



HAL
open science

Cyclic multiplicative-additive proof nets of linear logic with an application to language parsing

Vito Michele Vito Abrusci, Roberto Maieli

► **To cite this version:**

Vito Michele Vito Abrusci, Roberto Maieli. Cyclic multiplicative-additive proof nets of linear logic with an application to language parsing. 2015. hal-01120347

HAL Id: hal-01120347

<https://hal.science/hal-01120347>

Preprint submitted on 26 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cyclic multiplicative-additive proof nets of linear logic with an application to language parsing

Vito Michele Abrusci and Roberto Maieli
Department of Mathematics and Physics, Roma Tre University
Largo San Leonardo Murialdo 1 – 00146 Rome, Italy
{abrusci,maieli}@uniroma3.it

February 25, 2015

Abstract

This paper concerns a logical approach to natural language parsing based on proof nets (PNs), i.e. de-sequentialized proofs, of linear logic (LL). In particular, it presents a syntax for PNs of the cyclic multiplicative and additive fragment of linear logic (CyMALL). Any proof structure (PS), in Girards style, is weighted by boolean monomial weights, moreover, its conclusions Γ (a sequence of formulas occurrences) are endowed with a cyclic order σ , i.e., $\sigma(\Gamma)$. Naively, a CyMALL PS π with conclusions $\sigma(\Gamma)$ is correct if, for any slice $\varphi(\pi)$ (obtained by a boolean valuation φ of π) there exists an additive resolution (i.e. a multiplicative refinement of $\varphi(\pi)$) that is a CyMALL PN with conclusions $\sigma(\Gamma_r)$, where Γ_r is an additive resolution of Γ (i.e. a choice of an additive subformula for each formula of Γ). In its turn, the correctness criterion for CyMALL PNs can be considered as the non-commutative counterpart of the famous Danos-Regnier (DR) criterion for PNs of the pure multiplicative fragment (MALL) of LL. The main intuition relies on the fact that any DR-switching (i.e. any correction or test graph for a given PN) can be naturally viewed as a *seaweed*, i.e. a rootless planar tree inducing a cyclic order on the conclusions of the given PN. Unlike the most part of current syntaxes for non-commutative PNs our syntax allows a sequentialization for the full class of CyMALL PNs, without requiring these latter must be cut-free. Moreover, we give a characterization of CyMALL PNs for the extended (MALL) Lambek Calculus and thus a geometrical (non inductive) way to parse phrases or sentences. In particular additive Lambek PNs allow to parse phrases containing words with syntactical ambiguity (i.e. words with polymorphic type).

Contents

1	Introduction	3
1.1	Cyclic MALL	4
2	Cyclic MLL proof structures	4
2.1	Correctness	5
2.2	Cut reduction	6
2.3	Sequentialization	8
3	Cyclic MALL proof structures	11
3.1	Correctness	12
3.2	Cut reduction	13
3.2.1	Ready cut reduction	13
3.2.2	Commutative cut reduction	13
3.2.3	Stability of correctness under cut reduction	15
3.2.4	Strong cut-elimination	17
3.3	Sequentialization	17
4	Embedding Lambek Calculus into CyMALL PNs	18
5	Language parsing with Lambek CyMALL PNs	21
6	Conclusions and further works	23
6.1	Stability of sequentialization under cut reduction	23
6.2	Proof-structures with explicit n -ary links	23
6.3	Fully local cut reduction and parsing contraction	23
7	Complexity	24

1 Introduction

Proof nets (PNs) are one of the most innovative inventions of linear logic (LL, [Gir1987]): they are used to represent demonstrations in a geometric (i.e., non inductive) way, abstracting away from the technical bureaucracy of sequential proofs. Proof nets quotient classes of derivations that are equivalent up to some irrelevant permutations of inference rules instances.

Following this spirit, we present a syntax for PNs of the cyclic multiplicative and additive fragment of linear logic (CyMALL, Sections 1.1). This syntax, like the original Girard's one [Gir1996], is based on weighted (by boolean monomials) proof structures with explicit binary contraction links (Section 3). The conclusions Γ (a sequence of formulas occurrences) of a PS π are endowed with a cyclic order σ on Γ . Naively, a CyMALL PS π with conclusions $\sigma(\Gamma)$ is correct if, for any slice $\varphi(\pi)$ (obtained by a boolean valuation φ of π) there exists an additive resolution (a slight deformation of $\varphi(\pi)$) that is a CyMALL PN with conclusion $\sigma(\Gamma_r)$, where Γ_r is an additive resolution of Γ (i.e. a choice of an additive sub-formula for each formula of Γ). In its turn, correctness criterion for CyMALL PNs can be considered as the non-commutative counterpart of the famous Danos-Regnier criterion for proof nets of linear logic (see [DR1989] and [Dan1990]). The main intuition relies on the fact that any *DR-switching for a PS* (i.e. any correction or test graph, obtained by mutilating one premise of each disjunction ∇ -link) can be naturally viewed as a rootless planar tree, called *seaweed*, inducing a *cyclic ternary relation* on the conclusions of the given proof structure (Section 2.1).

Dislike some previous syntaxes for non-commutative logic, like e.g., [AR2000] and [Mai2003], this new syntax admits a *sequentialization* (i.e., a correspondence with sequential proofs) for the full class of CyMALL PNs including those ones with cuts (see Sections 2.2 and 2.3). Actually, unlike what happens in the standard (commutative) MLL case, the presence of cut links is "rather tricky" in the non-commutative case, since cut links are not equivalent, from a topological point of view, to tensor links; these latter make appear new conclusions that may disrupt the original (i.e., in presence of cut links) order of conclusions.

Moreover, CyMALL PNs satisfy both a simple convergent cut-elimination procedure in Laurent-Maieli's style [LM2008] (Section 3.2) and a sequentialization, in Girard's style (Section 3.3), stable under cut-reduction.

CyMALL can be considered as a conservative classical extension of Lambek Calculus (LC, see [Lam1958], [Abr2002] and [MR2012]) one of the ancestors of LL. The LC represents the first attempt of the so called *parsing as deduction*, i.e., parsing of natural language by means of a logical system. Following [AP1991], in LC parsing is interpreted as type checking in the form of theorem proving of Gentzen sequents. Types (i.e. propositional formulas) are associated to words in the lexicon; when a string $w_1 \dots w_n$ is tested for grammaticality, the types t_1, \dots, t_n associated with the words are retrieved from the lexicon and then parsing reduces to proving the derivability of a one-sided sequent of the form $\vdash t_n^\perp, \dots, t_1^\perp, s$, where s is the type associated with sentences. Moreover, forcing constraints on the Exchange rule, by e.g. allowing only *cyclic permutations* over sequents of formulas, gives the required computational control needed to view theorem proving as parsing in Lambek Categorical Grammar style. Anyway, LC parsing presents some syntactical ambiguity problems; actually, there may be:

1. (*non canonical proofs*) more than one cut-free proof for the same sequent;
2. (*lexical polymorphism*) more than one type associated with a single word.

Proof nets are commonly considered an elegant solution to the first problem of representing canonical proofs; in this sense, in Section 4, we give an *embedding* of extended MALL Lambek Calculus into Cyclic MALL PNs.

In Section 5 we propose a parsing approach based on CyMALL PNs that could be considered a step towards a proof-theoretical solution to the problem of lexical polymorphism; technically speaking, CyMALL proof nets allow to manage formulas (types) superposition (polymorphism) by means of the additive links $\&$ and \oplus . Following [Moo2002], we propose a parsing, by means of Lambek CyMALL PNs, of some sentences which make use of polymorphic words; naively, when a

word has two possible formulas A and B assigned then we can combine (or super-pose) these into a single additive formula $A\&B$.

1.1 Cyclic MALL

We briefly recall the necessary background of the *Cyclic MALL fragment* of LL, denoted *CyMALL*, without units (see [Abr2002]). We arbitrarily assume literals $a, a^\perp, b, b^\perp, \dots$ with a polarity: *positive* (+) for atoms, a, b, \dots and *negative* (−) a^\perp, b^\perp, \dots for their duals. A *formula* is built from literals by means of the two groups of connectives: *negative*, ∇ ("par") and $\&$ ("with") and *positive*, \otimes ("tensor") and \oplus ("plus"). For these connectives we have the following De Morgan laws: $(A \otimes B)^\perp = B^\perp \nabla A^\perp$, $(A \nabla B)^\perp = B^\perp \otimes A^\perp$, $(A \& B)^\perp = B^\perp \oplus A^\perp$, $(A \oplus B)^\perp = B^\perp \& A^\perp$. A CyMALL (resp., CyMLL) proof is any derivation tree built by the following (resp., only multiplicative) inference rules where sequents Γ, Δ are sequences of formulas occurrences endowed with a *total cyclic order* (lists with cyclic permutation) (see Definition 1):

$$\begin{array}{l}
\textit{identity:} \quad \frac{}{\vdash A, A^\perp} \textit{id} \qquad \frac{\vdash \Gamma, A \quad A^\perp \Delta}{\vdash \Gamma, \Delta} \textit{cut} \\
\textit{multiplicatives:} \quad \frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta} \otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \nabla B} \nabla \\
\textit{additives:} \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \& \qquad \frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus_i A_2} \vdash \oplus_{i=1,2}
\end{array}$$

Naively, a *total cyclic order* can be thought as follows; consider a set of points of an oriented circle; the orientation induces a total order on these points as follows: if a, b and c are three distinct points, then b is either between a and c ($a < b < c$) or between c and a ($c < b < a$). Moreover, $a < b < c$ is equivalent to $b < c < a$ or $c < a < b$.

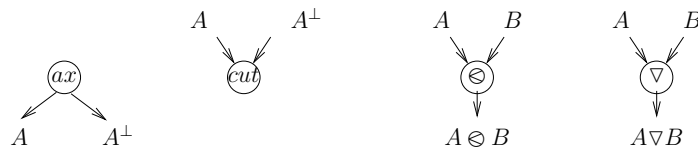
Definition 1 (total cyclic order) A total cyclic order is a pair (X, σ) where X is a set and σ is a ternary relation over X satisfying the following properties:

1. $\forall a, b, c \in X, \sigma(a, b, c) \rightarrow \sigma(b, c, a)$ (cyclic);
2. $\forall a, b \in X, \neg \sigma(a, a, b)$ (anti-reflexive);
3. $\forall a, b, c, d \in X, \sigma(a, b, c) \wedge \sigma(c, d, a) \rightarrow \sigma(b, c, d)$ (transitive);
4. $\forall a, b, c \in X, \sigma(a, b, c) \vee \sigma(c, b, a)$ (total).

Negative (or *asynchronous*) connectives correspond to *true determinism* in the way we apply bottom-up their corresponding inference rules. In particular, observe that Γ must appear as the same context (with same order) in both the premises the $\&$ -rule. Vice-versa, positive (or *synchronous*) connectives correspond to *true non-determinism* in the way we apply bottom-up their corresponding rules; in particular, there is not deterministic the way to split the context into Γ, Δ in the \otimes rule, as well there not exist a deterministic choice in the way we apply bottom up \oplus_1 or \oplus_2 rules.

2 Cyclic MLL proof structures

Definition 2 (CyMLL proof structure) A CyMLL proof structure (PS) is an oriented graph π , in which edges are labeled by formulas and nodes are labeled by connectives of CyMLL, built by juxtaposing the following special graphs, called links, in which incident (resp., emergent) edges are called premises (resp., conclusions):



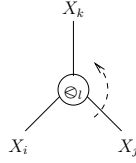
In a PS π each premise (resp., conclusion) of a link must be conclusion (resp., premise) of exactly (resp., at most) one link of π . We call conclusion of π any emergent edge that is not premises of any link.

