Proof nets for the Lambek calculus — an overview

François Lamarche and Christian Retoré
{lamarche,retore}@loria.fr
Projet Calligramme, INRIA-Lorraine & CRIN-C.N.R.S.
BP 101, 54 602 Villers-lès-Nancy, France

1 Introduction: the interest of proof nets for categorial grammar

There are both linguistic and mathematical reasons for studying proof nets the perspective of categorial grammar. It is now well known that the Lambek calculus corresponds to *intuitionnistic non-commutative multiplicative linear logic* — with no empty antecedent, to be absolutely precise. As natural deduction underlines the constructive contents of intuitionistic logic (Curry-Howard isomorphism), and the sequent calculus stresses the symmetries of classical logic, the syntax of proof nets [13,14] throws light on the symmetries and computational contents of linear logic. Thus it is a mathematically natural question to look for proof nets for the Lambek calculus, but which is also relevant in a more linguistic perspective.

The first expected mathematical outcome is a better understanding of the Lambek calculus. For example, cut-elimination [14] and interpolation [4] are particulary clear in this setting. The second mathematical outcome could be conclusive results on the complexity of the Lambek calculus. As conjectured by Chomsky in 1963 [6], and proved by Pentus in 1992 [23], the generative power of the Lambek calculus is exactly that of Contex-Free Grammars, which may be recognized in polynomial time, for example using the Cocke-Kasami-Younger algorithm [16]. On the other hand, the decision problems for the linear logic systems that are most closely related to the Lambek calculus are all NP-complete, when they are known, but next to nothing is known about the non-commutative variants. So the Lambek calculus has been conjectured by some to be polynomial, and by others to be NP-complete, but there is no answer yet.

From the linguistic angle, proof nets make the relation between various systems clearer. As it is known that the expressive power of the Lambek calculus is not sufficient for linguistic purposes, people look at extensions of the Lambek calculus. Linear logic offers several possibilities and suggestions in the search for a neat logical calculus, bringing along the ease of describing proofs through the proof net setting. Last comes a theoretical motivation, which is according to us the most important. In linguistics the structure which represents the analysis is highly important: see for example how Chomsky moved from Strings to Derivational Trees and back to Strings during the evolution of the theory [28,8]. We think that a proof net is in itself a linguistically relevant structure, and some existing work confirms our belief:

- M. E. Johnson [17] shows that syntactic analysis by the means of proof nets can catch a difficult linguistic notion: the instantaneous complexity in understanding a sentence with nested relatives. This measure consists in two integers easily computed from the proof net. It fits the statistical data, and it has been shown that although numerous parsers have been defined for Context-Free Grammars, none is able to exhibit the understanding of this form of complexity.

- De Groote and Retoré [10] show that proof nets are extremely convenient for computing the semantics of a sentence out of the semantics of the words and of the syntactical analysis. The reason why is that both lambda terms and proofs in the Lambek calculus can be formulated as proof nets. The transition from syntax to semantics is smoothened by the existence of a common logical framework. In fact the added expressiveness of proof nets leads one to believe that they may replace lambda terms, allowing for more subtle semantic recipes.

- Bechet and de Groote [4] show how proof nets can be used to construct the different possible phonological bracketings. The key technique is a Craig-like interpolation theorem, whose formulation is wholly dependent on proof nets.

- Abrusci, Fouqueré and Vauzeilles [1] show how Tree Adjoining Grammars may be represented in the Lambek calculus: without proof nets, it would have been much harder to find this representation, which is very perspicuous in the proof net syntax — because of the tree-like structure of proof nets.

For all these reasons proof nets are interesting for categorial grammar, and it is now time to explain what they are, and the various system they can be applied to.

2 The systems considered and their relations

Let $\mathcal{P} = \{\alpha, \beta, ...\}$ be a set of propositional variables.

2.1 The Lambek calculus (short reminder)

The language $\mathcal{L}$ for the Lambek calculus L is defined from the set of propositional variables $\mathcal{P}$ using three binary connectives: $\backslash$, $/$, $\cdot$.

$$\mathcal{L} ::= \mathcal{P} \mid \mathcal{L} \setminus \mathcal{L} \mid \mathcal{L} \,/\, \mathcal{L} \mid \mathcal{L} \cdot \mathcal{L}$$

The axioms of the calculus are

$$\alpha \vdash \alpha \quad \textbf{Ax-lr}$$

for every atomic variable α, and as usual there are two introduction rules for every connective, a left and a right one:

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \cdot B, \Gamma' \vdash C}\ \cdot\textbf{-l} \qquad \frac{\Gamma \vdash A \quad \Gamma' \vdash B,}{\Gamma, \Gamma' \vdash A \cdot B,}\ \cdot\textbf{-r}$$

$$\frac{A, \Gamma \vdash B}{\Gamma \vdash A \setminus B}\ \backslash\textbf{-r} \qquad \frac{\Gamma \vdash A \quad \Gamma_1, B, \Gamma_2 \vdash C}{\Gamma_1, \Gamma, A \setminus B, \Gamma_2 \vdash C}\ \backslash\textbf{-l}$$

$$\frac{\Gamma, B \vdash A}{\Gamma \vdash A \,/\, B}\ /\textbf{-r} \qquad \frac{\Gamma_1, A, \Gamma_2 \vdash C \quad \Gamma \vdash B}{\Gamma_1, A \,/\, B, \Gamma, \Gamma_2 \vdash C}\ /\textbf{-l}$$

In addition, there is the restriction that every sequent used in a derivation must have a nonempty list of formulas at the left of the turnstile; the calculus where this restriction is absent will be called L_ε. Sometimes there are reasons for extending the set of axioms to $A \vdash A$ for any formula A, but this can be derived very easily in the system.

As opposed to other well known logical systems, this calculus is:

linear i.e. there is neither the weakening nor the contraction rule: each hypothesis is used exactly once,

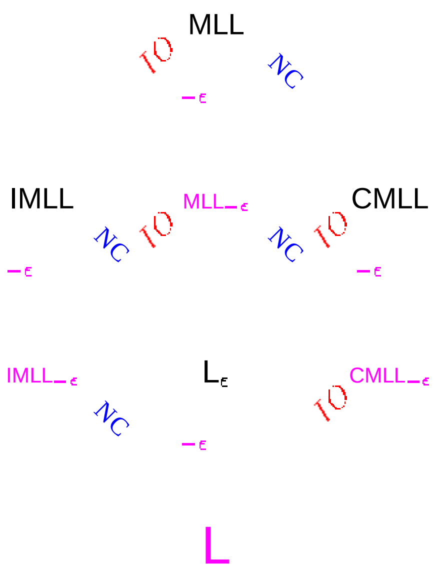
multiplicative i.e. the context of a conclusion sequent is always the disjoint union of the contexts appearing in the premiss sequents (modulo the symbol introduced),

intuitionistic i.e. exactly one formula appears on the right hand side of a sequent,

non-commutative i.e. there is no exchange rule, which prohibits the law $A \cdot B \equiv B \cdot A$

empty-sequence-free as explicitly stated in the rules — this is not very natural from a logical point of view, but it is clearly well motivated linguistically, and is easily handled in the logical system.

Among the intermediate systems related to the Lambek calculus, one deserves special interest: classical multiplicative linear logic MLL. This is because of its simplicity, especially with respect with the associated proof nets. MLL can be seen as the removal of the last three restrictions above from L, and the addition of an involutive negation operator, which makes it a “classical” system, as opposed to an “intuitionistic” one like L. The additional symmetry is what allows great simplifications in the presentation of proofs and related objects. Among the logical systems we will consider here, it is the most powerful one and should be seen as a universal system, in which the other ones, including L, will be conservatively embedded as subsystems. Thus, according to which of these three restrictions are added to MLL, we get eight possible systems, that can be arranged in a cube: below, prefixing with the letter I means adding the intuitionistic restriction, prefixing with C means adding non-commutativity (the reason for using this letter will be seen below), and the subscript $(-)_{-\varepsilon}$ means adding the no-empty-sequence-at-the-left rule… it is the inverse of the operation that makes L_ε out of L.



