Cyclic Multiplicative 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

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 simple and intuitive syntax for PNs of the cyclic multiplicative fragment of linear logic (CyMLL). The proposed correctness criterion for CyMLL PNs can be considered as the non-commutative counterpart of the famous Danos-Regnier (DR) criterion for PNs of the pure multiplicative fragment (MLL) 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. Dislike the most part of current syntaxes for non-commutative PNs, our syntax allows a sequentialization for the full class of CyMLL PNs, without requiring these latter must be cut-free. Moreover, we give a simple characterization of CyMLL PNs for Lambek Calculus and thus a geometrical (non inductive) way to parse phrases or sentences by means of Lambek PNs.

Keywords: Categorial grammars · Cyclic orders · Lambek calculus · Language parsing · Linear logic · Non-commutative logic · Proof nets · Sequent calculus

1 Introduction

Proof nets are one of the most innovative inventions of linear logic (LL, [5]): 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 first present a simple syntax for proof nets of the Cyclic Multiplicative fragment of LL (CyMLL PNs, Sect. 2). In particular, we introduce a new correctness criterion for CyMLL PNs which can be considered as the non-commutative counterpart of the famous Danos-Regnier (DR) criterion for proof nets of linear logic (see [4]). The main intuition relies on the fact that any DR-switching for a proof structure π (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 π (Sect. 2.1). Moreover, the proposed correctness criterion:

V. de Paiva et al. (Eds.): WoLLIC 2015, LNCS 9160, pp. 53–68, 2015.
DOI: 10.1007/978-3-662-47709-0_5

1. is shown to be stable under (or preserved by) cut elimination (Sect. 2.2);
2. dislike some previous syntaxes (e.g., [15], [2] or [11]) it admits a sequentialization (that is, a way to associate a (unique) sequent proof to each proof net) for the full class of CyMLL PNs including those ones with cuts (Sect. 2.3).

CyMLL can be considered as a classical extension of Lambek Calculus (LC, see [1,9,13]) 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 [3], 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 $w_1 ... w_n$ is tested for grammaticality, the types $t_1, ..., t_n$ 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 $\vdash t_n^{\perp}, ..., t_1^{\perp}, s$, where s is the type associated with sentences. Moreover, forcing constraints on the Exchange rule by allowing only *cyclic permutations* over sequents of formulas, gives the required computational control needed to view theorem proving as parsing in Lambek Categorial 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.

Now, proof nets are commonly considered an elegant solution to the first problem of representing canonical proofs; in this sense, in Sect. 3, we give an embedding of pure Lambek Calculus into Cyclic MLL proof nets; then, in Sect. 4, we show how to parse some linguistic examples that can be found in [14].

Unfortunately, there is not an equally brilliant solution to the second problem listed above. However, we retain that, as further work, extending parsing by means of additive proof nets (MALL) could be a step towards a proof-theoretical solution to the problem of lexical polymorphism; technically speaking, Cyclic MALL proof nets allow to manage formulas (types) superposition (polymorphism) by means of the additive connectives &and $\oplus$ (see Sect. 5, also [6,8,12]).

1.1 Cyclic MLL

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

$$\frac{}{\vdash A, A^{\perp}}\,id \qquad \frac{\vdash \Gamma, A \qquad A^{\perp}\Delta}{\vdash \Gamma, \Delta}\,cut \qquad \frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\,\otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \triangledown B}\,\triangledown$$

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 $c < a < b$.

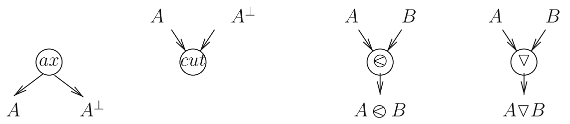
Definition 1 (Total Cyclic Order). *A* total cyclic order *is a pair* (X, σ) *where* X *is a set and* σ *is a ternary relation over* X *satisfying the following properties:*

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

Negative (or *asynchronous*) connectives correspond to a kind of *true determinism* in the way we apply bottom-up their corresponding inference rules (the application of $\triangledown$ rule is completely deterministic). Vice-versa, positive (or *synchronous*) connectives correspond to a kind of *true non-determinism* in the way we apply bottom-up their corresponding rules (there is no deterministic way to split the context Γ, Δ in the $\otimes$ rule).

