A lot of work has been done to extend (1) and (2) to MALL.

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

- without additive boxes (sequentiality)

# Proof Nets: state of the art

MLL PNs are the perfect setting:

1. a proof net is a canonical representative of a proof of the sequent calculus,
2. the (strong) cut elimination procedure is purely local: the reduction of a cut is given by only modifying the nodes connected to it.

A lot of work has been done to extend (1) and (2) to MALL.

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

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

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them.

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is

# Proof Nets: state of the art (continues)

… but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is
   – lazy: only (ready) cuts not involving additive contractions are reducible; as consequence, not all proof nets are normalizable;

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is
   – lazy: only (ready) cuts not involving additive contractions are reducible; as consequence, not all proof nets are normalizable;
   – not local;

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is
   – lazy: only (ready) cuts not involving additive contractions are reducible; as consequence, not all proof nets are normalizable;
   – not local;

Our goal here is:

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is
   – lazy: only (ready) cuts not involving additive contractions are reducible; as consequence, not all proof nets are normalizable;
   – not local;

Our goal here is:
– to provide an answer to the (monomial) cut elimination.

# Proof Nets: state of the art (continues)

... but Girard's proposal was not as good as for MLL:

1. w.r.t. canonicity: there exist proofs which de-sequentialize into two possible PNs with no way to discriminate them. This problem has been solved (in a perfectly satisfactory way) by D. Hughes and R. van Glabbeek (2003)
2. w.r.t. cut elimination: Girard's one is
   – lazy: only (ready) cuts not involving additive contractions are reducible; as consequence, not all proof nets are normalizable;
   – not local;

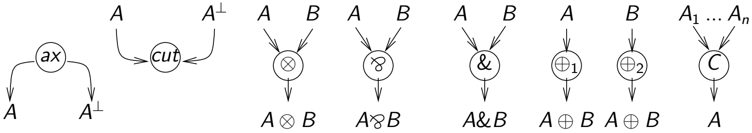
Our goal here is:

– to provide an answer to the (monomial) cut elimination.

– to allow a new kind of additive super-position (sharing nodes)

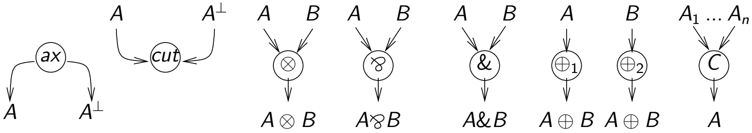
# MALL Pre-Proof Structures (PPS)

- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):



# MALL Pre-Proof Structures (PPS)

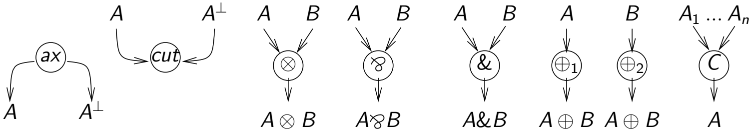
- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):




- in a contraction node $C$: $A = A_1 = \ldots = A_{n\geq 1}$

# MALL Pre-Proof Structures (PPS)

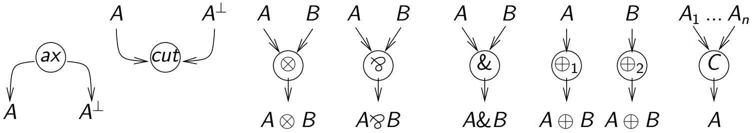
- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):




- in a contraction node $C$: $A = A_1 = \ldots = A_{n \geq 1}$
- entering edges are **premises** while the (possibly) emergent edges are *conclusions*

# MALL Pre-Proof Structures (PPS)

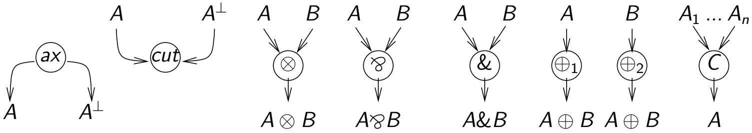
- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):




- in a contraction node $C$: $A = A_1 = \ldots = A_{n \geq 1}$
- entering edges are **premises** while the (possibly) emergent edges are *conclusions*
- two contraction nodes cannot have a common edge

# MALL Pre-Proof Structures (PPS)

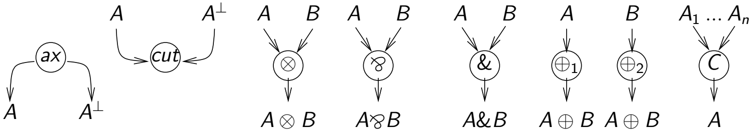
- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):




- in a contraction node $C$: $A = A_1 = \ldots = A_{n \geq 1}$
- entering edges are **premises** while the (possibly) emergent edges are *conclusions*
- two contraction nodes cannot have a common edge
- pending edges are called **conclusions** of $\pi$

# MALL Pre-Proof Structures (PPS)

- A **PPS** $\pi$ is an oriented graph built on the following nodes (edges are labelled by a MALL formulas):




- in a contraction node $C$: $A = A_1 = \ldots = A_{n \geq 1}$
- entering edges are **premises** while the (possibly) emergent edges are *conclusions*
- two contraction nodes cannot have a common edge
- pending edges are called **conclusions** of $\pi$
- a **link** is the graph made by a node together with its premise(s) and its (possibly) conclusion(s).