2.2 From the Lambek calculus with permutations to MLL...

The calculus LP, defined over the same language $\mathcal{L}$ as L and is obtained from it by adding the (intuitionistic) exchange rule:

$$\frac{A_1, A_2 \ldots, A_n \vdash B}{A_{\sigma(1)}, A_{\sigma(2)}, \ldots, A_{\sigma(n)} \vdash B}$$

where σ is any permutation of the set $\{1, 2, \ldots, n\}$. As above, it comes in two versions, LP and LP_ϵ, where in the latter case sequents are allowed to have an empty premiss. Now it is well known that any permutation can be obtained by composing transpositions, and that a circular permutation will generate enough transpositions from a single one to obtain all permutations. Thus the two following rules together are equivalent to the general exchange rule given above.

$$\frac{A_1, A_2, \Gamma \vdash B}{A_2, A_1, \Gamma \vdash B}\,\textbf{Exch-1-2} \qquad \frac{A_1, A_2, \Gamma \vdash B}{A_2, \Gamma, A_1 \vdash B.}\,\textbf{Cycl}$$

In this calculus the equivalence $A \cdot B \equiv B \cdot A$ trivially holds, and the distinction between the two implications of the Lambek calculus disappears. In other words we always have that $A \setminus B$ is provably equivalent to $B \,/\, A$; this is very easy to show.

What would a "classical" version of LP be? By this we mean a calculus that would have an involutive negation $(-)^{\perp}$:

$$(A^{\perp})^{\perp} \;\equiv\; A.$$

Such a calculus requires the sequents to have more than one formula at the right, since introducing a negation on a formula on the left side makes it migrate to the right:

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}\,\textbf{Neg-l-r}$$

Obviously we want the Exchange rule to the right of the turnstile as well as to the left. The involutivity of negation makes the symmetrical introduction rule very natural:

$$\frac{\Gamma \vdash A, \Delta}{A^{\perp}, \Gamma \vdash \Delta.}\,\textbf{Neg-r-l}$$

The conjunction in this calculus will be denoted $\otimes$; if we forget that there is now an arbitrary number of formulas at the right, it has the same introduction rules as the old $\cdot$, unary to the left and binary to the right:

$$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, A \otimes B, \Gamma' \vdash \Delta}\,\otimes\textbf{-l} \qquad \frac{\Gamma \vdash \Delta, A \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash \Delta, A \otimes B, \Delta'}\,\otimes\textbf{-r}$$

but there is a new operator $\parr$, a disjunction, which has the same relationship with the commas at the right of the turnstile as $\otimes$ has with the commas at the left (in particular giving meaning to the right-hand comma as a logical operator)

$$\frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, A \parr B, \Delta'}\,\parr\textbf{-r} \qquad \frac{\Gamma, A \vdash \Delta \quad B, \Gamma' \vdash \Delta'}{\Gamma, A \parr B, \Gamma' \vdash \Delta, \Delta'}\,\parr\textbf{-l}$$

Thus the two de Morgan laws are easily provable (exercise):

$$\begin{array}{rcl}(A \otimes B)^\perp & \vdash\dashv & B^\perp \parr A^\perp \\ (A \parr B)^\perp & \vdash\dashv & B^\perp \otimes A^\perp.\end{array}$$

Notice that we could have stated, say, the first law as $(A \otimes B)^\perp \equiv A^\perp \parr B^\perp$. Since here we have the exchange rule these two ways of presenting the first de Morgan law are perfectly valid, but the reason we favored the version that reverses the order is that it can be proved *without* the exchange rule; thus it is simpler. Since we want to go non-commutative later we want as much as possible of what we present to stay valid in that context.

Just as in ordinary Boolean logic, we can consider implication as a defined symbol; for the same reason as just stated we keep the two symbols, even if only one is necessary with the Exchange rule:

$$A \setminus B \;=\; A^\perp \parr B, \qquad A \,/\, B \;=\; A \parr B^\perp.$$

They have pretty much the same introduction rules as before (things can't be exactly the same because of the right-hand sides are different in the intuitionistic and classical systems), but now these rules are *derived* in our system, e.g. :

$$\dfrac{\dfrac{A, \Gamma \vdash B}{\Gamma \vdash A^\perp, B}\,\textbf{Neg-l-r}}{\Gamma \vdash A \setminus B}\,\backslash\textbf{-r} \qquad \dfrac{\dfrac{\Gamma \vdash A, \Delta}{\Gamma, A^\perp \vdash \Delta}\,\textbf{Neg-r-l} \qquad B, \Gamma' \vdash \Delta'}{\Gamma, A \setminus B, \Gamma' \vdash \Delta, \Delta'}\,\parr\textbf{-l}$$

All this is very nice but we are far from having made full use of the symmetries we have encountered. For instance, there is some amount of redundancy in the introduction rules, since the system is invariant under the automorphism that simultaneously exchanges $\otimes \leftrightarrow \parr$ along with left $\leftrightarrow$ right. Notice that, as in Boolean logic, every formula has a normal form equivalent to it, where the negation operator is only applied to variables: to find this normal form just apply de Morgan repeatedly, pushing the negations inwards until they reach the variables. This leads to the following formalization of MLL. First, to our set $\mathcal{P}$ of propositional variables/atomic formulas we add an isomorphic set $\mathcal{P}^\perp = \{\alpha^\perp, \beta^\perp, \ldots\}$ whose elements are the negated versions of the variables in $\mathcal{P}$.

The set of formulas $\mathcal{M}$ of MLL is now defined as:

$$\mathcal{M} ::= \mathcal{P} \cup \mathcal{P}^\perp \mid \mathcal{M} \parr \mathcal{M} \mid \mathcal{M} \otimes \mathcal{M}.$$

Thus negation in MLL is a *defined* operator, that obeys:

$$\begin{array}{rclcrcl}(\alpha)^\perp & = & \alpha^\perp, & \qquad & (\alpha^\perp)^\perp & = & \alpha, \\ (A \otimes B)^\perp & = & B^\perp \parr A^\perp, & \qquad & (A \parr B)^\perp & = & B^\perp \otimes A^\perp,\end{array}$$

(notice again the reversal of the orders… we have ulterior motives) and the equivalence of $(A^\perp)^\perp$ and A is now a *syntactical identity*. One outcome of this is that MLL has an extremely simple sequent calculus, where the sequents are one-sided (we think it is Tait [29] who first noticed that the two remarks in the paragraph above led to simplified calculi that used one-sided sequents. Naturally he did this in the context of classical logic). The axioms are of the form

$$\vdash \alpha^\perp, \alpha \quad \textbf{Ax-r},$$

the only structural rule is Exchange, and there are only two introduction rules: $\otimes$**-r** and $\parr$**-r**. Then it is trivial to show by induction that one can only prove sequents whose left parts are empty.

The two-sided rules that we first presented are now derived rules: a sequent $\Gamma \vdash \Delta$ is translated in MLL as $\vdash \Gamma^{\perp}, \Delta$ (where, given a list $\Gamma = A_1, A_2, \ldots, A_n$ of formulas, $\Gamma^{\perp}$ is the list $A_n^{\perp}, \ldots, A_2^{\perp}, A_1^{\perp}$, the reversal of order being in accordance with our convention on de Morgan duality… after all, comma is $\parr$. Remember also that a formula like $A^{\perp}$ is not the addition of a single symbol to A) For example, the following are all different representations for the *same* sequent of MLL.

$$C^{\perp}, B, A \vdash \qquad B, A \vdash C \qquad A \vdash B^{\perp}, C \qquad \vdash A^{\perp}, B^{\perp}, C.$$

2.3 …and back

We want to show that LP_ϵ can be thought of as a subsystem of MLL, via the standard translation. To every formula A of $\mathcal{L}$ we will associate a formula $A^\flat$ in $\mathcal{M}$ by (unsurprisingly):

$$\begin{array}{rclcrcl} \alpha^\flat & = & \alpha & \qquad & (X \cdot Y)^\flat & = & X^\flat \otimes Y^\flat \\ (X \setminus Y)^\flat & = & (X^\flat)^{\perp} \parr Y^\flat & & (X \,/\, Y)^\flat & = & X^\flat \parr (Y^\flat)^{\perp}. \end{array}$$

Remember again that here the negation operator is not a primitive symbol of our system, but defined using induction and de Morgan's law.

