



Normalization and sub-formula property for Lambek with product and PCMLL – Partially Commutative Multiplicative Linear Logic

Maxime Amblard, Christian Retoré

► To cite this version:

Maxime Amblard, Christian Retoré. Normalization and sub-formula property for Lambek with product and PCMLL – Partially Commutative Multiplicative Linear Logic. 2007. hal-00941206

HAL Id: hal-00941206

<https://hal.science/hal-00941206>

Preprint submitted on 3 Feb 2014

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

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

Normalization and sub-formula property for Lambek with product and PCMLL – Partially Commutative Multiplicative Linear Logic

Maxime Amblard, Christian Retoré

Loria (UMR 7503) Université de Lorraine, CNRS, INRIA Nancy Grand-Est
Labri -(UMR 5800) INRIA Bordeaux Sud-Ouest - Université de Bordeaux
amblard@loria.fr retore@labri.fr

Abstract

This paper establishes the normalisation of natural deduction or lambda calculus formulation of Intuitionistic Non Commutative Logic — which involves both commutative and non commutative connectives. This calculus first introduced by de Groote and as opposed to the classical version by Abrusci and Ruet admits a full entropy which allow order to be relaxed into any suborder. Our result also includes, as a special case, the normalisation of natural deduction the Lambek calculus with product, which is unsurprising but yet unproved. Regarding Intuitionistic Non Commutative Logic with full entropy does not have up to now a proof net syntax, and that for linguistic applications, sequent calculi which are only *more or less* equivalent to natural deduction, are not convenient because they lack the standard Curry-Howard isomorphism.

Keys words : Logic ; Intuitionistic Non Commutative Logic ; Lambek calculus ; normalisation

Contents

1	Presentation	3
1.1	Non commutative linear logics	3
1.2	Motivation for such calculi	4
2	Partially Commutative Linear Logic	6
2.1	Order and formulae	6
2.2	General definitions	7
3	A short example using PCMLL in Computational Linguistics	8
4	Normalisation du Lambek avec produit	9
4.1	Properties of L_{\odot}	9
4.2	Normalisation de L_{\odot}	13
4.3	Sub-formula property for L_{\odot}	15
5	Normalization of proofs of PCMLL	19
5.1	Property of PCMLL	19
5.2	Normalisation of PCMLL	25
5.3	Sub-formula property for PCMLL	30
6	Conclusion	33

We first define partially commutative linear logic as a natural deduction system, Lambek calculus with product being an easily identified fragment, [Ret05]. Next we sketch its use for a logical account of minimalist grammars.

Thereafter we prove normalisation for Lambek calculus with product which is a folklore result, assumed by the community but without any known proof. This enlightens the treatment of the product elimination rules to be intensively used for the complete calculus in combination with order rules.

Finally we give the proof of normalisation for the complete partially commutative calculus of de Groote in natural deduction. In addition to the difficulty of commutative conversion for product elimination rules, we are also faced with commutative conversions regarding the entropy rule acting on partial orders, [Amb07].

We conclude with a discussion of normal form which we already know to be unique for the Lambek calculus with product and to be "nearly unique" for the complete partially commutative calculus (they are all obtained by permuting commutative-product elimination among sequences of commutative-product eliminations). But this would have made the paper too lengthy.

We begin by introducing and studying the properties of mixed logic, introduced in [dG96], [dGR96] and [BdGR97]. This type of logic stems from the work of G. Gentzen, [Gen34a, Gen34b], [Gen36], depending on the direction given by Lambek, [Lam58]. For a detailed presentation of the evolution of these theories see [GLT88], [Gir97]. Then, we present a typical use of this calculus for computational linguistic. The two last sections present the normalization for Lambek with product and prove that the sub-formula property holds and the same for PCMLL.

1 Presentation

Non-commutative logics arise naturally both in the mathematical perspective and in the modelling of some real world phenomena. Mathematically non commutativity is a natural both from the truth valued semantics viewpoint (phase semantics, based on monoids which can be non commutative) and from a syntactical one (sequent calculus with sequences rather than sets of formulae, proof nets which can have well bracketed axiom links). Non commutativity also appears from real world application such as concurrency theory, like concurrent execution of Petri net, and in our favourite application, computational linguistic, and this goes back to the fifties and the apparition of the Lambek calculus. We first give a brief presentation of non commutative logics and then stress their interest for concurrency and computational linguistics.

1.1 Non commutative linear logics

Linear logic [Gir87] offered a logical view of the Lambek calculus [Lam58] and non commutative calculi. During many years, the difficulty was to integrate commutative connective and non commutative connectives. A first solution, without a term calculus, was Pomset Logic, now studied with extended sequent calculi called Calculus of Structures [Gug02].

Another kind of calculus was introduced as a sequent calculus by de Groote in [dG96], which has to be intuitionistic to work neatly. It consists in a superpo-

sition of the Lambek calculus (non commutative) and of Intuitionistic Linear Logic (commutative). For making a distinction between the two connectives it is necessary that the context includes two different commas mimicking the conjunctions, one being commutative and the other being non commutative. Hence we deal with series-parallel partial orders over multisets of formulae as sequent right hand side. Let us write (\dots, \dots) for the parallel composition and $\langle \dots; \dots \rangle$ for the non commutative: hence $\langle \{a, b\}; \{c, d\} \rangle$ stands for the finite partial order $a < c, b < c, a < d, b < d$. Of course we would like the two conjunctions to be related. Either the commutative product is stronger than the non commutative one, or the other way round. Surprisingly, the two options work as well, provided one direction is fixed once and for all. This relationship between the two products results from a structural rule modifying the order.

Now a difference should be underlined between the Abrusci-Ruet classical calculus and the intuitionistic one of de Groote's and concern precisely the order rule. The Abrusci-Ruet definitely has an intuitionistic version by limiting it to sequents with a single formula on the right and intuitionistic connectives, namely implications and conjunctions [AR99]. But there is an important difference with de Groote's calculus: what can be the order rule stigmatising the relation between the two conjunctions?

$$\frac{\Gamma \text{ ordered by } I \vdash C}{\Gamma \text{ ordered by } J \vdash C}$$

In de Groote's calculus J can be any order such that $J \subset I$ (as set of ordered pairs of formulae in Γ) while in Ruet J can only be obtained by turning some non commutative commas into commutative ones. This not equivalent to allowing as result any suborder as J . Indeed Bechet de Groote and the second author in [BdGR97] showed that four rewriting rules are needed to obtain all possible series-parallel partial suborders from some series parallel partial order. Here is a typical derivation that can be performed in de Groote's calculus and not in Ruet's.

$$\frac{(\{a, b\}; \{c, d\}) \vdash (a \otimes b) \odot (c \otimes d)}{\{(\{a, b\}; \{c, d\}) \vdash (a \otimes b) \odot (c \otimes d)\}}$$

Abrusci-Ruet calculus admit a proof net syntax, which can be restricted to the intuitionistic case. Regarding the more flexible de Groote calculus, there neither exists proof nets, nor natural deduction: it only exists a sequent calculus which has been proved to enjoy cut-elimination by semantical means in [dG96] and by a proof-theoretical method in [Ret04]. This is what we propose in this paper, with normalisation. Firstly we thus obtain a calculus which is more convenient, because of the Curry-Howard isomorphism for computational linguistics, and secondly it is possibly a first step towards proof net syntax.

1.2 Motivation for such calculi

Non commutativity in logic is rather natural in a resource consumption perspective. An hypothesis is viewed as a resource that can be used but then it is natural to think of how hypotheses are organised and accessible. As argued by Abrusci [Abr91] and others, linearity is a mandatory condition for non commutativity. Observe that the first non commutative calculus, Lambek calculus which was

invented long before linear logic, is a linear calculus, whose relation to other logical system, in particular intuitionistic has only been understood after the invention of linear logic by Girard.

Concurrency, in which the order of the computations or of the resources matters, is of course a natural application. In the framework of proofs as programs, with normalisation as the computational process; it is rather the pomset logic and the calculus of structure which are of some use, because the order applies to cuts that are the computations to be performed [Ret97, Gug02] But in the framework of proof search as computation, in the logic programming style of Miller, or in planning, the calculus studied in this paper with an order on hypotheses is of course important and process calculi can be encoded in non commutative calculi and this was the main motivation for Ruet's work. The second author also provided a description of the parallel execution of a Petri net in the calculus we are studying. It is a true concurrency approach, where $a||b$ is not reduced to $a;b \oplus b;a$ (where \oplus is the non deterministic choice). An execution according to a series parallel partial order corresponds to a proof in the partially commutative calculus that we study in this paper; in this order based approach of parallel computations any set of minimal transitions can be fired simultaneously. [Ret04]

Our main motivation for such calculi is computational linguistics and grammar formalisms, in particular the description of mildly context formalisms. They are assumed to be large enough for natural language constructs, go beyond context-free languages, but admit polynomial parsing algorithms. We are especially fond of logical description of grammar classes as introduced by Lambek because from a parse structure one is able to automatically compute the logical structure of the sentence. This especially true if the Lambek calculus or the partially commutative extensions that we are considering are given in a natural deduction format. Indeed, the syntactic categories can be turned into semantic categories on two types, individuals (e) and truth values (t), in such a way that the proof in the Lambek calculus (the syntactic analysis) can be turned into a proof intuitionistic logic, that is a lambda term describing a logical formula in Church's style.

Lambek calculus is definitely too restrictive as a syntactic formalism, in particular it only describes context free languages, and many common syntactic constructs are difficult to model. This is the reason to use partially commutative calculi. In particular Lecomte and the second author managed to give a logical presentation [LR01] of Stabler's minimalist grammars [Sta97] in this the de Groote calculus, presented in natural deduction to obtain semantic representation of the parsed sentences. In parsing as deduction paradigm and for other applications as well it is quite important to have normalisation, unicity of the normal form: indeed the normal form is the structure of the analysed sentence, and normalisation ensures the coherence of the calculus. The algorithm of normalisation, easily extracted from the proof is important as well: one define correct sentences as the ones such that some sequent can be proved, and both the parse structure and the semantic reading are obtained from the normal form.

2 Partially Commutative Linear Logic

2.1 Order and formulae

The sequent calculus for Partially Commutative Linear Logic (PCIMLL) was introduced by de Groote in [dG96]. It is a super imposition of between commutative intuitionistic multiplicative linear logic and the Lambek calculus with product, that is non commutative intuitionistic multiplicative linear logic. Formulae are defined from a set of propositional variables P , by the commutative conjunction (\otimes), the non commutative conjunction (\odot), the commutative implication (\multimap), the two non commutative implications ($/$ and \backslash):

$$L ::= P \mid L \odot L \mid L \otimes L \mid L / L \mid L \backslash L \mid L \multimap L$$

Left hand side are partially ordered multiset of formulae whose underlying order is series-parallel (sp), that is can be defined by two operation: disjoint union denoted by $(-, -)$ and conjoint union, denoted by $\langle -, - \rangle$ (the domain is the disjoint union of the two domain, and every formulae in the first component is before any formulae in the second component. They respect the following syntax:

$$CTX ::= L \mid \langle CTX; CTX \rangle \mid (CTX, CTX)$$

For example, the context $\langle \langle B; (A \multimap (B \backslash (D / C), A)); C \rangle$ denote the sp order, $Succ(B) = (A, A \multimap (B \backslash (D / C))$, $Succ(A) = Succ(A \multimap (B \backslash (D / C)) = C$ where $Succ(X)$ is the immediate successor of X and this function of the domain to parts of this domain determine a full finite order.

The term denoting an sp order is unique up to the commutativity of $(-, -)$ and to the associativity of both $(-, -)$ and $\langle -, - \rangle$. The term notation is only a short hand, a convenient notation for sp order. That is even if the sp term are different, the left hand side of two sequents are considered as equal whenever they are equal as partially ordered multiset.