# MALL Girard Proof Structures (GPS): weights

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \ldots,$

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \ldots,$
- a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of variables or negation of variables.

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \dots$,
- a **monomial weight** $w, v, \dots$ is a product "." (conjunction) of variables or negation of variables.
- $\epsilon_p$, for a variable $p$ or its negation $\overline{p}$;

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \ldots$,
- a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of variables or negation of variables.
- $\epsilon_p$, for a variable $p$ or its negation $\overline{p}$;
- 1, for the empty product;

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \ldots,$
- a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of variables or negation of variables.
- $\epsilon_p$, 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;

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \ldots$,
- a **monomial weight** $w, v, \ldots$ is a product "." (conjunction) of variables or negation of variables.
- $\epsilon_p$, 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 $v.w = 0$.

# MALL Girard Proof Structures (GPS): weights

- a set of Boolean variables denoted by $p, q, \dots$,
- a **monomial weight** $w, v, \dots$ is a product "." (conjunction) of variables or negation of variables.
- $\epsilon_p$, 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 $\epsilon_p$ appears in $w$;

# Girard MALL Proof Structures (GPS)

A MALL *GPS* $\pi$ is a PPS with associated weights as follows:

# Girard MALL Proof Structures (GPS)

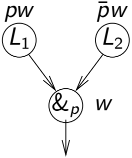
A MALL *GPS* $\pi$ is a PPS with associated weights as follows:

1. we associate a (different) *eigen weight* $p$, to each & node of $\pi$ (notation $\&_p$):

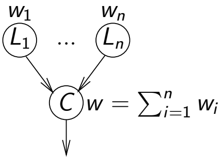
# Girard MALL Proof Structures (GPS)

A MALL *GPS* $\pi$ is a PPS with associated weights as follows:

1. we associate a (different) *eigen weight* $p$, to each & node of $\pi$ (notation $\&_p$):
2. we associate a weight $w \neq 0$ to each node; two nodes have the same weight if they have a common edge, except when:




$\epsilon_p$ does not occur in $w$

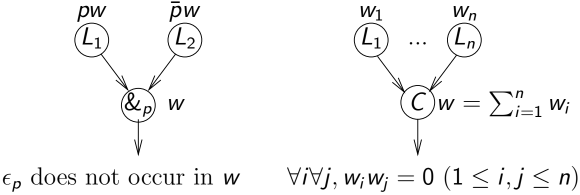



$\forall i \forall j, w_i w_j = 0 \ (1 \leq i, j \leq n)$

# Girard MALL Proof Structures (GPS)

A MALL *GPS* $\pi$ is a PPS with associated weights as follows:

1. we associate a (different) *eigen weight* $p$, to each & node of $\pi$ (notation $\&_p$):
2. we associate a weight $w \neq 0$ to each node; two nodes have the same weight if they have a common edge, except when:

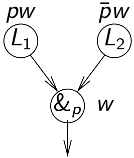


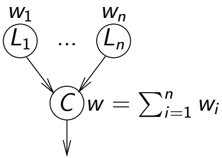

3. a conclusion node has weight 1;

# Girard MALL Proof Structures (GPS)

A MALL *GPS* $\pi$ is a PPS with associated weights as follows:

1. we associate a (different) *eigen weight* $p$, to each & node of $\pi$ (notation $\&_p$):
2. we associate a weight $w \neq 0$ to each node; two nodes have the same weight if they have a common edge, except when:



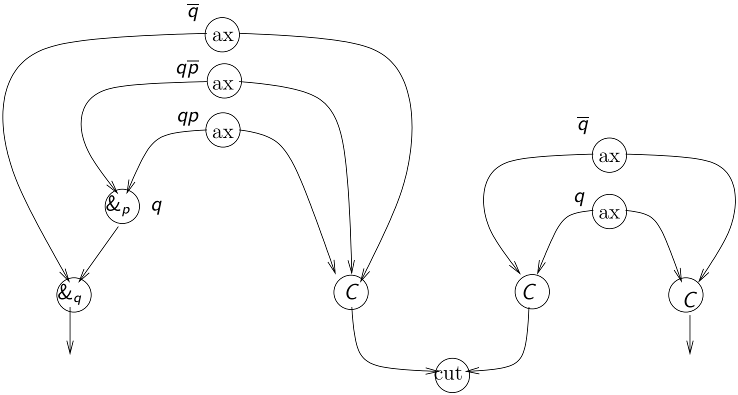




$\epsilon_p$ does not occur in $w$ $\qquad$ $\forall i \forall j, w_i w_j = 0 \ (1 \leq i, j \leq n)$

3. a conclusion node has weight 1;
4. **tech. cond.** if $w$ in $\pi$ depends on $p$, then $w \leq v$, where $v$ is the weight of the $\&_p$ node.

