Partially Commutative Linear Logic and Lambek Caculus with Product: Natural Deduction, Normalisation, Subformula Property

Maxime Amblard, Christian Retoré

Partially Commutative Linear Logic and Lambek Calculus with Product: Natural Deduction, Normalisation, Subformula Property

Maxime Amblard
Loria (UMR 7503) Université de Lorraine, CNRS, INRIA Nancy Grand-Est
amblard@loria.fr

Christian Retoré
LaBRI (UMR 5800) Université de Bordeaux, CNRS
retore@labri.fr

Abstract

This article defines and studies a natural deduction system for partially commutative intuitionistic multiplicative linear logic, that is a combination of intuitionistic commutative linear logic with the Lambek calculus, which is non-commutative, and was first introduced as a sequent calculus by de Groote.

In this logic, the hypotheses are endowed with a series-parallel partial order: the parallel composition corresponds to the commutative product, while the series composition corresponds to the noncommutative product. The relation between the two products is that a rule, called *entropy*, allows us to replace a series-parallel order with a sub series-parallel order — this rule (already studied by Retoré) strictly extends the entropy rule initially introduced by de Groote. A particular subsystem emerges when hypotheses are totally ordered: this is Lambek calculus with product, and when orders are empty it is is multiplicative linear logic.

So far only the sequent calculus and cut-elimination have been properly studied. In this article, we define natural deduction with product elimination rules as Abramsky proposed long ago. We then give a brief illustration of its application to computational linguistics and prove normalisation, firstly for the Lambek calculus with product and then for the full partially ordered calculus. We show that normal proofs enjoy the subformula property, thus yielding another proof of decidability of these calculi.

The authors wish thank Jiří Maršík (LORIA, Nancy) for his prompt and efficient rereading.

This logic was shown to be useful for modelling the truly concurrent execution of Petri nets and for minimalist grammars in computational linguistics. Regarding this latter application, natural deduction and the Curry-Howard isomorphism is extremely useful since it leads to the semantic representation of analysed sentences.

Keywords: Logic; Intuitionistic Noncommutative Logic; Lambek calculus; normalisation

1 Presentation

Non-commutative logic arises both as a natural mathematical generalisation of commutative logic and in the modeling of some computational phenomena that require some noncommutativity.

Both truth-value semantics (phase semantics, based on monoids which can be non commutative) and syntax (sequent calculus with sequences rather than sets of formulae, proof nets with non-crossing axiom links) suggest the study of noncommutative linear logic — which fits less well with proof semantics, except for Pomset logic.

Non commutativity is also appealing from a real world application perspective such as in concurrency theory, like the truly concurrent execution of Petri net, and in our favourite application, computational linguistics. This goes back to the fifties and the apparition of the Lambek calculus. We first give a brief presentation of noncommutative logics and then stress their interest with respect to concurrency theory and to computational linguistics, before introducing and studying natural deduction for this calculus.

1.1 Noncommutative linear logics

Linear logic [10] offered a logical view of the Lambek calculus [14] and noncommutative calculi. During many years, the difficulty was to integrate commutative connectives and noncommutative connectives.

The first solution, inspired by denotational semantics (coherence spaces) was Pomset Logic of Retoré (1993) [19, 20]. This logic was defined as an extension of proof net syntax with a faithful interpretation in the category of coherence spaces. In addition to the multiplicative conjunction and disjunction Pomset logic has a noncommutative and self-dual connective called “before". By now it has been generalised and is now studied with extended sequent calculi called Calculus of Structures [12].

Another kind of calculus was introduced as a sequent calculus by de Groote in [8]. Let us stress that it is an intuitionistic calculus (several hypotheses, a single conclusion) and that the classical extension by [3] is quite difficult. The intuitionistic calculus introduced by de Groote, called partially commutative linear logic, consists of a superposition of the Lambek calculus (noncommutative) and of Intuitionistic Linear Logic (commutative). For making a distinction between the two connectives it is necessary that the context includes two different commas that mimic the conjunctions, one being commutative and the other being noncommutative. Hence we deal with series-parallel partial orders over multisets of formulae as right-hand side of sequents. Let us write $(...,...)$ for the parallel composition and $\langle ...;... \rangle$ for the noncommutative one: 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 entails the noncommutative one, or the converse. Surprisingly enough, the two options work just as well — provided one direction is fixed once and for all, of course! This relation between the two products results from a structural rule acting on the order.

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

According to our view of this calculus, J can be any order such that $J \subset I$ (ordered are viewed as sets of ordered pairs of formulae in Γ). That's the version of this calculus that we already used for a logical description of Petri net firing, and for viewing derivations in minimalist grammars as proofs. Indeed Bechet, de Groote and Retoré, in [7], showed that only four rewriting rules are needed to obtain all possible series-parallel partial suborders from some series-parallel partial order.

In de Groote's calculus as well as in Abrusci and Ruet's calculus, the resulting order J can only be obtained by replacing some noncommutative commas/products with commutative ones. This is not equivalent to our formulation, indeed Ruet showed in his PhD dissertation [23] that there cannot exist a classical calculus with the classical analogue of our order rule.

Here is an example of a derivation that can be performed in our calculus but not in theirs:

$$\dfrac{\dfrac{\langle (a,b);(c,d) \rangle \vdash (a \otimes b) \odot (c \otimes d)}{(\langle a;c \rangle, \langle b;d \rangle) \vdash (a \otimes b) \odot (c \otimes d)}}{(a \odot c) \otimes (b \odot d) \vdash (a \otimes b) \odot (c \otimes d)}$$

Up to now our calculus can only be presented with a sequent calculus: there exist neither proof nets, nor natural deduction, only a sequent calculus that has

been proven to enjoy cut-elimination in [21]. This is the reason why this paper proposes a natural deduction system, as well as a notion of normal proof, and a normalisation theorem that entails the subformula property.

Commutation of rules is tricky in this calculus. Hence, although we are absolutely convinced that there is a form of confluence and a form of strong normalisation, the possibility to swap these rules results in a lengthy and complicated proof, although there surely is nothing deep into these forgotten proofs.