Uppercase Greek letters are used for contexts. An expression $\Gamma[\]$ represents a context in which we distinguish a specific element $[*]$, where for an expression $\Gamma[\Delta]$ the element $[*]$ is replaced by the context Δ . More details can be found in [BdGR97] and [Ret04].

Figure 1 shows all rules of PCIMLL. It uses the classical rules of commutative multiplicative intuitionistic linear logic and of non-commutative multiplicative intuitionistic linear logic. Both bring introductions and eliminations for its implicative(s) connectives and its product connective. Moreover, we use the axiom rule and an entropy rule (\sqsubset) which correspond to order's inclusions (weakness of order).

This calculus deserve some explanation and comments, especially the later rule: the entropy:

$\Gamma' \sqsubset \Gamma$ whenever these contexts that are sp partially ordered multisets of formulae have the same multiset domain $|\Gamma| = |\Gamma'|$, and whenever considering each occurrence of a formula as distinct if $A < B$ in Γ' then $A < B$ in Γ as well. The inclusion \sqsubset of series parallel partial orders can be viewed as a rewriting relation (modulo commutativity and associativity) on the sp term denoting them as shown in [BdGR97] – see also [Ret04] where the order rule is used the other way round, but as said in the introduction, it does not change normalisation.

$$\begin{array}{c}
\frac{\Gamma \vdash A \quad \Delta \vdash A \setminus C}{< \Gamma; \Delta > \vdash C} [\setminus_e] \quad \frac{\Delta \vdash A / C \quad \Gamma \vdash A}{< \Delta; \Gamma > \vdash C} [/_e] \quad \frac{\Gamma \vdash A \quad \Delta \vdash A \multimap C}{(\Gamma, \Delta) \vdash C} [\multimap_e] \\
\\
\frac{< A; \Gamma > \vdash C}{\Gamma \vdash A \setminus C} [\setminus_i] \quad \frac{< \Gamma; A > \vdash C}{\Gamma \vdash C / A} [/_i] \quad \frac{(A, \Gamma) \vdash C}{\Gamma \vdash A \multimap C} [\multimap_i] \\
\\
\frac{\Delta \vdash A \odot B \quad \Gamma, < A; B >, \Gamma' \vdash C}{\Gamma, \Delta, \Gamma' \vdash C} [\odot_e] \quad \frac{\Delta \vdash A \otimes B \quad \Gamma, (A, B), \Gamma' \vdash C}{\Gamma, \Delta, \Gamma' \vdash C} [\otimes_e] \\
\\
\frac{\Delta \vdash A \quad \Gamma \vdash B}{< \Delta; \Gamma > \vdash A \odot B} [\odot_i] \quad \frac{\Delta \vdash A \quad \Gamma \vdash B}{(\Delta, \Gamma) \vdash A \otimes B} [\otimes_i] \\
\\
\frac{}{A \vdash A} [axiom] \quad \frac{\Gamma \vdash C}{\Gamma' \vdash C} [\text{entropy} \text{ --- whenever } \Gamma' \sqsubset \Gamma]
\end{array}$$

Figure 1: Rules of PCMLL.

In the \otimes and \odot_e rules, A and B must be equivalent:

$$\forall X \neq A, B \left\{ \begin{array}{l} X < A \Leftrightarrow X < B \\ X > A \Leftrightarrow X > B \end{array} \right.$$

In the \otimes_e case A and B are equivalent and uncomparable while in the \odot_e case, they are equivalent and $A < B$. In the conclusion of rules, they are replaced by the context which had produced $A \otimes B$ (i.e. in figure 1 Δ).

Our formulation in a lambda calculus style of the elimination of the multiplicative linear logic conjunction is due Abramsky in [Abr93] — on the term side it corresponds to *let* $x = (u, v)$ *in* $t(u, v)$ construct.

Although we do have normalisation and sub-formula property (next sections) we do not have complicated rules of the kind introduced in [Neg02] for MLL. We assume her rules are motivated by other properties as well, and work for the complete linear calculus with additive and exponentials.

2.2 General definitions

Lets δ be a proof, S_j denotes an instance of a sequent in δ , $|S_j|$ denotes for the corresponding sequent, and $|S_j|^r$ the formula in the right hand side of this sequent.

In a proof δ , $B(S_0)$ is the **principal branch** outcome of an occurrence S_0 as a sequent, $|B(S_0)|$ is the smallest path which contains S_0 and closed by the following:

1. If $S \in B(S_0)$ is obtained by an unary rule R as an occurrence S' of a sequent, then $S' \in B(S_0)$.
2. If $S \in B(S_0)$ is obtained by a product elimination \odot_e (resp. \otimes_e), then the premise which brings the marker S' , with $|S'| = \Gamma[\langle A, B \rangle] \vdash C$ (resp. $|S'| = \Gamma[\langle A, B \rangle] \vdash C$) is also in $B(S_0)$.
3. If $S \in B(S_0)$ is obtained by an implicative elimination rule \setminus_e (resp. $/_e$, \multimap_e), then the premisewhich brings the marker S' , with $|S'| = \Delta \vdash A \setminus C$ (resp. $|S'| = \Delta \vdash A / C$, $|S'| = \Delta \vdash A \multimap C$) is also in $B(S_0)$.

For every path in a principal branch $B(S)$ from S to S_i such that $|S|^r = |S_i|^r$, if $|S|$ is an elimination rule and $|S_i|$ an introduction one, they are over the same formula and they are said **conjoined**

3 A short example using PCMLL in Computational Linguistics

Before we prove the normalisation and subformula property of PCMLL, let us illustrate briefly our use of this calculus for computational linguistics — for more details see [LR01]. As said above, Lambek calculus is too restricted to describe natural language syntax, hence we try to logically describe richer formalisms, the logic being an easy way to extract semantic readings that are higher order logical formula describing the meaning of the sentence. We focused on Stabler’s minimalist grammars because they have good computational properties (polynomial parsing) are rather close to Lambek grammars and formalise Chomsky’s recent minimalist program (hence they inherit from a good coverage of lot of syntactic constructs), [Sta97, Amb07, ALR04].

As Lambek grammars our categorial minimalist grammars are lexicalised: a lexicon maps every word into a PCMLL formula which describes its interaction with other words. We do not use all the rules of PCMLL but only some of them, grouped in derived rules. There are strings of words and variables labelling every formula of every sequent in the proof. We first derive natural deduction trees from axioms $x : A \vdash x : A$ and proper axioms $\vdash w : T$ when T is the type of the word w yielding a proof of *sentence* : C .

$$\text{MERGE} \quad \frac{\frac{\frac{\vec{x} : \Delta \vdash w : A \quad \vec{x}' : \Gamma \vdash w' : A \setminus C}{\vec{x} : \Delta; \vec{x}' : \Gamma \vdash ww' : C} [\setminus_e]}{\vec{x} : \Delta, \vec{x}' \Gamma \vdash ww' : C} [entropy] \quad \frac{\frac{\vec{x}' : \Gamma \vdash w' : C / A \quad \vec{x}\vec{x} : \Delta \vdash w : A}{\vec{x}' : \Gamma; \vec{x} : \Delta \vdash w'w : C} [/_e]}{\vec{x}' : \Gamma; \vec{x} : \Delta \vdash w'w : C} [entropy]$$

The *merge* rule is completely similar to the residuation law of AB grammars of Lambek grammars.

$$\text{MOVE} \quad \frac{\vec{y} : \Gamma \vdash s : A \otimes B \quad \vec{x} : \Delta, x : A, y : B, \vec{x}' : \Delta' \vdash t : C}{\vec{x} : \Delta, \vec{y} : \Gamma, \vec{x}' : \Delta' \vdash t[s / x, \epsilon / y] : C} [\otimes_e]$$

The *move* rule is typical from Chomskyan linguistics: our construct mimics the movement of the constituent / string s from the y place to the x place.

Here is an example with a tiny lexicon, especially simple because Italian allows null subjects, exemplifying the movement of an interrogative noun phrase to a leftmost position:

$$\begin{array}{llllll} \text{que} & wh \otimes (k \otimes d) / n & \text{cosa} & n & \epsilon & k \otimes d \\ \text{fai} & k \setminus d \setminus v / d & \text{infl} & k \setminus t / v & \text{comp} & wh \setminus c / t \end{array}$$

Our interest for such an analysis is that we are able from such a structure to automatically compute the semantic reading with correct quantifier scope, which is, for our example: $\exists?(\lambda x (\wedge(cosa(x))(far(tu, x)))) \equiv \exists?x(cosa(x) \wedge far(tu, x))$

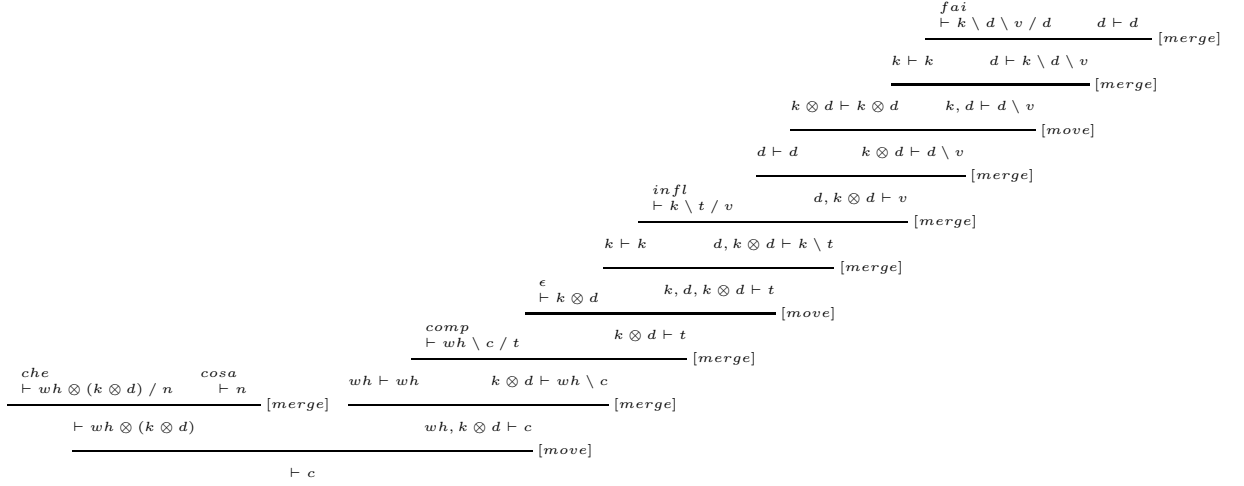


Figure 2: "che cosa e fai ?" analyze

Of course our representation of minimalist grammars and its semantic-interpretation outcome is extremely sketchy, and details can be found in [LR01, Amb06]. Indeed the objective of this paper is the normalisation of this calculus.

4 Normalisation du Lambek avec produit

4.1 Properties of L_{\odot}

Lambek calculus with product (L_{\odot}) is the restriction of PCMLL to the connectives: \backslash , $/$ and \odot . Furthermore, it uses only the order $\langle \dots; \dots \rangle$, *i.e.* it treats sequences of formulae (total order). That wy, the order is not mark. The entropy rule is not use here.

Property 1. *Let R a product elimination \odot_e of $\Gamma[\Delta] \vdash C$ between a proof δ_0 with $\Delta \vdash A \odot B$ as conclusion and a proof with $\Gamma[A, B] \vdash C$ as conclusion obtained by a rule R' between a proof δ_1 of a sequent $\Theta[A, B] \vdash X$ (and if R' is a binary rule, a second proof δ_2 with $\Psi \vdash U$ as conclusion). This description of the structure of theses proofs is presented in figure 3.*

Thus, we can obtain a proof for the same sequent $\Gamma[\Delta] \vdash C$ by first apply the rule R between the proof δ_0 with $\Delta \vdash A \odot B$ as conclusion and the proof δ_1 with $\Theta[A, B] \vdash X$ as conclusion, which produces the sequent $\Theta[\Delta] \vdash X$. By applying the rule R' to this new proof and the proof δ_2 , we obtain the same sequent $\Gamma[\Delta] \vdash C$.