Let us characterize the image in $\mathcal{M}$ of that translation: let first $\mathcal{M}^\circ$ denote the set of all formulas of the form $A^\flat$, and $\mathcal{M}^\bullet$ the set of all formulas of the form $(A^\flat)^{\perp}$. We say that a formula $X \in \mathcal{M}^\circ$ has polarity Output (and when this is the case we write X°), while an $X \in \mathcal{M}^\bullet$ has polarity Input (which we express by $X^\bullet$). There are formulas of $\mathcal{M}$ that cannot have a polarity, for instance $\alpha \parr \alpha$. It is easy to see that $\mathcal{M}^\circ$ and $\mathcal{M}^\bullet$ can be given the following recursive definition:

$$\begin{array}{rcll} \mathcal{M}^\circ & ::= & \mathcal{P} & \mid \mathcal{M}^\circ \otimes \mathcal{M}^\circ \mid \mathcal{M}^\circ \parr \mathcal{M}^\bullet \mid \mathcal{M}^\bullet \parr \mathcal{M}^\circ \\ \mathcal{M}^\bullet & ::= & \mathcal{P}^{\perp} & \mid \mathcal{M}^\bullet \parr \mathcal{M}^\bullet \mid \mathcal{M}^\bullet \otimes \mathcal{M}^\circ \mid \mathcal{M}^\circ \otimes \mathcal{M}^\bullet, \end{array}$$

and since no clause appears above more than once, that $\mathcal{M}^\circ \cap \mathcal{M}^\bullet$ is empty. In other words polarity is a partial funtion $\mathcal{M} \to \{\circ, \bullet\}$. It is also quite obvious that negation reverses polarity. Moreover, the translation is injective:

Proposition 2.1 *If X is a formula of $\mathcal{M}^\circ$ then there is a unique $X^\sharp$ in $\mathcal{L}$ such that $(X^\sharp)^\flat = X$.*

The proof is by induction. If X is a variable, we take $X^\sharp = X$. If $X = Y \otimes Z$, then both Y, Z have polarity Output, and we take $X^\sharp = Y^\sharp \cdot Z^\sharp$. If $X = Y \parr Z$, then Y and Z have opposite polarity. If Y° then $(Z^{\perp})^\circ$ and we take $X^\sharp = Y^\sharp \,/\, (Z^{\perp})^\sharp$. Similarly if $Y^\bullet$ we take $X^\sharp = (Y^{\perp})^\sharp \setminus Z^\sharp$.

It is only natural, given a sequent $A_1, \ldots, A_n \vdash B$ in LP, to translate it as $\vdash (A_n^\flat)^{\perp}, \ldots, (A_1^\flat)^{\perp}, B^\flat$. Notice that if we replace every comma by a $\parr$, turning a sequent into a formula, in whatever order we want, sequents that have exactly one formula of polarity Output are exactly those that become formulas of polarity Output.

Thus we can define Intuitionistic Multiplicative Linear Logic (IMLL)[1] as the fragment of MLL whose sequents are of the form $\vdash X_1, \ldots, X_n, Y$, where Y° and the $X_i^\bullet$. The introduction rules are those of MLL, but the restriction that premiss and conclusion sequents have to respect the polarity rules make it so that the introduction rules cannot be applied as

[1] To the best of our knowledge the use of polarities to characterize the image of inutitionisitic linear logic into the classical system was inaugurated by van de Wiele in some unpublished work. The first published use of polarities seems to be in Roorda [27] and Bellin and Scott [5].

freely as before. In LP or LP_ϵ the formula at the right is distinguished by its *position*, while in IMLL it will be distinguished by its *polarity*: because of the exchange rule it can appear at the beginning or the middle of the sequent, but is nonetheless distinguished from the other formulas. The other formulas form the "left part" of the intuitionistic sequent.

Proposition 2.2 *A sequent $\Gamma \vdash B$ is provable in LP_ϵ iff $\vdash (\Gamma^\flat)^\perp, B^\flat$ is provable in IMLL.*

The proof is simply the construction of a bijective correspondence between the sets of introduction rules in the two calculi. A left (resp. right) introduction rule in LP will be associated to the introduction of an Input (resp. Output) connective in IMLL. All the unary rules in LP translate as $⅋$-introduction, the different connectives being distinguished solely by the polarities of the two subformulas of the formula introduced. In the same manner all the binary rules translate as $\otimes$-introductions. For example the **$\backslash$-l** rule we have given translates as (here all the formulas should be decorated with $\flat$)

$$\dfrac{\dfrac{\vdash \Gamma_2^\perp, B^\perp, \Gamma_1^\perp, C^\circ}{\vdash \Gamma_1^\perp, C, \Gamma_2^\perp, B^\perp}\,\textbf{Cycl} \qquad \dfrac{\vdash \Gamma^\perp, A^\circ}{\vdash A, \Gamma^\perp}\,\textbf{Cycl}}{\dfrac{\vdash \Gamma_1^\perp, C, \Gamma_2^\perp, B^\perp \otimes A, \Gamma^\perp}{\vdash \Gamma_2^\perp, B^\perp \otimes A, \Gamma^\perp, \Gamma_1^\perp, C}\,\textbf{Cycl}}\,\otimes\textbf{-r}$$

the conclusion being exactly the translation of $\Gamma_1, \Gamma, A \setminus B, \Gamma_2 \vdash C$. Notice that we have used the cyclic exchange rule, but only that one. Conversely, given an instance of the **$\otimes$-r** rule in a deduction of IMLL, by looking at the polarity of the two premiss subformulas one can tell if it is the translation of a **$\backslash$-l** or a **$\backslash$-r** rule and recover the intuitionistic two-sided sequents.

Therefore, we have the right to *identify* the calculus LP_ϵ with IMLL, and in the same manner, corresponding to the calculus LP there is a one-sided equivalent $\mathsf{IMLL}_{-\epsilon}$, which is just IMLL with the added rule that *every sequent must have more than one formula*.

Note Since we take MLL to be the fundamental system, we tend to translate weaker logics into it, and we end up specializing tensor and par. But there is another notational approach, due to Roorda [27], that allows one to give a one-sided-sequent presentation of intuitionistic systems like L and LP without having direct recourse to the classical system. The idea is to use two versions each for the connectors $\cdot, \backslash, /$, i.e. an Output or Positive version, and an Input or Negative version. This has some advantages if one wants to restrict oneself to purely intuionistic systems, but hides some of the symmetries that are present there.

2.4 Non-commutativity

What notion of non-commutativity may also apply to a classical system? notice that non-commutatity may not be defined by saying that the conjunction is non-commutative. This is a consequence of the rules, which may be derived or not.

We now want to discuss the potential "classical" extensions of the Lambek calculus, i.e. possibilities of systems with no or weak versions of the exchange rule, but some kind of negation operator(s) which would provide added symmetries like de Morgan laws. This is done by allowing several formulas in the right hand side, and looking for the possible rules of negation. It is easier to think of it on the double sided sequent calculus, since in this case negation appears as a connective, defined by rules, which moves a formula from one side to another.

In the same manner as before, a classical extension of the Lambek calculus will introduce a disjunction, which we will also write $\wp$, corresponding to the comma on the right hand side, and a negation $*$ than makes, say, $\backslash$ definable as $A^* \wp B$. As it is well known that $A \backslash B \backslash C$ is $(B \cdot A) \backslash C$, one has $(A \cdot B)^* = B^* \wp A^*$, i.e. *negation necessarily reverses the order of its consituents*.