1.2 Applications of noncommutative (linear) logics

Noncommutativity in logic is rather natural in a resource consumption perspective. A 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 [2] and others, linearity is a mandatory condition for noncommutativity. The first noncommutative logical calculus, namely the Lambek calculus designed for the grammatical description of natural language, was invented long before linear logic. It is nevertheless a particular system of linear logic, whose relation to other logical calculi, in particular to intuitionistic, was better understood with the help of linear logic.

Concurrency, in which the order of the computations or of the resources matters, is of course a natural application of noncommutative logic(s). In the framework of proofs as programs, and normalisation as computation, Pomset Logic and the subsequent calculus of structures are easier to understand because the order on the conclusions also concerns cuts, which are the computations to be performed [20, 12].

The noncommutative logical calculus that we study in this paper, as well as those by Lambek, Abrusci, Ruet *etc.*, better matches proof search as computation that is at work in linear logic programming [13], planning [16], Petri net firing or parsing in computational linguistics [21]. For instance, Retoré 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 nondeterministic choice and ";" sequential composition). 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 [21].

Our preferred motivation for such calculi is computational linguistics and grammar formalisms, in particular the deductive description of mildly context-sensitive formalisms. They are assumed to be large enough for natural language constructs, go beyond context-free languages, but admit polynomial parsing algorithms. Logical descriptions of grammar classes as introduced by Lambek are especially appealing because the parse structure and a semantic lexicon automatically lead to a formula

which represents the semantics of the analysed sentence. This is especially simple when the Lambek calculus or the partially commutative extensions that we are considering here are described as natural deduction systems. Indeed, the syntactic categories can be turned into semantic categories over 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 in intuitionistic logic, that is a lambda term: when the semantic lambda terms are inserted at the place occupied by the corresponding words, one obtains a lambda term that reduces to the semantics of the sentence, *i.e.* a logical formula written as a lambda term.

Lambek calculus is definitely too restrictive as a syntactic formalism, in particular it only describes context-free languages, thus leaving out many common syntactic constructions. This is the reason to use partially commutative calculi instead of Lambek calculus. In particular Lecomte and Retoré managed to give a logical presentation [15] of Stabler's minimalist grammars [24] in partially commutative linear logic, presented in natural deduction in order to obtain semantic representations of the parsed sentences. In such a framework and for other applications as well, normalisation is quite important : 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 proven, and both the parse structure and the semantic reading are obtained from the normal form.

2 Partially Commutative Intuitionnistic Multiplicative Linear Logic (pcIMLL)

2.1 Series-parallel ordered multisets of formulae

Formulae and contexts are defined as in the initial work of de Groote in [8].

Formulae are defined from a set of propositional variables P, by the commutative conjunction ($\otimes$), the noncommutative conjunction ($\odot$), the commutative implication ($\multimap$), the two noncommutative implications ($/$ and $\backslash$):

$$\mathrm{L} ::= \mathrm{P} \mid \mathrm{L} \odot \mathrm{L} \mid \mathrm{L} \otimes \mathrm{L} \mid \mathrm{L} \,/\, \mathrm{L} \mid \mathrm{L} \setminus \mathrm{L} \mid \mathrm{L} \multimap \mathrm{L}$$

Contexts, that are left-hand side of sequents, are multisets of formulae endowed with a series-parallel (SP) partial order. Contexts are denoted by upper case Greek letters. Such orders can be defined by two operations: disjoint union, or parallel

$$\frac{\Gamma \vdash A \quad \Delta \vdash A \setminus C}{\langle \Gamma; \Delta \rangle \vdash C}\,[\setminus_e] \qquad \frac{\Delta \vdash C \,/\, A \quad \Gamma \vdash A}{\langle \Delta; \Gamma \rangle \vdash C}\,[/_e] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A \multimap C}{(\Gamma, \Delta) \vdash C}\,[\multimap_e]$$

$$\frac{\langle A; \Gamma \rangle \vdash C}{\Gamma \vdash A \setminus C}\,[\setminus_i] \qquad \frac{\langle \Gamma; A \rangle \vdash C}{\Gamma \vdash C \,/\, A}\,[/_i] \qquad \frac{(A, \Gamma) \vdash C}{\Gamma \vdash A \multimap C}\,[\multimap_i]$$

$$\frac{\Delta \vdash A \odot B \quad \Gamma[\langle A; B \rangle] \vdash C}{\Gamma[\Delta] \vdash C}\,[\odot_e] \qquad \frac{\Delta \vdash A \otimes B \quad \Gamma[(A, B)] \vdash C}{\Gamma[\Delta] \vdash C}\,[\otimes_e]$$

$$\frac{\Delta \vdash A \quad \Gamma \vdash B}{\langle \Delta; \Gamma \rangle \vdash A \odot B}\,[\odot_i] \qquad \frac{\Delta \vdash A \quad \Gamma \vdash B}{(\Delta, \Gamma) \vdash A \otimes B}\,[\otimes_i]$$

$$\frac{}{A \vdash A}\,[axiom] \qquad \frac{\Gamma \vdash C}{\Gamma' \vdash C}\,[\text{entropy — whenever } \Gamma' \sqsubset \Gamma]$$

Figure 1: Rules of pcIMLL.

composition, denoted by (Γ, Δ) and series composition denoted by $\langle \Gamma; \Delta \rangle$: the domain is the disjoint union of Γ and Δ, and every formula in Γ comes before any formula in Δ. Contexts obey the following syntax:

$$\text{CTX} ::= \text{L} | \langle \text{CTX}; \text{CTX} \rangle \mid (\text{CTX}, \text{CTX})$$

For example, the context $\langle \langle B; (A \multimap (B \setminus (D \,/\, C), A) \rangle; C \rangle$ denotes the sp order, $Succ(B) = (A, A \multimap (B \setminus (D \,/\, C))$, $Succ(A) = Succ(A \multimap (B \setminus (D \,/\, C)) = \{C\}$ where $Succ(X)$ is the multiset of the immediate successors of X (we only consider finite orders).

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: even though the sp terms denoting them are different, the left-hand sides of two sequents are considered equal whenever they are equal as partially ordered multisets.