Proof. The proof of this proposition is a case study based on the type of the rule above the product elimination. The following presents the different cases:

- Rising over \backslash_e :
- hypotheses in the left premise of \backslash_e :

$$\begin{array}{c}
\begin{array}{c}
\vdots \delta_0 \\
\Delta \vdash A \odot B
\end{array}
\quad
\frac{
\begin{array}{c}
\vdots \delta_2 \quad \vdots \delta_1 \\
\Psi \vdash U \quad \Theta[A, B] \vdash X \\
\Gamma[A, B] \vdash C
\end{array}
}{\Gamma[\Delta] \vdash C}
\begin{array}{c}
R' \\
R
\end{array}
\Rightarrow \\
\begin{array}{c}
\vdots \delta_2 \quad \vdots \delta_0 \quad \vdots \delta_1 \\
\Psi \vdash U \quad \Delta \vdash A \odot B \quad \Theta[\Delta] \vdash X \\
\Theta[\Delta] \vdash X
\end{array}
\begin{array}{c}
R \\
R'
\end{array}
\end{array}$$

Figure 3: Proof's structure allowing the rise of the product in L_\odot .

$$\frac{
\Gamma \vdash A \odot B \quad \frac{
A, B \vdash D \quad \Delta \vdash D \setminus C
}{A, B, \Delta \vdash C} [\setminus_e]
}{\Gamma, \Delta \vdash C} [\odot_e]
\Rightarrow
\frac{
\Gamma \vdash A \odot B \quad A, B \vdash D
}{\Gamma \vdash D} [\odot_e]
\quad \Delta \vdash D \setminus C
}{\Gamma, \Delta \vdash C} [\setminus_e]$$

- hypotheses in the right premise of \setminus_e :

$$\frac{
\Gamma \vdash A \odot B \quad \frac{
\Delta \vdash D \quad A, B \vdash D \setminus C
}{\Delta, A, B \vdash C} [\setminus_e]
}{\Delta, \Gamma \vdash C} [\odot_e]
\Rightarrow
\frac{
\Delta \vdash D \quad \frac{
\Gamma \vdash A \odot B \quad A, B \vdash D \setminus C
}{\Gamma \vdash D \setminus C} [\odot_e]
}{\Delta, \Gamma \vdash C} [\setminus_e]$$

- Rising over \setminus_i :

$$\frac{
\Gamma \vdash A \odot B \quad \frac{
D, \Delta, A, B, \Delta' \vdash C
}{\Delta, A, B, \Delta' \vdash D \setminus C} [\setminus_i]
}{\Delta, \Gamma, \Delta' \vdash D \setminus C} [\odot_e]
\Rightarrow
\frac{
\Gamma \vdash A \odot B \quad D, \Delta, A, B, \Delta' \vdash C
}{\frac{
D, \Delta, \Gamma, \Delta' \vdash C
}{\Delta, \Gamma, \Delta' \vdash D \setminus C} [\setminus_i]} [\odot_e]$$

- Rising over \odot_e :

- hypotheses in the left premise of $/_e$:

$$\frac{
\Gamma \vdash A \odot B \quad \frac{
\Delta \vdash C / D \quad A, B \vdash D
}{\Delta, A, B \vdash C} [/_e]
}{\Delta, \Gamma \vdash C} [\odot_e]
\Rightarrow
\frac{
\Delta \vdash C / D \quad \frac{
\Gamma \vdash A \odot B \quad A, B \vdash D
}{\Gamma \vdash D} [\odot_e]
}{\Delta, \Gamma \vdash C} [/_e]$$

- hypotheses in the right premise of $/_e$:

$$\frac{
\Gamma \vdash A \odot B \quad \frac{
A, B \vdash C / D \quad \Delta \vdash D
}{A, B, \Delta \vdash C} [/_e]
}{\Gamma, \Delta \vdash C} [\odot_e]
\Rightarrow
\frac{
\Gamma \vdash A \odot B \quad A, B \vdash C / D
}{\frac{
\Gamma \vdash C / D
}{\Gamma, \Delta \vdash C}} [\odot_e]
\quad \Delta \vdash D
}{\Gamma, \Delta \vdash C} [/_e]$$

◦ Rising over $/_i$:

$$\frac{\Gamma \vdash A \odot B \quad \frac{\Delta, A, B, \Delta', D \vdash C}{\Delta, A, B, \Delta', \vdash C / D} [/_i]}{\Delta, \Gamma, \Delta' \vdash C / D} [\odot_e] \Rightarrow \frac{\Gamma \vdash A \odot B \quad \Delta, A, B, \Delta', D \vdash C}{\frac{\Delta, \Gamma, \Delta', D \vdash C}{\Delta, \Gamma, \Delta' \vdash C / D} [/_i]} [\odot_e]$$

◦ Rising over \odot_e :

- hypotheses in the left premise of the first \odot_e :

$$\begin{aligned} & \frac{\Gamma \vdash A \odot B \quad \frac{\Delta, A, B, \Delta' \vdash C \odot D \quad \Phi, C, D, \Phi' \vdash E}{\Phi, \Delta, A, B, \Delta', \Phi' \vdash E} [\odot_e]}{\Phi, \Delta, \Gamma, \Delta', \Phi' \vdash E} [\odot_e] \\ & \Rightarrow \frac{\frac{\Gamma \vdash A \odot B \quad \Delta, A, B, \Delta' \vdash C \odot D}{\Delta, \Gamma, \Delta' \vdash C \odot D} [\odot_e] \quad \Phi, C, D, \Phi' \vdash E}{\Phi, \Delta, \Gamma, \Delta, \Phi' \vdash E} [\odot_e] \end{aligned}$$

- hypotheses in the right premise of the first \odot_e :

$$\begin{aligned} & \frac{\Gamma \vdash A \odot B \quad \frac{\Delta \vdash C \odot D \quad \Phi, A, B, C, D, \Phi' \vdash E}{\Phi, A, B, \Delta, \Phi' \vdash E} [\odot_e]}{\Phi, \Gamma, \Delta, \Phi' \vdash E} [\odot_e] \\ & \Rightarrow \frac{\Delta \vdash C \odot D \quad \frac{\Gamma \vdash A \odot B \quad \Phi, A, B, C, D, \Phi' \vdash E}{\Phi, \Gamma, C, D, \Phi' \vdash E} [\odot_e]}{\Phi, \Gamma, \Delta, \Phi' \vdash E} [\odot_e] \end{aligned}$$

◦ Rising over \odot_i :

- hypotheses in the left premise of the first \odot_i :

$$\frac{\Gamma \vdash A \odot B \quad \frac{\Delta, A, B, \Delta' \vdash C \quad \Phi \vdash D}{\Delta, A, B, \Delta', \Phi \vdash C \odot D} [\odot_i]}{\Delta, \Gamma, \Delta', \Phi \vdash C \odot D} [\odot_e] \Rightarrow \frac{\Gamma \vdash A \odot B \quad \Delta, A, B, \Delta' \vdash C}{\Delta, \Gamma, \Delta' \vdash C} [\odot_e] \quad \Phi \vdash D}{\Delta, \Gamma, \Delta', \Phi \vdash C \odot D} [\odot_i]$$

- hypotheses in the right premise of the first \odot_i :

$$\frac{\Gamma \vdash A \odot B \quad \frac{\Delta \vdash C \quad \Phi, A, B, \Phi' \vdash D}{\Delta, \Phi, A, B, \Phi' \vdash C \odot D} [\odot_i]}{\Delta, \Phi, \Gamma, \Phi' \vdash C \odot D} [\odot_e] \Rightarrow \frac{\Delta \vdash C \quad \frac{\Gamma \vdash A \odot B \quad \Phi, A, B, \Phi' \vdash D}{\Phi, \Gamma, \Phi' \vdash D} [\odot_e]}{\Delta, \Phi, \Gamma, \Phi' \vdash C \odot D} [\odot_i]$$

All possible cases of combinations of rules have been examined. The product elimination has the ability to rise above any rule if the hypothesis used are in the same premise. \square

Definition 1. A \odot_e rule over the sequent $A \odot B$ is said **as high as possible** if in the premise of the rule above, the two hypothesis A and B are not in the same formula.

We say that a rule is **at the bottom** of the proof if it is the rule that provides the sequent conclusion of the proof.

Let us call a **redex** in a proof the immediate succession of an introduction rule and its conjoined elimination.

In this calculus, there are four *redexes*: $/$ one, \backslash one and \odot two, depending in which premise the \odot_i takes place. We give the proof schemes

- Redex $_/$: introduction $/_i$ and immediate elimination of $/_e$.

$$\frac{\frac{\frac{D}{\vdots} C}{C/D} [/_i] \quad \frac{\frac{\vdots}{D} \delta_1}{D} [\backslash_e]}{C} [/_e] \Rightarrow \frac{\frac{\vdots}{D} \delta_1}{D} [/_e] \quad \frac{\vdots}{C}$$

- Redex $_{\backslash}$: introduction \backslash_i and immediate elimination of \backslash_e .

$$\frac{\frac{\frac{\vdots}{D} \delta_1}{D} \quad \frac{\frac{\frac{D}{\vdots} C}{D \backslash C} [\backslash_i]}{D \backslash C} [\backslash_e]}{C} [/_e] \Rightarrow \frac{\frac{\vdots}{D} \delta_1}{D} [/_e] \quad \frac{\vdots}{C}$$

- Redex $_{\odot}$: introduction \odot_i and immediate elimination of \odot_e on the left. gauche.

$$\frac{\frac{\frac{\frac{\vdots}{A} \delta_1}{A} \quad \frac{\frac{\vdots}{B} \delta_2}{B}}{A \odot B} [\odot_i] \quad \frac{\frac{\frac{A}{\vdots} B}{D} [\odot_e]}{D} [\odot_e]}{D} [\odot_e] \Rightarrow \frac{\frac{\vdots}{A} \delta_1}{A} \quad \frac{\frac{\vdots}{B} \delta_1}{B}$$

- Redex $_{\odot}$: introduction \odot_i and immediate elimination of \odot_e on the right.

$$\frac{\frac{\frac{\frac{\vdots}{A \odot B} \delta_1}{A \odot B} \quad \frac{\frac{A}{\vdots} B}{A \odot B} [\odot_i]}{A \odot B} [\odot_e]}{A \odot B} [\odot_e] \Rightarrow \frac{\frac{\vdots}{A \odot B} \delta_1}{A \odot B}$$

From the notion of redex, we define the **k-extended-redex** on a proof.

Every path of a principal branch $B(S_0)$ of length k from S_0 to S_n with $|S_0|^r = |S_n|^r$, such that $|S_0|$ marks an elimination rule R_e and S_n is the conclusion of an introduction rule R_i , is called a **k-extended-redex**. Note that 0-extended-redexes are redexes of PCMLL, already presented.

Proposition 1. *A k-extended-redex only contains \odot_e rules or a k'-extended-redex, with $k' < k$.*

Proof. For an instance of X which become a X/U , X will necessary be produced by an elimination of U (for the “same” X). In this case, the k -extended-redex contains a smaller k' -extended-redex. Only the \odot_e rule allows to keep in the conclusion one of premises, it can be used an unspecified number of times without changing the concept of the same X . \square

4.2 Normalisation de L_\odot

A **normal proof** is a proof which contains no k -extended-redexes and where all \odot_e rules are as higher as possible.