Comma on the left hand side is $\cdot$, while comma on the left hand side is $\wp$. Let $A_1, \ldots, A_n \vdash B_1, \ldots, B_p$ be a sequent. If negation were allowed to take any formula and put it anywhere, then we would be able to derive an exchange rule. But there are only four places that can be defined independently from the sequent itself: *left_first*, *left_last*, *right_first*, *right_last*. Thus a negation can only act on these places. Firstly, let us see that it can not move something from the *left_first* place to the *right_last* place. The sequent $A, B, \Gamma \vdash \Delta$ is equivalent to the sequent $A \cdot B, \Gamma \vdash \Delta$. If a negation takes the *left_first* formula and move it to the *right_last* place, the first sequent would lead to $\Lambda \vdash \Delta, B^*, A^*$ which should be the same as $\Lambda \vdash \Delta, A^* \wp B^*$ and the second one to $\Lambda \vdash \Delta, (A \otimes B)^*$, which should be, as explained above, the same as $\Lambda \vdash \Delta, B^* \wp A^*$. Thus this would lead to the commutativity of $\cdot$ and $\wp$, which forces the exchange rule. Via a similar argument, it is prohibited too that a negation rule move a formula from *left_last* to *right_first*, from *right_last* to *left_first* form *right_first* to *left_last*. So negation only moves a formula from a *left_first* place to a *right_first* place (and conversely), and from a *left_last* place to a *right_last* place. Let us assume these four rules are done by four different negations, and see which ones have to be identified.

This yields the following rules:

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash A^1, \Delta}\ \textbf{neg-1} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \Delta, A^2}\ \textbf{neg-2} \qquad \frac{\Gamma \vdash A, \Delta}{A^3, \Gamma, \vdash \Delta}\ \textbf{neg-3} \qquad \frac{\Gamma \vdash \Delta, A}{\Gamma, A^4 \vdash \Delta}\ \textbf{neg-4}$$

We have as usual that a round trip from one place to the other and back changes nothing, i.e.:

$A^{1\,3} \equiv A^{3\,1} \equiv A$ (negations 1 and 3 are the inverse of each other) $A^{2\,4} \equiv A^{4\,2} \equiv A$ (negations 2 and 4 are the inverse of each other)

These rules allow the following two proofs we call $P_{4\,1}$ and $P_{3\,2}$, valid for every formula A:

$$\frac{\dfrac{A \vdash A}{A, A^4 \vdash}\ \textbf{neg-4}}{A^4 \vdash A^1}\ \textbf{neg-1} \qquad \frac{\dfrac{A \vdash A}{A^3, A \vdash}\ \textbf{neg-3}}{A^3 \vdash A^2}\ \textbf{neg-2}$$

Applying $P_{4\,1}$ to $A = B^3$, yields $B^{3\,4} \vdash B^{3\,1}$, thus for any formula B: $B^{3\,4} \vdash B$.
Applying $P_{4\,1}$ to $A = B^2$, yields $B^{2\,4} \vdash B^{2\,1}$, thus for any formula B: $B \vdash B^{2\,1}$.
Applying $P_{3\,2}$ to $A = B^1$, yields $B^{1\,3} \vdash B^{1\,2}$, thus for any formula B: $B \vdash B^{1\,2}$.
Applying $P_{3\,2}$ to $A = B^4$, yields $B^{4\,3} \vdash B^{4\,2}$, thus for any formula B: $B^{4\,3} \vdash B$.
Therefore, by transitivity of $\vdash$, we have $B^{3\,4} \vdash B^{2\,1}$. From this we obtain:

$$\frac{\dfrac{\dfrac{B^{4\,3} \vdash B^{1\,2}}{B^{4\,3}, B^{1\,2\,4} \vdash}\ \textbf{neg-4}}{B^{1\,2\,4} \vdash B^{4\,3\,1}}}{B^1 \vdash B^4}\ \text{i.e.} \qquad \frac{\dfrac{\dfrac{B^{3\,4} \vdash B^{2\,1}}{B^{2\,1\,3}, B^{3\,4} \vdash}\ \textbf{neg-3}}{B^{2\,1\,3} \vdash B^{3\,4\,2}}}{B^2 \vdash B^3}\ \text{i.e.}$$

Thus negations 1 and 4 should be identified, as well as negations 3 and 2.

Is it possible to identify them further, i.e. to have a single negation written A^*, which would be its own inverse? Yes, but in this forces the cyclic exchange rule **Cycl** on right handed sequents:

$$\dfrac{\dfrac{\dfrac{\vdash X, \Delta}{X^\perp \vdash \Delta}}{\vdash \Delta, X^{**}}\ \textbf{neg-*}}{\vdash \Delta, X}\ \text{i.e.} \qquad \dfrac{\dfrac{\dfrac{\vdash \Delta, X}{X^* \vdash \Delta}}{\vdash X^{**}, \Delta}\ \textbf{neg-*}}{\vdash X, \Delta}\ \text{i.e.}$$

So we have here a justification for the two non-commutative classical multiplicative calculi that have been studied in depth:

- The calculus of M. Abrusci [2,3], which does not have any form of exchange rule, i.e. which is closer to the Lambek calculus. In this calculus $A^1 = A^4$ is denoted $A^\perp$, and $A^2 = A^3$ is written ${}^\perp A$. Thus ${}^\perp(A^\perp) \equiv ({}^\perp A)^\perp \equiv A$.

- The calculus of Yetter [31], obtained from the one of M. Abrusci by identifyling the two negations, which can be presented by a one-sided-sequent calculus that has the rule **Cycl**.

The calculus of Yetter being, regarding proof nets, simpler, we shall concentrate on it. However, one should read the paper by M. Abrusci in this volume, for more details.

The considerations above allow us to define CMLL as the calculus which is identical to MLL except for the one fact that the exchange rule is restricted to cyclic exchange. Notice that we introduced the cyclic exchange rule on the left side of ordinary intuitionistic (and therefore two-sided) sequents, but that so far we've only used it (and only want to use it) on *one-sided* sequents; the use of this rule must be restricted to one-sided sequents if we want it to be a weakening of the full exchange rule, and anyway it loses its reason for being in a two-sided classical calculus, since it can be derived there.

2.5 Combining restrictions

Notice that the change from MLL to $\mathsf{IMLL} = \mathsf{LP}_\epsilon$ was obtained by restricting the *formulas* under considerations, while CMLL is obtained from MLL by restricting the *structural rule* of Exchange. Naturally we can combine these two restrictions and get a new calculus, CIMLL, which turns out to be equivalent to L_ϵ:

Proposition 2.3 *Let $A_1, \ldots, A_n \vdash B$ be a sequent of L_ϵ. Then*

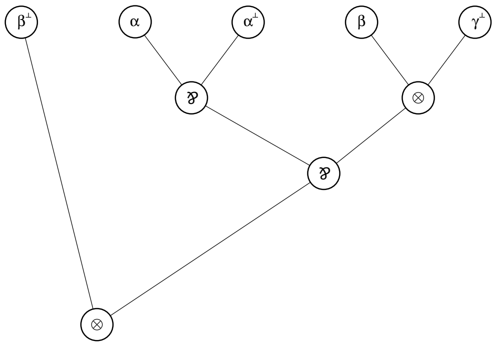
1. *it is provable in L_ϵ iff $\vdash (A_n^\flat)^\perp, \ldots, (A_1^\flat)^\perp, B^\flat$ is probable in CIMLL.*
2. *it is provable in L iff $\vdash (A_n^\flat)^\perp, \ldots, (A_1^\flat)^\perp, B^\flat$ has a proof in CIMLL where every sequent involved contains formulas of polarity Input.*

The proof is obtained by reading the proof of 2.2 again, noticing that full rule of Exchange was never used, only **Cycl**.

Therefore from now on we identify $\mathsf{CIMLL}$with L_ϵ, and $\mathsf{CIMLL}_{-\epsilon}$ with L.

3 The core: proof nets for MLL

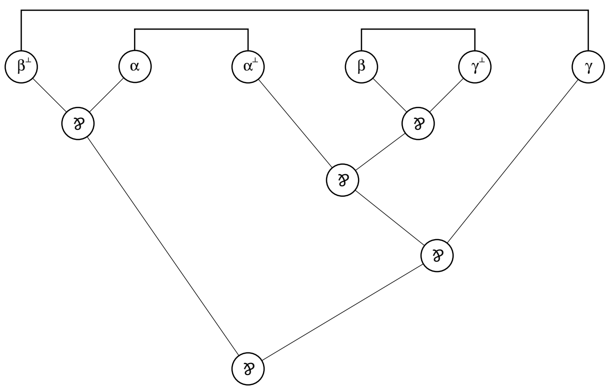
Proof nets are a very natural way of representing proofs in MLL, and have the same relationship to it as the simply typed lambda calculus has to the logic whose only connector is implication. But first let us recall some well-known elementary stuff [30]. Let X be a formula of $\mathcal{M}$. To X is associated a tree structure, whose leaves are the atomic subformulas (the convention in linear logic is to draw trees with the leaves at the top and the root at the bottom), and whose other nodes are decorated with the two connectors $\otimes, \parr$. For example the formula $\beta^\perp \otimes ((\alpha \parr \alpha^\perp) \parr (\beta \otimes \gamma^\perp))$ is seen as



Every node corresponds to a subformula, easily read by looking at everything above it (and vice versa). When we do not bother with expanding a tree in full, we will decorate a leaf node with a subformula.