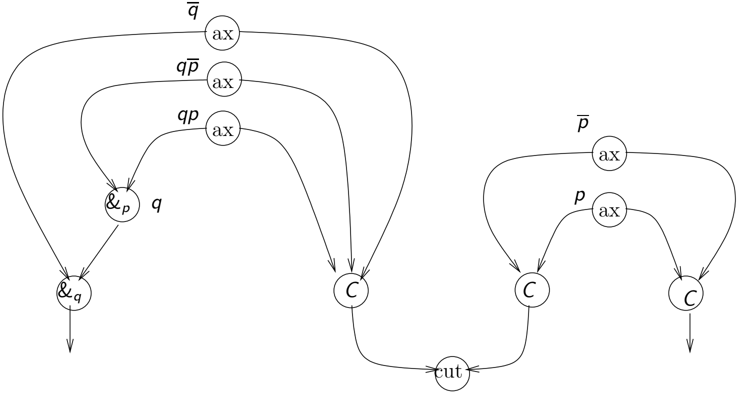
# Girard MALL Proof Structures: example 1

The following is a GPS:



# Girard MALL Proof Structures: example 2

The following is not a GPS:




it violates the *technical condition* of GPS definition: there exists a (axiom) node whose weight is $\bar{p}$ but $\overline{p} \not\leq q$, where $q$ is the weight of the (unique) node $\&_p$.

# Correctness Criterion: valuation, slices, switchings

# Correctness Criterion: valuation, slices, switchings

- a **valuation** $\varphi$ for $\pi$ is a function s.t.:

$$\varphi : p \mapsto \{0, 1\} \;;\; \varphi : w \mapsto \{0, 1\}$$

# Correctness Criterion: valuation, slices, switchings

- a **valuation** $\varphi$ for $\pi$ is a function s.t.:

$$\varphi : p \mapsto \{0,1\} \; ; \; \varphi : w \mapsto \{0,1\}$$

- a **slice** $\varphi(\pi)$ is the graph obtained from $\pi$ by keeping only those nodes (together its emerging edges) whose weight is 1;

# Correctness Criterion: valuation, slices, switchings

- a **valuation** $\varphi$ for $\pi$ is a function s.t.:

$$\varphi : p \mapsto \{0, 1\} \; ; \; \varphi : w \mapsto \{0, 1\}$$

- a **slice** $\varphi(\pi)$ is the graph obtained from $\pi$ by keeping only those nodes (together its emerging edges) whose weight is 1;
- a **switching** $S$ for $\pi$ is what remains of a slice $\varphi(\pi)$ after that:
  - for each $\invamp$-node we take only one premise and we cut the remaining one (left or right);

# Correctness Criterion: valuation, slices, switchings

- a **valuation** $\varphi$ for $\pi$ is a function s.t.:

$$\varphi : p \mapsto \{0,1\} \; ; \; \varphi : w \mapsto \{0,1\}$$

- a **slice** $\varphi(\pi)$ is the graph obtained from $\pi$ by keeping only those nodes (together its emerging edges) whose weight is 1;
- a **switching** $S$ for $\pi$ is what remains of a slice $\varphi(\pi)$ after that:
  - for each $\parr$-node we take only one premise and we cut the remaining one (left or right);
  - for each $\&_p$ node we cut the (unique) premise in $\varphi(\pi)$ and we add an oriented edge (a **jump**) from this $\&_p$ node to a node whose weight depends on $p$.

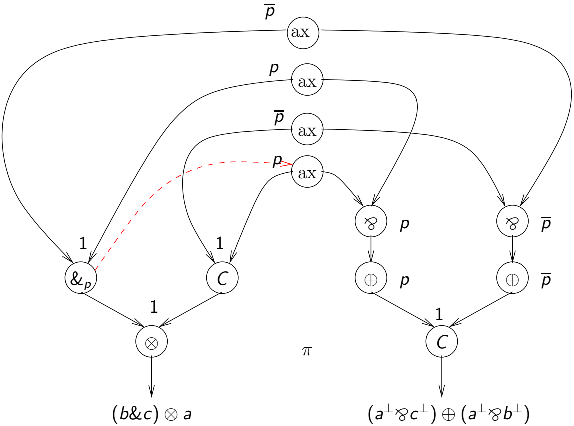
# Girard's Proof Net (GPN)

Definition: a GPS $\pi$ is **correct**, it is a GPN, if any switching, induced by a valuation $\varphi$ for $\pi$, is ACC.

# Girard's Proof Net (GPN)

Definition: a GPS $\pi$ is **correct**, it is a GPN, if any switching, induced by a valuation $\varphi$ for $\pi$, is ACC.