Lets δ a proof, we define $PE(\delta)$ as the set of occurrences of \odot_e rules in δ . For $R \in PE(\delta)$, we define the two following integers:

1. the integer $g(R)$ is the number of rules if there is a k -extended-redex in $B(S_0)$ and 0 otherwise with as conclusion premise S_0 ;
2. the integer $d_{conj}(R)$ is the number of rules do not belong to $PE(\delta)$ between R and the rule which relies hypothesis A and B erased by R .

We define $h(\delta)$ as $\sum_{R \in PE(\delta)} d_{conj}(R)$ and $g(\delta)$ as $\min_{R \in PE(\delta)} (g(R))$ equal to 0 is and only if δ no longer possesses k -extended-redex product. We note $n(\delta)$ the number of rules of δ .

In order to prove the normalization of L_\odot , we define the following measure over the proof δ be be a triplet of integers, which respect the lexicographic order:

$$|\delta| = \langle n(\delta), h(\delta), g(\delta) \rangle$$

Property 2. *A proof δ is normal if and only if it does not contain 0-extended-redex and if $h(\delta) = 0$ and $g(\delta) = 0$.*

Proof. Let δ a proof of L_\odot , belongs to $|\delta|$:

- the first integer is the classical one for Lambek calculus normalisation. It is minimal if the proof does not contain 0-extended-redex.
- the second integer give the process to rise \odot_e to their higher position. In this phase, k -extended-redexes $/$ and \backslash appears and can be cancelled. If every \odot_e have their higher position in the proof then $h(\delta) = 0$. Only k -extended-redexes \odot remains in δ . This case is presented in example 1 of Figure 4.
- the third integer represents the number of rules in a k -extended-redex \odot . When it is null, there is no more k -extended-redex \odot in δ . This case is presented in example 2 of Figure 4.

example 1	example 2
$\frac{\frac{\frac{\vdash C}{\vdash E \odot F} \quad \frac{C, E, F \vdash A / B}{E, F \vdash C \setminus (A / B)} [\setminus_i]}{\vdash C \setminus (A / B)} [\odot_e]}{\vdash A / B} [\setminus_e]$	$\frac{\frac{\frac{\vdash C}{\vdash E \odot F} \quad \frac{E \vdash (C \setminus A) / B \quad F \vdash B}{E, F \vdash C \setminus A} [\setminus_e]}{\vdash A / B} [\odot_e]$
$g(\setminus_e) = 1 \Rightarrow$ $k\text{-extended-redex}$	$h(\odot_e) = 1 \Rightarrow$ \odot_e does not have its position

Figure 4: Examples of proof which are not in normal form

□

Theoreme 1. *Every proof δ in L_{\odot} calculus has a unique normal form.*

Proof. We proceed by induction on $|\delta|$. By induction hypotheses, every proof δ' of size $|\delta'| < \langle r, d, g \rangle$ has a unique normal form. Given a proof δ of size $= \langle r, d, g \rangle$, let us show that δ has a unique normal form as well.

- if δ has a redex, we could reduce it. The induced proof δ' , $n(\delta') < n(\delta)$, hence $|\delta'| < \langle r, d, g \rangle$ and by induction, δ' has a unique normal form, then δ as well.

- Else:

If $d \neq 0$: let R the lowest rule \odot_e such that $\neq 0$. Hence, there exists a rule $R' \neq \odot_e$ higher than R , and R can move upwards over all \odot_e and over R' , next R can rise over R' . the induced proof δ' is such that $n(\delta') = n(\delta)$ and $h(\delta') = h(\delta) - 1$. Values of $d_{conj}(R_i)$, for $R_i \odot_e$ rules below R , stands null because R do not contribute to $d_{conj}(-)$. Therefore $\delta' < \langle r, d, g \rangle$, and by induction, δ' has a unique normal form, then δ as well.

Else:

If $g \neq 0$: let R' such that $g(R') = g$. This rule can move upwards above its left premise. The number of rules and the sum stand the same. In its left part, we change a \odot_e by a \odot_e and g decrease of 1. Hence the proof δ' is such that $|\delta'| < |\delta|$. By induction, δ' has a unique normal form, then δ as well.

Else: using the property 2, the proof is in normal form.

\odot_e can appear only after any rule with two premises which links its two hypothesis A and B . Moreover, only one \odot_e can be "as high as possible" after this rule use. Indeed, a rule can connect only two hypotheses: the rightmost one of the left premise and the leftmost one of the right premise.. We give a unique position to each \odot_e thus the normal form is unique.

□

All proofs have a unique normal form which could be performed using the strategy inside the proof above. Normal forms of the two previous examples, figure 4 are the two following proofs:

<p>example 1</p> $ \begin{array}{c} \frac{\frac{\frac{\vdash E \odot F}{\vdash C \setminus (A / B)} [\odot_e] \quad \frac{\frac{\vdash C \quad \frac{C, E, F \vdash A / B}{E, F \vdash C \setminus (A / B)} [\setminus_i]}{\vdash A / B} [\setminus_e]}{\vdash C \setminus (A / B)} [\odot_e]}{g(\setminus_e) = 0} \end{array} $	<p>example 2</p> $ \begin{array}{c} \frac{\frac{\frac{\vdash C \quad \frac{\vdash E \odot F}{\vdash A / B}}{E, F \vdash A} [/_e] \quad \frac{\frac{\frac{E \vdash (C \setminus A) / B \quad F \vdash B}{E, F \vdash C \setminus A} [/_e]}{\vdash C \setminus (A / B)} [\odot_e]}{h(\odot_e) = 0} \end{array} $
--	--

4.3 Sub-formula property for L_\odot

Theoreme 2. *All proof of Lambek calculus with product on normal form δ of a sequent $\Gamma \vdash C$ holds the sub-formula property: every formulae in a normal proof are sub-formula of some hypotheses Γ or the conclusion of the proof C .*

Proof. We proceed by induction :

We use a stronger definition of the sub-formula property: every formulae in a normal sub-proof are formulae of some hypotheses or the conclusion of the proof and if the last rule used is an \setminus_e or $/_e$ every sub-formulae are sub-formulae of some hypotheses only. Note that the axiom rule clearly satisfies the property of the sub-formula because the sequent is then one of the hypotheses.

We check the induction hypothesis after the use of each rule.

1. \setminus_e : let the proof δ , where Γ_i is the set of hypotheses used in the sub-proof δ_i , for $i \in [2]$:

$$\frac{\frac{\frac{\Gamma_1 \quad \vdots \quad \delta_1}{C}}{\vdots} \quad \frac{\frac{\Gamma_2 \quad \vdots \quad \delta_2}{C \setminus D} [R]}{D} [\setminus_e]$$

Using the induction hypothesis:

- In δ_1 every formulae are sub-formulae of C or Γ_1 ;
- In δ_2 every formulae are sub-formulae of $C \setminus D$ or Γ_2 .

The conclusion D and the premise C are direct sub-formulae of the premise $C \setminus D$. We have to check the rule $[R]$ above this premise:

- if R is $/_e$ or \setminus_e : we use the induction hypothesis, we conclude that $C \setminus D$ is a sub-formula of Γ_2 . Then every formula of δ is a sub-formula of Γ_2 .
- if R is \setminus_i : it is impossible because the rule should be a 0-extended-redex, or δ is in normal form.

- if R is $/_i$: this case is structurally impossible because we could not derive $C \setminus D$ with this rule.
- if R is \odot_i : this case is also impossible because we could not derive $C \setminus D$ with this rule.
- if R is \odot_e . Once again, we must check the property with the rule R' above:

$$\begin{array}{c}
\Gamma_1 \\
\vdots \delta_1 \\
C \\
\hline
D
\end{array}
\quad
\frac{
\begin{array}{c}
\Gamma_2[A, B] \\
\vdots \delta_2 \\
\hline
A \odot B \quad \frac{}{C \setminus D} [R'] \\
\hline
C \setminus D \quad [\odot_e]
\end{array}
}{C \setminus D} [\odot_e]
\quad
\frac{}{D} [\setminus_e]$$

If R' is \setminus_e or $/_e$, by the induction hypothesis, $C \setminus D$ is a sub-formula of some hypotheses.

If R' is \setminus_i : impossible because it should introduce a 1-extended-redex, or δ is in normal form... then it is not possible.

If R' is one of the other introduction rules (\setminus_i or \odot_i): these case are structurally impossible. We could not derive $C \setminus D$ with this rule.

If R' is \odot_e , once again, we check the property on the rule above. Remark that the number of rules above this rule is finite and they constitute a sequence such that:

$$\begin{array}{c}
\Gamma_1 \\
\vdots \delta_1 \\
C \\
\hline
D
\end{array}
\quad
\frac{
\begin{array}{c}
\Gamma_2[A_1, \dots, A_n, B_1, \dots, B_n] \\
\vdots \delta_2 \\
\hline
A_n \odot B_n \quad \frac{}{C \setminus D} [R] \\
\hline
C \setminus D \quad [\odot_e]
\end{array}
}{C \setminus D} [\odot_e]
\quad
\frac{}{D} [\setminus_e]$$

In this case, we have:

- either there is only \odot_e in this sequence then $C \setminus D$ is one of the hypotheses.
- either it exist a rules R'' different than \odot_e inside then using the argument of the corresponding case above we prove that $C \setminus D$ is sub-formula oh the hypotheses.

In every case, the conclusion of \setminus_e is a sub-formula of the hypotheses. The induction hypothesis is checked.

2. $/_e$: let the proof δ , where Γ_i is the set of hypotheses used in the sub-proof δ_i , for $i \in [2]$:

$$\frac{\frac{\Gamma_2 \vdots \delta_2}{D/C} [R] \quad \Gamma_1 \vdots \delta_1}{C} [\backslash_e] \quad D$$

The conclusion D and the premise C are direct sub-formulae of the premise D/C . We have to check the rule $[R]$ above this premise whose conclusion is D/C : this case is the same as \backslash_e . In the same way, we prove that D/C is sub-formula of Γ_2 .

3. \backslash_i : let the proof δ , where Γ_1 is the set of hypotheses used in the sub-proof δ_1 :

$$\frac{C, \Gamma_1 \vdots \delta_1 \quad D}{C \setminus D} [\backslash_i]$$

In δ_1 every formula is a sub-formula of D or of C and Γ_1 . Furthermore, D is a sub-formula of $C \setminus D$. Then, every formula of δ is sub-formula of C, Γ_1 or $C \setminus D$.

4. $/_i$: let the proof δ , where Γ_1 is the set of hypotheses used in the sub-proof δ_1 :

$$\frac{C, \Gamma_1 \vdots \delta_1 \quad D}{D/C} [/_i]$$

This case is strictly symmetrical to that of \backslash_i : D is a sub-formula of D/C , and every formula of δ_1 is sub-formula of C, Γ_1 or D . By transitivity, any formula of δ is sub-formula of C, Γ_1 or D/C .

5. \odot_i : let the proof δ , where Γ_i is the set of hypotheses used in the sub-proof δ_i , for $i \in [2]$:

$$\frac{\Gamma_1 \vdots \delta_1 \quad C \quad \Gamma_2 \vdots \delta_2 \quad D}{C \odot D} [\odot_i]$$

- In δ_1 every formula is a sub-formula of C or of Γ_1 .
- In δ_2 every formula is a sub-formula of D or of Γ_2 .

Futhermore , C and D are sub-formulae of $C \odot D$. By transitivity, every formula of δ is sub-formula of Γ_1, Γ_2 or of $C \odot D$.

6. \odot_e : let the proof δ , where Γ_i is the set of hypotheses used in the sub-proof δ_i , for $i \in [2]$:

$$\frac{\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ A \odot B \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ D \end{array}}{D} [\odot_e]$$

- In δ_1 , every formula is a sub-formula of $A \odot B$ or of Γ_1 .
- In δ_2 , every formula is a sub-formula of D or of Γ_2 .