The same goes if $\vdash A_1, \ldots, A_n$ is a one-sided sequent, except that now we get a forest with n roots instead of a single tree.

A *proof structure* over a sequent $\vdash \Gamma$ is the addition to its forest of a bijective matching that associates to every leaf labeled α a unique leaf labeled $\alpha^\perp$ (same variable, but negated version), and vice versa: every $\alpha^\perp$ must be matched by a unique α. Such a pair $\alpha, \alpha^\perp$ is called an *axiom link*, and is represented by adding the necessary edge to the forest graph of $\vdash \Gamma$. Obviously not every sequent can be turned into a proof structure, if the variable count is not right. Here is an example of a proof structure over a one-formula sequent.

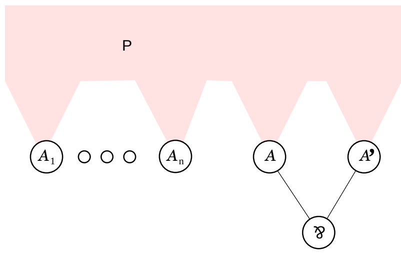


Example 3.1

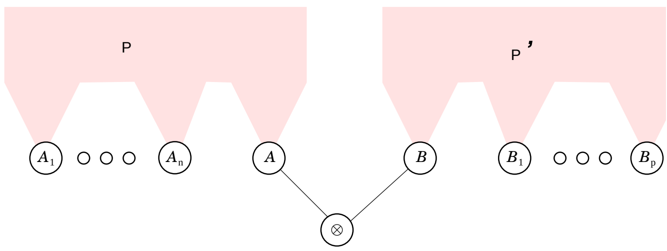
To every proof in the sequent calculus one associates a proof structure, which is called a *proof net*. The point of the distinction between nets and structures, naturally, is that there are proof structures that cannot be obtained that way. Proof nets are constructed by induction:

1. The proof net associated to an axiom $\vdash \alpha^\perp, \alpha$ is

2. if Π is a proof for the sequent $\vdash A_1, \ldots, A_n, A, A'$ (we will also call its associated proof net Π), and Π is extended by if a $⅋$-introduction to give $\vdash A_1, \ldots, A_n, A ⅋ A'$, one gets the proof net associated to the extended proof by simply adding a $⅋$-node to Π:



3. If a $\otimes$-introduction is done on Π (a proof of $\vdash A_1, \ldots, A_n, A$) and Π' (a proof of $\vdash B, B_1, \ldots B_p$), to yield the sequent $\vdash A_1, \ldots A_n, A \otimes B, B_1, \ldots, B_p$, one obtains a new proof net by putting the two nets together and adding a $\otimes$-node under A, B:



There is no need to interpret the Exchange rule, since its ultimate meaning is that there is no order structure whatsoever on the conclusions of a sequent… therefore there is no need for additional structure on the set of roots of the syntactical forest.

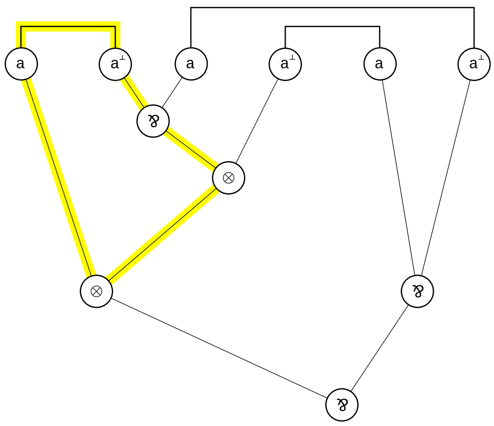
Thus the basic slogan is that the essence of a proof is contained in the axioms that were used in it… in addition to the sequent that's been proved itself! Different proofs in the sequent calculus can give rise to the same proof net. This should be ascribed to the well-known tendency of the sequent calculus towards overdeterminacy: one often finds that different sequent proofs should be considered as "the same", and there is a whole tradition of searching for normal forms by the rearranging the order of introduction rules. Proof nets give an alternative approach to the search for normal forms: a proof net is its own normal form. It naturally factors out from the sequent calculus what should be. This had already been noticed by category theorists [18] in the late sixties, as they were trying to construct what is known as free Symmetric Monoidal closed categories (here a morphism in such a category being seen as an equivalence class of proofs, the crucial thing being that two proofs should be identified *exactly* when they define the same proof structure/net.

What was not known at the time is that there are geometric conditions that allow one to tell if a given proof structure is a proof net or not. The first such *correctness criterion* was given by Girard [13], and uses a set of paths on the proof structure, such that every subformula/node is visited by these paths twice. A simpler version of that criterion was later given by Danos-Regnier [9] and it goes as follows:

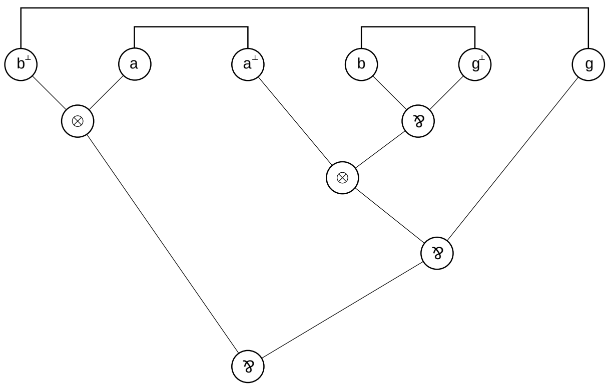
Definition 3.1 *Given a proof structure on sequent* $\vdash \Gamma$, *a* switching *is a map* S *from its set of* $\parr$*-nodes of to the set* $\{\text{left}, \text{right}\}$.

Theorem 1 *A proof structure is a proof net iff for every possible switching* S *the graph obtained by removing, for every* $\parr$*-link, the branch above it determined by* S*, is* acyclic *and* connected.

For example, the proof structure below is not a proof net, and this for two reasons: Every switching that removes the right branch above the upper left $\parr$-node will give rise to a cycle, shown highlighted. Moreover, any switching that in addition removes the right branch above the rightmost $\parr$-node will make the graph disconnected.



As an easy exercise, the reader should try to turn this proof structure into a proof net simply by changing some pars to tensors and vice versa. Another example of a proof structure which is not a net is the one given in Example 3.1. The following structure is a proof net, as the reader can check. Another easy exercise is to give a sequent derivation for that net.



How does one go about proving the theorem above? Obviously, a sensible strategy, given a proof structure that satisfies the Danos-Regnier correctness criterion (henceforth we will call such proof structures *correct*), would be to try to decompose it into smaller and smaller correct structures, in such a way that to every such decomposition step one could associate an application of one of the introduction rules, thus reconstructing a proof tree in the sequent calculus from the bottom (root) up. When we get a proof structure which is just an axiom link and nothing else, we have obviously reached an axiom (a leaf) of the proof tree and we are done.

One such decomposition steps is trivial: if a correct proof structure is associated to a sequent $\vdash \Gamma, A \parr B$ then it is immediate that the removal of the $\parr$-node under A, B will yield a correct structure, and the proof under construction will simply be incremented by a $\parr$-introduction. Thus if one does this systematically, removing all outermost pars, one is left with a correct structure whose syntactic forest has only tensors and (neg)atomic variables as roots. If there are no tensors, we have only (neg)atomic variables and there can only be two of them forming an axiom (otherwise the structure would not be connected). So in that case we are done. Otherwise, in order to go ahead, we need the non-trivial part of the proof:

Lemma 3.1 (Splitting tensor lemma) *Given a correct structure not an axiom link, all whose roots in the syntactic forest are tensors or variables, then there exists one tensor root whose removal will create two disjoint proof structures (it is easy to show that they will also be correct).*

Thus the proof being constructed by induction can be extended by a $\otimes$-introduction. This lemma is where the geometric properties of absence from cycles is used.