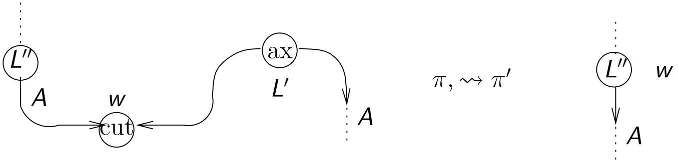
Examples: The GPS in the Ex. 1 is correct, while the next one is not so:



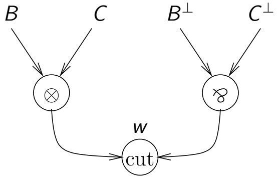
# Cut Elimination

… Girard's cut elimination is only the lazy (ready) one!

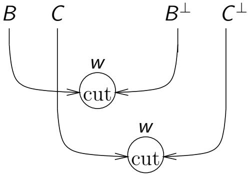
# Ready Cut Elimination: *ax*-step



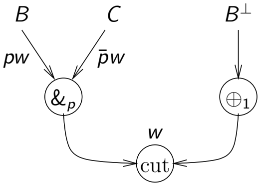
# Ready Cut Elimination: $(\otimes/\parr)$-step



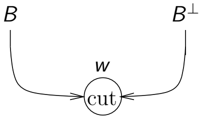

$\pi \rightsquigarrow \pi'$



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



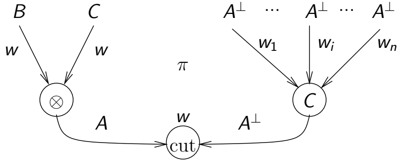

$\pi \rightsquigarrow \pi'[p/1]$



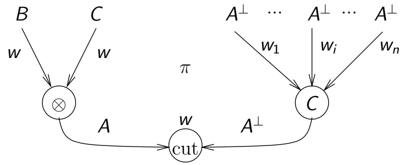

$\pi'$ is what is still nonzero in $\pi$, once $p = 1$ (resp., $\bar{p} = 0$).

... Girard's cut elimination stops here!

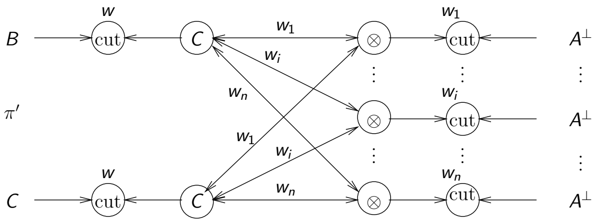
# Commutative Cut Elimination: ($\otimes$/$C$)-step



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

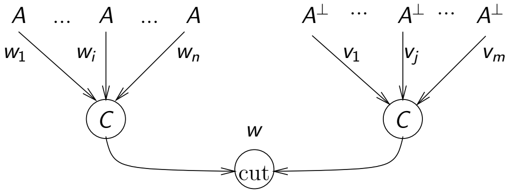



reduces to (the "$\leftrightarrow$" edges are axiom links):

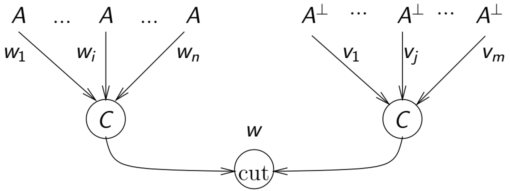



the step $(\invamp/C)$ is similar

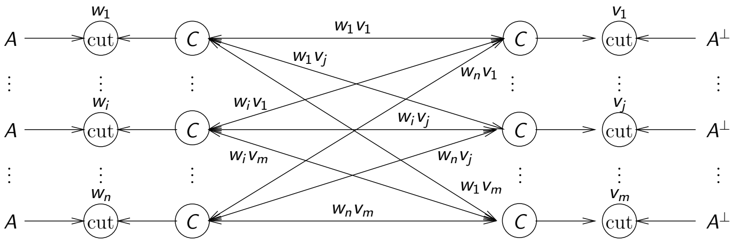
# Commutative Cut Elimination: $(C/C)$-step



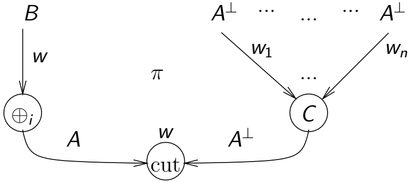
# Commutative Cut Elimination: $(C/C)$-step



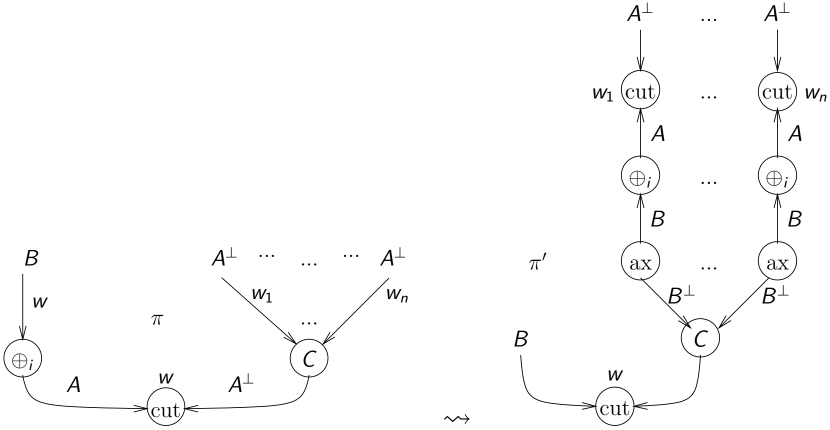

reduces to:



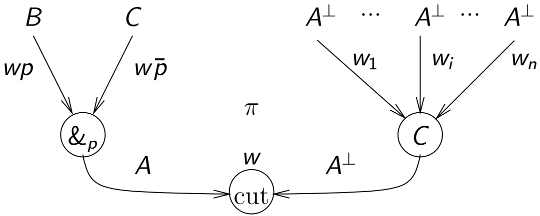
# Commutative Cut Elimination: $(\oplus_i/C)$-step



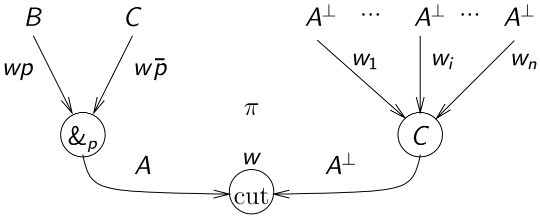
# Commutative Cut Elimination: $(\oplus_i / C)$-step



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

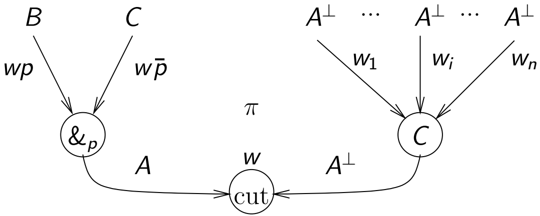


# Commutative Cut Elimination: $(\&/C)$-step




two possible solutions:

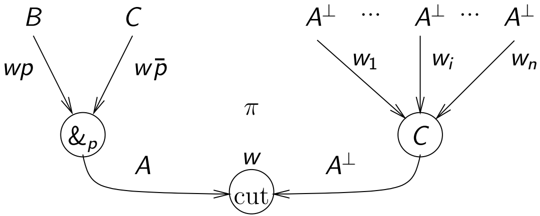
# Commutative Cut Elimination: $(\&/C)$-step




two possible solutions:

1. global solution : replace $\&_p$ by $\overbrace{\&_{p_1}, \ldots, \&_{p_n}}^{\text{different } p_i}$

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



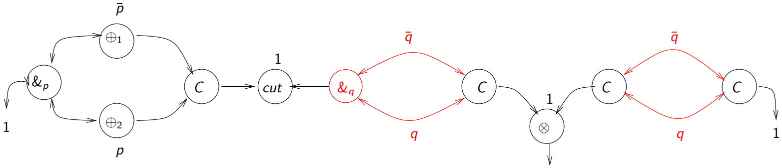

two possible solutions:

1. global solution : replace $\&_p$ by $\overbrace{\&_{p_1}, \ldots, \&_{p_n}}^{\textit{different } p_i}$

2. local solution : replace $\&_p$ by $\overbrace{\&_p, \ldots, \&_p}^{n-\textit{times the same } p}$

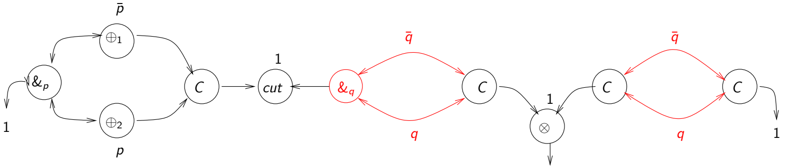
# $(\&/C)$-Cut Elimination: the global solution

Idea: *q-dependency graph*: the sub-graph of $\pi$ depending on $q$

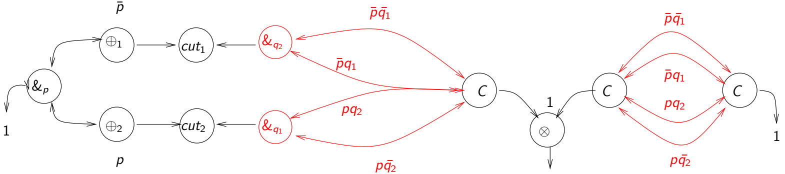


# $(\&/C)$-Cut Elimination: the global solution

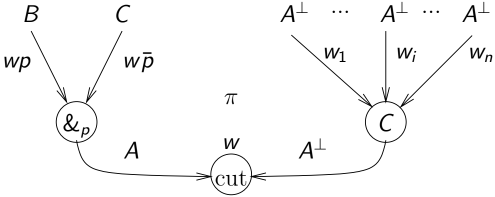
Idea: *q-dependency graph*: the sub-graph of $\pi$ depending on $q$