2 Proof Structures

Definition 2 (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*):*



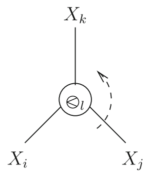
In a PS π *each premise (resp., conclusion) of a link must be conclusion (resp., premise) of exactly (resp., at most) one link of* π*. We call* conclusion of π *any emergent edge that is not premises of any link.*

2.1 Correctness of Proof Structures

We characterize those CyMLL PSs that are images of CyMLL proofs. Actually, there exist several syntaxes for CyMLL proof nets, like those ones of [2,11]; 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 and connected switching for π; $S(\pi)$ is the rootless planar tree*[1] *whose nodes are labeled by $\otimes$-nodes, and whose leaves $X_1, ..., X_n$ (with $\Gamma \subseteq X_1, ..., X_n$) are the terminal, i.e., pending, edges of $S(\pi)$; $S(\pi)$ is a ternary relation, called* seaweed, *with* support *$X_1, ..., X_n$; 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 a node $\otimes_l$;*
  - *the three paths $X_i\otimes_l, X_j\otimes_l$ and $X_k\otimes_l$ are in this cyclic order while moving anti-clockwise around the $\otimes_l$-node like below*

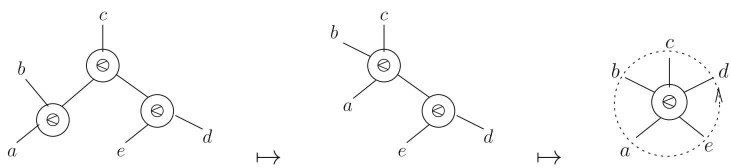


If A is an edge of the seaweed $S(\pi)$, then $S_i(\pi)\downarrow^A$ is the restriction of the seaweed *$S(\pi)$, that is, the sub-graph of $S(\pi)$ obtained as follows:*

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 $X_1, ..., X_n$; 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.*

Naively, we may *contract* a seaweed (by associating the $\otimes$-nodes) until we get a collapsed single n-ary $\otimes$-node with n pending edges (its support), like in the example below:



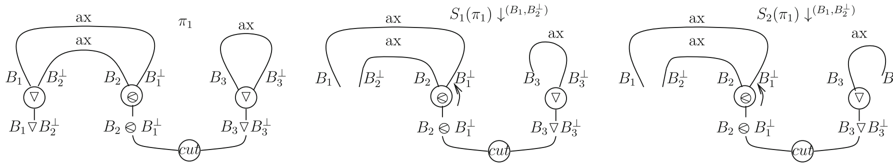
Definition 4 (CyMLL Proof Net). *A PS π is* correct, *i.e. it is a CyMLL proof net (PN), iff:*

---
[1] In any switching we can consider as a single edge any axiom, cut or ∇-link obtained after the mutilation of one of the two premises.

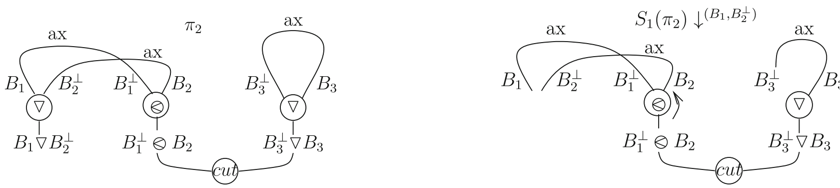
1. π is a standard MLL PN, that is, any switching $S(\pi)$ is a connected and acyclic graph (therefore, $S(\pi)$ is a seaweed);

2. for any $\triangledown$-link $\frac{A \quad B}{A \triangledown B}$ the triple (A, B, C) must occur in this cyclic order in any seaweed $S(\pi)$ restricted to A, B, i.e., $(A, B, C) \in S(\pi) \downarrow^{(A,B)}$, for all pending leaves C (if any) in the support of the restricted seaweed.

Example 1. We give an instance of CyMLL proof net π_1 with its 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.



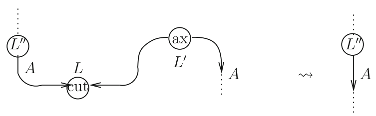
On the opposite, the following instance of proof structures π_2 is not correct (it is not a proof net) since condition 2 of Definition 4 is violated: there exists a $\triangledown$-link $\frac{B_1 \quad B_2^\perp}{B_1 \triangledown B_2^\perp}$ and a seaweed $S_1(\pi_2)$ s.t. $\neg\forall C$ pending, $(B_1, B_2^\perp, C) \in S_1(\pi_2) \downarrow^{(B_1, B_2^\perp)}$; 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

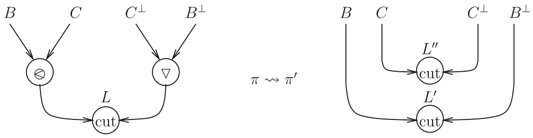
Definition 5 (Cut Reduction). *Let L be a cut link in a proof net π whose premises A and $A^\perp$ are, resp., conclusions of links L', L''. Then we define the result π' (called* reductum*) of reducing this cut in π (called* redex*), as follows:*