An expression $\Gamma[*]$ stands for a context in which we distinguish a specific element $[*]$ and an expression $\Gamma[\Delta]$ denotes the context obtained be replacing $*$ in $\Gamma[*]$ by the context Δ in $\Gamma[*]$.

Figure 1 shows the rules of pcIMLL. It uses the standard rules of commutative multiplicative intuitionistic linear logic and of non-commutative multiplicative intuitionistic linear logic. Both have introductions and eliminations for the implicative connectives and the product connectives. In addition there is the axiom rule and the entropy rule ($\sqsubset$) which correspond to inclusions of orders.

This calculus deserves some explanation and comments:

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 system (modulo commutativity and associativity) on the sp term denoting them as shown in [7] – see also [21] where the order rule is used the other way round, but as said in the introduction, it does not change normalisation.

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

$$\forall X \neq A, B \begin{cases} X < A \Leftrightarrow X < B \\ X > A \Leftrightarrow X > B \end{cases}$$

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 conclusions of rules, they are replaced by the context which lead to $A \otimes B$ (*i.e.* Δ in figure 1). Abramsky introduced similar rules long ago in [1].

Although we do have normalisation and the subformula property (see next sections) we avoided complicated rules of the kind introduced in [18] for MLL. We assume her rules are motivated by other properties; indeed, they work for the complete linear calculus with additive and exponentials. But in the restricted multiplicative case, our simple rules are preferable.

2.2 Principal branches

Let δ be a proof of this calculus, and let S_j be an occurrence of a sequent $|S_j|$ in δ, and $|S_j|^r$ be the conclusion of this sequent, that is the unique formula in the right-hand side of this sequent.

We write $B(S_0)$ for the **principal branch** starting with the sequent S_0 — see e.g. [11, p. 75] or [17, p. 35]. It is the smallest path which contains S_0 and is closed under the following operations:

1. If $S \in B(S_0)$ is obtained by a unary rule R from an occurrence of a sequent S', 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 $|S'| = \Gamma[\langle A, B\rangle] \vdash C$ (*resp.* $|S'| = \Gamma[(A, B)] \vdash C$) is also in $B(S_0)$.

3. If $S \in B(S_0)$ is obtained by an implicative elimination rule $\backslash_e$ (*resp.* $/_e$, $\multimap_e$), then the premise $|S'| = \Delta \vdash A \setminus C$ (*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 the conclusion of an elimination rule and $|S_i|$ the conclusion of introduction rule, the two sequents are said to be **conjoined** — if there are no rules in-between these two these two rules, they define a redex.

3 A brief example using pcIMLL in Computational Linguistics: categorial minimalist grammars

Before we prove the normalisation and subformula property of pcIMLL, let us illustrate briefly our use of this calculus in computational linguistics — for more details see [15, 4, 6]. As said above, Lambek calculus is too restricted to describe some constructions in natural language syntax, hence, we need a richer logical calculus, in order to remain in the parsing-as-deduction paradigm and to have a simple syntax semantic interface. The calculus presented in this paper, namely partially commutative intuitionistic multiplicative linear logic, pcIMLL, is able to account of many more syntactical phenomena, especially when viewed as a natural deduction system.

Indeed, this logical calculus can account for Stabler's minimalist grammars, which are an elegant and computationally efficient formalisation of Chomsky's minimalist program. As first observed in [9], such a view of syntax is not too far from categorial grammar like Lambek grammars. [24, 22, 4, 6]. Minimalist grammars cover all (or most of) syntactic constructions, but lacks a simple connection to semantics because it is not a deductive system.

Therefore we defined *categorial minimalist grammars*, a deductive grammatical formalism that resembles Lambek grammars: a lexicon maps every word into a formula of partially commutative linear logic which describes its interaction with other words. The main difficulty was to mimic the Chomskyan notion of movement: the proper word order is recovered in a second step that labels the proof nodes. We do not use all the possible combinations of rules of pcIMLL but only combination of fixed sequences of rules, that correspond to minimalist-grammar rules. Thus parse structures are derivations in pcIMLL where every formula of every sequent in the proof is labelled with strings of words and variables. Parsing consists in deriving in natural deduction *sentence* : C from axioms $x : A \vdash x : A$ and proper axioms $\vdash w : T$ where T is the formula associated with w in the lexicon.

The MERGE rule is almost like residuation rules in categorial grammars with *noncommutative* implications:

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

The MOVE rule was trickier to encode and required the *commutative* product:

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

The MOVE rule is typical of Chomskyan linguistics: our encoding in pcIMLL mimics the movement of the constituent / string s from the place y to the place x .

Here is a simple example involving movement with a few entries from an Italian lexicon — null subjects are allowed in this language, it makes the example simpler. Observe how an interrogative noun phrase is moved to the leftmost position:

que	$wh \otimes (k \otimes d) \,/\, n$	cosa	n	ϵ	$k \otimes d$
fai	$k \setminus d \setminus v \,/\, d$	infl	$k \setminus t \,/\, v$	comp	$wh \backslash c / t$

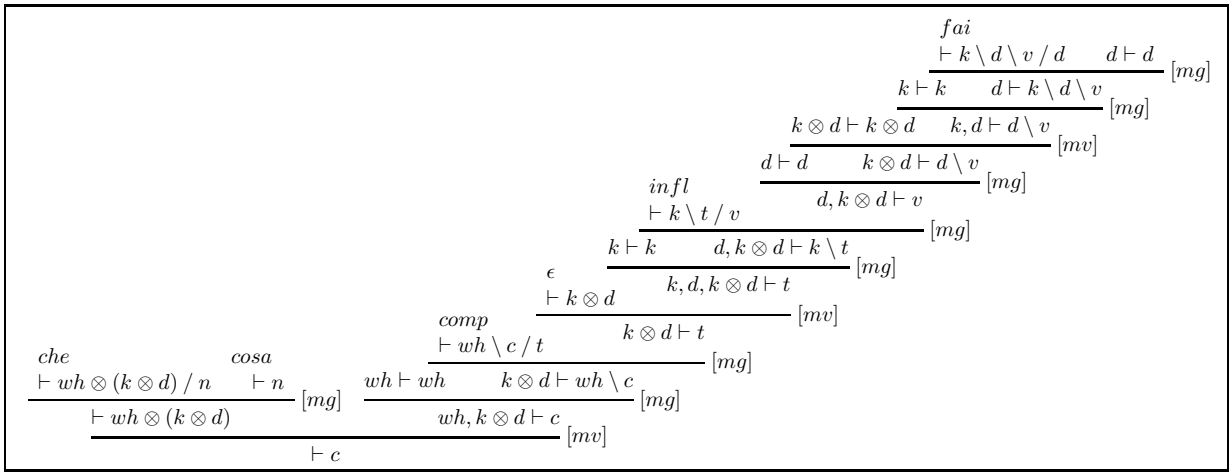


Figure 2: Analysis of "che cosa ϵ fai ?"