The conclusion of δ is the conclusion of one premise, then the property stands for the part of the proof which the conclusion belongs, *i.e.* δ_2 . We check the property for the second part of the proof. here, we have to check that $A \odot B$ is sub-formula of hypothesis of δ_1 . For this, we analyze the rule R above:

- if R is \backslash_e ou $/_e$, by induction and th type of the rule, $A \odot B$ is sub-formula of de Γ_1
- if R is \backslash_i ou $/_i$: this case is structurally impossible because $A \odot B$ could not be produced after these lasts.
- if R is \odot_i : this case is absurd because the succession of these two rules imply a 0-extended-redex, but the proof is in normal form.
- if R is \odot_e . We check the property for the rule R above:

$$\frac{\begin{array}{c} \Gamma_1[E, F] \\ \vdots \\ \delta_1 \\ \vdots \\ E \odot F \end{array} \quad \frac{\begin{array}{c} \Gamma_2[A, B] \\ \vdots \\ \delta_2 \\ \vdots \\ A \odot B \end{array} \quad \frac{\begin{array}{c} E \odot F \\ \vdots \\ A \odot B \end{array} \quad \frac{\begin{array}{c} A \odot B \\ \vdots \\ D \end{array}}{D} [\odot_e]}{D} [\odot_e]$$

- If $A \odot B$ is not an hypothesis: in this case, a conjoined rule \odot_i exists for the analyzed rule. This implies that a k -extended-redex is in the proof, but we assume that it is in normal form
- If $A \odot B$ is an hypothesis of Γ_1 , then $A \odot B$ is a sub-formula of hypothesis.

In any possible case, $A \odot B$ is sub-formula of hypothesis. Thereby, the sub-formula property stands for \odot_e .

□

In L_{\odot} , every proof have a unique normal form which check the sub-formula property. We observe that unlike [Neg02], rules use are the usual one for this calculus.

5 Normalization of proofs of PCMLL

Now, we present a normalization for proofs of PCMLL, from which we prove that the sub-formula property holds. In the same way for L_{\odot} , the normalization uniquely positioned eliminations of non-commutative product, and build sequence of commutative product eliminations. The relative position of a commutative product removal in a sequence is not unique

5.1 Property of PCMLL

Property 3 (product eliminations could rise in the proof). *Let R a product elimination \otimes_e (resp. a rule \odot_e) of $\Gamma[\Delta] \vdash C$ between a proof δ_0 $\Delta \vdash A \otimes B$ and a proof with conclusion $\Gamma[(A, B)] \vdash C$ (resp. $\Gamma[\langle A; B \rangle] \vdash C$) obtained by a rule R' of a proof δ_1 of a sequent $\Theta[(A, B)] \vdash X$ (resp. $\Theta[\langle A; B \rangle] \vdash X$) (and if R' is a binary rule of a second proof δ_2 with $\Psi \vdash U$ as conclusion).*

Then, we can obtain a proof for the same sequent $\Gamma[\Delta] \vdash C$ by first applying the rule \otimes_e (resp. the rule \odot_e) between the proof δ_0 with $\Delta \vdash A \otimes B$ as conclusion and the proof δ_1 with $\Theta[(A, B)] \vdash X$ as conclusion (resp. $\Theta[\langle A; B \rangle] \vdash X$) giving the sequent $\Theta[\Delta] \vdash X$. By applying the rule R' to this new proof and potentially the proof δ_2 , we get the same sequent $\Gamma[\Delta] \vdash C$.

This derivation is presented in figure 5

$$\begin{array}{c}
 \begin{array}{c} \vdots \delta_0 \\ \Delta \vdash A \otimes B \end{array} \quad \frac{\begin{array}{c} \vdots \delta_2 \\ \Psi \vdash U \end{array} \quad \frac{\begin{array}{c} \vdots \delta_1 \\ \Theta[(A, B)] \vdash X \end{array}}{R'} \quad \frac{\Gamma[(A, B)] \vdash C}{R} \\
 \hline
 \Gamma[\Delta] \vdash C \quad \Rightarrow
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c} \vdots \delta_2 \\ \Psi \vdash U \end{array} \quad \frac{\begin{array}{c} \vdots \delta_0 \\ \Delta \vdash A \otimes B \end{array} \quad \frac{\begin{array}{c} \vdots \delta_1 \\ \Theta[(A, B)] \vdash X \end{array}}{R}}{\Theta[\Delta] \vdash X} \quad \frac{\Theta[\Delta] \vdash X}{R'} \\
 \hline
 \Gamma[\Delta] \vdash C
 \end{array}$$

Figure 5: Proof's structure allowing the rise of the product in PCMLL.

Proof. The proof is similar to property 1. This is a case study according to rule over the product elimination. This elimination could only rise when hypothesis which must be cancelled are in the same premise and hold their respective position according order required by the elimination.

Let check every cases for \otimes_e .

◦ Rise over \setminus_e .

- if hypothesis are in the left premise of \setminus_e :

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Gamma[(A, B)] \vdash D \quad \Phi \vdash D \setminus C}{< \Gamma[(A, B)]; \Phi > \vdash C} [\setminus_e]}{< \Gamma[\Delta]; \Phi > \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\frac{\Delta \vdash A \otimes B \quad \Gamma[(A, B)] \vdash D}{\Gamma[\Delta] \vdash D} [\otimes_e] \quad \Phi \vdash D \setminus C}{< \Gamma[\Delta]; \Phi > \vdash C} [\setminus_e]$$

- if hypothesis are in the right premise of \setminus_e :

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Gamma \vdash D \quad \Phi[(A, B)] \vdash D \setminus C}{< \Gamma; \Phi[(A, B)] > \vdash C} [\setminus_e]}{< \Gamma; \Phi[\Delta] > \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\Gamma \vdash D \quad \frac{\Delta \vdash A \otimes B \quad \Phi[(A, B)] \vdash D \setminus C}{\Phi[\Delta] \vdash D \setminus C} [\otimes_e]}{< \Gamma; \Phi[\Delta] > \vdash C} [\setminus_e]$$

◦ Rise over $/_e$.

- if hypothesis are in the right premise of $/_e$:

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Phi \vdash C / D \quad \Gamma[(A, B)] \vdash D}{< \Phi; \Gamma[(A, B)] > \vdash C} [/_e]}{< \Phi; \Gamma[\Delta] > \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\Phi \vdash C / D \quad \frac{\Delta \vdash A \otimes B \quad \Gamma[(A, B)] \vdash D}{\Gamma[\Delta] \vdash D} [\otimes_e]}{< \Phi; \Gamma[\Delta] > \vdash C} [/_e]$$

- if hypothesis are in the left premise of $/_e$:

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Phi[(A, B)] \vdash C / D \quad \Gamma \vdash D}{< \Phi[(A, B)]; \Gamma > \vdash C} [/_e]}{< \Phi[\Delta]; \Gamma > \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\frac{\Delta \vdash A \otimes B \quad \Phi[(A, B)] \vdash C / D}{\Phi[\Delta] \vdash C / D} [\otimes_e] \quad \Gamma \vdash D}{< \Phi[\Delta]; \Gamma > \vdash C} [/_e]$$

◦ Rise over $\neg\circ_e$

- if hypothesis are in the left premise of $\neg\circ_e$:

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Gamma[(A, B)] \vdash D \quad \Phi \vdash D \neg\circ C}{(\Gamma[(A, B)], \Phi) \vdash C} [\neg\circ_e]}{(\Gamma[\Delta]; \Phi) \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\frac{\Delta \vdash A \otimes B \quad \Gamma[(A, B)] \vdash D}{\Gamma[\Delta] \vdash D} [\otimes_e] \quad \Phi \vdash D \multimap C}{(\Gamma[\Delta], \Phi) \vdash C} [-\circ_e]$$

- if hypothesis are in the right premise of \multimap_e :

$$\frac{\Delta \vdash A \otimes B \quad \frac{\Gamma \vdash D \quad \Phi[(A, B)] \vdash D \multimap C}{(\Gamma, \Phi[(A, B)]) \vdash C} [-\circ_e]}{(\Gamma, \Phi[\Delta]) \vdash C} [\otimes_e]$$

$$\Rightarrow \frac{\Gamma \vdash D \quad \frac{\Delta \vdash A \otimes B \quad \Phi[(A, B)] \vdash D \multimap C}{\Phi[\Delta] \vdash D \multimap C} [\otimes_e]}{(\Gamma, \Phi[\Delta]) \vdash C} [-\circ_e]$$

- Rise over $/_i$:

$$\frac{\Delta \vdash A \otimes B \quad \frac{< \Gamma[(A, B)]; D > \vdash C}{\Gamma[(A, B)] \vdash C / D} [/_i]}{\Gamma[\Delta] \vdash C / D} [\otimes_e] \Rightarrow \frac{\Delta \vdash A \otimes B \quad < \Gamma[(A, B)]; D > \vdash C}{< \Gamma[\Delta]; D > \vdash C} [\otimes_e]}{\Gamma[\Delta] \vdash C / D} [/_i]$$

- Rise over \setminus_i :

$$\frac{\Delta \vdash A \otimes B \quad \frac{< D; \Gamma[(A, B)] > \vdash C}{\Gamma[(A, B)] \vdash D \setminus C} [\setminus_i]}{\Gamma[\Delta] \vdash D \setminus C} [\otimes_e] \Rightarrow \frac{\Delta \vdash A \otimes B \quad < D; \Gamma[(A, B)] > \vdash C}{< D; \Gamma[\Delta] > \vdash C} [\setminus_i]}{\Gamma[\Delta] \vdash D \setminus C} [\setminus_i]$$

- Rise over \multimap_i :

$$\frac{\Delta \vdash A \otimes B \quad \frac{(\Gamma[(A, B)], D) \vdash C}{\Gamma[(A, B)] \vdash D \multimap C} [-\circ_i]}{\Gamma[\Delta] \vdash D \multimap C} [\otimes_e] \Rightarrow \frac{\Delta \vdash A \otimes B \quad (\Gamma[(A, B)], D) \vdash C}{(\Gamma[\Delta], D) \vdash C} [\otimes_e]}{\Gamma[\Delta] \vdash D \multimap C} [-\circ_i]$$

- Rise over \otimes_e :

- if hypothesis are in the right premise of \otimes_e :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{\Delta \vdash C \otimes D \quad (\Phi, (A, B), (C, D), \Phi') \vdash E}{(\Phi, (A, B), \Delta, \Phi') \vdash E} [\otimes_e]}{(\Phi, \Gamma, \Delta, \Phi') \vdash E} [\otimes_e]$$

$$\Rightarrow \Delta \vdash C \otimes D \frac{\frac{\Gamma \vdash A \otimes B \quad (\Phi, (A, B), (C, D), \Phi') \vdash E}{(\Phi, \Gamma, (C, D), \Phi') \vdash E} [\otimes_e]}{(\Phi, \Gamma, \Delta, \Phi') \vdash E} [\otimes_e]$$

- if hypothesis are in the left premise of \otimes_e :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{(\Delta, (A, B), \Delta') \vdash C \otimes D \quad (\Phi, (C, D), \Phi') \vdash E}{(\Phi, \Delta, (A, B), \Delta', \Phi') \vdash E} [\otimes_e]}{(\Phi, \Delta, \Gamma, \Delta', \Phi') \vdash E} [\otimes_e]$$

$$\Rightarrow \frac{\frac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C \otimes D}{(\Delta, \Gamma, \Delta') \vdash C \otimes D} [\otimes_e] \quad (\Phi, (C, D), \Phi') \vdash E}{(\Phi, \Delta, \Gamma, \Delta, \Phi') \vdash E} [\otimes_e]$$

- Rise over \otimes_i :