***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 to L'' (resp., to L') the other conclusion of L' (resp., L'') as new conclusion.*



($\otimes/\triangledown$)-cut: *if L' is a $\otimes$-link with premises B and C and L'' is a $\triangledown$-link with premises $C^\perp$ and $B^\perp$, then π' is obtained by removing in π the formulas A*

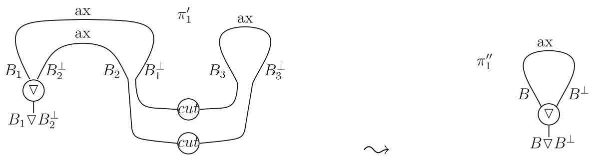
and $A^{\perp}$ as well as the cut link L with L' and L'' and by adding two new cut links with, resp., premises B, $B^{\perp}$ and $C, C^{\perp}$, as follows:



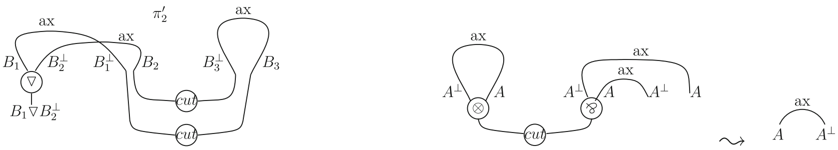
Theorem 1 (Stability of PNs Under Cut Reduction). *If π is a CyMLL PN that reduces to π' in one step of cut reduction, $\pi \rightsquigarrow \pi'$, then π' is a CyMLL PN.*

See proof in Appendix A.1.

Example 2. W.r.t. Example 1, π_1 reduces to π_1' and also π_1' to π_1'' as below; both π_1' and π_1'' are correct since condition 2 of Definition 4 is void for both of them:



Moreover, w.r.t. Example 1, π_2 is a non correct PS that reduces to the 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 CyMLL PNs and sequential proofs. A first sequentialisation result for non commutative (CyMLL) cut-free proof nets can be found in [17].

Lemma 1 (Splitting). *Let π be a CyMLL PN with at least a $\otimes$-link (resp., a 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 $\frac{A \quad B}{A \otimes B}$ (resp., a cut-link $\frac{A \quad A^{\perp}}{}$) that splits π in two CyMLL PNs, π_A and π_B (resp., π_A and $\pi_{A^{\perp}}$).*

See proof in Appendix A.2.

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.

See proof in Appendix A.3

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 $\sigma(\Gamma)$.*

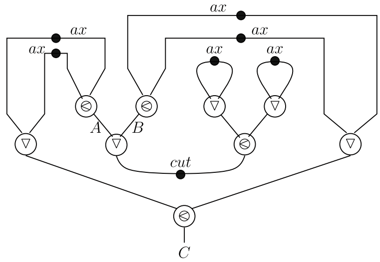
Theorem 2 (Adequacy of CyMLL PNs). *Any CyMLL cut-free proof of a sequent $\sigma(\Gamma)$* de-sequentializes *into a CyMLL PN with same conclusions $\sigma(\Gamma)$.*

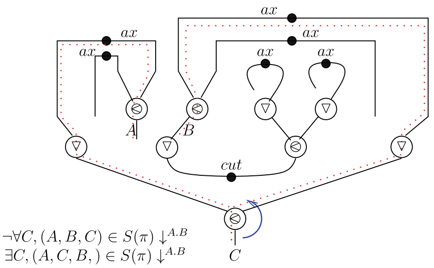
Proof. By induction on the height of the given sequential proof of $\sigma(\Gamma)$.

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

Proof. By induction on the size $\langle \sharp vertexes, \sharp edges \rangle$ of π via Lemmas 1 and 2.

Example 3 (Melliès 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 PNs without assuming these must be cut-free. It is enough to require the cut-free condition only in the adequacy part (Theorem 2). In particular, observe that *Melliès proof structure* below is not a correct proof net according to our correctness criterion (thus, it is not sequentializable) since there exists a $\frac{A \quad B}{A \nabla B}$ 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: actually, following the crossing red dotted lines in right hand side seaweed, you can easily verify there exists a pending C (a conclusion, indeed) s.t.$(A, C, B) \in S(\pi) \downarrow^{(A,B)}$.





Observe that Melliès's proof structure becomes correct (therefore sequentializable) after cut reduction. Reader may refer to [10,13] (pp. 223-224) for a discussion of this example and to [16] for a discussion about incorrect proof nets that reduce to correct proof nets from a denotational semantics viewpoint.

3 Embedding Lambek Calculus into CyMLL PNs

In this section we characterize those CyMLL PNs that correspond to Lambek proofs. The first (sound) notion of Lambek cut-free proof net, without sequentialization, was given in [18]; see also [13,17] for an original discussion on the embedding of Lambek Calculus into PNs.