Our interest for such analyses is to automatically compute the semantic representation (usually, a formula of first-order or higher-order logic) with correct quantifier scope. The semantic representation of our example is:

$$\exists ?(\lambda x\ (\wedge(cosa(x))(far(tu, x)))) \equiv \exists ?x(cosa(x) \wedge far(tu, x)$$

Our presentation of categorial minimalist grammars in pcIMLL is extremely brief and sketchy, but it gives an idea of why the logical calculus in this paper and its

natural deduction formulation is relevant to computational linguistics. Details can be found in [15, 4, 5].

Regarding the application to true concurrency and Petri nets, see [21]

4 Normalisation of Lambek calculus with product ($L_\odot$)

Before to give an algorithm for normalisation for the full pcIMLL calculus, let us deal with $L_\odot$, that is the Lambek calculus with product. For this calculus we shall define normal natural deductions, give a normalisation algorithm and prove that it always reach a normal proof. This restricted case is simpler, in particular *entropy* rule is never used. Nevertheless this simpler case shows the difficulties of rule commutations in such calculi, which will be tougher for full pcIMLL in the next section.

4.1 Some properties of the Lambek calculus with product $L_\odot$

Lambek calculus with product ($L_\odot$) is the restriction of pcIMLL to the connectives: $\backslash$, / and $\odot$. Furthermore, contexts are always totally ordered multi sets of formulae, that are sequences of formulae: hence the associative $\langle ...; ... \rangle$ braces are omitted. Observe that in this setting, the entropy rule cannot be applied.

Property 1. *Let R be a product elimination rule $\odot_e$ yielding $\Gamma[\Delta] \vdash C$ from a proof δ_0 of $\Delta \vdash A \odot B$ and a proof of $\Gamma[A, B] \vdash C$ obtained by a rule R' from a proof δ_1 of $\Theta[A, B] \vdash X$ and from a second proof δ_2 of $\Psi \vdash U$ — if R' is a binary rule.*

A proof for the same sequent $\Gamma[\Delta] \vdash C$ can be derived by the following two steps:

1. *apply R between the proof δ_0 of $\Delta \vdash A \odot B$ and the proof δ_1 of $\Theta[A, B] \vdash X$ as conclusion, yielding $\Theta[\Delta] \vdash X$;*

2. *apply R' to this new proof and to the proof δ_2.*

In other words, provided that the hypotheses A, B that are cancelled by the R product elimination rule are in the same premise of the R' rule, the R product elimination rule can swing over R', as shown in figure 3.

Proof. The proof of this proposition is rather lengthy but simple: it is a case study of all the possible rules R' above the product elimination R. Here are the possible cases:

- R' is $\backslash_e$:

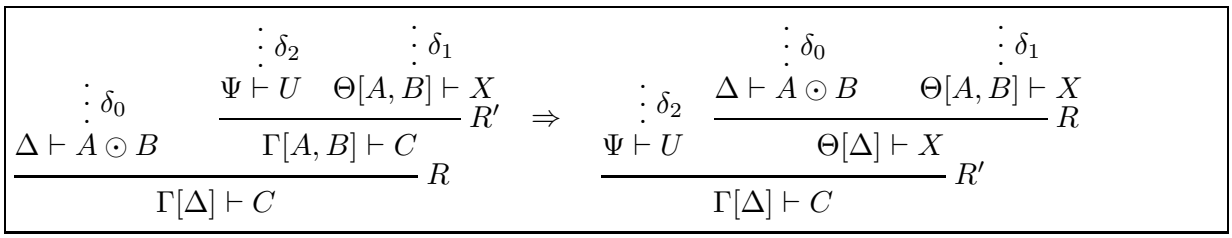


Figure 3: A product elimination rule R swings over a rule R' in $\mathrm{L}_{\odot}$.

- hypotheses A, B are in the left premise of $\backslash_e$:

$$\dfrac{\dfrac{\Gamma \vdash A \odot B \qquad \dfrac{A, B \vdash D \quad \Delta \vdash D \setminus C}{A, B, \Delta \vdash C}\,[\backslash_e]}{\Gamma, \Delta \vdash C}\,[\odot_e]}{} \quad \Rightarrow \quad \dfrac{\dfrac{\dfrac{\Gamma \vdash A \odot B \quad A, B \vdash D}{\Gamma \vdash D}\,[\odot_e] \qquad \Delta \vdash D \setminus C}{\Gamma, \Delta \vdash C}\,[\backslash_e]}{}$$

- hypotheses A, B are in the right premise of $\backslash_e$:

$$\dfrac{\dfrac{\Gamma \vdash A \odot B \qquad \dfrac{\Delta \vdash D \quad A, B \vdash D \setminus C}{\Delta, A, B \vdash C}\,[\backslash_e]}{\Delta, \Gamma \vdash C}\,[\odot_e]}{} \quad \Rightarrow \quad \dfrac{\Delta \vdash D \qquad \dfrac{\dfrac{\Gamma \vdash A \odot B \quad A, B \vdash D \setminus C}{\Gamma \vdash D \setminus C}\,[\odot_e]}{\Delta, \Gamma \vdash C}\,[\backslash_e]}{}$$

- R' is $/_e$ — symmetrical to the previous case.
- R' is $\backslash_i$:

$$\dfrac{\Gamma \vdash A \odot B \qquad \dfrac{D, \Delta, A, B, \Delta' \vdash C}{\Delta, A, B, \Delta' \vdash D \setminus C}\,[\backslash_i]}{\Delta, \Gamma, \Delta' \vdash D \setminus C}\,[\odot_e]$$

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

- R' is $/_i$ — symmetrical to the previous case.