2.1 Correctness

We characterize those proof structures that are images of proofs. Actually, there exist several syntaxes for CyMLL proof nets, like those ones of [AR2000] and [Mai2003]; for sequentialization reasons we prefer the latter one.

Definition 3 (switchings and seaweeds) Assume π is a CyMLL PS with conclusions Γ .

- A Danos-Regnier switching S for π , denoted $S(\pi)$, is the non oriented graph built on nodes and edges of π with the modification that for each ∇ -node we take only one premise, that is called left or right ∇ -switch.
- Let $S(\pi)$ be an acyclic an connected switching for π ; $S(\pi)$ is the rootless planar tree whose nodes are labeled by \ominus -nodes, and whose leaves X_1, \dots, X_n are the terminal (pending) edges of $S(\pi)$; $S(\pi)$ is a ternary relation, called seaweed, with support X_1, \dots, X_n ; an ordered triple (X_i, X_j, X_k) belongs to the seaweed $S(\pi)$ iff:
 - the intersection of the three paths X_iX_j , X_jX_k and X_kX_i is the node \ominus_l ;
 - the three paths $X_i\ominus_l$, $X_j\ominus_l$ and $X_k\ominus_l$ are in this cyclic order while moving anti-clockwise around the \ominus -node as below

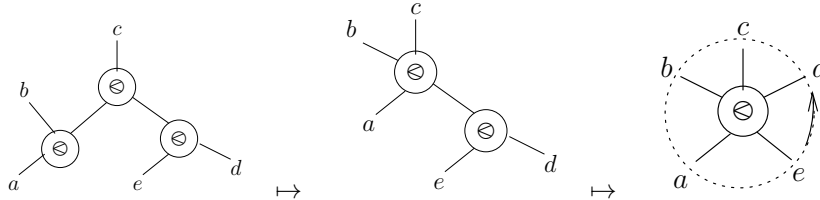


If A is an edge of the seaweed $S(\pi)$, then $S_i(\pi) \downarrow^A$ is the restriction of the seaweed $S(\pi)$, that is, the sub-graph of $S(\pi)$ obtained as follows:

1. disconnect the graph below (w.r.t. the orientation of π) the edge A ;
2. delete the graph not containing A .

Fact 1 (seaweeds as cyclic orders) Any seaweed $S(\pi)$ can be viewed as a cyclic total order (Definition 1) on its support X_1, \dots, X_n ; in other words, if a triple $(X_i, X_j, X_k) \in S(\pi)$, then $X_i < X_j < X_k$ are in cyclic order.

Intuitively, we may contract a seaweed (by associating the \ominus -nodes) until it collapses into single n -ary \ominus -node with n pending edges (its support), as follows:

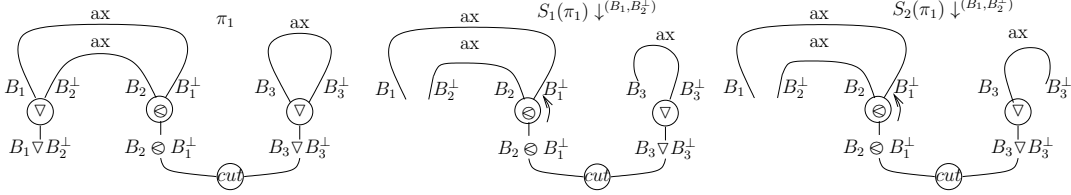


Definition 4 (CyMLL proof net) A CyMLL PS π is correct, i.e. it is a CyMLL proof net (PN), iff:

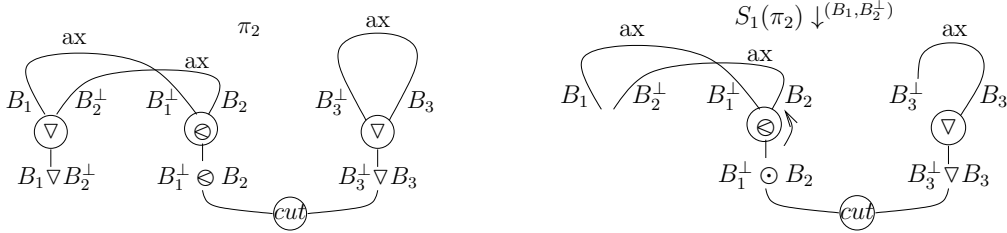
1. π is a standard MLL PN, that is, any switching $S(\pi)$ is a connected and acyclic graph (therefore, a seaweed);

2. for any ∇ -link $\frac{A \ B}{A \nabla B}$ the triple (A, B, C) must occur in this cyclic order in any seaweed $S(\pi)$ restricted to A, B , i.e., $(A, B, C) \in S(\pi) \downarrow^{(A, B)}$, where C is any pending leaf (if it exists) in the support of the restricted seaweed.

Example 1 We give an instance of CyMLL proof net π_1 with, in particular, the two restricted seaweeds, $S_1(\pi_1) \downarrow^{(B_1, B_2^\perp)}$ and $S_2(\pi_1) \downarrow^{(B_1, B_2^\perp)}$, both satisfying condition 2 of Definition 4.



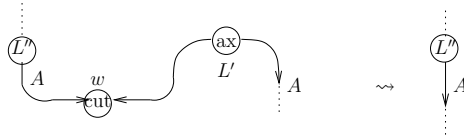
Vice-versa, the following instance of proof structures π_2 is not correct (it is not a proof net), since condition 2 of Definition 4 is violated: there exists a ∇ -link $\frac{B_1 \ B_2^\perp}{B_1 \nabla B_2^\perp}$ and a seaweed $S_1(\pi_2)$ s.t. ∇C pending, $(B_1, B_2^\perp, C) \in S_1(\pi_2) \downarrow^{(B_1, B_2^\perp)}$; actually, if we take $C = B_3^\perp$ then $(B_1, C, B_2^\perp) \in S_1(\pi_2) \downarrow^{(B_1, B_2^\perp)}$ as follows



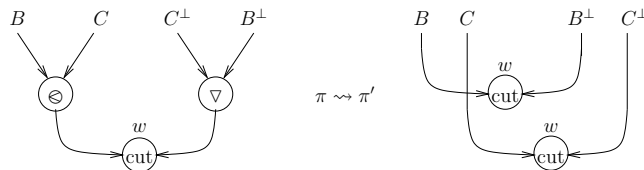
2.2 Cut reduction

Definition 5 (multiplicative cut reduction) Let L be a cut link in a proof net π whose premises A and A^\perp are, resp., conclusions of links L', L'' . Then we define the result π' (called reductum) of reducing this cut in π (called redex), as follows:

Ax-cut: if L' (resp., L'') is an axiom link then π' is obtained by removing in π both formulas A, A^\perp (as well as L) and giving as new conclusion to L'' (resp., L') the other conclusion of L' (resp., L'').



(\otimes/∇)-cut: if L' is a \otimes -link with premises B and C and L'' is a ∇ -link with premises C^\perp and B^\perp , then π' is obtained by removing in π the formulas A and A^\perp as well as cut link L with L' and L'' and by adding two new cut links with, resp., premises B, B^\perp and C, C^\perp , as follows:

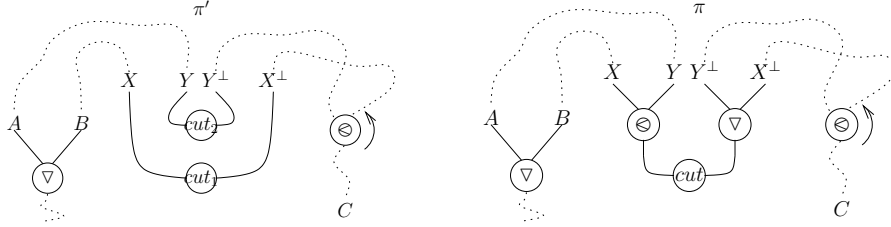


Theorem 1 (stability of PNs under cut reduction) *If π is a CyMLL PN that reduces to π' in one step of cut reduction, $\pi \rightsquigarrow \pi'$, then π' is a CyMLL PN.*

PROOF — Observe that w.r.t. Definition 4 condition 1 follows as an almost immediate consequence of the next graph theoretical property ([Gir2006], pages 250-251):

Property 1 (Euler-Poincaré invariance) *Given a graph \mathcal{G} , then $(\#CC - \#Cy) = (\#V - \#E)$, where $\#CC$, $\#Cy$, $\#V$ and $\#E$ denotes the number of, respectively, connected components, cycles, vertexes and edges of \mathcal{G} .*

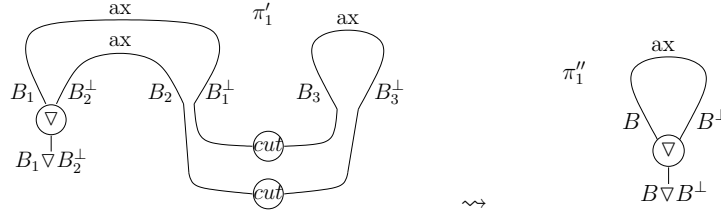
Condition 2 of Definition 4 follows by calculation. Assume π reduces to π' after the reduction of a cut between $(X \otimes Y)$ and $(Y^\perp \nabla X^\perp)$ and assume, by absurdum, there exist a ∇ -link labeled by a formula $A \nabla B$ s.t. the triple (A, C, B) occurs in this wrong cyclic order in a seaweed $S(\pi')$ restricted to A, B , $S(\pi') \downarrow^{(A,B)}$, for a pending leave C occurring in this restriction, i.e., $(A, C, B) \in S(\pi') \downarrow^{(A,B)}$. Then, two of the three paths $A \otimes$, $B \otimes$ and $C \otimes$ must go through (i.e., they must contain) the two (sub)cut-links, $cut_1 \frac{X}{X^\perp}$ and $cut_2 \frac{Y}{Y^\perp}$, resulting from the cut reduction, otherwise π would already be violating condition 2 of Definition 4; so, assume path $B \otimes$ goes through cut_1 and path $A \otimes$ goes through cut_2 , like in the next left hand side picture



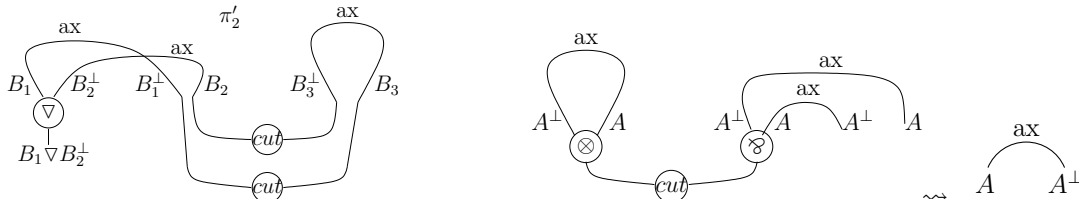
This means there exist a seaweed $S'(\pi)$, a link $Y^\perp \nabla X^\perp$ and a triple (Y^\perp, C, X^\perp) s.t. $(Y^\perp, C, X^\perp) \in S'(\pi) \downarrow^{(Y^\perp, X^\perp)}$, violating condition 2 and so contradicting correctness of π (see the right hand side picture given before; observe that, since any switching of π is acyclic, deleting the subgraph below $Y^\perp \nabla X^\perp$ does not make disappear C).