Definition 6 ((pure-)Lambek Formulas and Sequents of CyMLL). *Assume A and S are, respectively, a formula and a sequent of CyMLL.*

1. *A is a* (pure) Lambek formula (LF) *if it is a CyMLL formula recursively built according to the following grammar*

$$A \ := \ \textit{positive atoms} \mid A \otimes A \mid A^{\perp} \triangledown A \mid A \triangledown A^{\perp}.$$

2. *S is a* Lambek sequent of CyMLL *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).

3. *A* (pure) Lambek proof *is any derivation built by means of the CyMLL inference rules in which premise(s) and the conclusions are Lambek sequents.*

Definition 7 (Lambek CyMLL Proof Net). *We call* Lambek CyMLL proof net *any CyMLL PN whose edges are labeled by pure LFs or negation of pure LFs and whose conclusions form a Lambek sequent.*

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

Proof. Consequence of Theorem 1. Any reduction step preserves the property that each edge (resp., the conclusion) of the reductum is labeled by a Lambek formula or by a negation of a Lambek formula (resp., by a Lambek sequent).

Theorem 4 (Adequacy of Lambek CyMLL PNs). *Any cut-free proof of a Lambek sequent* $\vdash \sigma(\Gamma^{\perp}, A)$ *can be de-sequentialized in to a Lambek CyMLL PN with same conclusions* $\sigma(\Gamma^{\perp}, A)$*.*

Proof. by induction on the height of the given sequent proof.

Theorem 5 (Sequentialization of Lambek CyMLL PNs). *Any Lambek CyMLL proof net of* $\sigma(\Gamma^{\perp}, A)$ *sequentializes into a Lambek CyMLL proof of the sequent* $\vdash \sigma(\Gamma^{\perp}, A)$*.*

See proof in Appendix A.4.

4 Parsing via Lambek CyMLL PNs

In this section we reformulate, in our syntax, some examples of linguistic parsing suggested by Richard Moot in his PhD thesis [14]. 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 $w_1...w_n$ is tested for grammaticality, the types $t_1, ..., t_n$ 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 $t_1, ..., t_n \vdash s$. Remind that proving a two sided Lambek derivation $t_1, ..., t_n \vdash s$ is equivalent to prove the one-sided sequent $\vdash t_n^{\perp}, ...t_1^{\perp}, s$ where $t_i^{\perp}$ is the dual (i.e., linear negation) of type t_i. Any phrase or sentence should be read like in a mirror (with opposite direction).

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

1. Vito $= np$; Sollozzo $= np$; him $= np$;
2. trusts $= (np \multimap s) \circ\!- np \ = \ (np^{\perp} \triangledown s) \triangledown np^{\perp}$.

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:

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

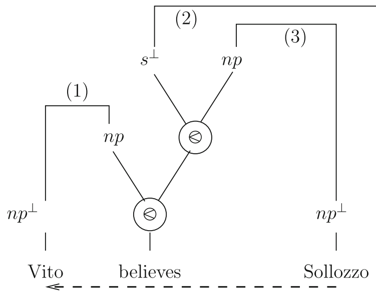
$$Sollozzo\ believes\ Vito\ trusts\ him. \tag{2}$$

We can express this verb ambiguity by two lexical assignments as follows:

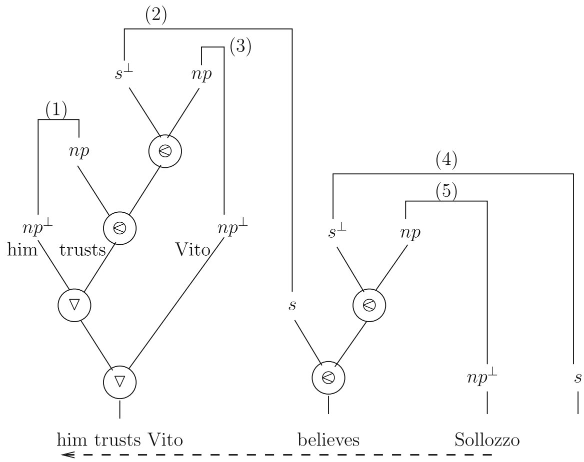
3. believes $= (np \multimap s) \circ\!- np \ = \ (np^{\perp} \triangledown s) \triangledown np^{\perp}$;
4. believes $= (np \multimap s) \circ\!- s \ = \ (np^{\perp} \triangledown s) \triangledown s^{\perp}$.