- R' is $\odot_e$:

  - hypotheses A, B are in the left premise of R':

$$\dfrac{\Gamma \vdash A \odot B \quad \dfrac{\Delta, A, B, \Delta' \vdash C \odot D \quad \Phi, C, D, \Phi' \vdash E}{\Phi, \Delta, A, B, \Delta', \Phi' \vdash E}\,[\odot_e]}{\Phi, \Delta, \Gamma, \Delta', \Phi' \vdash E}\,[\odot_e]$$

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

  - hypotheses A, B are in the right premise of R':

$$\dfrac{\Gamma \vdash A \odot B \quad \dfrac{\Delta \vdash C \odot D \quad \Phi, A, B, C, D, \Phi' \vdash E}{\Phi, A, B, \Delta, \Phi' \vdash E}\,[\odot_e]}{\Phi, \Gamma, \Delta, \Phi' \vdash E}\,[\odot_e]$$

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

- R' is $\odot_i$:

  - hypotheses A, B are in the left premise of R':

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

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

- hypotheses in the right premise of R':

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

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

All possible cases of combinations of rules have been examined. The product elimination has the ability to swing over any rule provided the cancelled hypotheses are in the same premise. □

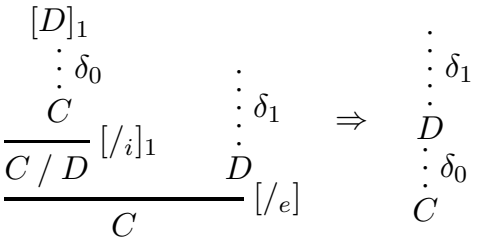
Definition 1. *Let R be a product elimination rule:*

$$\dfrac{\begin{array}{cc} \vdots\,\delta_0 & \dfrac{\begin{array}{cc} \vdots\,\delta_2 & \vdots\,\delta_1 \\ \Gamma_2 \vdash C_2 & \Gamma_1 \vdash C_1 \end{array}}{\Gamma[A, B] \vdash C}\,R' \\ \Delta \vdash A \odot B & \end{array}}{\Gamma[\Delta] \vdash C}\,R$$

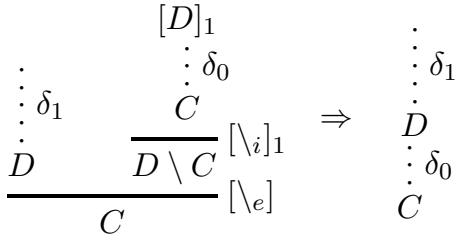
*A $\odot_e$ rule R cancelling the two hypotheses A and B is said to be **as high as possible** if R' is binary, and A and B are not in the same sequent, i.e. A is in Γ_2 and B is in Γ_1.*

As usual, a **redex** consists in an introduction rule of a given connective immediately followed by the elimination rule of the same connective. Hence, in this calculus, there are four *redexes*: one for /, one for \ and two for $\odot$ (depending in which premise it takes place). The reductions patterns are given below. For simplicity we only write the conclusions of the sequents and leave out the contexts: this is unambiguous given that contexts are plain sequences of formulae.

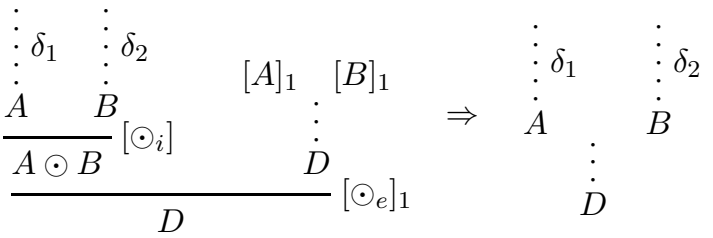
- Redex$_/$: $/_i$ immediately followed by $/_e$.



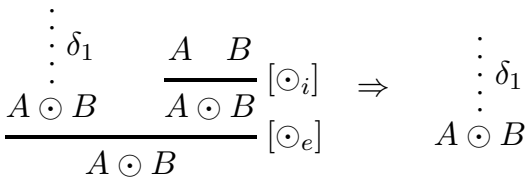
- Redex$_{\backslash}$: $/_i$ immediately followed by $/_e$: symmetrical.



- Redex$_{\odot}$: introduction $\odot_i$ (left) immediately followed by elimination $\odot_e$.



- Redex$_{\odot}$: introduction $\odot_i$ (right) immediately followed by elimination $\odot_e$.



From the notion of redex, we define a generalisation that we call a **k-extended-redex**.

Definition 2. *Every path of a principal branch $B(S_0)$ of length k from S_0 to S_n with $|S_0|^r = |S_n|^r$ (the conclusions of those two sequents are the same), such that $|S_0|$ is the conclusion of an elimination rule R_e and S_n is the conclusion of an introduction rule R_i of the same connective, is called a* **k-*extended-redex*.** *Note that* 0-extended-redexes *are* redexes.

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

Proof. Assume that one of the occurrences of X results from a $/_e$ rule between $X \,/\, U$, and U. In this case, the k-extended-redex contains a smaller k'-extended-redex, between this elimination and same the introduction rule. Otherwise only $\odot_e$ rules preserve the conclusion of the sequent, and they can be used an unspecified number of times without changing the conclusion of the sequent (*i.e.* X). □

4.2 Normalisation of $\mathbf{L}_\odot$

Definition 3. *A **normal proof** is a proof which contains no k-extended-redexes and where all $\odot_e$ rules are as high as possible.*

Given a proof δ, and $PER(\delta)$ its $\odot_e$ rules, we define for a product elimination rule R in $PE(\delta)$ the two following integers:

1. $g(R)$ which is k if there is a k-extended-redex in the principal branch $B(S_0)$ starting from the conclusion of R and 0 otherwise.