The remaining case when path $C \otimes$ goes through cut_1 (resp., through cut_2) and either path $A \otimes$ or path $B \otimes$ goes through cut_2 (resp., through cut_2) is treated similarly and so omitted. \square

Example 2 *Observe that w.r.t. Example 1, π_1 reduces to π'_1 and so π'_1 to π''_1 below; both π'_1 and π''_1 are correct since condition 2 of Definition 4 is void in both π'_1 and π''_1 :*



Moreover, w.r.t. Example 1, π_2 is a non correct PS that reduces to a correct one, π'_2 , after a cut reduction step (see the left hand side picture below); this is an already well known phenomenon in the standard MLL case where we can easily find non correct MLL PSs that become correct after cut reduction, like that one on the right hand side below:



We use indexed formulas B_1, B_2, B_3 to distinguish different occurrences of B .

Cut reduction is trivially *convergent* (i.e., terminating and confluent).

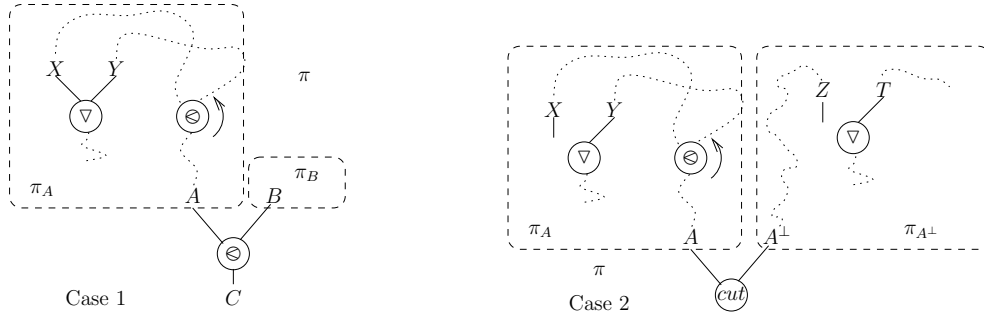
2.3 Sequentialization

We show a correspondence (*sequentialization*) between PNs and sequential proofs.

Lemma 1 (splitting) *Let π be a CyMLL PN with at least a \otimes -link or cut-link and with conclusions Γ not containing any terminal ∇ -link (so, we say π is in splitting condition); then, there must exist a \otimes -link $\frac{A \ B}{A \otimes B}$ (resp., a cut-link $\frac{A \ A^\perp}{A \ \text{cut}}$) that splits π into two CyMLL PNs, π_A and π_B (resp., π_A and π_{A^\perp}).*

PROOF — Assume π is a CyMLL PN in splitting condition, then by the Splitting Lemma for standard commutative MLL PNs ([Gir1987]) π must split either at a \otimes -link or a *cut*-link. We reason according to these two cases.

1. Assume π splits at $\frac{A \ B}{A \otimes B}$ in two components π_A and π_B ; we know that both components satisfy condition 1 (they are MLL PNs); assume by absurdum π_A is not a CyMLL PN, i.e., π_A violates condition 2 of Definition 4. This means there exists a $\frac{X \ Y}{X \ \nabla \ Y}$ and a restricted seaweed $S(\pi_A) \downarrow^{(X,Y)}$ containing the triple X, A, Y in the wrong order, i.e., $(X, A, Y) \in S(\pi_A) \downarrow^{(X,Y)}$ like in Case 1 of next figure. But then there exists a restricted seaweed $S(\pi) \downarrow^{(X,Y)}$ containing X, Y and C (where $C = A \otimes B$) in the wrong cyclic order, i.e., $(X, C, Y) \in S(\pi) \downarrow^{(X,Y)}$, contradicting the correctness of π .
2. Assume π splits at the cut link $\frac{A \ A^\perp}{A \ \text{cut}}$ in two components π_A and π_{A^\perp} ; assume by absurdum π_A is not a CyMLL PN, hence π_A must be violating condition 2 of Definition 4. Moreover, assume π is such a minimal PN in cut-splitting condition whose subproof π_A is not a CyMLL PN. This means, as before, there exists a $\frac{X \ Y}{X \ \nabla \ Y}$ and a restricted seaweed $S(\pi_A) \downarrow^{(X,Y)}$ containing the triple X, A, Y in the wrong order, i.e., $(X, A, Y) \in S(\pi_A) \downarrow^{(X,Y)}$ like in Case 2 of next figure. Then, by correctness π , π_{A^\perp} must have A^\perp as unique conclusion, otherwise we can find a restricted seaweed for π , $S(\pi) \downarrow^{(X,Y)}$, containing a triple X, C, Y with wrong order for a conclusion $C \neq A^\perp$; moreover, π_{A^\perp} cannot contain any cut, otherwise, by Theorem 1, we could replace in π the redex π_{A^\perp} by its reductum π'_{A^\perp} , contradicting the minimality of π . Now, observe the equality $\#\nabla - \#\otimes = 1$, relating the number of \otimes -nodes with the number of ∇ -nodes, holds for any cut free proof net with an unique conclusion. Therefore, π_{A^\perp} must contain at least a ∇ -link, let us say $\frac{Z \ T}{Z \ \nabla \ T}$. But then we can easily find a restricted seaweed for π , $S(\pi) \downarrow^{(X,Y)}$, and a triple (X, Z, Y) occurring in $S(\pi) \downarrow^{(X,Y)}$ with the wrong cyclic order, contradicting the correctness of π , like in Case 2 of the picture below:



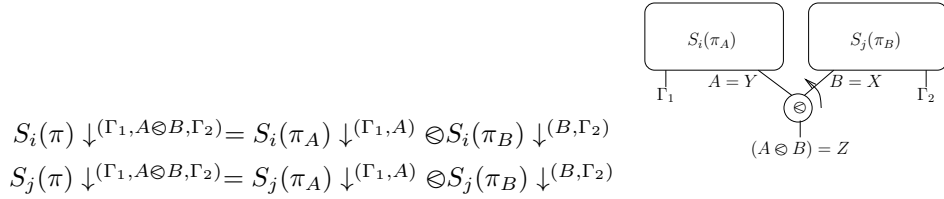
□

Lemma 2 (PN cyclic order conclusions) *Let π be a CyMLL PN with conclusions Γ , then all seaweeds $S_i(\pi) \downarrow^\Gamma$, restricted to Γ , induce the same cyclic order σ on Γ , denoted $\sigma(\Gamma)$ and called the (cyclic) order of the conclusions of π .*

PROOF — By induction on the size of π , i.e. $\langle \#edges, \#vertices \rangle^1$.

1. If π is reduced to an axiom link, then obvious.
2. π contains at least a conclusion $A \nabla B$, then $\Gamma = \Gamma', A \nabla B$; by hypothesis of induction the sub-proof net π' with conclusion Γ', A, B has cyclic order $\sigma(\Gamma', A, B)$, and so, by condition 2 of Definition 4 applied to π , we know that each restricted seaweed $S_i(\pi) \downarrow^{(\Gamma', A, B)}$ induces the same cyclic order $\sigma(\Gamma', A, B)$; finally, by substituting $[A/A \nabla B]$ (resp., $[B/A \nabla B]$) in the restriction $S_i(\pi) \downarrow^{(\Gamma', A)}$ (resp., $S_i(\pi) \downarrow^{(\Gamma', B)}$), we get that each seaweed $S_i(\pi) \downarrow^{(\Gamma', A \nabla B)}$ induces the same cyclic order $\sigma(\Gamma', A \nabla B)$.
3. Otherwise π must contain a terminal splitting \otimes or *cut* link. Assume π contains a splitting \otimes -link, $\frac{A}{A \otimes B}$, and assume by absurdum that π is such a minimal (w.r.t. the size) PN with at least two seaweeds $S_i(\pi)$ and $S_j(\pi)$ s.t. $(X, Y, Z) \in S_i(\pi)$ and $(X, Y, Z) \notin S_j(\pi)$. We reason according the following two sub-cases.

- (a) It cannot be the case $X = B, Y = A$ and $Z = C$ otherwise, by definition of seaweeds, $S_i(\pi)$ and $S_j(\pi)$ will be as follows:

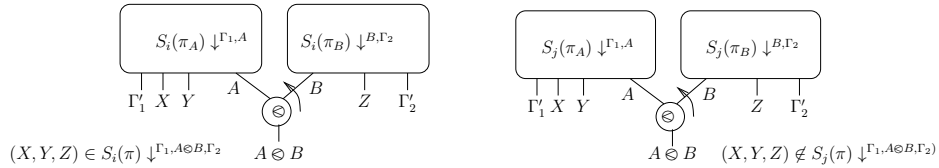


Now, by hypothesis of induction, all seaweeds on π_A (resp., all seaweeds on π_B) induce the same order on Γ_1, A (resp., Γ_2, B), then in particular,

$$S_i(\pi_A) \downarrow^{(\Gamma_1, A)} = S_j(\pi_A) \downarrow^{(\Gamma_1, A)} \quad \text{and} \quad S_i(\pi_B) \downarrow^{(B, \Gamma_2)} = S_j(\pi_B) \downarrow^{(B, \Gamma_2)}$$

but this implies $S_i(\pi) \downarrow^{(\Gamma_1, A \otimes B, \Gamma_2)} = S_j(\pi) \downarrow^{(\Gamma_1, A \otimes B, \Gamma_2)}$.

- (b) So, assume both X and Y belong to π_A (resp., π_B) and Z belongs to π_B (resp., π_A); moreover, assume for some i, j , $(X, Y, Z) \in S_i(\pi) \downarrow^{(\Gamma_1, Z, \Gamma_2)}$ and $(X, Y, Z) \notin S_j(\pi) \downarrow^{(\Gamma_1, Z, \Gamma_2)}$; by Splitting Lemma 1, each seaweed for π must appear as follows, in particular $S_i(\pi)$ and $S_j(\pi)$:



then, by restriction on seaweeds, $(X, Y, A) \in S_i(\pi_A) \downarrow^{\Gamma_1, A}$ and $(X, Y, A) \notin S_j(\pi_A) \downarrow^{\Gamma_1, A}$, contradicting the assumption (by minimality) that π_A is a correct PN with a cyclic order on its conclusions $\Gamma_1, X, Y, A = \Gamma_1, A$.

The remaining case, π contains a splitting cut, is similar and so omitted.

□

Next Corollary states that Lemma 2 is preserved by cut reduction.

Corollary 1 (stability of PN order conclusions under cut reduction) *If π , with conclusions $\sigma(\Gamma)$, reduces in one step of cut reduction to π' , then also π' has conclusions $\sigma(\Gamma)$.*

¹Number of edges and number of vertices.