Finally, parsing of sentences (1) and (2) corresponds to the following Lambek CyMLL proofs with associated their corresponding proof nets:

$$\dfrac{\dfrac{}{np^{\perp}, np}\,id_1 \qquad \dfrac{\dfrac{}{s^{\perp}, s}\,id_2 \qquad \dfrac{}{np, np^{\perp}}\,id_3}{s^{\perp} \olessthan np, np^{\perp}, s}\,\olessthan}{np^{\perp}, np \olessthan (s^{\perp} \olessthan np), np^{\perp}, s}\,\olessthan$$



$$\dfrac{\dfrac{\dfrac{\dfrac{}{np^{\perp}, np}\,id_1 \qquad \dfrac{\dfrac{}{s, s^{\perp}}\,id_2 \qquad \dfrac{}{np, np^{\perp}}\,id_3}{s^{\perp} \olessthan np, np^{\perp}, s}\,\olessthan}{np^{\perp}, np \olessthan (s^{\perp} \olessthan np), np^{\perp}, s}\,\olessthan \qquad \dfrac{\dfrac{}{s, s^{\perp}}\,id_4 \qquad \dfrac{}{np, np^{\perp}}\,id_5}{s^{\perp} \olessthan np, np^{\perp}, s}\,\olessthan}{\dfrac{np^{\perp}, np \olessthan (s^{\perp} \olessthan np), np^{\perp}, s \olessthan (s^{\perp} \olessthan np), np^{\perp}, s}{np^{\perp} \triangledown (np \olessthan (s^{\perp} \olessthan np)), np^{\perp}, s \olessthan (s^{\perp} \olessthan np), np^{\perp}, s}\,\triangledown}\,\olessthan}{(np^{\perp} \triangledown (np \olessthan (s^{\perp} \olessthan np))) \triangledown np^{\perp}, s \olessthan (s^{\perp} \olessthan np), np^{\perp}, s}\,\triangledown$$



5 Conclusions and Further Works

In this paper we presented a correctness criterion for cyclic pure multiplicative (CyMLL) proof nets satisfying a sequentialization for the full class of proof nets, including those ones with cut links.

As future work, we aim at studying the complexity of both correctness verification and sequentialization. Moreover, in order to capture lexical ambiguity we aim at embedding the extended CyMALL Lambek calculus [1] into MALL proof nets (see e.g., [6,8,12]). Additive connectives, &and $\oplus$, allow superpositions of formulas (types); in particular, as suggested by [14], we could collapse the previous assignments 3 and 4 into the following single additive assignment:

5. believes = $((np \multimap s) \circ\!\!- np)\&((np \multimap s) \circ\!\!- s) = ((np^{\perp} \triangledown s) \triangledown np^{\perp})\&((np^{\perp} \triangledown s) \triangledown s^{\perp})$.

Acknowledgements. The authors thank the anonymous reviewers and Richard Moot for their useful comments and suggestions. This work was partially supported by the PRIN Project *Logical Methods of Information Management*.

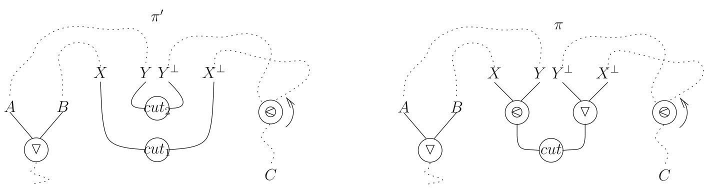
A Technical Appendices

A.1 Proof of Theorem 1: Stability of PNs under Cut Reduction

Proof. Observe that condition 1 of Definition 4 follows as an almost immediate consequences of the next graph theoretical property (see pages 250-251 of [7]):

Property 1 (Euler-Poicaré invariance). Given a graph $\mathcal{G}$, then $(\sharp CC - \sharp Cy) = (\sharp V - \sharp E)$, where $\sharp CC$, $\sharp Cy$, $\sharp V$ and $\sharp E$ denotes the number of, respectively, connected components, cycles, vertices 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 $(Y^{\perp} \triangledown X^{\perp})$ and assume, by absurdum, there exist a $\triangledown$-link labeled by a formula $A \triangledown B$ s.t. the triple (A, C, B) occurs in this wrong cyclic order in a seaweed $S(\pi')$ restricted to A, B, i.e., $S(\pi') \downarrow^{(A,B)}$, 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, $cut_1 \frac{X \quad X^{\perp}}{}$ and $cut_2 \frac{Y \quad Y^{\perp}}{}$, resulting from the cut reduction, otherwise π would already be violating condition 2 of Definition 4; assume path $B\otimes$ (resp., $A\otimes$) goes through cut_1 link (resp., cut_2 link) as follows