2. the integer $d_{conj}(R)$ is the number of rules other than product elimination rules (*i.e.* not in $PER(\delta)$) between R and the rule which gathers in the same sequent the hypotheses A and B that are cancelled by R.

With those measures, we define the measure of a proof that will be used to established the normalisation. It is a lexically ordered triple of integers:

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

where:

- $n(\delta)$ is the number of rules of δ. The number of rules decreases when one reduces a redex as in any linear calculus.

- $h(\delta) = \sum_{R \in PER(\delta)} d_{conj}(R)$ when PER is the set of Proudct Elimination rules. The number $h(\delta)$ is 0 when all product elimination rules are as high as possible (cf. definition 1).

- $g(\delta) = \min_{R \in PER(\delta)}(g(R))$ which is 0 if and only if δ contains no k-extended-redex (cf. definition 3).

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

Proof. Let δ a proof of $L_{\odot}$,

- $h(\delta)$ is there to make sure that $\odot_e$ rule reach their highest possible position. During this phase, k-extended-redexes$_{/}$ and $_{\backslash}$ may appear and are reduced afterwards. If every $\odot_e$ is as high as possible then $h(\delta) = 0$. Only $\odot$ k-extended-redexes may still exist in δ. This case is presented in example 1 of Figure 4.
- $g(\delta)$ represents the number of rules in a $\odot$ k-extended-redex. When it is zero, there is no more k-extended-redex$\odot$ in δ. This case is presented in example 2 of Figure 4.

□

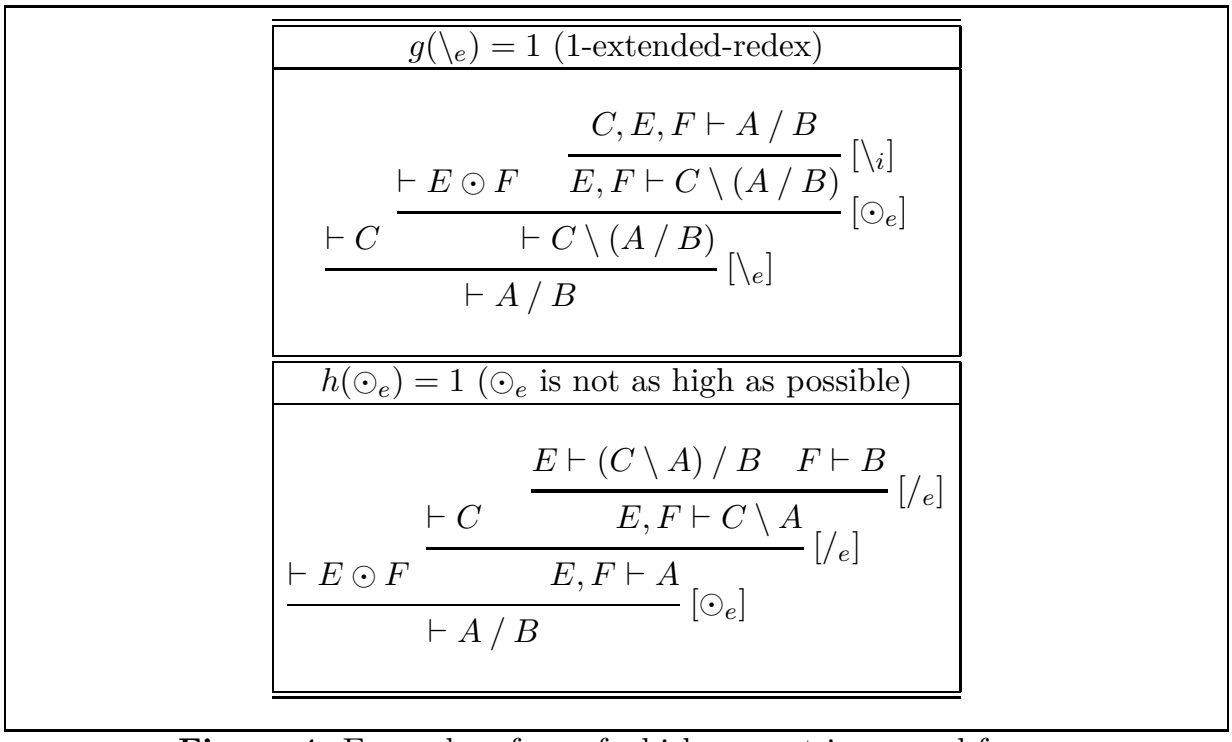


Figure 4: Examples of proof which are not in normal form

Theorem 1. *Every proof δ in $L_{\odot}$ calculus can be turned into a normal form which is unique.*

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 $|\delta| <= \langle r, d, g\rangle$, let us show that δ has a unique normal form as well.

1. If δ contains a redex, we can reduce it, and let δ', be the reduced proof. We have $n(\delta') < n(\delta)$, hence $|\delta'| < \langle r, d, g\rangle$ and by induction, δ' has a unique normal form, and therefore so does δ.

2. Else, if δ contains no redex,

   (a) If $d \neq 0$: let R be the lowest $\odot_e$ rule such that $d(R) \neq 0$. Hence, there exists a rule $R' \neq \odot_e$ higher than R, and R can be lifted above all $\odot_e$ and finally R can swing over R'. The induced proof δ' is such that $n(\delta') = n(\delta)$ and $h(\delta') = h(\delta) - 1$. The number $d_{conj}(R_i)$, for an $\odot_e$ rule R_i below R, is zero because R does not contribute to $d_{conj}(\_)$. Therefore $\delta' < \langle r, d, g\rangle$, and by induction , δ' has a unique normal form, hence so does δ.

   (b) Else:

      i. If $g \neq 0$: let R' such that $g(R') = g$. This rule can swing over its left premise. The number of rules and the sum remain unchanged. In its part, the places of the $\odot_e$ rules are just exchanged, hence g decrease of 1. Hence the proof δ' is such that $|\delta'| < |\delta|$. By induction, δ' has a unique normal form, hence so does δ.