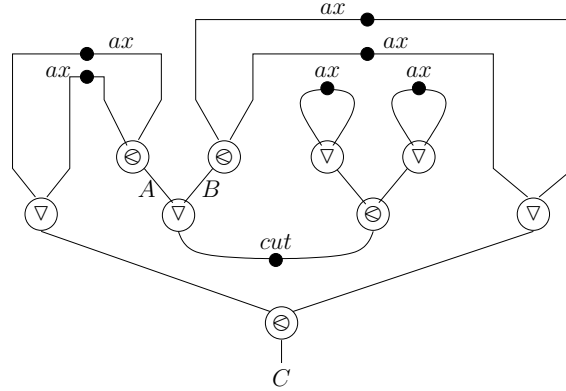
Theorem 2 (adequacy of CyMLL PN) *Any CyMLL proof of a sequent $\sigma(\Gamma)$ de-sequentializes into a CyMLL PN with same conclusions $\sigma(\Gamma)$.*

PROOF — By induction on the height of the given sequent proof of $\sigma(\Gamma)$. □

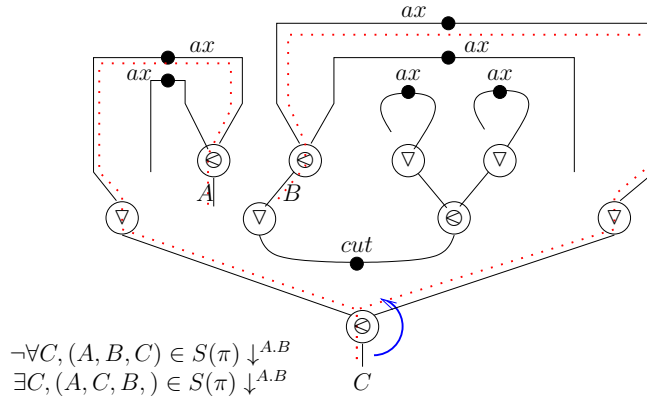
Theorem 3 (sequentialization of CyMLL PN) *Any CyMLL PN with conclusions $\sigma(\Gamma)$ sequentializes into a CyMLL sequent proof with same cyclic conclusions $\sigma(\Gamma)$.*

PROOF — by induction on the size of the given proof net π via Lemmas 1 and 2. □

Example 3 (Mellies’s proof structure) *Observe that, dislike what happens in the commutative MLL case, the presence of cut links is ”quite tricky” in the non-commutative case, since cut links are not equivalent, from a topological point of view, to tensor links: these latter make appear new conclusions that may disrupt the original (i.e., in presence of cut links) order of conclusions. By the way, unlike the most part of correctness criteria for non-commutative proof nets, our syntax enjoys a sequentialization for the full class of CyMLL PN without assuming these must be cut-free. In particular the following Mellies’s proof structure², given a page 224 of [MR2012],*



is not correct according to our correctness criterion (thus it is not sequentializable) since there exists a $\frac{A}{A \nabla B}$ link and a switching $S(\pi)$ s.t. $\neg \forall C, (A, B, C) \in S(\pi) \downarrow^{(A, B)}$, contradicting condition 2 of Definition 4: following the crossing red dotted lines of next figure you can easily verify $\exists C, (A, C, B) \in S(\pi) \downarrow^{(A, B)}$.

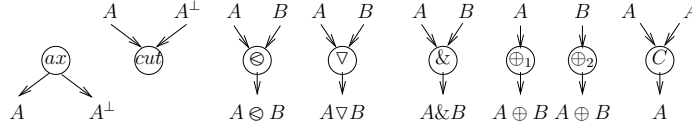


Observe, Mellies’s PS becomes correct (and so sequentializable) after cut reduction. Our criterion satisfies, indeed, all four requirements that, according to page 223 of [MR2012], any ”satisfactory” criterion should enjoy.

²In the related literature, this example is commonly considered a ”measure of the satisfiability degree” of correctness criteria of non commutative proof nets. Let us say, a ”good” correctness criterion for non-commutative logic should not recognize this structure as a correct proof net.

3 Cyclic MALL proof structures

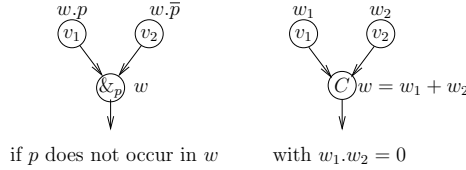
Definition 6 (CyMALL proof structure) A CyMALL proof structure (PS) is an oriented graph π , in which edges are labeled by formulas while nodes are labeled by connectives of MALL, built by juxtaposing the following special graphs, called link, in which incident (resp., emergent) edges are called premises (resp., conclusions):



In a PS π each premise (resp., conclusion) of a link must be conclusion (resp., premise) of exactly (resp., at most) one link of π . Finally, we call conclusion of π any emergent edge that is not premises of any link.

Definition 7 (Girard CyMALL proof structure) A Girard proof structure (GPS) is a PS with weights associated as follows (weights assignment):

1. first we associate a boolean variable, called eigen weight p , to each $\&$ -node (eigen weights are supposed to be different);
2. then we associate a weight, a product of (negation of) boolean variables ($p, \bar{p}, q, \bar{q}, \dots$) to each node, with the constraint that two nodes have the same weight if they have a common edge, except when the edge is the premise of a $\&$ or C -node, in these cases we do like below:



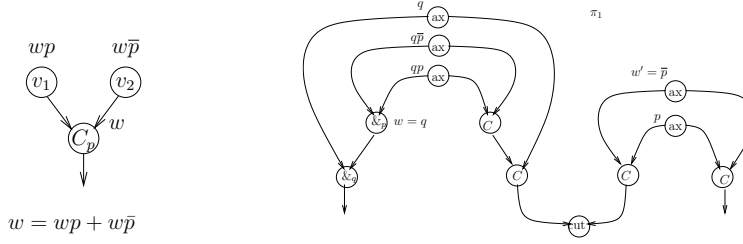
3. a conclusion node has weight 1;
4. if w is the weight of a $\&$ -node, with eigen weight p , and w' is a weight depending on p and appearing in the proof structure then $w' \leq w$ (we say that a weight w depends on p when p or \bar{p} occurs in w).

Finally we say that:

- a node L with weight w depends on the eigen weight p if w depends on p or L is a C -node and one of the weights just above it depends on p ;
- the signature of a weight w is the set of variables occurring in w ; e.g., the following weights have the same signature $\{p, q\}$: $pq, \bar{p}q, p\bar{q}$ and $\bar{p}\bar{q}$.

Remark 1 Observe that:

1. – since the weights associated to a PS are products (monomials) of the Boolean algebra generated by the eigen weights associated to a proof structure, then, for each weight w associated to a binary contraction node, there exists a unique eigen weight p that splits w into $w_1 = wp$ and $w_2 = w\bar{p}$. We sometimes index a C -link with its splitting variable p , like in the left hand side picture below;
2. – the graph π_1 is not a GPS since it violates condition 4 of Definition 7; actually, if $w = q$ is the weight of the $\&$ -link and $w' = \bar{p}$ is a weight depending on p and appearing in the proof-structure then $\bar{p} \not\leq q$.



3.1 Correctness

Definition 8 (slices, switchings, resolutions) Let π be a CyMALL GPS.

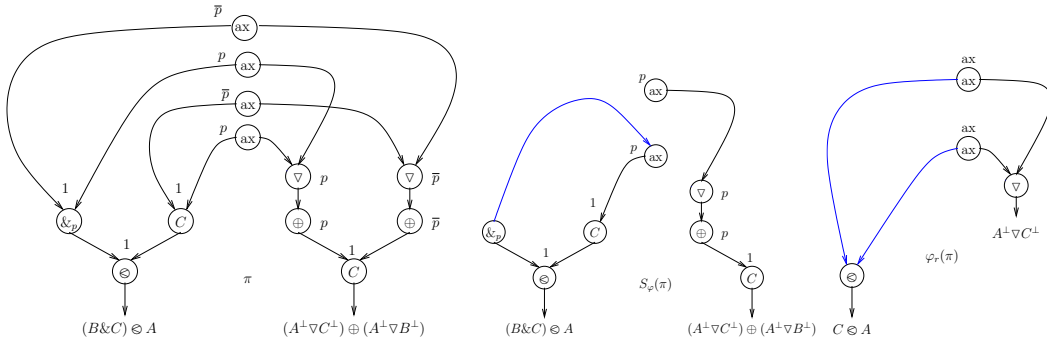
- A valuation φ of π is a function from the set of all weights of π into $\{0, 1\}$.
- Fixed a valuation φ for π , the slice $\varphi(\pi)$ is the graph obtained from π by keeping only those nodes with weight 1 together with its incident edges.
- Fixed a slice $\varphi(\pi)$ an additive switching is a multiplicative switching of the slice $\varphi(\pi)$, denoted $S_\varphi(\pi)$, in which for each $\&_p$ -node we erase the (unique) premise in $\varphi(\pi)$ and we add an oriented edge, called jump, from the $\&_p$ -node to an L-node whose weight depends on the eigen weight p of the $\&_p$ -node.
- An additive resolution $\varphi_r(\pi)$ for a slice $\varphi(\pi)$ is the graph obtained by replacing in $\varphi(\pi)$ each unary link L (a link that after the valuation has a single premise) by a single edge labeled by the formula labeling the unique premise of L . In particular, each conclusion of $\varphi_r(\pi)$ is labeled by a multiplicative formula.

We call *additive resolution* of a CyMALL sequent Γ what remains of Γ after erasing one subformula in each additive formula occurrence of Γ .

Definition 9 (CyMALL proof net) We call correct (or proof net (GPN)) any CyMALL GPS π whose conclusions Γ are endowed with a cyclic order $\sigma(\Gamma)$ and s.t. for any valuation φ of π :

1. each additive switching $S_\varphi(\pi)$ is an acyclic and connected graph (ACC);
2. there exists an additive resolution $\varphi_r(\pi)$ for $\varphi(\pi)$ that is a CyMALL PN with cyclic order conclusions $\sigma(\Gamma_r)$, where Γ_r is an additive resolution of Γ .

Example 4 Observe, the proof structure π on the left hand side below is not correct: actually, fixed a valuation φ s.t. $\varphi(p) = 1$, there exists an additive switching $S_\varphi(\pi)$ (with a jump) that is not ACC (see the middle side figure). Nevertheless, any slice $\varphi(\pi)$ is ACC; indeed, for each slice $\varphi(\pi)$ there exists an additive resolution $\varphi_r(\pi)$ that is a CyMALL PN like that one, on the rightmost hand side, with conclusion $C \otimes A, A^\perp \nabla C^\perp$ that is an additive resolution of the sequent $(B \& C) \otimes A, (A^\perp \nabla C^\perp) \oplus (A^\perp \nabla B^\perp)$.

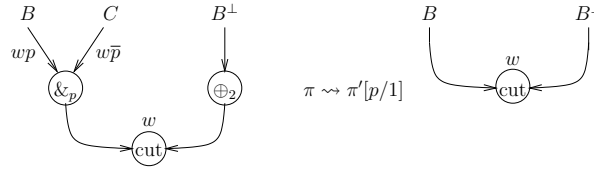


3.2 Cut reduction

3.2.1 Ready cut reduction

Definition 10 (additive ready cut reduction) Let L be a cut in a proof net π whose premises A and A^\perp are the respective conclusions of links L', L'' both different from the contraction C . Then we define the result π' (reductum) of reducing this additive ready cut in π (redex), as follows:

($\&/\oplus$)-cut: if L' is a $\&_p$ -link with premises B and C and L'' is a \oplus_1 -link (resp., a \oplus_2 -link) with premise B^\perp (resp., C^\perp), then π' is obtained in three steps: first remove in π both formulas A, A^\perp as well as the cut link L with L' and L'' , then replace the eigen weight p by 1 (resp., by 0) and keep only those links (vertexes and edges) that still have nonzero weight; finally we add a cut between B and B^\perp (resp., between C and C^\perp).