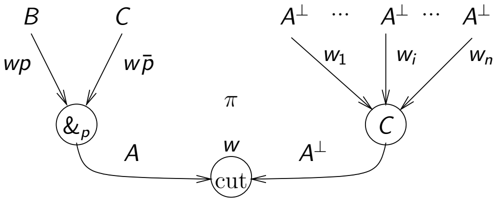

reduces to



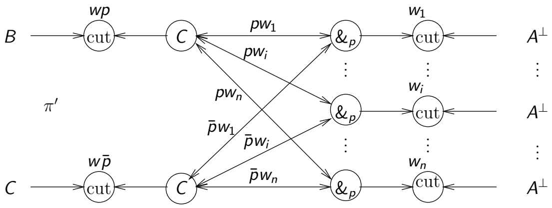
# (&/$C$)-Cut Elimination: the local solution



# $(\&/C)$-Cut Elimination: the local solution

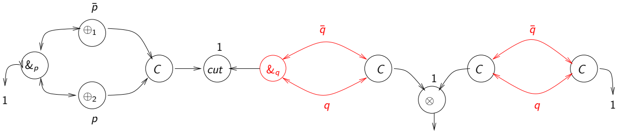



reduces to

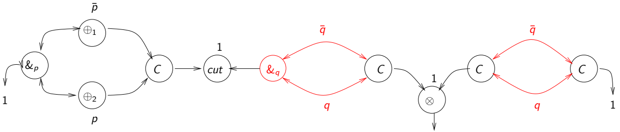



but this step does not preserve the notion GPS !

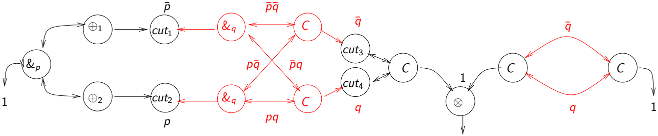
# $(\&/C)$-Cut Elimination: problems with the local solution



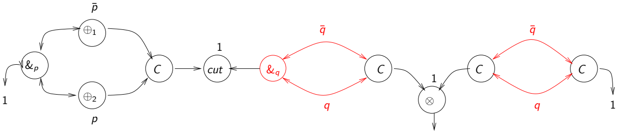
# $(\&/C)$-Cut Elimination: problems with the local solution



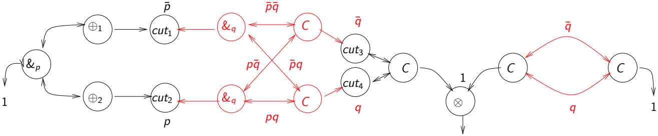

1) $\pi$ reduces to a $\pi'$ that is not a PS (by technical condition: $q \leq ?$)



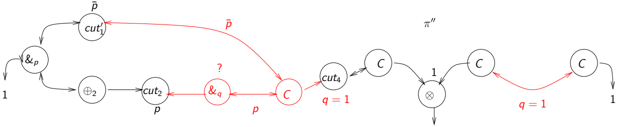
# $(\&/C)$-Cut Elimination: problems with the local solution




1) $\pi$ reduces to a $\pi'$ that is not a PS (by technical condition: $q \leq ?$)




2) $\pi'$ reduces $(cut_1)$ to $\pi''[q = 1;\ \bar{q} = 0]$ that is not even a PPS !



# MALL PS: *nouvelle syntax*

A MALL **proof structure** (*EPS*), is a pair $\langle \pi, E \rangle$ where:

- $E = \{\epsilon_p.w = 0 \mid \epsilon_p$ is a prefix $\wedge\ w$ is a weight $\epsilon_p$-free$\}$;
- $\pi$ is a GPS with the following modifications:

# MALL PS: *nouvelle syntax*

A MALL **proof structure** (*EPS*), is a pair $\langle \pi, E \rangle$ where:

- $E = \{\epsilon_p.w = 0 \mid \epsilon_p \text{ is a prefix } \wedge w \text{ is a weight } \epsilon_p\text{-free}\}$;
- $\pi$ is a GPS with the following modifications:
  - the eigen weights are not supposed to be different

# MALL PS: *nouvelle syntax*

A MALL **proof structure** (*EPS*), is a pair $\langle \pi, E \rangle$ where:

- $E = \{\epsilon_p.w = 0 \mid \epsilon_p$ is a prefix $\wedge\, w$ is a weight $\epsilon_p$-free$\}$;
- $\pi$ is a GPS with the following modifications:
    - the eigen weights are not supposed to be different
    - if $v_1(\&_p), \ldots, v_n(\&_p)$, then $v_i.v_j = 0$ for all $1 \leq i \leq j \leq n$

# MALL PS: *nouvelle syntax*

A MALL **proof structure** (*EPS*), is a pair $\langle \pi, E \rangle$ where:

- $E = \{\epsilon_p.w = 0 \mid \epsilon_p$ is a prefix $\wedge\ w$ is a weight $\epsilon_p$-free$\}$;
- $\pi$ is a GPS with the following modifications:
  - the eigen weights are not supposed to be different
  - if $v_1(\&_p), \ldots, v_n(\&_p)$, then $v_i.v_j = 0$ for all $1 \leq i \leq j \leq n$
  - all weights are considered *modulo* $E$;

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or
- $w$ occurs in an equation of $E$

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or
- $w$ occurs in an equation of $E$

then

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

where :

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or
- $w$ occurs in an equation of $E$

then

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

where :

- $v_i$, $1 \leq i \leq n$, is :
  - either the weight of a node $\&_p$
  - or the suffix of an equation $\epsilon_p.v_i = 0$ of $E$;

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or
- $w$ occurs in an equation of $E$

then

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

where :

- $v_i$, $1 \leq i \leq n$, 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 $E$);

# MALL PS: *nouvelle syntax* (continues)

(new) technical condition: if $w$ is a weight depending on $p$ and s.t.

- $w$ belongs to a node of $\pi$, or
- $w$ occurs in an equation of $E$

then

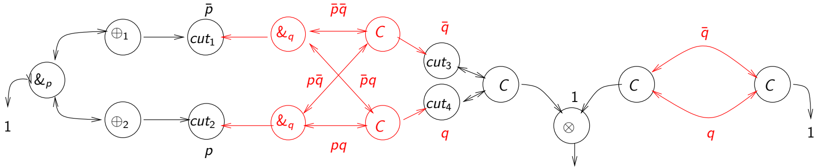
$$w \leq (\sum_{i=1}^{n} v_i) \mod E$$

where :

- $v_i$, $1 \leq i \leq n$, 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 $v_1, \ldots v_n$ are pairwise disjoint.

# MALL EPS : example

The pair $\langle \pi, \emptyset \rangle$ is (now) a proof structure ($q$ or $\bar{q} \leq p + \bar{p}$)



# Correctness Criterion: EPNs

## Definition (EPN)

An EPS is correct if all local switchings are ACC.

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

# Correctness Criterion: EPNs

## Definition (EPN)

An EPS is correct if all local switchings are ACC.

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

## Theorem

*A EPN with conclusion* $\Gamma$ *can be sequentialized into a sequent proof with same conclusion* $\Gamma$ *and vice-versa.*

# Correctness Criterion: EPNs

## Definition (EPN)

An EPS is correct if all local switchings are ACC.

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

## Theorem

*A EPN with conclusion* $\Gamma$ *can be sequentialized into a sequent proof with same conclusion* $\Gamma$ *and vice-versa.*

## Proof.

- we exploit an *expansion procedure* which allows us to unfold each EPN into a GPN;

# Correctness Criterion: EPNs

## Definition (EPN)

An EPS is correct if all local switchings are ACC.

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

## Theorem

*A EPN with conclusion $\Gamma$ can be sequentialized into a sequent proof with same conclusion $\Gamma$ and vice-versa.*

## Proof.

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

□

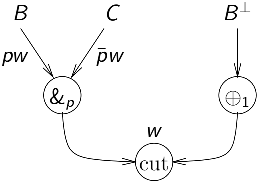
# Cut Elimination

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

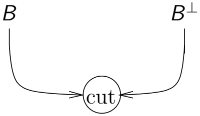
when $R$ is one of the reduction steps defined before for GPS:

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

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

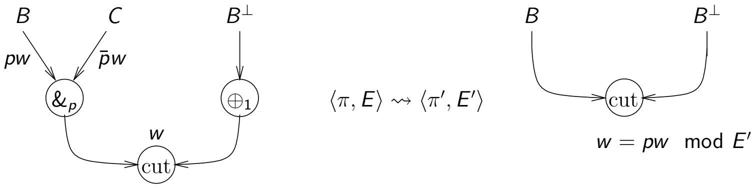



$$\langle \pi, E \rangle \rightsquigarrow \langle \pi', E' \rangle$$



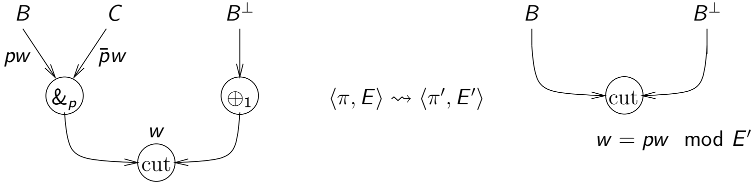

$$w = pw \mod E'$$

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



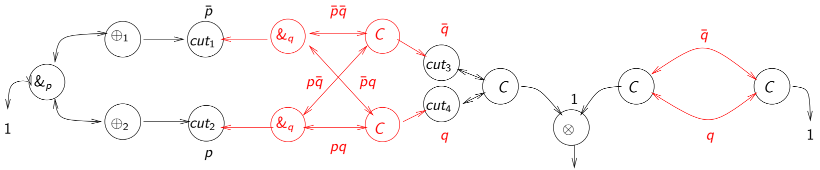

- $E' = E \cup \{\bar{p}.w = 0\}$;