Let us conclude this section by saying that many other correctness criteria have been given, some being variations on this one. The Danos-Regnier criterion seems to take exponential time to verify, which one may think is too much: after all, we are simply checking whether a syntactical object is well-formed or not. But there are many other criteria e.g. [26] (and variations of Danos-Regnier) that are polynomial in time.

There are also alternative methods of proving the sequentialization theorem, where the search for a splitting tensor node is replaced by the search for something else to split the net in two.

4 An intrinsically Non-Commutative characterization

We show here how ordinary correctness criteria can be applied to non-commutative proof structures (and in our case this means CMLL), and also give a criterion, due to A. Fleury ([11], Chapitre 5), that only works in the non-commutative world. Although it applies to various non-commutative systems, we consider it for cyclic linear logic, and then for the Lambek calculus. There are other criteria specialized to CMLL, we mention in particular [22].

4.1 Proof nets for cyclic linear logic

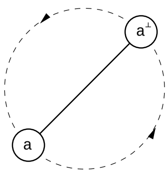
Due to the added constraint on the Exchange rule, the theory of proof nets for CMLL takes an even more geometric character.

Let $\vdash A_1, \ldots, A_n$ be a sequent of CMLL. It is endowed with a total order, but this is order is far from canonical, because of cyclic exchange. We want to think of the formulas arranged in a circle instead of a line, and this can be done by thinking of the sequent as a set $\{A_1, \ldots, A_n\}$ endowed with a *cyclic order*: a ternary relation $R \subset E^3$ (the meaning of $R(x, y, z)$ being "when going counterclockwise, x, y, z are met in that order") such that forall $a \in E$ $R_a(x, y) \equiv R(a, x, y)$ is a strict (antireflexive) total order and $R(x, y, z) \Leftrightarrow R(y, z, x)$.

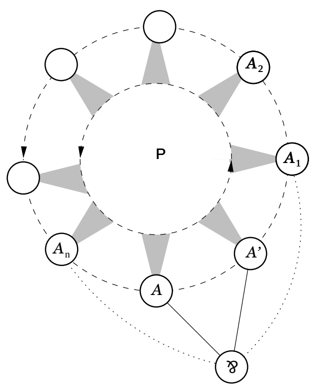
Notice that a cyclic order on conclusions induces a cyclic order on the (occurrences of) atoms involved in the A_i. One natural way to look at things (see pictures) is to think of a "crown", each one of whose "spikes" is a syntactic tree for a formula, the end of the spikes forming an outer circle, given by the cyclic order on conclusions, where the bases of the spikes are the collection of atomics/negatomics, which thus form an inner circle (with the induced cyclic order mentioned above). Thus a proof structure is simply the addition of axiom links, and these can be thought of lying *inside* the inner circle.

We now get the following inductive definition of a proof net:

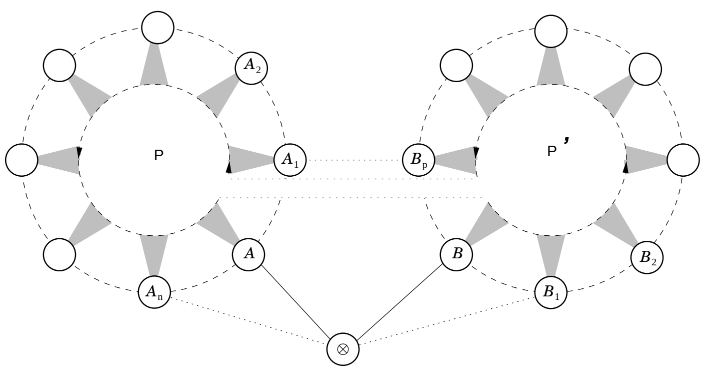
axiom



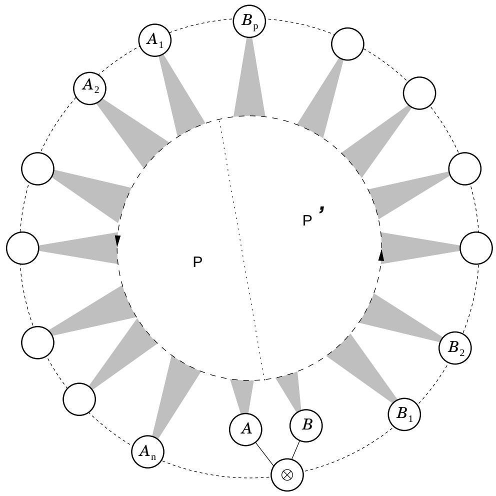
par rule



tensor rule



An easy rearrangement giving the following crown:



It is easy to see that a proof net thus constructed will obey a fundamental property: *it can be drawn on the plane in such a way that no two edges of the proof structure will cross*. Obviously the edges associated to syntactical trees cannot cross, and axiom links cannot cross syntactical edges since they are inside the inner circle; finally the fact that no two axiom links ever cross (if drawn with the least amount of care) is easy to see by induction. In other words the theory of cyclic proof nets can be expressed in terms of *planar graphs*. But the machinery of planar graph theory is rather formidable and leads to a lot of hair splitting, if one wants to be formal. Our aim is to stay as intuitive and informal as possible, and we will use only one easy-to-visualize notion:

Definition 4.1 *Let Π be a proof structure, drawn as above. It splits the plane in different connected components (puzzle pieces). A* face *is one such connected component.*

Since a planar graph is bounded, there is exactly one face which is *infinite*. The other faces will be called *internal*. For example, the proof structure below has two internal faces in addition to the infinite one (the circular order has been made linear for ease of drawing).

Given a sequent, it completely determines the "crown" part of the graph, and it is natural to restrict the definition of a proof structure for the cyclic calculus by requiring that the axiom links can be drawn inside the inner circle without crossing. This fact can actually expressed in a fashion less "topological," less dependent on *how* the links are drawn. Notice that if the circular order is opened anywhere and made into a line (as in the diagram below) then the axiom links form a *correct bracketing*, in the usual sense. And there is a notion of correct bracketing for circular orders, independent of what endpoints are chosen: let $\alpha_1, \ldots, \alpha_{2n}$ be the sequence of the atoms of the sequent; as said in the definition of ordinary proof structure, we can represent the axiom links as a bijective mapping $\mathcal{A}$ from $\{\alpha_1, \ldots, \alpha_{2n}\}$ to itself satisfying $\forall \alpha_i\ \mathcal{A}(\mathcal{A}(\alpha_i)) = \alpha_i \wedge \mathcal{A}(\alpha_i) \neq \alpha_i$. $\mathcal{S}$ is said to be a correct bracketing whenever it respects the cyclic order on $\alpha_1, \ldots, \alpha_{2n}$, i.e.

$$\forall \alpha, \beta\ R(\alpha, \beta, \mathcal{A}(\alpha)) \rightarrow R(\alpha, \mathcal{A}(\beta), \mathcal{A}(\alpha)).$$

Hence, given a sequent of CMLL we define a *cyclic* proof structure on it to be a set of axiom links which is correctly bracketed as above.

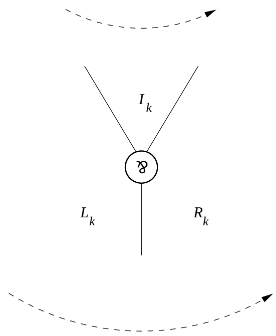
Proposition 4.1 *A cyclic proof structure which obeys the Danos-Regnier criterion (or any correctness criterion for* MLL*) is a proof net in* CMLL*.*

The proof is as before: conclusion pars are removed as usual, and the main additional observation to be made is that when using the tensor splitting lemma, the correct bracketing will guarantee that the two subnets obtained will split the big net in the right fashion, in other words that a sequent like $\vdash A, B \otimes A', B'$ will not be split into $\vdash A, A'$ and B, B'.