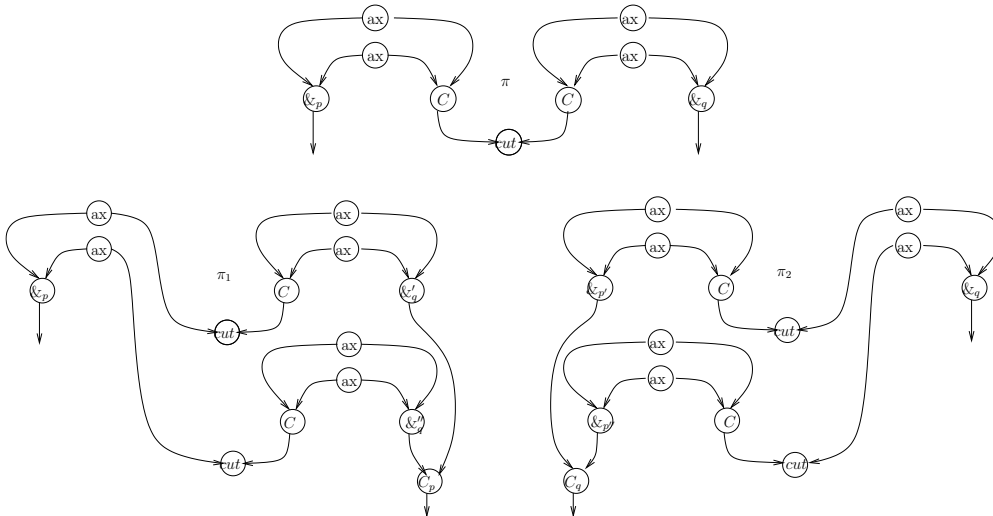


Theorem 4 (stability of GPN under ready cut reduction) Assume π is GPN that reduces to π' in one step of multiplicative (Definition 5) or additive (Definition 10) ready cut reduction, then π' is GPN too.

PROOF — Stability of multiplicative ready cut reduction (Definition 5) follows by Theorem 1. Stability of additive ready cut reduction (Definition 10) is immediate. Trivially, conditions 1 and 2 of Definition 9 are preserved by additive ready cut reduction steps. \square

3.2.2 Commutative cut reduction

(The confluence problem). In general, reducing a cut involving a contraction link as (at least) one of its premises from a proof structure π may lead to several solutions, depending on which subgraph of π we decide to duplicate. For instance, as illustrated in the following picture, reducing the commutative cut of π may lead to solution π_1 as well solution π_2 , depending on which additive box ($\&_q$ or $\&_p$) we decide to duplicate. Of course these two resulting proof nets are different and there is no a-priori way to make them equal. Girard does not give in [Gir1996] a solution to this problem which has been proposed later by Laurent and Maieli in [LM2008]. In the following we present a commutative cut reduction that simplifies this latter.



Definition 11 (empire and spreading) Assume a proof structure π , an eigen weight p and a weight w , then:

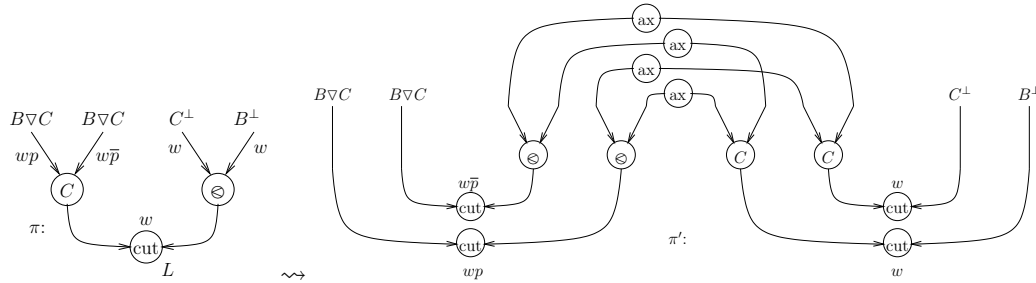
- dependency graph of p w.r.t. π , denoted \mathcal{E}_p , is the (possibly disconnected) subgraph of π made by all links depending on p .
- the spreading of w over π , denoted by $w.[\pi]$, is the product of w for π , i.e., π where we replaced each weight v with the product of weights wv .

Lemma 3 (dependency graph) If a $\&_p$ -node belongs to the dependency graph \mathcal{E}_q , then $\mathcal{E}(p) \subseteq \mathcal{E}(q)$.

PROOF — If the $\&_p$ -node belongs to \mathcal{E}_q then the weight w of the $\&_p$ -node depends on q (i.e., $w = w'q$ or $w = w'\bar{q}$) then trivially, by the dependency condition 4 of Definition 7, each node v whose weight depends on p will also depends on q . \square

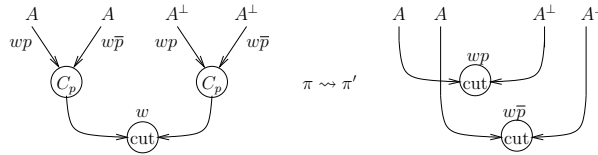
Definition 12 (commutative cut reduction) Let L be cut link in a proof net π whose premises A and A^\perp are the respective conclusions of links L' and L'' s. t. at least one of them is a contraction link C . Then we define the result π' (reductum) of reducing this commutative cut L in π (redex), as follows:

(C/⊗)-cut: if L' is a C-link and L'' is a ⊗-link, then π reduces in one (C/⊗) step to π' (the (C/∇) step is analogous) as follows:

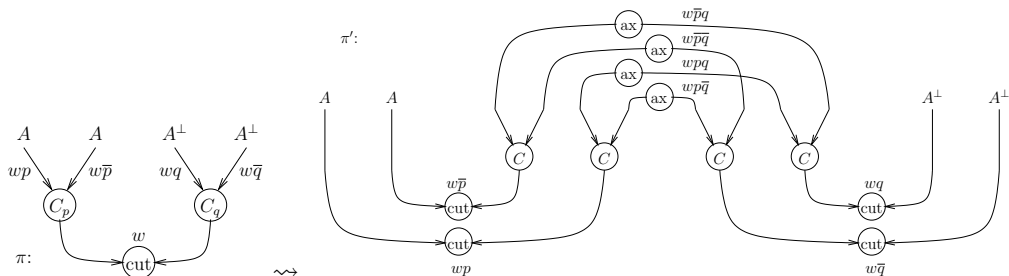


(C/C)-cut: if both L' and L'' are both C-links, then there are two cases:

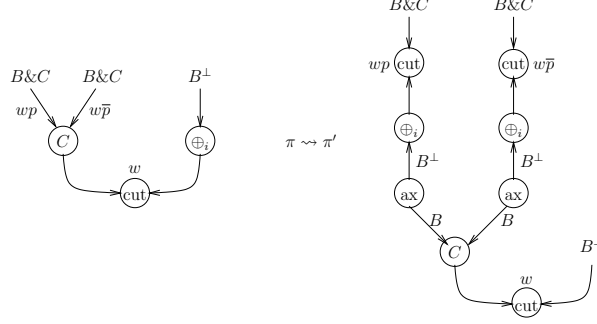
1. either the weight w of both L' and L'' splits on a same p variable, then π reduces in one (C_p/C_p) step to π' as follows



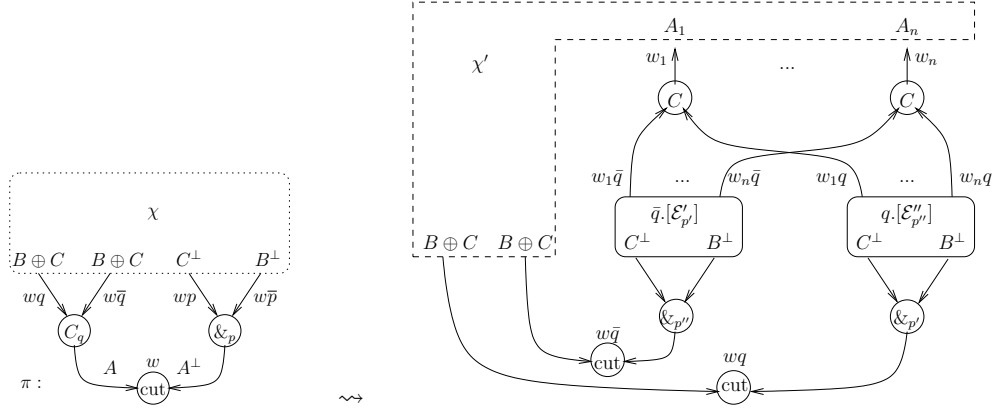
2. or the weight w of L' , resp., L'' , is split by two different variables, p and q , then π reduces in one (C_p/C_q) step to π' , like follows



(C/\oplus_i) -cut: if L' is a C -link and L'' a $\oplus_{i=1,2}$ -link, then π reduces in one (C/\oplus) step to π' , as follows



$(C/\&_p)$ -cut: if L' is a C -link and L'' a $\&_p$ -link, then π reduces in one $(C/\&)$ step to π' , like follows



with the assumptions that graphs $\bar{q}.\mathcal{E}'_{p'}$ and $q.\mathcal{E}''_{p''}$ are obtained as follows:

1. we take two copies, \mathcal{E}'_p and \mathcal{E}''_p , of the dependency graph of p , \mathcal{E}_p ;
2. we replace in \mathcal{E}'_p (resp., in \mathcal{E}''_p) the variable p with a new p' (resp., p'');
3. we spread \bar{q} (resp., q) over $\mathcal{E}'_{p'}$ (resp., over $\mathcal{E}''_{p''}$).

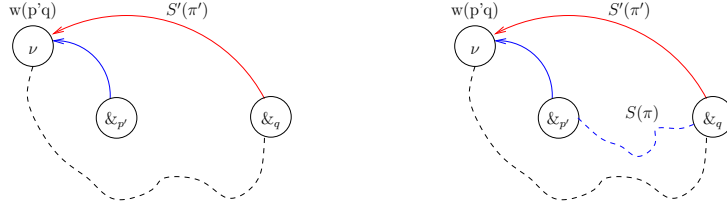
3.2.3 Stability of correctness under cut reduction

Theorem 5 (stability under commutative cut reduction: I part) *If π is a proof net s.t. it reduces to π' in one step of commutative cut reduction that is different from the $C/\&$ -case, then π' is still a proof net.*

PROOF — (sketch) All cases are more or less immediate consequences of the Euler-Poincaré Property 1. Trivially, conditions 1 and 2 of Definition 9 are preserved by these commutative cut reduction steps. \square

Lemma 4 (conservative/extensive switching) *Assume π is a proof net that reduces to π' in one step of commutative cut reduction $C_q/\&_p$. We say that a switching S' for π' is extensive (resp., conservative) w.r.t. q if there exists (resp., there do not exist) in S' at least a jump going from the $\&_q$ -node to a node that was not depending on q in π . Then, for any extensive (resp., conservative) switching for π' that is not ACC we can find a corresponding switching for π that is not ACC too.*

PROOF — (sketch) Assume π reduce to π' after a commutative cut reduction $C_q/\&_p$ and assume there exists a cycle in an extensive switching $S'(\pi')$ like the left hand side picture below



This cycle goes through a node ν that depends on both variables p' and q , where p' is a fresh variable copying p of π , according to the definition $C_q/\&\epsilon_p$ cut reduction step. Then we can easily find a switching for $S(\pi)$ with a cycle like the blue-black one of the following picture: replace in π , the fresh variable p' of π' with the original one p . Similarly, if we assume that $S'(\pi')$ is not connected. \square

Lemma 5 (separation) *Assume π is a proof net containing two nodes, $\&\epsilon_{p'}$ and $\&\epsilon_{p''}$, with weights, resp., $w'q$ and $w''\bar{q}$, for some eigen weight q of π . Then, there cannot exist in π a node whose weight depends both on p' and on p'' .*

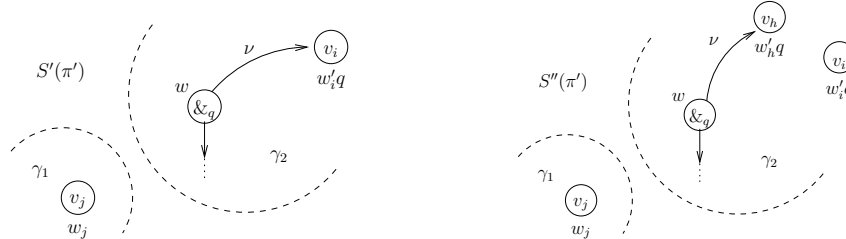
PROOF — By absurdum, assume a node v whose weight w depends both on p' and p'' ; let's say, for instance, $w = w_1.p'.p''$. Now fix an evaluation φ for π s.t. $\varphi(w) = 1$, then by the dependency condition 4 of Definition 7, we have both $w_1.p'.p'' \leq w'q$ and $w_1.p'.p'' \leq w''\bar{q}$; this is only possible when $\varphi(w) = 0$, contradicting $\varphi(w) = 1$. \square

Theorem 6 (stability under commutative cut reduction: II part) *If π is a proof net that reduces to π' in one step of cut reduction $C/\&$, then π' is a proof net too.*

PROOF — (*sketch*) First observe that each cut reduction step $C_q/\&\epsilon_p$ preserves the property of being a proof structure. This follows by construction of π' . In particular Lemmas 3 and 5, ensure that we can safely rename the sets of eigen weights in \mathcal{E}' and \mathcal{E}'' and get still a proof structure.