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

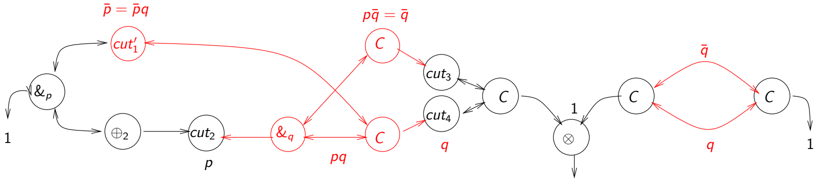



- $E' = E \cup \{\bar{p}.w = 0\}$;
- $\pi'$ is what (of $\pi$) remains still nonzero modulo $E'$:
  in particular, we remove all nodes whose weight $v \leq_{E'} \bar{p}.w$;
  (i.e., we remove the slice $\bar{p}$ rooted at $w$).

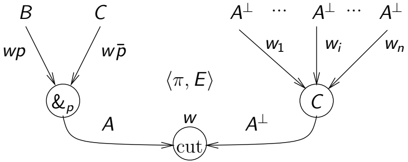
# $(\&/C)$-Cut Elimination: example 3



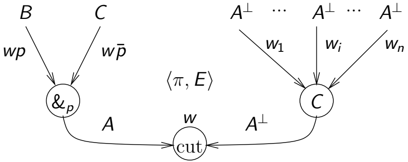

$\langle \pi, \emptyset \rangle$ reduces $(cut_1)$ to $\langle \pi', \{\bar{q}.\bar{p} = 0\} \rangle$ (that is still an EPS)



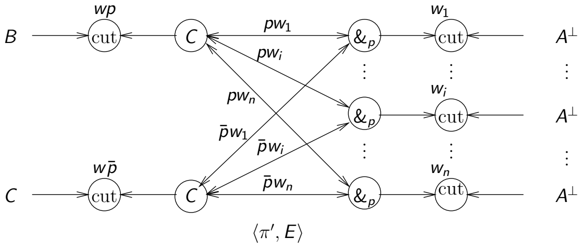
# Cut Elimination: the "local" $(\&/C)$-step



# Cut Elimination: the “local” $(\&/C)$-step




reduces to:



# Stability under the Cut Elimination

**Theorem (Stability of EPS)**

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

# Stability under the Cut Elimination

Theorem (Stability of EPS)

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

Theorem (Stability of EPN)

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

# Strong Cut Elimination

## Theorem

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

# Strong Cut Elimination

## Theorem

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

## Proof.

The proof is by lexicographic induction on the *cut complexity sequence*

$$\sharp 0, \sharp 1, ..., \sharp n$$

# Strong Cut Elimination

## Theorem

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

## Proof.

The proof is by lexicographic induction on the *cut complexity sequence*

$$\sharp 0, \sharp 1, \ldots, \sharp n$$

- $n$ is the number of Boolean variables occurring in $\langle \pi, E \rangle$;

# Strong Cut Elimination

## Theorem

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

## Proof.

The proof is by lexicographic induction on the *cut complexity sequence*

$$\sharp 0, \sharp 1, ..., \sharp n$$

- $n$ is the number of Boolean variables occurring in $\langle \pi, E \rangle$;
- $\sharp i$, with $0 \leq i \leq n$, is the sum of the logical complexities of all cuts whose *depth* is $i$.

# Strong Cut Elimination

### Theorem

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

### Proof.

The proof is by lexicographic induction on the *cut complexity sequence*

$$\sharp 0, \sharp 1, ..., \sharp n$$

- $n$ is the number of Boolean variables occurring in $\langle \pi, E \rangle$;
- $\sharp i$, with $0 \leq i \leq n$, is the sum of the logical complexities of all cuts whose *depth* is $i$.
- the depth $\delta(L)$ of a node $L$ is $\max(|w_1|, |w_2|)$, if
  - $w_1$ and $w_2$ are equivalent (modulo $E$) weights of $L$ and
  - $|w_j|$, for $j = 1, 2$, is the length (the number of possibly variables or negations of variables) of $w_j$.

# Confluence

## Theorem (local confluence)

*Let* $\langle \pi, E \rangle$ *be a proof net with two cut nodes,* $L_1$ *and* $L_2$*, and let*

- $\alpha$ *be the cut reduction* $\langle \pi, E \rangle \rightsquigarrow_{L_1} \langle \pi_1, E_1 \rangle$ *and*
- $\beta$ *be the cut reduction* $\langle \pi, E \rangle \rightsquigarrow_{L_2} \langle \pi_2, E_2 \rangle$*,*

*then there exists a proof net* $\langle \pi^*, E^* \rangle$ *which* $\langle \pi_i, E_i \rangle$*, for* $1 \leq i \leq 2$*, reduces to in at most one step.*

fine