- if hypothesis are in the left premise of \otimes_i :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{(\Delta, (A, B), \Delta') \vdash C \quad \Phi \vdash D}{(\Delta, (A, B), \Delta', \Phi) \vdash C \otimes D} [\otimes_i]}{(\Delta, \Gamma, \Delta', \Phi) \vdash C \otimes D} [\otimes_e]$$

$$\Rightarrow \frac{\frac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C}{(\Delta, \Gamma, \Delta') \vdash C} [\otimes_e] \quad \Phi \vdash D}{(\Delta, \Gamma, \Delta', \Phi) \vdash C \otimes D} [\otimes_i]$$

- if hypothesis are in the right premise of \otimes_i :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{\Delta \vdash C \quad (\Phi, (A, B), \Phi') \vdash D}{(\Delta, \Phi, (A, B), \Phi') \vdash C \otimes D} [\otimes_i]}{(\Delta, \Phi, \Gamma, \Phi') \vdash C \otimes D} [\otimes_e]$$

$$\Rightarrow \frac{\Delta \vdash C \quad \frac{\Gamma \vdash A \otimes B \quad (\Phi, (A, B), \Phi') \vdash D}{(\Phi, \Gamma, \Phi') \vdash D} [\otimes_e]}{(\Delta, \Phi, \Gamma, \Phi') \vdash C \otimes D} [\otimes_i]$$

- Rise over \odot_e :

- if hypothesis are in the right premise of \odot_e :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{\Delta \vdash C \odot D \quad (\Phi, (A, B), \Psi, < C; D >, \Psi', \Phi') \vdash E}{(\Phi, (A, B), \Psi, \Delta, \Psi', \Phi') \vdash E} [\odot_e]}{(\Phi, \Gamma, \Psi, \Delta, \Psi', \Phi') \vdash E} [\odot_e]$$

$$\Rightarrow \Delta \vdash C \odot D \quad \frac{\Gamma \vdash A \otimes B \quad (\Phi, (A, B), \Psi, < C; D >, \Psi', \Phi') \vdash E}{(\Phi, \Gamma, \Psi, < C; D >, \Psi', \Phi') \vdash E} [\otimes_e] \\ \frac{}{(\Phi, \Gamma, \Psi, \Delta, \Psi' \Phi') \vdash E} [\odot_e]$$

- if hypothesis are in the left premise of \odot_e :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{(\Delta, (A, B), \Delta') \vdash C \odot D \quad (\Phi, \Psi, < C; D >, \Psi', \Phi') \vdash E}{(\Phi, \Psi, \Delta, (A, B), \Delta', \Psi', \Phi') \vdash E} [\odot_e]}{(\Phi, \Psi, \Delta, \Gamma, \Delta', \Psi', \Phi') \vdash E} [\otimes_e] \\ \Rightarrow \frac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C \odot D}{(\Delta, \Gamma, \Delta') \vdash C \odot D} [\otimes_e] \quad (\Phi, \Psi, < C; D >, \Psi', \Phi') \vdash E \\ \frac{}{(\Phi, \Psi, \Delta, \Gamma, \Delta', \Psi', \Phi') \vdash E} [\odot_e]$$

- Rise over \odot_i :

- if hypothesis are in the left premise of \odot_i :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{(\Delta, (A, B), \Delta') \vdash C \quad \Phi \vdash D}{< (\Delta, (A, B), \Delta'); \Phi > \vdash C \odot D} [\odot_i]}{< (\Delta, \Gamma, \Delta'); \Phi > \vdash C \odot D} [\otimes_e] \\ \Rightarrow \frac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C}{(\Delta, \Gamma, \Delta') \vdash C} [\otimes_e] \quad \Phi \vdash D \\ \frac{}{< (\Delta, \Gamma, \Delta'); \Phi > \vdash C \odot D} [\odot_i]$$

- if hypothesis are in the right premise of \odot_i :

$$\frac{\Gamma \vdash A \otimes B \quad \frac{\Delta \vdash C \quad (\Phi, (A, B), \Phi') \vdash D}{< \Delta; (\Phi, (A, B), \Phi') > \vdash C \odot D} [\odot_i]}{< \Delta; (\Phi, \Gamma, \Phi') > \vdash C \odot D} [\otimes_e] \\ \Rightarrow \frac{\Delta \vdash C \quad \frac{\Gamma \vdash A \otimes B \quad (\Phi, (A, B), \Phi') \vdash D}{(\Phi, \Gamma, \Phi') \vdash D} [\otimes_e]}{< \Delta; (\Phi, \Gamma, \Phi') > \vdash C \odot D} [\odot_i]$$

- Rise over \sqsubset :

$$\frac{\frac{\Gamma \vdash A \otimes B}{\Gamma' \vdash A \otimes B} [\sqsubset] \quad \Delta[A, B] \vdash D}{\Delta[\Gamma'] \vdash D} [\otimes_e] \quad \Rightarrow \quad \frac{\Gamma \vdash A \otimes B \quad \Delta[A, B] \vdash D}{\Delta[\Gamma] \vdash D} [\otimes_e] \\ \frac{}{\Delta[\Gamma'] \vdash D} [\sqsubset]$$

(the removal of hypothesis in Γ does not modify the order)

The check of the property for non-commutativity is an extension of the property 1 which is similar to the previous case. \square

The procedure is analogous to the case studied for L_{\odot} . To do this we introduce **redexes** of the calculus. Mix logic contains seven redexes: one for each implicative connective and two for each product connective, the conjoined introduction could be in the left premise or in the right premise.

The following present the seven redexes:

- Redex_/: introduction $/_i$ and direct elimination of $/_e$.

$$\frac{\frac{\frac{\vdots}{\langle \Gamma; D \rangle \vdash C} [\wedge_i]}{\Gamma \vdash C / D} \quad \frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Gamma; \Delta \rangle \vdash C} [\wedge_e]}{\langle \Gamma; \Delta \rangle \vdash C} \Rightarrow \frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Gamma; \Delta \rangle \vdash C}$$

- Redex_\: introduction \backslash_i and direct elimination of \backslash_e .

$$\frac{\frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Delta; \Gamma \rangle \vdash C} \quad \frac{\frac{\frac{\vdots}{\langle D; \Gamma \rangle \vdash C}}{\Gamma \vdash D \backslash C} [\backslash_i]}{\langle \Delta; \Gamma \rangle \vdash C} [\backslash_e]}{\langle \Delta; \Gamma \rangle \vdash C} \Rightarrow \frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Delta; \Gamma \rangle \vdash C}$$

- Redex_¬: introduction \multimap_i and direct elimination of \multimap_e .

$$\frac{\frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Delta, \Gamma \rangle \vdash C} \quad \frac{\frac{\frac{\vdots}{(D, \Gamma) \vdash C}}{\Gamma \vdash D \multimap C} [\multimap_i]}{\langle \Delta, \Gamma \rangle \vdash C} [\multimap_e]}{\langle \Delta, \Gamma \rangle \vdash C} \Rightarrow \frac{\frac{\vdots}{\Delta \vdash D} \delta_1}{\langle \Delta, \Gamma \rangle \vdash C}$$

- Redex_⊙: introduction \odot_i and direct elimination of \odot_e on the left.

$$\frac{\frac{\frac{\frac{\vdots}{\Delta_1 \vdash A} \delta_1 \quad \frac{\frac{\vdots}{\Delta_2 \vdash B} \delta_2}{\langle \Delta_1; \Delta_2 \rangle \vdash A \odot B} [\odot_i]}{\Gamma[\langle \Delta_1; \Delta_2 \rangle] \vdash D} \quad \frac{\frac{\vdots}{\Gamma[\langle A; B \rangle] \vdash D}}{\Gamma[\langle \Delta_1; \Delta_2 \rangle] \vdash D} [\odot_e]}{\Gamma[\langle \Delta_1; \Delta_2 \rangle] \vdash D} \Rightarrow \frac{\frac{\frac{\vdots}{\Delta_1 \vdash A} \delta_1 \quad \frac{\frac{\vdots}{\Delta_2 \vdash B} \delta_2}{\Gamma[\langle A; B \rangle] \vdash D}}{\Gamma[\langle \Delta_1; \Delta_2 \rangle] \vdash D}$$

- Redex_⊙: introduction \odot_i and direct elimination of \odot_e on the right.

$$\frac{\frac{\frac{\vdots \delta_1}{\Gamma \vdash A \odot B} \quad \frac{A \vdash A \quad B \vdash B}{\langle A; B \rangle \vdash A \odot B} [\odot_i]}{\Gamma \vdash A \odot B} [\odot_e] \Rightarrow \frac{\vdots \delta_1}{\Gamma \vdash A \odot B}$$

◦ Redex_⊗: introduction \otimes_i and direct elimination of \otimes_e on the left.

$$\frac{\frac{\frac{\vdots \delta_1}{A} \quad \frac{\vdots \delta_2}{B}}{A \otimes B} [\otimes_i] \quad \frac{A \quad B}{D} [\otimes_e]}{D} [\otimes_e] \Rightarrow \Gamma[(\frac{\vdots \delta_1}{A}, \frac{\vdots \delta_2}{B})] \vdash D$$

◦ Redex_⊗: introduction \otimes_i and direct elimination of \otimes_e on the right.

$$\frac{\frac{\frac{\vdots \delta_1}{\Gamma \vdash A \otimes B} \quad \frac{A \vdash A \quad B \vdash B}{(A, B) \vdash A \otimes B} [\otimes_i]}{\Gamma \vdash A \otimes B} [\otimes_e] \Rightarrow \frac{\vdots \delta_1}{\Gamma \vdash A \otimes B}$$

Again, we use the notion of k -extended-redex, that we give back to this calculus. Every path of a principal branch $B(S_0)$ of length k from S_0 to S_n with $|S_0|^r = |S_n|^r$, such that $|S_0|$ is an elimination rule R_e and S_n is the conclusion of an introduction rule R_i is called a **k -extended-redex**. Note that 0 -extended-redex are the usual redexes of PCMLL, presented above.

A proof is in **normal form** if it does not contain any k -extended-redexes, $\forall k \in \mathbb{N}$.

5.2 Normalisation of PCMLL

A proof is in **normal form** if it does not contain any k -extended-redex. In the same way for L_\odot , we define the different parts of a measure used for the normalisation.

1. For a rule R , *implicative elimination* (\backslash_e , $/_e$ or \multimap_e), with S_0 as conclusion, the integer $e(R)$ is k if there is a k -extended-redex in $B(S_0)$ called a k -extended-redex over R , and otherwise 0.

In this measure, we count every rules, including entropy rules and product eliminations. (this is exactly the number k for k -extended-redexes wich contains implicative eliminations).

2. For a rule R , *product elimination*, with S_0 as conclusoin, the integer $g(R)$ is k if there is a k -extended-redex in $B(S_0)$ called a k -extended-redex over R and otherwise 0. (this is exactly the number k for k -extended-redexes which contains product elimination).

Let δ a proof of PCMLL, we define $IER(\delta)$ (*resp.* $PER(\delta)$) as the number of occurrences of implicative elimination rules (*resp.* product) in δ .

We define $e(\delta)$ as $\min_{R \in IER(\delta)}(e(R))$ and $g(\delta)$ as $\min_{R \in PER(\delta)}(g(R))$ equal to 0 if and only if δ does not contain implicative k -extended-redex any more (*resp.* product). Moreover, we define $r(\delta)$ as the number of rule in δ . We introduce the measure of the proof δ , wrote $|\delta|$, as the triplet of integers, with the lexicographic order:

$$\langle r(\delta), e(\delta), g(\delta) \rangle$$

Property 4. *A proof with the measure $\langle n, 0, 0 \rangle$ is in normal form.*

Proof. Let δ a proof of PCMLL, in the measure of δ :

- the first integer is the number of rules which is the number of rules which is minimal if δ does not contain any k -extended-redex.
- the second integer is the distance between each part of an implicative k -extended-redex (\backslash , $/$ or \multimap). If its value is null, δ contains no more.
- the third integer is the distance between each part of a product k -extended-redex (\odot ou \otimes). If its value is null, δ contains no more.