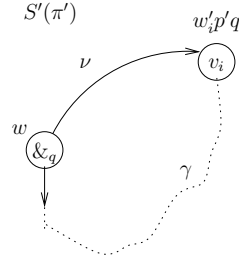
Moreover, by Lemma 4 it is enough to only verify that all the extensive switchings, w.r.t. q , for π' are ACC. In the rest of proof, whenever it is not explicitly declared, each switching for π' is meant to be extensive w.r.t. q .

Connectedness - Assume by absurdum there exists a switching $S'(\pi')$ that is disconnected. Then there exists at least a node v_j occurring in a connected component γ_1 that is separated from the component γ_2 that contains the $\&\epsilon_q$ -node together with the jump ν directed to a node v_i , like in the next left hand side picture where γ_1 and γ_2 are separated by dotted lines (in $S'(\pi')$ there is no path from v_j to v_i).



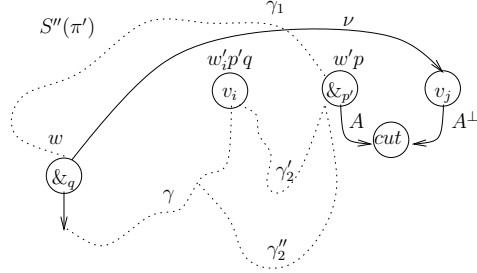
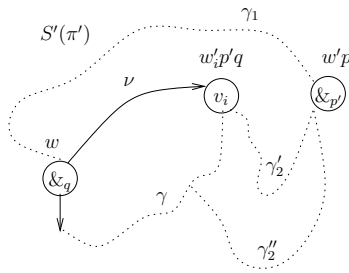
Now let $S''(\pi')$ be an other switching that is a copy of $S'(\pi')$ except for the jump ν from the $\&\epsilon_q$ -node to a v_h node that was already depending on q in π (in other words, S'' is a conservative switching, w.r.t. q , which differs from S' only for the jump ν , like in the previous right hand picture). Clearly in $S''(\pi')$ there is no connection path between v_i and v_j , otherwise this path would go through the ν -jump (and so through the $\&\epsilon_q$ -node) contradicting the assumption that the $\&\epsilon_q$ -node and v_j were disconnected in $S'(\pi')$. But this contradicts, by Lemma 4, the hypothesis that π was correct.

Acyclicity - Assume there exists a cycle in $S'(\pi')$ going through a jump ν from the $\&\epsilon_q$ -node to a node v_i whose weight w_i depends on q as follows



where γ is the path in $S'(\pi')$ from the conclusion of the $\&_q$ -node to v_i . Now, the new variable p' or its negation (resp., p'' or its negation) occurs by definition of $C_q/\&_p$ -cut reduction step in the weight w_i : let us say $w_i = w'_i p' q$. So, by condition 4 of Definition 7 and by connectedness of $S'(\pi')$ showed before, the $\&_{p'}$ -node (resp., the $\&_{p''}$ -node) and the node v_i must be connected in $S'(\pi')$ in two possible ways:

1. either by a path going through the $\&_q$ -node without crossing γ , like path γ_1 of the next left hand side picture;
2. or by a path γ'_2 not going through the $\&_q$ -node (and possibly crossing γ as γ''_2) like in the left hand side picture below:



In both cases we can set a conservative switching $S''(\pi')$ that is identical to $S'(\pi')$ except for the jump ν going from the $\&_q$ -node to the node v_j whose conclusion A^\perp is the premise of the cut link of π' obtained after reducing the cut $C_q/\&_p$ in π (see the right hand side picture above). Then we get a cycle, contradicting, by Lemma 4, the hypothesis that π is correct. \square

3.2.4 Strong cut-elimination

Technical details of proofs of next Theorems 7 and 8 can be found in [Mai2008].

Theorem 7 (termination) *We can always reduce a proof net π in to a proof net π' that is cut-free, by iterating the reduction steps of Definition 5 and 12.*

Theorem 8 (confluence) *Assume π is proof net s.t. it reduces in one step α to π' ($\pi \rightsquigarrow_\alpha \pi'$) and it reduces in another step β to π'' ($\pi \rightsquigarrow_\beta \pi''$); then, there exists a proof net σ such that π' reduces, in a certain number of steps, to σ ($\pi' \rightsquigarrow^* \sigma$) and π'' reduces, in a certain number of steps, to σ ($\pi'' \rightsquigarrow^* \sigma$).*

3.3 Sequentialization

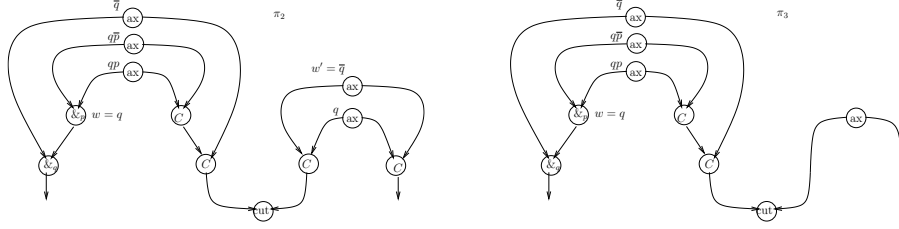
Theorem 9 (sequentialization) *A CyMALL GPN of Γ can be sequentialized into a CyMALL sequent proof with same cyclic conclusions $\sigma(\Gamma)$ and vice-versa (de-sequentialization).*

PROOF — *Sequentialization-part:* any CyMALL proof net π can be sequentialized into a proof Π , by induction on the number of slices (or, equivalently, by the number of $\&$ links). The base of induction corresponds to the sequentialization of the CyMALL proof nets (Theorem 3). The

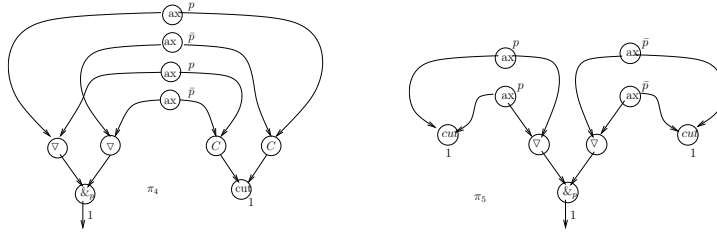
induction step follows from the sequentialization of standard MALL PN (see [Gir1996]). The only novelty is to show that whenever a PN π contains a terminal $\&$ -link L , then π can be *toggled*³ at L in two sub-proof nets preserving conditions 1 and 2 of Definition 9.

De-sequentialization-part: any CyMALL proof Π of $\sigma(\Gamma)$ can be de-sequentialized into a PN π of $\sigma(\Gamma)$, by induction of the height of Π derivation. \square

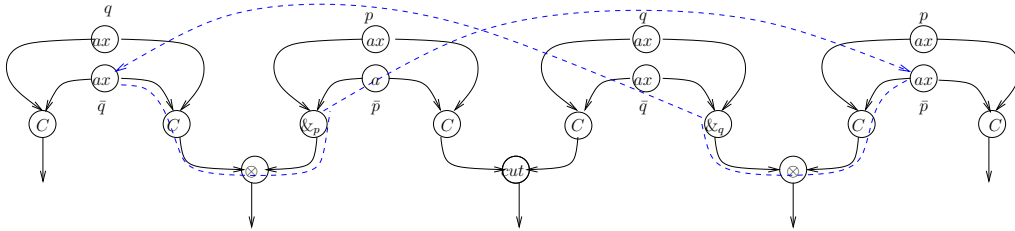
Example 5 In general, there may exist different GPNs to denote the same sequential proof. Although π_2 and π_3 below are both correct proof structures with same conclusions, π_3 quotients “more” sequential proofs than π_2 .



Vice-versa, π_4 and π_5 are both correct proof structures (with same conclusions) that sequentialize into the same proof, although π_4 is not the de-sequentialization of any proof.



Finally, π_6 is not a correct proof structure: you can easily get an additive switching with a cycle like that one in dashed line.



4 Embedding Lambek Calculus into CyMALL PN

Definition 13 (Lambek formulas and sequents of CyMALL) Assume A and S are, respectively, a formula and a sequent of CyMALL.

1. A is a pure Lambek formula (PLF) if it is a CyMALL formula recursively built according to the following grammar

$$A ::= + \text{ atoms} \mid A \otimes A \mid A^\perp \nabla A \mid A \nabla A^\perp.$$

³We say that a conclusion link $\&_p$ of a GPN π is *toggling* whenever the restriction of π w.r.t. p and the restriction of π w.r.t. \bar{p} are both correct GPSs. We call restriction of π w.r.t. p (resp., \bar{p}) denoted $\pi \upharpoonright^p$ (resp., $\pi \upharpoonright^{\bar{p}}$), what remains of π when we replace p with 1 (resp., \bar{p} with 1) and keep only those vertexes and edges whose weights are still non zero (see details in [Gir1996]).

2. A is an additive Lambek formula (ALF or simply LF) if it is a CyMALL formula recursively built according to the following grammar

$$A := a \text{ PLF} \mid A \& A \mid A \oplus A.$$

3. S is a Lambek sequent of CyMALL iff

$$S = (\Gamma)^\perp, A$$

where A is a non void LF and $(\Gamma)^\perp$ is a possibly empty finite sequence of negations of LFs (i.e., Γ is a possibly empty sequence of LFs and $(\Gamma)^\perp$ is obtained by taking the negation of each formula in Γ).

4. A Lambek proof is any derivation built by means of the CyMALL inference rules whose premise(s) and conclusions are CyMALL Lambek sequents.

Definition 14 (Lambek CyMALL proof net) We call Lambek CyMALL proof net any CyMALL PN whose edges are labeled by LFs or negation of LFs and whose conclusions are a Lambek sequent.

Corollary 2 Any Lambek CyMALL proof net π is stable under cut reduction, i.e., if π reduces in one step to π' , then π' is a Lambek CyMALL proof net too.