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

Consequently an $\odot_e$ rule R cancelling two hypotheses A and B may only appear below the binary rule R_b which gathers the two hypothesis A and B in one sequent. Moreover, only one $\odot_e$ can be "as high as possible", *i.e.* immediately below the binary rule R_b. Indeed, an $\odot_e$ rule can only cancel two adjacent free hypotheses: the rightmost hypothesis of the left premise of R_b and the leftmost hypothesis of the right premise os R_b. We thus assign a *unique* position to each $\odot_e$ rule, and therefore the normal form is unique.

□

All proofs have a unique normal form which can be computed using the aforementioned strategy. The normal forms of the two previous examples, figure 4, are the two following proofs:

4.3 Subformula property for $\mathbf{L}_{\odot}$

Theorem 2. *Any normal proof δ of the Lambek calculus with product of a sequent $\Gamma \vdash C$ satisfies the subformula property: every formula in δ is a subformula of some hypothesis in Γ or of the conclusion C.*

Proof. Here we prove a stronger result than the plain subformula property:

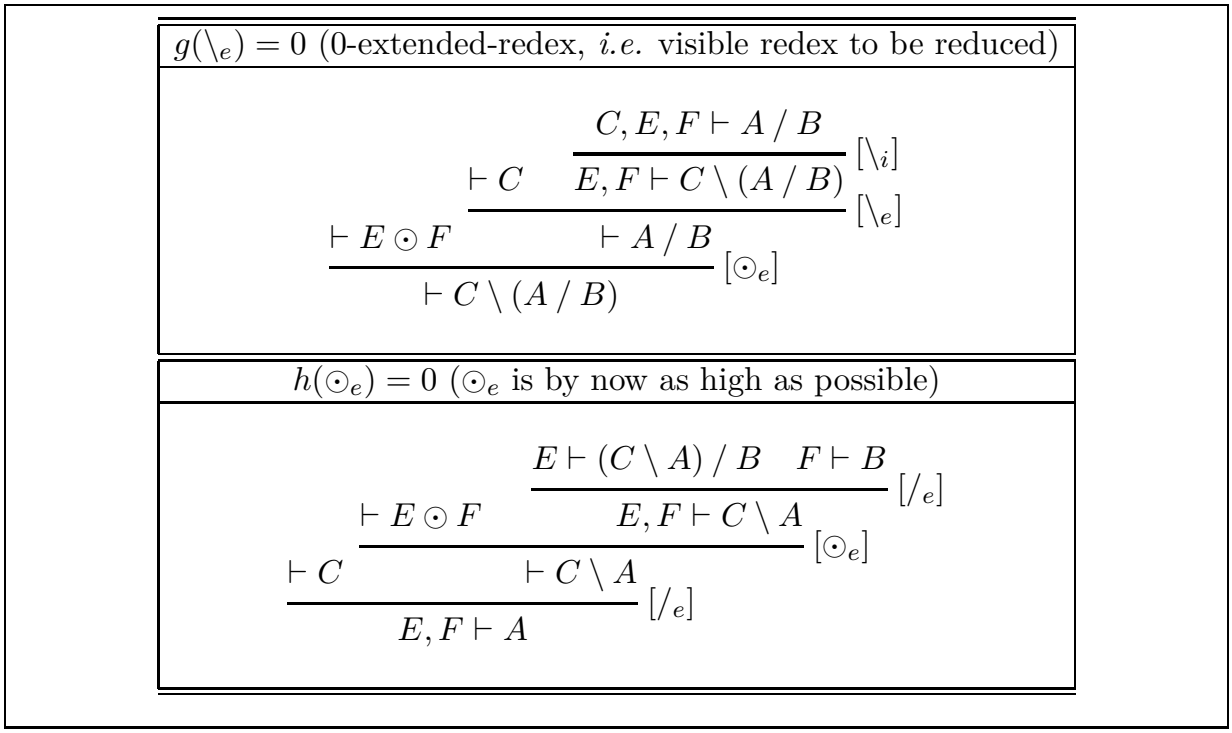


Figure 5: Applying a generalised reduction step to the examples of figure 4

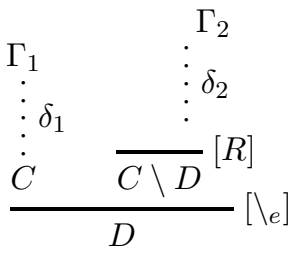
1. every formula in a normal sub-proof is a formula of some hypotheses or of the conclusion of the proof ;

2. and if the last rule used is an $\backslash_e$ or $/_e$ every subformula is a subformulae of some (uncancelled) hypothesis.

We proceed by a standard induction on the height of the proof, according to the nature of the last rule:

1. A proof consisting in an axiom clearly enjoys the two properties.

2. If the last rule R is $\backslash_e$: let δ be the following proof, where Γ_i is the set of hypotheses used in the sub-proof δ_i, for $i \in [1, 2]$:



Using the induction hypothesis:

- In δ_1 every formula is subformula of C or Γ_1;
- In δ_2 every formula is subformula of $C \setminus D$ or Γ_2.

The conclusion D and the premise C are direct subformulae of the premise $C \setminus D$. We have to consider 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 subformula of Γ_2. Then every formula of δ is a subformula of Γ_2.
- if R is $\setminus_i$: it is impossible because the rule should be a 0-extended-redex, while δ is in normal form.
- if R is $/_i$: this case is structurally impossible because this rule cannot derive $C \setminus D$.
- if R is $\odot_i$: this case is also impossible because this rule cannot derive $C \setminus D$.
- if R is $\odot_e$. Once more, it depends on the rule R' above R:

$$\dfrac{\Gamma_1 \quad \vdots\ \delta_1 \quad C \qquad \dfrac{A \odot B \qquad \dfrac{\Gamma_2[A,B] \quad \vdots\ \delta_2 \quad C \setminus D}{C \setminus D}\,[R']}{C \setminus D}\,[\odot_e]}{D}\,[\setminus_e]$$