Note that the two other redexes could only be 0-extended-redexes. The figure 6 shows two examples of proofs which are not in normal form. \square

<p>example 1</p> $ \begin{array}{c} \frac{\frac{\frac{\vdash C}{\vdash A / B} [\backslash_e] \quad \frac{\frac{\frac{(C, \langle E; F \rangle) \vdash A / B}{\langle E; F \rangle \vdash C \backslash (A / B)} [\backslash_i] \quad \vdash E \odot F}{\vdash C \backslash (A / B)} [\odot_e]}{\vdash A / B} [\backslash_e]}{e(\backslash_e) = 1 \Rightarrow} \end{array} $	<p>example 2</p> $ \begin{array}{c} \frac{\frac{\frac{\vdash A \otimes B \quad \frac{\frac{A \vdash E \quad B \vdash F}{(A, B) \vdash E \otimes F} [\otimes_i]}{\vdash E \otimes F} [\otimes_e] \quad (E, F) \vdash D}{\vdash D} [\otimes_e]}{g(\odot_e) = 1 \Rightarrow} \end{array} $
--	---

Figure 6: Examples of proofs of PCMLL which are not in normal form

Property 5. *A k -extended-redex $S_0 \cdots S_k$, with an implicative elimination rule contains a k' -extended-redex, with $k > k'$.*

Proof. Let δ a proof in normal form. It goes through a principal branch from the conclusion and we exhibit the possibilities encountered in this derivation:

One of minimal k -extended-redexes has the following structure:

$$\begin{array}{c}
\frac{}{X} [introduction] \\
\vdots \delta_3 \\
U \\
\hline U / A [/i] \\
\vdots \delta_2 \\
U / A \quad A \\
\hline U \quad A [/e] \\
\vdots \delta_1 \\
X \\
\hline [elimination]
\end{array}$$

Then, we define:

- δ_1 as a sequence of implicative elimination rules and entropy;
- δ_2 as a sequence of product elimination rules and entropy.

U is the formula produced by the higher implicative elimination rule. For this derivation, the number of symbols in U is over the number of symbols in X . Then, in the principal branch, δ_2 is a sequence of product eliminations and entropy.

The only rule above which could give the formula U / A is an introduction rule, because it is the only type which decreases the number of symbols in the formula.

On examples, the only introduction rule that we could structurally use is $/i$ on the formula A . Because this introduction is in the principal branch, it must be the conjoined rule of the previous introduction. Then, we have a new k' -extended-redex inside the k -extended-redex. k' is the number of rules in δ_2 and because δ_2 is a sub-part of the full proof, $k > k'$. \square

We could propose a consequence of this property.

Lemma 1. *In the proof δ , a k -extended-redex which minimize $e(\delta)$ nonzero only contains product elimination rules and entropy.*

Proof. If the k -extended-redex is minimal, with the property 5, there is no elimination rule. Moreover, if we do not use elimination rules, the number of symbols in the formula could not decrease. It must be constant in the k -extended-redex.

In this case, the only rules that we could use are rules with the conclusion is one of the premises. The sequence of rules could only contain product elimination and entropy rules: \odot_e, \otimes_e or \sqsubset . \square

Property 6. *Product eliminations and entropy rules could go under implicative elimination rules.*

Let R a product elimination \otimes_e (resp. a rule \odot_e) of $\Gamma[\Delta] \vdash C$ between a proof δ_0 with $\Delta \vdash A \otimes B$ as conclusion and a proof δ_1 with $\Gamma[(A, B)] \vdash C$ as conclusion (resp. $\Gamma[\langle A; B \rangle] \vdash C$). This proof is combined with a proof δ_2

with $\Theta \vdash U$ as conclusion, by an implicative elimination rule R' . Then, the conclusion is $\langle \Theta; \Gamma[\Delta] \rangle \vdash V$ if R' is \backslash_e (resp. $\langle \Gamma[\Delta]; \Theta \rangle \vdash V$ if R' is $/_e$ and $(\Gamma[\Delta], \Theta) \vdash V$ if R' is $\neg o_e$). Figure 7 presents the case where R' is \backslash_e .

Then, we can obtain a proof for the same sequent which depends on R' by applying first the rule R' between the proof δ_2 with $\Theta \vdash U$ as conclusion and the proof δ_1 with $\Gamma[(A, B)] \vdash X$ as conclusion (resp. $\Gamma[(A; B)] \vdash X$) giving the sequent $\langle \Theta; \Gamma[(A, B)] \rangle \vdash V$ (resp. $\langle \Gamma[(A; B)]; \Theta \rangle \vdash V$ and $(\Theta, \Gamma[(A, B)]) \vdash V$). Applying the rule R on this new proof, we get the same sequent $\langle \Theta; \Gamma[\Delta] \rangle \vdash V$ (resp. $\langle \Gamma[\Delta]; \Theta \rangle \vdash V$ and $(\Theta, \Gamma[\Delta]) \vdash V$).

$$\begin{array}{c}
\begin{array}{c} \vdots \delta_2 \\ \Theta \vdash U \end{array} \quad \frac{\begin{array}{c} \vdots \delta_0 \quad \vdots \delta_1 \\ \Delta \vdash A \otimes B \quad \Gamma[(A; B)] \vdash C \end{array}}{\Gamma[\Delta] \vdash C} R \Rightarrow \\
\hline
\langle \Theta; \Gamma[\Delta] \rangle \vdash V \quad [\backslash_e]
\end{array}$$

$$\begin{array}{c}
\begin{array}{c} \vdots \delta_0 \\ \Delta \vdash A \otimes B \end{array} \quad \frac{\begin{array}{c} \vdots \delta_2 \quad \vdots \delta_1 \\ \Theta \vdash U \quad \Gamma[(A; B)] \vdash C \end{array}}{\langle \Theta; \Gamma[(A; B)] \rangle \vdash V} [\backslash_e] \\
\hline
\langle \Theta; \Gamma[\Delta] \rangle \vdash V \quad R
\end{array}$$

Figure 7: The go down of the product on the rule \backslash_e in PCMLL.

Proof. Implicative eliminations do not modify the order between formulas of a same premise and do not use them. Product elimination and entropy rules do not modify formulas but only hypothesis. Then, these rules could be used in any order. \square

Theoreme 3. *Every proof δ of PCMLL has a normal form.*

Proof. Let δ a proof such that $|\delta| = \langle n, e, g \rangle$.

We proceed by induction on the measure of proof.

By induction, every proof δ' of measure $|\delta'| < \langle n, e, g \rangle$ has a normal form.

If δ contains a redex: reducing this redex reduces the number of rules in δ , then the resulting proof δ' is such that $n(\delta') < n(\delta)$, hence $|\delta'| < |\delta|$. By induction δ' has a normal form, then δ also.

Else:

If $e(\delta) \neq 0$, then there is an implicative elimination rule S such that S is in a $e(\delta)$ -extended-redex. This $e(\delta)$ -extended-redex must be minimal, and the property 1 implies that it contains only product elimination and entropy rules. Moreover, the property 6 allows to rise S over the rule above it (which corresponds to lower a product elimination or entropy rules below an implicative elimination). The proof obtained δ' is such that $n(\delta') = n(\delta)$ and $e(\delta') = e(\delta) - 1$. The measure reduces and the induction allows to conclude that δ' has a normal form and then δ .

Else $e(\delta) = 0$:

If $g(\delta) \neq 0$: then it exists a product elimination rule R such that R is in a $g(\delta)$ -extended-redex. In this case, δ does not contain any implicative extended-redex and is it is minimal, it only contains product elimination and entropy rules. R could rise over the left premise – the rule conjoined \otimes_i is necessary in this part of the proof.

But for those rules, product eliminations can still rise above as shown in the property 3.

Then, we obtain the proof δ' such that $n(\delta') = n(\delta)$.

We check that none new k -extended-redex based on a rule of $IEP(\delta)$ appears:

δ has the form:

$$\begin{array}{c}
 \begin{array}{ccc}
 & A \odot B & \\
 \vdots & \vdots & \vdots \\
 \vdots \delta_2 & \vdots \delta_3 & \vdots \\
 \vdots & \vdots & \vdots \delta_1 \\
 X & A \odot B & \\
 \hline
 A \odot B & [R] & D \\
 \hline
 D & [\odot_e] & \\
 \vdots & & \\
 \vdots \delta_4 & & \\
 \vdots & &
 \end{array}
 \end{array}$$

- every principal branch of δ_3 to δ_4 does not contain extended-redexes because δ_3 is in the left part of \odot_e
- every principal branch of δ_1 to δ_4 can contain extended-redexes.
- every principal branch of δ_2 to δ_4 does not contain extended-redexes because δ_2 is in the left part of \odot_e (only for product elimination rules).

The reduction scheme of the redex then gives the new structure of the proof δ' :

$$\begin{array}{c}
 \begin{array}{ccc}
 & A \odot B & \\
 \vdots & \vdots & \vdots \\
 \vdots \delta_2 & \vdots \delta_3 & \vdots \delta_1 \\
 \vdots & \vdots & \vdots \\
 X & A \odot B & D \\
 \hline
 & D & [\odot_e] \\
 \hline
 D & [R] & \\
 \vdots & & \\
 \vdots \delta_4 & & \\
 \vdots & &
 \end{array}
 \end{array}$$

In this new proof:

- every principal branch of δ_3 to δ_4 does not contain extended-redexes because δ_3 is in the left part of \odot_e
- every principal branch of δ_1 to δ_4 does not contain new extended-redexes, and the measure for these extended-redexes decrease of 1.
- every principal branch of δ_2 to δ_4 does not contain extended-redexes because δ_2 is in the left part of R (which is necessary a product elimination rule).

The proof does not contain new k extended-redex ; then the proof does not contain new implicative extended-redex. Then, we have $e(\delta') = e(\delta)$ and $g(\delta') = g(\delta) - 1$. Thus $|\delta'| < |\delta|$ and by induction δ' has a normal form, and therefor δ also has one.

Else: we have $e(\delta) = g(\delta) = 0$, because of the property 4, δ is in normal form.

□

Normal forms of the previous examples are presented in the figure 6. they are derived from the underlying algorithm of the previous proof.

<p>example 1</p> $\frac{\frac{\frac{\vdash E \odot F}{\vdash C \setminus (A/B)} [\odot_e] \quad \frac{\frac{(C, \langle E; F \rangle) \vdash A/B}{\langle E; F \rangle \vdash C \setminus (A/B)} [\setminus_i] \quad \vdash C}{\vdash A/B} [\setminus_e]}{\vdash C \setminus (A/B)} [\odot_e] \quad e(\setminus_e) = 0$	<p>example 2</p> $\frac{\frac{\frac{A \vdash E \quad B \vdash F}{(A, B) \vdash E \otimes F} [\otimes_i] \quad (E, F) \vdash D}{(A, B) \vdash D} [\otimes_e] \quad \vdash A \otimes B}{\vdash D} [\otimes_e] \quad g(\odot_e) = 0$
---	---

Now, for proofs in normal form, we check the sub-formula property.

5.3 Sub-formula property for PCMLL

Theoreme 4. *The sub-formula property holds for PCMLL: in a normal proof δ of a sequent $\Gamma \vdash C$, every formulae of a sequent is a sub-formula of some hypothesis (Γ) or of the conclusion (C).*

Proof. We proceed by induction on the number of rules of the proof. Once again, we use a stronger definition of the property:

every formulae in a normal proof are sub-formulae of some hypotheses or of the conclusion of the proof and if the last rule used is an implicative elimination \setminus_e , $/_e$ or \multimap_e every sub-formulae are sub-formulae of some hypotheses only.