PROOF — Consequence of Theorems 4, 5 and 6. Trivially, each reduction step preserves the property that each edge of the reductum is labeled by a Lambek formula or the negation of a Lambek formula. \square

Theorem 10 (de-sequentialization of Lambek CyMALL proofs) Any proof of a CyMALL Lambek sequent $\vdash \sigma(\Gamma^\perp, A)$ can be de-sequentialized in to a Lambek CyMALL PN with conclusions $\sigma(\Gamma^\perp, A)$.

PROOF — By induction on the height of the given sequent proof (similarly to the de-sequentialization part of Theorem 9). \square

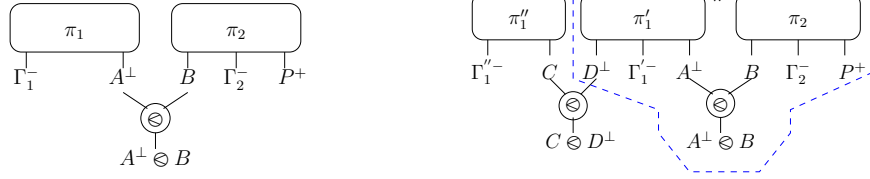
Theorem 11 (sequentialization of pure Lambek CyMALL PNs) Any Lambek CyMALL PN of $\sigma(\Gamma^\perp, A)$ sequentializes into a Lambek CyMALL proof of the sequent $\vdash \sigma(\Gamma^\perp, A)$.

PROOF — Sequentialization follows by induction on the number of $\&$ -links of the given PN. The base of induction follows from next Theorem 12. \square

Theorem 12 (sequentialization of Lambek CyMALL PNs) Any Lambek CyMALL PN of $\sigma(\Gamma^\perp, A)$ sequentializes into a Lambek CyMALL proof of $\vdash \sigma(\Gamma^\perp, A)$.

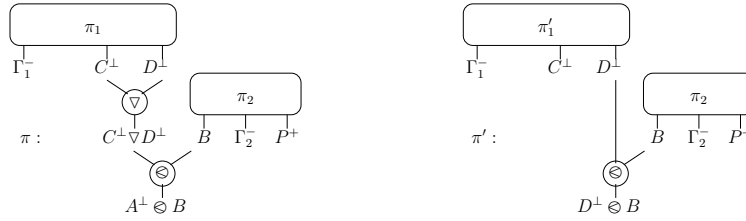
PROOF — Assume by absurdum there exists a pure Lambek CyMALL proof net π that does not sequentialize into a Lambek CyMALL proof. We can chose π minimal w.r.t. the size. Clearly, π cannot be reduced to an axiom link; moreover π contains neither a negative conclusion of type $A^\perp \nabla B^\perp$ nor a positive conclusion of type $A^\perp \nabla B$ (resp., $A \nabla B^\perp$), otherwise, we could remove this terminal ∇ -link and get a strictly smaller (than π) proof net π' that is sequentializable, by minimality of π ; this implies that also π is sequentializable (last inference rule of the sequent proof will be an instance of ∇ -rule) contradicting the assumption. For same reasons (minimality), the unique positive conclusion (e.g. $A \otimes B$) of π cannot be splitting. Therefore, since π is not an axiom link $\frac{A^\perp}{A^\perp} \frac{A}{A}$, by Lemmas 1 and 2, there must exist either a (negative) splitting \otimes -link (Case 1) or a splitting cut-link (Case 2).

Case 1. Assume a negative splitting conclusion $A^\perp \otimes B$ (resp., $A \otimes B^\perp$). By minimality, π must split like in the next left hand side picture (we use A^+ , resp. A^- , to denote positive, resp., negative, LF and Γ^- for sequence of negative LFs):

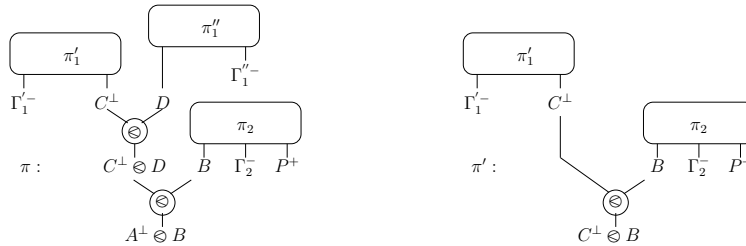


Now, let us reason on π_1 (reasoning on π_2 is symmetric): by minimality of π , π_1 cannot be reduced to an axiom link (otherwise Γ_1^- would not be negative); moreover, none of Γ_1^- is a (negative) splitting link, like e.g., $C \otimes D^\perp$, otherwise we could easily restrict to consider the sub-proof-net π' , obtained by erasing from π the sub-proof-net π''_1 (with conclusions $\Gamma_1^{\prime\prime-}, C$) together with the $C^\perp \otimes D$ -link, like the graph enclosed in the dashed line above. Clearly, π' would be a non sequentializable Lambek proof net strictly smaller than π . In addition, π_1 must be cut-free, otherwise by minimality, after a cut-step reduction we could easily build a non sequentializable reductum PN π' , strictly smaller than π , (π' will have same conclusions of π). Therefore, there are only two sub-cases:

1. either $A^\perp = C^\perp \nabla D^\perp$, then we can easily get, from the PN π on the l.h.s. below, the non sequentializable PN π' , on the r.h.s.; π' is strictly smaller than π , contradicting the minimality assumption:

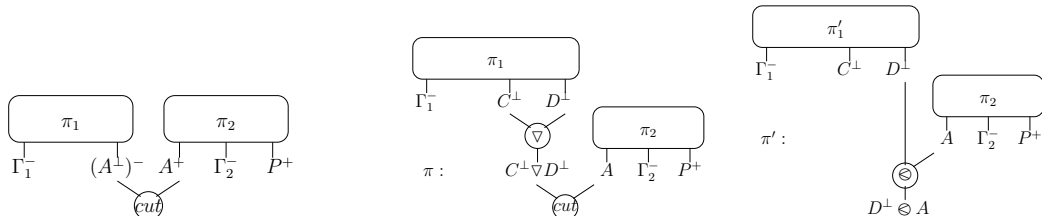


2. or $A^\perp = C^\perp \otimes D$, then this $C^\perp \otimes D$ -link must split by Lemma 1, since π_1 is a cut-free PN in splitting condition without other \otimes -splitting conclusion in Γ_1^- ; so from π on the l.h.s., we can easily get the non sequentializable PN π' on r.h.s.; π' is strictly smaller of π , contradicting the minimality assumption:

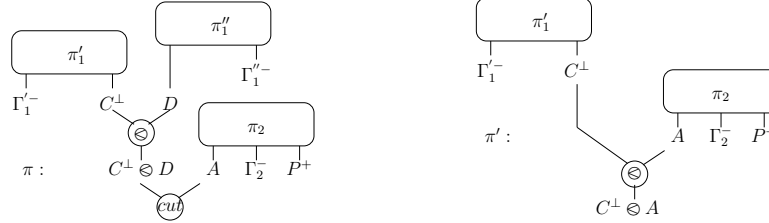


Case 2. Assume π contains a splitting cut link, like the leftmost hand side picture below, then we proceed like in Case 1. We reason on π_1 with two sub-cases:

1. either $A^\perp = C^\perp \nabla D^\perp$, then we can easily get, starting from the PN π on the middle side below, a non sequentializable PN π' , like the rightmost hand side picture; π' is strictly smaller than π , contradicting the minimality assumption:



2. or $A^\perp = C^\perp \otimes D$, then this A^\perp -link must be splitting by Lemma 1, since π_1 is a cut-free PN in splitting condition without any other \otimes -splitting conclusion in Γ_1^- ; so, we can easily get, starting from the PN π on the l.h.s., a non sequentializable PN π' that is strictly smaller than π (on the r.h.s.), contradicting the minimality assumption.



□

5 Language parsing with Lambek CyMALL PNs

In this section we reformulate, in our syntax, some examples of linguistic parsing suggested by Richard Moot in his PhD thesis [Moo2002]. We use s, np and n as the types expressing, respectively, a *sentence*, a *noun phrase* and a *common noun*. According to the “parsing as deduction style”, when a string $w_1 \dots w_n$ is tested for grammaticality, the types t_1, \dots, t_n associated with the words are retrieved from the lexicon and then parsing reduces to proving the derivability of a two-sided sequent of the form $t_1, \dots, t_n \vdash s$. Remind that proving a two sided Lambek derivation $t_1, \dots, t_n \vdash s$ is equivalent to prove the one-sided sequent $\vdash t_n^\perp, \dots, t_1^\perp, s$ where t_i^\perp is the dual (i.e., linear negation) of each type t_i . Any phrase or sentence should be read like in a mirror (with opposite direction).

Assume the next lexicon, where *linear implication* \multimap (resp., \multimap) is traditionally used for expressing types in two-sided sequent parsing:

1. $\text{Vito} = np; \text{Sollozzo} = np; \text{him} = np;$
2. $\text{trusts} = (np \multimap s) \multimap np = (np^\perp \nabla s) \nabla np^\perp.$

Cases of lexical ambiguity follow to words with several possible formulas A and B assigned it. For example, a verb like “to believe” can express a relation between two persons, np 's in our interpretation, or between a person and a statement, interpreted as s , as in the following examples:

$$\text{Sollozzo believes Vito.} \tag{1}$$

$$\text{Sollozzo believes Vito trusts him.} \tag{2}$$

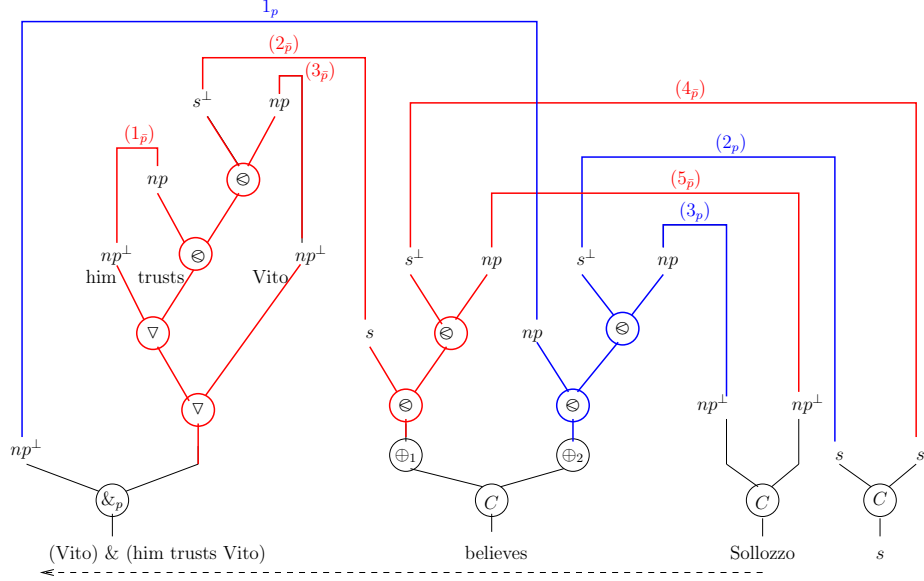
We can express this polymorphism by two lexical assignments as follows:

3. $\text{believes} = (np \multimap s) \multimap np = (np^\perp \nabla s) \nabla np^\perp;$
4. $\text{believes} = (np \multimap s) \multimap s = (np^\perp \nabla s) \nabla s^\perp.$