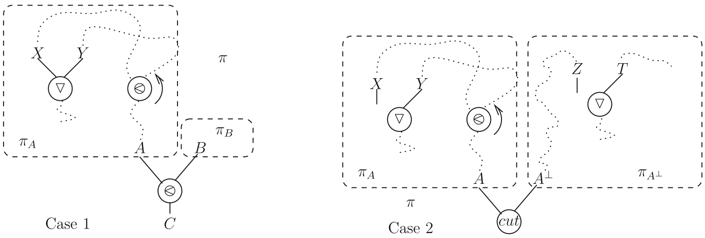
This means there exist a seaweed $S'(\pi)$, a link $Y^\perp \triangledown X^\perp$ and a triple $Y^\perp, C, X^\perp$ s.t. $(Y^\perp, C, X^\perp) \in S'(\pi) \downarrow^{(Y^\perp, X^\perp)}$, violating condition 2 and so contradicting correctness of π (see the right hand side picture above; since any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

A.2 Proof of Lemma 1: Splitting

Proof. Assume π is a CyMLL PN in splitting condition, then by the Splitting Lemma for standard commutative MLL PNs ([5]) π must split either at a $\otimes$-link or a *cut*-link. We reason according these two cases.

1. Assume π splits at $\frac{A \quad B}{A \otimes B}$ in two components π_A and π_B; we know that both components satisfy condition 1 (they eare 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 $\frac{X \quad Y}{X \triangledown Y}$ 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 Case 1 in picture below.



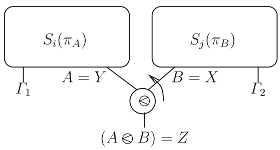
   This means there exists a restricted seaweed $S(\pi) \downarrow^{(X,Y)}$ containing X, Y and C (where $C = A \otimes B$) in the wrong cyclic order, i.e., $(X, C, Y) \in S(\pi) \downarrow^{(X,Y)}$, contradicting the correctness of π.
2. Assume π splits at the cut link $\frac{A \quad A^\perp}{}$ in two components π_A and $\pi_{A^\perp}$; 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 (w.r.t. the *size*, $\langle \sharp V, \sharp E \rangle$) PN in cut-splitting condition whose subproof π_A is not a CyMLL PN. This means, as before, there exists a $\frac{X \quad Y}{X \triangledown Y}$ 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 Case 2 of the previous picture. Then, by correctness π, $\pi_{A^\perp}$ must have $A^\perp$ as its unique conclusion, otherwise there exists a restricted seaweed for π, $S(\pi) \downarrow^{(X,Y)}$, containing a triple X, C, Y in the wrong order for a conclusion $C \neq A^\perp$. Moreover, $\pi_{A^\perp}$ cannot contain any cut, otherwise, by Theorem 1, we could replace in π the redex $\pi_{A^\perp}$ by its reductum $\pi'_{A^\perp}$, contradicting the minimality of π. Now, observe this equality

$\sharp\triangledown - \sharp\otimes = 1$, relating the number of $\otimes$-nodes with the number of $\triangledown$-nodes, holds for any cut free proof net with an unique conclusion. Therefore, $\pi_{A^\perp}$ must contain at least a $\triangledown$-link, let us say $\frac{Z \quad T}{Z\triangledown T}$. 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.

A.3 Proof of Lemma 2: Cyclic Order Conclusions of a PN

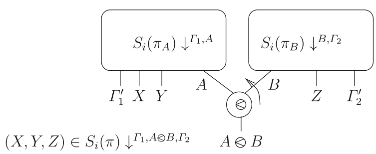
Proof. By induction on the size $\langle \sharp V, \sharp E\rangle$ of π.

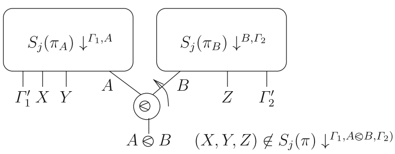
1. If π is reduced to an axiom link, then obvious.
2. If π contains at least a conclusion $A\triangledown B$, then $\Gamma = \Gamma', A\triangledown B$; by hypothesis of induction the sub-proof net π' with conclusion Γ', A, B has cyclic order $\sigma(\Gamma', A, B)$, and so, by condition 2 of Definition 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 $[A/A\triangledown B]$ (resp., $[B/A\triangledown B]$) in the restriction $S_i(\pi)\downarrow^{(\Gamma',A)}$ (resp., $S_i(\pi)\downarrow^{(\Gamma',B)}$), we get that each seaweed $S_i(\pi)\downarrow^{(\Gamma',A\triangledown B)}$ induces the same cyclic order $\sigma(\Gamma', A\triangledown B)$.
3. Otherwise π must contain a terminal splitting $\otimes$-link or *cut*-link. Assume π contains a splitting $\otimes$-link, $\frac{A \quad B}{A\otimes B}$, 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 follow 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 appear as follows:



$$S_i(\pi)\downarrow^{(\Gamma_1,A\otimes B,\Gamma_2)} = S_i(\pi_A)\downarrow^{(\Gamma_1,A)} \otimes S_i(\pi_B)\downarrow^{(B,\Gamma_2)}$$
$$S_j(\pi)\downarrow^{(\Gamma_1,A\otimes B,\Gamma_2)} = S_j(\pi_A)\downarrow^{(\Gamma_1,A)} \otimes S_j(\pi_B)\downarrow^{(B,\Gamma_2)}$$