Thus the standard theory of proof nets specializes to CMLL. But there exist criteria that are specialized to this particular case, and that make use of the geometrical considerations we mentioned above, i.e. the notion of face.

It is easily observed, by induction, that each face of a cyclic proof net contains a unique $\parr$-link. Unfortunately, this is not enough to characterise the correct proof nets.

Given a $\parr$-link $\parr_k$ let us define its Inner face I_k, its Left face L_k, its Right face R_k as follows:



Notice that L_k and R_k can sometimes be identical for proof nets (and also I_k for proof structures). We define the relation $F_i < F_j$ on faces by:

$$F_i < F_j \text{ iff } \exists k(F_i = L_k \vee F_i = R_k) \wedge F_j = I_k$$

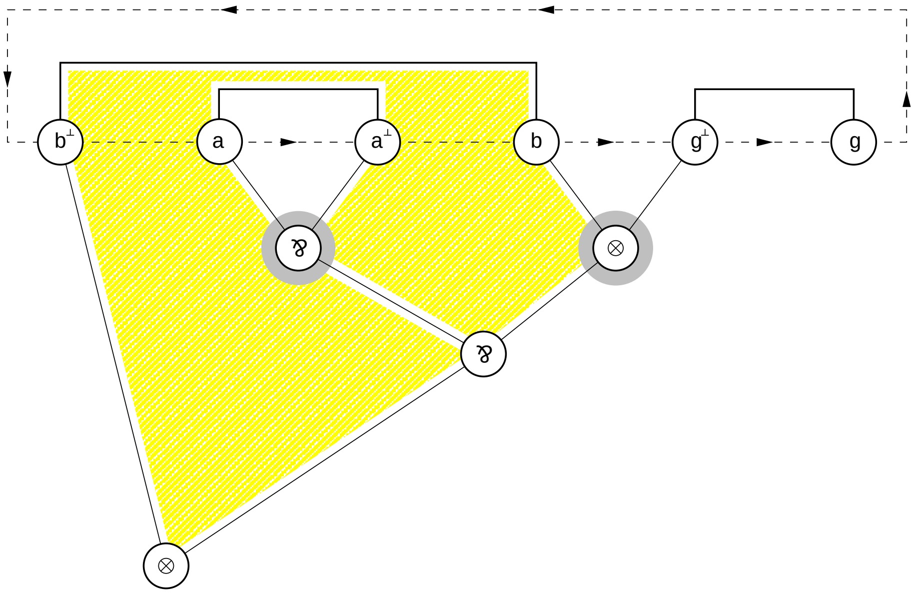
By induction we observe that $<$ is an acyclic relation.

In fact this characterises the cyclic proof nets:

Theorem 2 *A cyclic proof structure when drawn in the plane as above is a proof net for* CMLL *if and only if:*

1. *each internal face is the internal face I_k of a* unique *$\parr$-link $\parr_k$*
2. *the relation $<$ is acyclic — its transitive closure is a strict order.*

It should be observed that the first point is not enough to guarantee the correctness of a proof net. The following example does not fullfil the Danos-Regnier criterion: cutting the left branch of the rightmost par gives rise to both a cycle and a disconnected structure. But the face shaded face F satifies $F < F$, which refutes the second point.



Remark 4.1 • *A cyclic proof structure is checked to be a cyclic proof net in linear time.*

- *This criterion implies the Danos-Regnier criterion, for cyclic proof structures.*
- *Actually 1 may be replaced by the weaker 1': the number of internal faces is equal to the number of* $\parr$*-links.*
- *In presence of 1 or 1', 2 is equivalent to: each internal face* F_i *may be reached from the infinite face by* $<$*.*

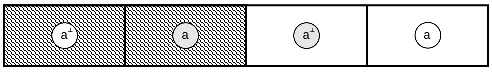
4.2 Proof nets for the extended Lambek calculus

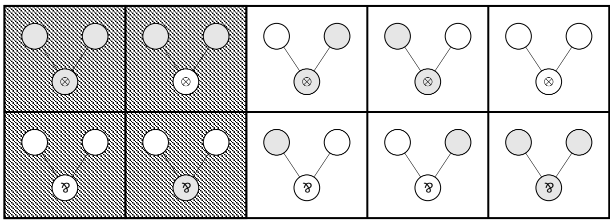
Given the conservativity result of 2.3, one can take a proof of a sequent $\Gamma \vdash A$ in the extended Lambek calculus L_ϵ and construct from it a cyclic proof net for $\vdash \Gamma^{\flat\perp}, A^\flat$, and vice versa: a proof net for a sequent of CIMLL is simply a cyclic proof net as above, for a sequent that obeys the polarity rules, and it is the representation of a proof in L_ϵ. As usual, the conclusion formula is distinguished by its polarity, and given this its associated intuitionistic sequent is uniquely defined. Then there are two possible natural total orderings on the conclusions: putting the Output formula A at the beginning, or at the end. In our translation of L_ϵ into CIMLL we have favored the second representation, but the first one is just as good; what matters is that the induced total ordering on $\Gamma^\flat$ is invariant. For example a proof net for $\vdash A_1^\bullet, A_2^\bullet, A_3^\circ, A_4^\bullet$ can only be the translation of a proof of $X_2, X_1, X_4 \vdash X_3$, where $A_4 = X_4^\flat$ and $A_i = X_i^{\flat\perp}$, $i \neq 4$.

5 An intrinsically Intuitionistic characterisation

Obviously the set of proof structures for IMLL is a subset of the one for MLL, since the distinction between the two logics is strictly about the set of formulas and sequents, and not about the connectors and their introduction rules. Therefore, given the conservativity result Proposition 2.2, a proof net for IMLL is simply a proof net for MLL whose sequent happens to lie in IMLL. But there are correctness criteria for IMLL that use the intuitionistic character of IMLL in an essential way, and therefore that cannot be extended to MLL. We will present such a criterion [19,20,27].

Given a sequent $\vdash \Gamma$ in IMLL, its syntactic forest will look just as before, except that now its nodes/subformulas should be decorated with their polarity. We use shading to do this, gray being $\bullet$ and white $\circ$. Therefore the permissible polarities for connectors will look like this (here the forbidden combinations are shaded, the permitted ones not).





Notice that the polarity information is sufficient to recover the connectives: inspection of that table shows that the combination of the polarity of a connective and that of its two sons determines if it's a tensor of a par. Notice also that there is always a son of the same polarity as the father. In particular from any node there is always a monochromatic branch all the way to the top.

Let $\vdash X_1^\bullet, \ldots, X_n^\bullet, Y^\circ$ be a sequent of IMLL. A proof structure for it is defined just as for MLL: it is the addition of a pairing of atomic and negatomic formulas to the syntactical forest. Given such a proof structure, a *path* for it is a sequence of nodes that obeys the following rules:

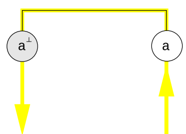
– A path always starts at the root of Y°.

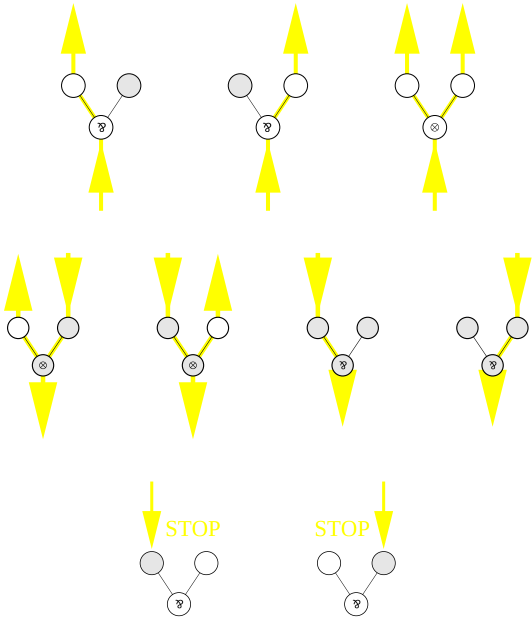
– If a path ends at a node x and there is z° immediately above it, then the path can be extended by z.