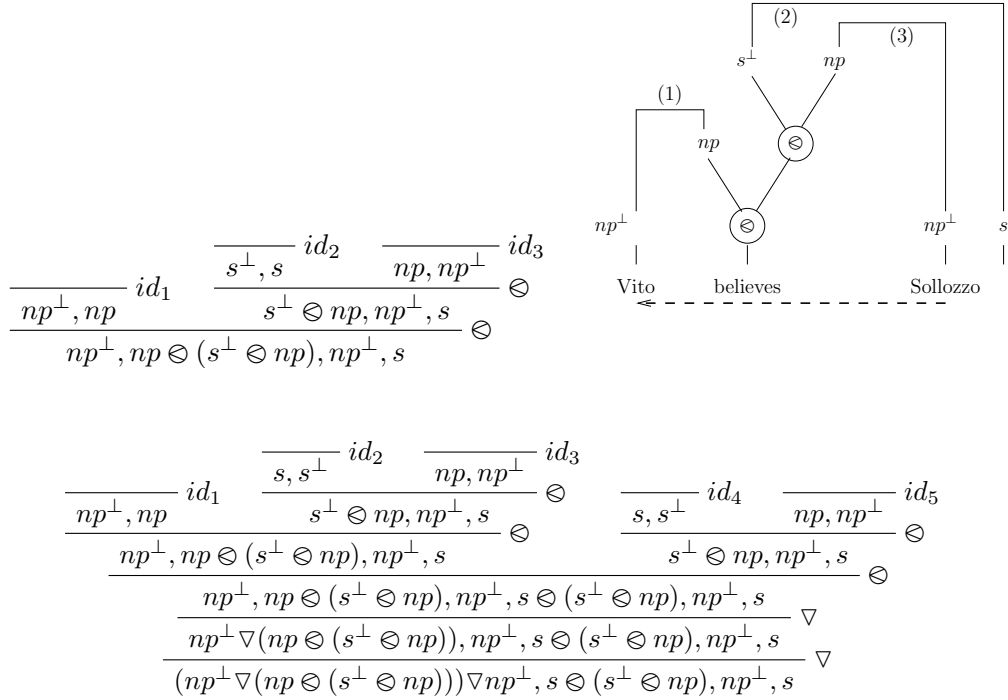
Typically, additives are used for capturing cases of lexical ambiguity. When a word has two possible formulas A and B assigned it, we can combine these into a single additive formula $A \& B$ (resp., $A \oplus B$). Thus, we can collapse assignments 3 and 4 into a single additive formula assignment as follow:

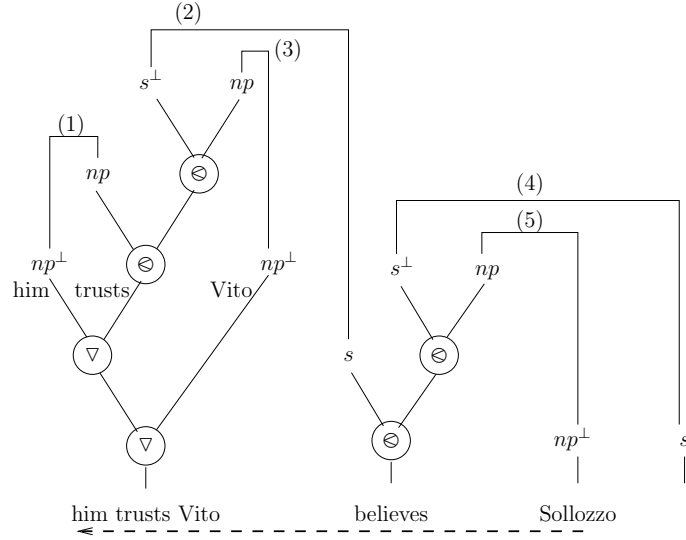
$$\text{believes} = ((np \multimap s) \multimap np) \& ((np \multimap s) \multimap s) = ((np^\perp \nabla s) \nabla np^\perp) \& ((np^\perp \nabla s) \nabla s^\perp).$$

Using that, we can move lexical ambiguity into proof net as follows: we give the Lambek CyMALL PN corresponding to the parsing of the additive superposition of sentences (1) and (2).



Observe that for slice $\varphi_p(\pi)$ (resp., $\varphi_{\bar{p}}(\pi)$) there exists an additive resolution that is a CyMLL PN corresponding to parsing of sentence (1), (resp., sentence (2); see the red subgraph); these PNs sequentialize as follows:





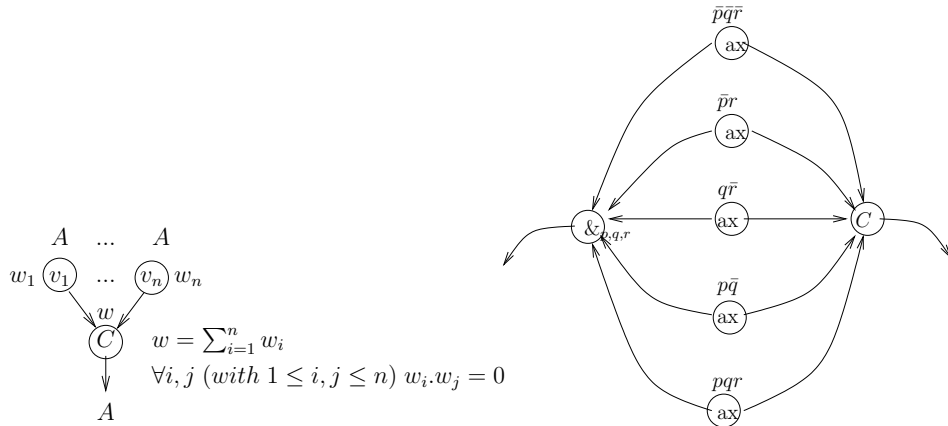
6 Conclusions and further works

6.1 Stability of sequentialization under cut reduction

Problem: try to give a direct sequentialization avoiding Girard's one.

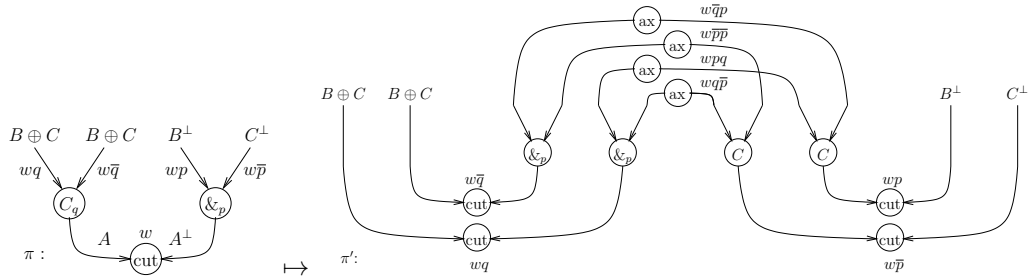
6.2 Proof-structures with explicit n -ary links

Proof-structures with explicit n -ary (& or Contraction) links differ from those one defined in Definition 7 only for the use of n -ary contractions instead of the binary ones, like the next left hand side picture. Then, the notion of slices and switchings (Definitions 8) and proof-nets (Definitions 9) remain unchanged as well the (De-)Sequentialization Theorem 9. Observe that in general the syntax with n -ary contraction or & links is not equivalent to the former one with binary ones. Actually, the picture on the right hand side below shows an example of proof-net with an n -ary contraction (resp., &) link whose monomial sum, $\overline{pqr} + \overline{pr} + q\overline{r} + p\overline{q} + pqr$, cannot be factorized in such a way to reduce this n -ary contraction into a sum of only monomial binary sums (contractions).



6.3 Fully local cut reduction and parsing contraction

Problem: try to define a retraction system stable under a strongly normalizing cut reduction that implements a commutative cut elimination step $\&_p/C$ that does not duplicate the empire \mathcal{E}_p as follows



Observe that, dislike the $C/\&$ cut reduction step given before, this new $C/\&$ cut reduction step breaks the monomial condition GPS. This problem has been formulated and solved in [LM2008]. It is not clear whether this new syntax has effects on Lambek proof nets.

7 Complexity

We aim at studying the class complexity of the CyMALL correctness criterion. What complicate matters from a computational point of view is the fact that, dislike the multiplicative case, we could have chosen different additive proof nets for capturing the polymorphic lexical behavior described by the proof net above.

References

- [Abr2002] Abrusci, V. M.. Classical conservative extensions of Lambek calculus *Studia Logica* 71 (3):277 - 314 (2002).
- [AR2000] Abrusci, V. M. and Ruet, P.. Non-commutative logic I: the multiplicative fragment. *Annals of Pure and Applied Logic* 101(1): 29-64, 2000.
- [AP1991] Andreoli J.-M. and Pareschi, R.: From Lambek Calculus to word-based parsing. In *Proc. of Workshop on Substructural Logic and Categorical Grammar*, CIS Munchen, Germany, 1991.
- [DR1989] Danos, V. and Regnier, L. The Structure of Multiplicatives. *Archive for Mathematical Logic*, 28:181-203, 1989,
- [Dan1990] Danos, V. La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ -calcul). *PhD Thesis*, Univ. Paris VII, Juin 1990.
- [Gir1987] Girard, J-Y. Linear Logic. *Theoretical Computer Science*, 50:1-102, 1987
- [Gir1996] Girard, J-Y.. *Proof-nets: the parallel syntax for proof theory*. Logic and Algebra. Marcel Dekker, 1996.
- [Gir2006] Girard, J-Y. Le point aveugle. *Cours de Logique. Volume I, Vers la Perfection*. Ed. Hermann, 2006, Paris.
- [GM2001] Guerrini, S. and Masini, A. *Parsing MELL proof nets*. Theoretical Computer Science, 254:317-335, 2001
- [HVG2003] Hughes, D. and van Glabbeek, R. Proof Nets for Unit-free Multiplicative-Additive Linear Logic. In *Proc. of IEEE Logic in Computer Science*, 2003.
- [Laf1995] Lafont, Y. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont and L. Regnier eds. *Advanced in Linear Logic*, pp. 225-247. Cambridge Press, 1995.

- [Lam1958] Lambek, J.. The mathematics of sentence structure. *American Mathematical Monthly*, 65, 1958.
- [Lau1999] Laurent, O.. Polarized Proof Nets: Proof Nets for LC (Extended Abstract). In J.-Y. Girard, editor, *Typed Lambda Calculi and Applications 1999*,
- [LM2008] Laurent, O. and Maieli, R.. Cut Elimination for Monomial MALL Proof Nets. In *Proc. of IEEE Logic in Computer Science*, pp 486-497, 2008. Pittsburgh, USA.
- [Mai2003] Maieli, R.. A new correctness criterion for multiplicative non commutative proof-nets.. *Archive for Mathematical Logic*, vol.42, 205-220, Springer-Verlag, 2003.
- [Mai2007] Maieli, R.. Retractable Proof Nets of the Purely Multiplicative and Additive Fragment of Linear Logic. *14th LPAR 2007*. Springer, LNAI 4790, pp. 363-377, 2007.
- [Mai2008] Maieli, R.. Cut Elimination for Monomial Proof Nets of the Purely Multiplicative and Additive Fragment of Linear Logic. IAC-CNR Report n.140 (2/2008); available at <http://logica.uniroma3.it/~maieli/iac140.pdf>.
- [MR2012] Moot, R. and Retoré Ch.. The logic of categorial grammars: a deductive account of natural language syntax and semantics. Springer LNCS 6850, 2012.
- [Moo2002] Moot, R.. *Proof nets for linguistic analysis. PhD thesis*. Utrecht University, 2002.