   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) 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, $(X, Y, Z) \in S_i(\pi)\downarrow^{(\Gamma_1,A\otimes B,\Gamma_2)}$ and $(X, Y, Z) \notin S_j(\pi)\downarrow^{(\Gamma_1,A\otimes B,\Gamma_2)}$; by Splitting Lemma 1, each seaweeds for π, $S_i(\pi)$ and $S_j(\pi)$, must appear as follows:





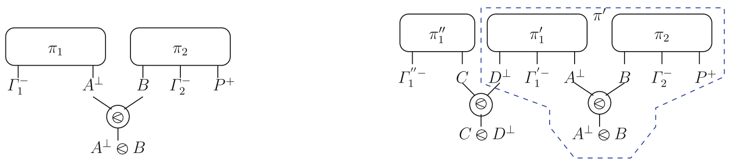
so, by restriction, $(X, Y, A) \in S_i(\pi_A) \downarrow^{\Gamma_1, A}$ and $(X, Y, A) \notin S_j(\pi_A) \downarrow^{\Gamma_1, A}$, contradicting the assumption (by minimality) that π_A is a correct PN with a cyclic order on its conclusions $\Gamma_1', X, Y, A = \Gamma_1, A$.

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

A.4 Proof of Theorem 5: Sequentialization of Lambek CyMLL PNs

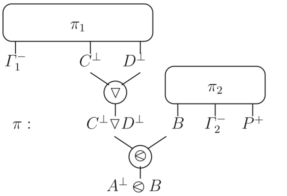
Proof. Assume by absurdum there exists a pure Lambek CyMLL proof net π that does not sequentialize into a Lambek CyMLL proof. We can chose π minimal w.r.t. the size. Clearly, π cannot be reduced to an axiom link; moreover π contains neither a negative conclusion of type $A^\perp \triangledown B^\perp$ nor a positive conclusion of type $A^\perp \triangledown B$ (resp., $A \triangledown B^\perp$), otherwise, we could remove this terminal $\triangledown$-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 $\triangledown$-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 $\overline{A^\perp \quad A}$, by Lemmas 1 and 2, there must exist either a (negative) splitting $\otimes$-link (Case 1) or a splitting cut-link (Case 2).

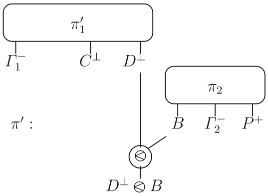
Case 1. Assume a negative splitting conclusion $A^\perp \otimes B$ (resp., $A \otimes B^\perp$). By minimality, π must split like in the next left hand side picture (we use A^+, resp. A^-, to denote positive, resp., negative, LF and Γ^- for sequence of negative LFs):



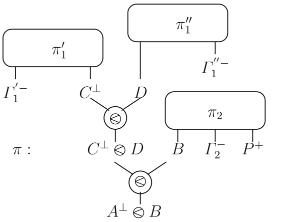
Now, let us reason on π_1 (reasoning on π_2 is symmetric): by minimality of π, π_1 cannot be reduced to an axiom link (otherwise Γ_1^- would not be negative); moreover, none of Γ_1^- is a (negative) splitting link, like e..g., $C \otimes D^\perp$, otherwise we could easily restrict to consider the sub-proof-net π', obtained by erasing from π the sub-proof-net π_1'' (with conclusions $\Gamma_1''^-, C$) 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:

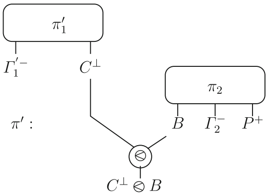
1. either $A^\perp = C^\perp \triangledown D^\perp$, then from the PN π on the l.h.s. of the next figure, we can easily get the non sequentializable PN π' (on the r.h.s.); π' is strictly smaller than π, contradicting the minimality assumption:





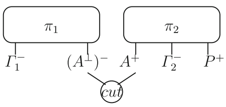
2. or $A^\perp = C^\perp \otimes D$, then this $C^\perp \otimes D$-link must split by Lemma 1, since π_1 is a cut-free PN in splitting condition without other $\otimes$-splitting conclusion in Γ_1^-; so from π on the l.h.s., we can easily get the non sequentializable PN π' on r.h.s.; π' is strictly smaller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