– If a path ends with a variable, say α°, it is matched by an axiom link to an $\alpha^{\perp\bullet}$, and the path can be extended by this negatomic.

– If a path ends with a node $x^\bullet$ whose father $z^\bullet$ has the same polarity, then it can be extended by z.

Thus the basic slogan is: start at the root of the output, always go up in polarity Output until you reach a leaf, then cross the axiom link and go down in polarity Input. You can always start going up again when there is something of polarity Output immediately above you. The rules can be summarized by:



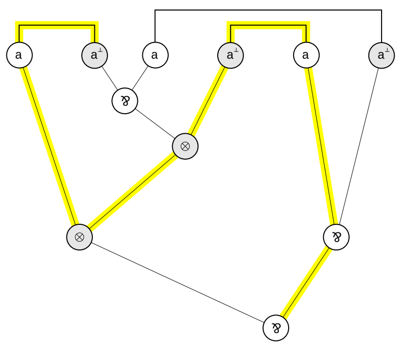


In particular the only nodes where there is a choice of which direction to go are the three tensors, where paths bifurcate. The only paths that cannot be extended are those that end at the root of a conclusion $X_i^\bullet$ or at the premiss of polarity $\bullet$ of a $⅋^\circ$.

Theorem 3 *A proof structure is a proof net iff*

1. *Every node can be reached by a path.*
2. *Every path is finite, i.e. there are no loops.*
3. *Given a configuration $x^\bullet ⅋ z^\circ$ then every path that ends in x goes through z.*

For example the following is not correct, the reason being that the third condition is not met: the higlighted path ends above a par node that it has not visited.



This criterion can also be used for the extended Lambek calculus L_ϵ, since all one has to do is cyclically order the conclusions and add the correct bracketing condition for the axiom links.

6 Excluding the empty sequence

This section describes the proof nets that come from the calculi where the empty antecedents are not allowed. As the technique applies to the various systems covered by this paper, namely MLL, IMLL, CMLL, and L_ϵ, we treat them all together, although this restriction is, up to now, only useful for L_ϵ, which thus yields the Lambek calculus L. Let us recall that this restriction is necessary for linguistic reasons — as already observed by Lambek in his seminal paper [21]. Assume that a word (or phrase) requires on its right (or left) a word (or phrase) whose type is a tautology of L. Then one could provide, instead of it, the empty word, thus leading to an incorrect phrase or sentence: for instance one could prove `a`:sn/n `very`:$(n/n)/(n/n)$ `paper`:$n \vdash sn$, where the word `good`: (n/n) has been omitted.

In the sequent calculus, one avoids the empty sequence by simply stating it. How do we express such a property in syntax of proof nets? First every sequent that appears in an ordinary proof is a "sub sequent" of the conclusion sequent (the meaning of this can be formalized easily). Given a proof structure, there is also a notion of "sub proof structure", which is simply a subforest of the syntactic forest (an up-closed subset of nodes), such that axiom links stay inside the substructure: an axiom link that has one end in the subforest will also have the other end in it. It is easy to see, using one's favorite correctness criterion, that if a proof structure is a proof net, then every sub structure will also be a net.

Therefore, given a proof in the sequent calculus, it gives rise not only to a proof net, but to a set of subnets, whose (reverse) ordering exactly mimics the tree ordering of the proof.

Thus, trivially, one could say that:

(*) *a proof net Π for sequent $\vdash \Gamma$ in the system S is a proof net of the system $S_{-\epsilon}$ if there exists a sequent proof of $\vdash \Gamma$ such that every associated subnet has two conclusions.*

Notice that, for intuitionistic systems, the correctness of such a sub proof net makes sure that exactly one of these conclusions is an output, and thus the list of hypotheses in the sequent is not empty.

However, this is not very much in the spirit of proof nets, since it explicitely refers to the underlying sequent calculus.

However, as proved in [25], (*) is equivalent to saying that *any* sub-proof net (or even sub proof structure), admits at least two conclusions, which is definitely a sensible statement within proof net theory.

References

[1] Michele Abrusci, Christophe Fouqueré, and Jacqueline Vauzeilles. Tree adjoining grammar and non-commutative linear logic. In [24], pages 13–17, 1996.

[2] V. Michele Abrusci. Phase semantics and sequent calculus for pure non-commutative classical linear logic. *Journal of Symbolic Logic*, 56(4):1403–1451, December 1991.

[3] V. Michele Abrusci. Non- commutative proof nets. In [15]. 1995.

[4] Denis Bechet and Philippe de Groote. Constructing different phonological bracketings from a proof net. In [24], pages 1—6, 1996.

[5] Gianluigi Bellin and P. J. Scott. On the π-calculus and linear logic (with an introduction by S. Abramsky). Technical report, University of Edimbourgh, 1992.

[6] Noam Chomsky. Formal properties of grammars. In *Handbook of Mathematical Psychology*, volume 2, pages 323 – 418. Wiley, New-York, 1963.

[7] Noam Chomsky. *La nouvelle syntaxe*. Seuil, Paris, 1987.

[8] Noam Chomsky. *The minimalist program*. MIT Press, Cambridge, MA, 1995.

[9] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1990.

[10] Philippe de Groote and Christian Retoré. Semantic readings of proof nets. In Geert-Jan Kruijff, Glyn Morrill, and Dick Oehrle, editors, *Formal Grammar*, pages 57–70, Prague, August 1996. FoLLI.

[11] Arnaud Fleury. *La règle d'échange: logique linéaire multiplicative tréssée*. Thèse de Doctorat, spécialité Mathématiques, Université Paris 7, Novembre 1996.

[12] J.-Y. Girard, M. Okada, and A. Scedrov, editors. *Linear'96*, volume 3 of *Electronic Notes in Computer Science*. Elsevier, 1996. (available from http://pigeon.elsevier.nl/mcs/tcs/pc/menu.html).

[13] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[14] Jean-Yves Girard. Linear logic: its syntax and semantics. In [15]. 1995.

[15] Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors. *Advances in Linear Logic*. London Mathematical Society Lecture Notes. Cambridge University Press, 1995.

[16] John E. Hopcroft and Jeffrey Ullman. *Introduction to automata theory, languages and computation*. Addison Wesley, 1979.

[17] Mark E. Johnson. Resources in grammar and processing. (Talk given at the 1996 CUNY conference and forthcoming paper), 1996.

[18] G.M. Kelly and S. MacLane. Coherence in closed categories. *J. Pure and Applied Algebra*, 1(1):97–140, 1971.

[19] François Lamarche. Proof nets for intuitionistic linear logic: Essential nets. 35 page technical report available by FTP from the Imperial College archives, April 1994.

[20] François Lamarche. From proof nets to games. *In* [12], 1996.

[21] Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, pages 154–170, 1958.

[22] Misao Nagayama and Mitsuhiro Okada. A graph-theoretic characterization theorem for multiplicative fragment of non-commutative linear logic. *In* [12], 1996.

[23] Matti Pentus. Lambek grammars are context-free. In *Logic in Computer Science*. IEEE Computer Society Press, 1993.

[24] C. Retoré, editor. *Logical Aspects of Computational Linguistics*. CRIN-C.N.R.S. & INRIA-Lorraine, 1996.

[25] Christian Retoré. Des réseaux de démonstration pour la linguistique: une introduction à la logique linéaire. *Traitement Automatique des Langues*, 1996. to appear.

[26] Christian Retoré. Perfect matchings and series-parallel graphs: multiplicative proof nets as R&B-graphs. *In* [12], 1996.

[27] Dirk Roorda. *Resource logic: proof theoretical investigations*. PhD thesis, FWI, Universiteit van Amsterdam, 1991.

[28] A. Rouveret. Présentation de [7], 1987.

[29] W. Tait. Normal derivability in classical logic. In J. Barwise, editor, *Syntax and Semantics of Infinitary Languages*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1968.

[30] Dirk van Dalen. *Logic and Structure*. Universitext. Springer-Verlag, second edition, 1983.

[31] D. N. Yetter. Quantales and (non-commutative) linear logic. *Journal of Symbolic Logic*, 55:41–64, 1990.