Remark that entropy conserve the premise as conclusion, then entropy checked the sub-formula property for every case. And moreover, the axiom rule checks the property because the sequent is an hypothesis.

We check the induction hypothesis after the use of each other rule:

1. for the rule \setminus_e .

$$\frac{\begin{array}{c} \Delta_1 \\ \vdots \delta_1 \\ \Delta \vdash C \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \delta_2 \\ \Gamma \vdash C \setminus D \end{array}}{\langle \Delta; \Gamma \rangle \vdash D} [\setminus_e]$$

By hypothesis induction, every formulae in δ_1 are sub-formulae of hypotheses Δ_1 or of the conclusion C . And every formulae in δ_2 are sub-formulae of hypotheses Γ_2 or of the conclusion $C \setminus D$. However C is sub-formula of $C \setminus D$ and D too. One needs that $C \setminus D$ was sub-formula of hypothesis of δ_2 hypotheses.

Let us look the rule R which produce $C \setminus D$:

- if R is \setminus_i : impossible because it is a 0-extended-redex and the proof is in normal form.
- if R is $/_i$, $\neg o_i$, \otimes_i or \odot_i : these cases are structurally impossibles beause because they can not produce $C \setminus D$.
- if R is \setminus_e , $/_e$ or $\neg o_e$: we use the induction hypothesis and $C \setminus D$ is sub-formula of Γ_2 .
- if R is \otimes_e , \odot_e or entropy: keep as conclusion the premise, thus we have to check the rule above.

If it is one of the previous rule, we use the same argument. Else, the proof is a finite sequence of \otimes_e , \odot_e and entropy. They keep as conclusion a premise and then $C \setminus D$ is one of the hypothesis of Γ_2 .

2. it is strictly symmetrical for $/_e$ and $\neg o_e$:

$$\frac{\begin{array}{c} \Gamma_2 \\ \vdots \delta_2 \\ \Gamma \vdash D / C \end{array} \quad \begin{array}{c} \Delta_1 \\ \vdots \delta_1 \\ \Delta \vdash C \end{array}}{\langle \Gamma; \Delta \rangle \vdash D} [/_e] \quad \frac{\begin{array}{c} \Delta_1 \\ \vdots \delta_1 \\ \Delta \vdash C \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \delta_2 \\ \Gamma \vdash C \neg o D \end{array}}{\langle \Delta; \Gamma \rangle \vdash D} [\neg o_e]$$

Every formulae are sub-formulae of D / C (*resp.* $C \neg o D$) or of hypotheses and D / C (*resp.* $C \neg o D$) is sub-formula of hypothesis Γ_2 .

3. For all the introduction rules, the conclusion of each part of the proof are sub-formula of the conclusion. If the rule R is \setminus_i , Let the proof δ :

$$\frac{\begin{array}{c} \Gamma \\ \vdots \delta_1 \\ \langle \Gamma; C \rangle \vdash D \end{array}}{\Gamma \vdash C \setminus D} [\setminus_i]$$

Using the induction hypothesis, every formula in δ_1 are sub-formulae of hypothesis Γ or conclusion D . However D is sub-formula of $C \setminus D$ thus in d every formulae of δ are sub-formulae of Γ or of the conclusion $C \setminus D$.

4. for the rule $/_i$, let the proof δ :

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \end{array} \quad \begin{array}{c} < \Gamma; C > \vdash D \end{array}}{\Gamma \vdash D / C} [\text{I}]$$

Every formulae in δ_1 are sub-formulae of hypotheses Γ ou of the conclusion D . D is sub-formula of D / C , then every formula of δ is sub-formula of hypotheses Γ or of the conclusion D / C . The property is checked in δ .

5. for the rule \neg_o , let theproof δ :

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \end{array} \quad \begin{array}{c} < \Gamma; C > \vdash D \end{array}}{\Gamma \vdash C \neg_o D} [\neg_o]$$

Every formulae in δ_1 are sub-formulae of hypotheses Γ ou of the conclusion D . D is sub-formula of $D \neg_o C$, then every formula of δ is sub-formula of hypotheses Γ or of the conclusion D / C . The property is checked in δ .

6. for the rule \otimes_i , let the proof δ :

$$\frac{\begin{array}{c} \Delta_1 \\ \vdots \\ \delta_1 \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \end{array} \quad \begin{array}{c} \Delta \vdash C \quad \Gamma \vdash D \end{array}}{(\Delta, \Gamma) \vdash C \otimes D} [\otimes_i]$$

- every formulae in δ_1 are sub-formulae of hypotheses Δ_1 or of the conclusion C .
- every formulae in δ_2 are sub-formulae of hypotheses Γ_2 or of the conclusion D .
- however C and D are sub-formulae of $C \otimes D$, then in Γ , every formulae are sub-formulae of hypotheses Δ and Γ or of the conclusion $C \otimes D$.

7. for the rule \otimes_e :

$$\frac{\begin{array}{c} \Delta_1 \\ \vdots \\ \delta_1 \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \end{array} \quad \begin{array}{c} \Delta \vdash A \otimes B \quad \Gamma \vdash D \end{array}}{D} [\otimes_e]$$

- every formulae of δ_1 are sub-formulae of hypotheses Δ_1 or of the conclusion $A \otimes B$.
- every formulae of δ_2 are sub-formulae of hypotheses Γ_2 or of the conclusion D .
- moreover, D is the conclusion of δ . Thus, every formulae of δ_2 are sub-formulae of hypotheses Γ_2 or of the conclusion of the proof δ : D .

To check that the property holds for the other part of the proof, we must prove that $A \otimes B$ is sub-formula of hypotheses of δ_1 . We look at the rule R above:

- if R is $\backslash_e, /_e$ or \multimap_e , using the induction hypothesis $A \otimes B$ is sub-formula of hypotheses Δ_1 .
- if R is \otimes_i : this case is impossible because it might be a 0-extended-redex and the proof is in normal form.
- if R is $\backslash_i, /_i, \multimap_i$ ou \odot_i : these case are structurally impossible because these rules can not produce $A \otimes B$.
- if \otimes_e, \odot_e or entropy: they keep one premise as conclusion, we analyze the rule above:
 - either there is one of the previous rules, thus using the same arguments we conclude.
 - either, the proof containing a finite number of rules, the sequence of rules is finite. Moreover, it contains only \otimes_e, \odot_e and entropy rules, thus the formula is a hypothesis in Δ_1 .

In every possible case, $A \otimes B$ or $A \odot B$ is sub-formula of hypotheses.

In PCMLL, all proofs have a normal form and enjoy the sub-formula property. \square

6 Conclusion

Motivated by concurrency and computational linguistics we have been defining PCMLL in natural deduction and proved normalisation. For Lambek calculus with product, a subcalculus of PCMLL, we also characterized the unique normal proof. We only sketched the normal form(s) for the complete calculus and writing this is our next job.

Next we'll look forward a proof net syntax for PCMLL, which also allows to easily compute lambda terms and semantic reading. Despite existence of proof nets for MLL and the Lambek calculus (of which PCMLL is the superimposition), and for intuitionistic NL of Abrusci and Ruet, because of the more flexible entropy rule that we are using, there is not yet any proof net calculus for PCNLL. This work can be viewed as a first step in this direction.

The uniqueness of the normal form for PCMLL can be done because what distinguishes two proofs in normal form of the same proof, is the relative position of \otimes_e in the sequences in which they belong. In reality, there exists a unique normal form but not canonical. The algorithm of normalization that we propose retains the starting order between the different \otimes_e . This gives a unique form. However, proofs with inversions of \otimes_e in sequences of \otimes_e provide equivalent proofs.

An extension of this work to a version as proof network of PCMLL (which also provide λ -terms and semantic representations) is a later stage will continue this work.

With respect to computational linguistic application, we look forward a simpler translation from PCMLL formulae to arrow types on e and t and thus from parse structures that are PCMLL deduction to intuitionistic deduction, which are semantic readings. This open the question to interpret noun phrase and generalised quantifiers as the combination of k (case) and d (entities). The linear logic view on syntax also introduce the possibility to use game semantics, Girard's ludics, rather than model theoretic semantics for interpreting parse

structures, and this ongoing work is part a French national research program called Prelude.

References

- [Abr91] V. Michele Abrusci. Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. *The Journal of Symbolic Logic*, 56(4):1403–1451, December 1991.
- [Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.
- [ALR04] Maxime Amblard, Alain Lecomte, and Christian Retoré. Synchronization syntax semantic for a minimalism theory. *Journée Sémantique et Modélisation*, 2004.
- [Amb06] Maxime Amblard. Treating clitics with minimalist grammars. *Formal grammar - Malaga*, 2006.
- [Amb07] Maxime Amblard. *Calcul de représentations sémantiques et syntaxe générative: les grammaires minimalistes catégorielles*. PhD thesis, université de Bordeaux 1, 2007.
- [AR99] V. Michele Abrusci and Paul Ruet. Non-commutative logic I: The multiplicative fragment. *Annals of pure and applied logic*, 101(1):29–64, 1999.
- [BdGR97] Denis Bechet, Philippe de Groote, and Christian Retoré. A complete axiomatisation of the inclusion of series-parallel partial orders. In H. Comon, editor, *Rewriting Techniques and Applications, RTA '97*, volume 1232 of *LNCS*, pages 230–240. Springer Verlag, 1997.
- [dG96] Philippe de Groote. Partially commutative linear logic: sequent calculus and phase semantics. In Vito Michele Abrusci and Claudia Casadio, editors, *Third Roma Workshop: Proofs and Linguistics Categories – Applications of Logic to the analysis and implementation of Natural Language*, pages 199–208. Bologna:CLUEB, 1996.
- [dGR96] 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, 1996. FoLLI.
- [Gen34a] Gehrard Gentzen. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, 39:176–210, 1934. Traduction Française de R. Feys et J. Ladrière: Recherches sur la déduction logique, Presses Universitaires de France, Paris, 1955.
- [Gen34b] Gehrard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39:405–431, 1934. Traduction française de J. Ladrière et R. Feys: Recherches sur la déduction logique, Presses Universitaires de France, Paris, 1955.

- [Gen36] Gerhard Gentzen. Die Widerspruchsfreiheit der reine Zahlentheorie. *Mathematische Annalen*, 112:493–565, 1936.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
- [Gir97] Jean-Yves Girard. Du pourquoi au comment : la théorie de la démonstration de 1950 à nos jours. unpublished manuscript, 1997.
- [GLT88] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Number 7 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1988.
- [Gug02] Alessio Guglielmi. A system of interaction and structure, September 25 2002.
- [Lam58] Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, pages 154–170, 1958.
- [LR01] Alain Lecomte and Christian Retoré. Extending Lambek grammars: a logical account of minimalist grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, ACL 2001*, pages 354–361, Toulouse, July 2001. ACL.
- [Neg02] Sara Negri. A normalizing system of natural deduction for intuitionistic linear logic. *Archive for Mathematical Logic*, 2002.
- [Ret97] Christian Retoré. Pomset logic: a non-commutative extension of classical linear logic. In Philippe de Groote and James Roger Hindley, editors, *Typed Lambda Calculus and Applications, TLCA’97*, volume 1210 of *LNCS*, pages 300–318, 1997.
- [Ret04] Christian Retoré. A description of the non-sequential execution of petri nets in partially commutative linear logic. In Jan van Eijck, Vincent van Oostrom, and Albert Visser, editors, *Logic Colloquium 99*, Lecture Notes in Logic, pages 152–181. ASL and A. K. Peters, 2004.
- [Ret05] Christian Retoré. The logic of categorial grammars – lecture notes. Research Report 5703, INRIA, 2005. 108 pp.
- [Sta97] Edward Stabler. Derivational minimalism. *Logical Aspect of Computational Linguistic*, 1328, 1997.