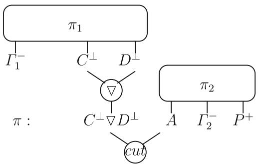


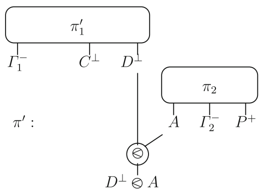


Case 2. Assume π contains a splitting cut link, like the leftmost hand side picture below, then we proceed like in Case 1. We reason on π_1 with two sub-cases:

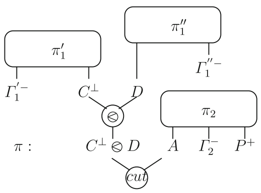
1. either $A^\perp = C^\perp \triangledown D^\perp$, then we can easily get, starting from the PN π on the middle side below, a non sequentializable PN π', like the rightmost hand side picture; $\pi'$is strictly smaller than π, contradicting the minimality assumption:

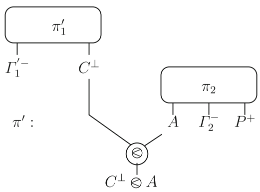






2. or $A^\perp = C^\perp \otimes D$, then this $A^\perp$-link must be splitting by Lemma 1, since π_1 is a cut-free PN in splitting condition without any other $\otimes$-splitting conclusion in Γ_1^-; so, we can easily get, starting from the PN π on the l.h.s., a non sequentializable PN π' that is strictly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





References

1. Abrusci, V.M.: Classical conservative extensions of Lambek calculus. Stud. Logica. **71**(3), 277–314 (2002)
2. Abrusci, V.M., Ruet, P.: Non-commutative logic I: the multiplicative fragment. Ann. Pure Appl. Logic **101**(1), 29–64 (2000)
3. Andreoli J.-M. and Pareschi, R.: From Lambek calculus to word-based parsing. In: Proceedings of Workshop on Substructural Logic and Categorial Grammar, CIS Munchen, Germany (1991)
4. Danos, V., Regnier, L.: The structure of multiplicatives. Arch. Math. Logic **28**, 181–203 (1989)
5. Girard, J.-Y.: Linear logic. Theoret. Comput. Sci. **50**, 1–102 (1987)
6. Girard, J.-Y.: Proof-nets: the parallel syntax for proof-theory. In: Agliano, U. (ed.) Logic and Algebra (1996)
7. Girard, J.-Y.: Le point aveugle. Cours de Logique. Vers la Perfection. Edward Hermann, Paris (2006)
8. Hughes D., van Glabbeek, R.: Proof nets for unit-free multiplicative-additive linear logic. In: Proceedings of the 18th IEEE Logic in Computer Science. IEEE Computer Society Press, Los Alamitos (2003)
9. Lambek, J.: The mathematics of sentence structure. Am. Math. Montly **65**, 154–170 (1958)
10. Melliès, P.-A.: A topological correctness criterion for multiplicative non commutative logic. In: Ehrhard, T., Girard, J.-Y., Ruet, P., Scott, P. (eds.) Linear Logic in Computer Science. London Mathematical Society Lecture Notes, vol. 316, pp. 283–321. Cambridge University Press, Cambridge (2004). Chapter 8
11. Maieli, R.: A new correctness criterion for multiplicative non commutative proof-nets. Arch. Math. Logic **42**, 205–220 (2003). Springer-Verlag
12. Maieli, R.: Retractile proof nets of the purely multiplicative and additive fragment of linear logic. In: Dershowitz, N., Voronkov, A. (eds.) LPAR 2007. LNCS (LNAI), vol. 4790, pp. 363–377. Springer, Heidelberg (2007)
13. Moot, R., Retoré, C.: The Logic of Categorial Grammars: A Deductive Account. LNCS, vol. 6850. Springer, Heidelberg (2012)
14. Moot, R.: Proof nets for linguistic analysis. Ph.D. thesis, Utrecht University (2002)
15. Pogodalla, S., Retoré, C.: Handsome non-commutative proof-nets: perfect matchings, series-parallel orders and hamiltonian circuits. Technical Report RR-5409, INRIA. In: Proceedings of Categorial Grammars, Montpellier, France (2004)
16. Retoré, C.: A semantic characterization of the correctness of a proof net. Math. Struct. Comput. Sci. **7**(5), 445–452 (1997)
17. Retoré, C.: Calcul de Lambek et logique linéaire. Traitement Automatique des Langues **37**(2), 39–70 (1996)
18. Roorda, D.: Proof nets for Lambek calculus. J. Logic Comput. **2**(2), 211–233 (1992)