  - If R' is $\setminus_e$ or $/_e$, by the induction hypothesis, $C \setminus D$ is a subformula of some hypotheses.
  - If R' is $\setminus_i$: impossible because it would result in a 1-extended-redex, while δ is in normal form.
  - If R' is one of the other introduction rules ($\setminus_i$ or $\odot_i$): these cases are structurally impossible since these rules cannot derive $C \setminus D$.
  - If R' is $\odot_e$, once more, we must consult the rule above. As the number of rules above a given rule is finite, the proof contains a sequence of rules that necessarily matches the following pattern:

$$\dfrac{\Gamma_1 \quad \dfrac{A_1 \odot B_1 \qquad \dfrac{A_n \odot B_n \qquad \dfrac{}{C \setminus D}[R''] \quad \Gamma_2[A_1, \cdots, A_n, B_n, \cdots, B_1] \\ \vdots \delta_2 \\ C \setminus D}{C \setminus D}[\odot_e] \\ \vdots \delta_1 \\ C \qquad\qquad C \setminus D}{D}[\setminus_e]$$

Then one of the following case applies:

- There are only $\odot_e$ rules in this sequence of rules, and $C \setminus D$ is one of the hypotheses.
- Otherwise the path stops on a rule R^n which according to what we said about R' above, can only be $/_e$ or $\setminus_e$: therefore, by induction hypothesis $C \setminus D$ is subformula of one of the hypotheses.

In every case, the conclusion of $\setminus_e$ is a subformula of the hypotheses and the property holds.

3. R is $/_e$: this case is similar to R is $\setminus_e$.

4. R is $\setminus_i$: let δ be the following proof, where Γ_1 is the set of hypotheses used in the sub-proof δ_1:

$$\dfrac{\begin{array}{c} C, \Gamma_1 \\ \vdots \delta_1 \\ D \end{array}}{C \setminus D}[\setminus_i]$$

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

5. R is $/_i$: is symmetrical to the previous case.

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

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots\,\delta_1 & \vdots\,\delta_2 \\ C & D\end{array}}{C \odot D}\,[\odot_i]$$

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

Furthermore, C and D are subformulae of $C \odot D$. Hence, every formula of δ is subformula of Γ_1, Γ_2 or of $C \odot D$.

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

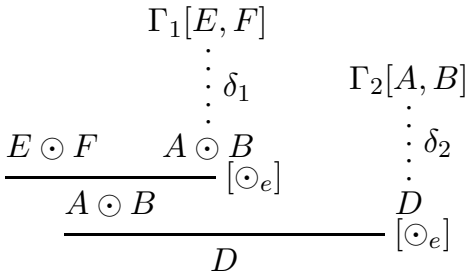
$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots\,\delta_1 & \vdots\,\delta_2 \\ A \odot B & D\end{array}}{D}\,[\odot_e]$$

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

The conclusion of δ is the conclusion of one premise, hence the property holds for the part of the proof which the conclusion belongs to, *i.e.* δ_2 and we only have to show that formulae in δ_1 are subformulae of a conclusion or of an hypothesis of δ.

Let us show that $A \odot B$ is a subformula of an hypothesis of Γ_1, which entails the result. What may be the rule R above $A \odot B$?

- if R is $\backslash_e$ or $/_e$, because of the induction hypothesis, and because $A \odot B$ is the conclusion of such a rule, $A \odot B$ is a subformula of Γ_1.
- if R is $\backslash_i$ or $/_i$: this case cannot happen because $A \odot B$ may not be a possible conclusion of those rules.
- if R is $\odot_i$: this case cannot happen: there would exist a 0-extended-redex, *i.e.* a redex and this impossible in a normal proof.
- if R is another $\odot_e$ rule:



There are two cases:

- If $A \odot B$ can be traced up to an hypothesis of Γ_1, then $A \odot B$ is a subformula of some hypothesis (itself).
- Otherwise, $A \odot B$ is not an hypothesis, but there exists a rule $\odot_i$ above it which generated the formula $A \odot B$. This $\odot_i$ introduction rule is conjoined to the $\odot_e$ rule under discussion, and the proof would contain a k-extended-redex: this case is ruled out.

In any case, $A \odot B$ is subformula of some hypothesis and the subformula property holds for $\odot_e$.

□

Thus in $L_\odot$, every proof have a unique normal form which satisfies the subformula property. We observe that unlike [18], rules use are the usual ones for this calculus.

5 Normalisation of proofs of pcIMLL

Now, we present a notion of normal proof with the subformula property, and a normalisation algorithm for proofs of pcIMLL.

As in the previous section about $L_\odot$, the normalisation assigns a unique place to the eliminations of non-commutative product, and build sequence of commutative product eliminations. Nevertheless the relative position of each rule in a sequence of elimination rules for commutative product is free, hence not unique unless we accept n-ary rules.

5.1 Some properties of pcIMLL

Property 3 (product elimination rules can swing over any other rule)**.** *Let R be a product elimination rule $\otimes_e$ (resp. $\odot_e$) yielding $\Gamma[\Delta] \vdash C$ from a proof δ_0 of $\Delta \vdash A \odot B$ and a proof of $\Gamma[A,B] \vdash C$ obtained by a rule R' from a proof δ_1 of*

$\Theta[A, B] \vdash X$ *(resp.* $\Gamma[\langle A; B\rangle] \vdash C$*) and from a second proof* δ_2 *of* $\Psi \vdash U$ *— if* R' *is a binary rule.*

A proof for the same sequent $\Gamma[\Delta] \vdash C$ *can be derived by the following two steps:*

1. *apply* R *between the proof* δ_0 *of* $\Delta \vdash A \odot B$ *and the proof* δ_1 *of* $\Theta[A, B] \vdash X$ *(resp.* $\Theta[\langle A; B\rangle] \vdash X$*) as conclusion, yielding* $\Theta[\Delta] \vdash X$*;*

2. *apply* R' *to this new proof and to the proof* δ_2*.*

In other words, provided that the hypotheses A, B *that are cancelled by the* R *product elimination rule are in the same premise of the* R' *rule, the* R *product elimination rule can swing over* R'*, as shown in figure 6.*

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

Figure 6: The product elimination R swings over a rule R' in PCIMLL.

Proof. The proof is similar to the one of property 1. This is a case study according to the rule over the product elimination. This elimination rule can only float up when the hypotheses that must be cancelled are in the same premise and occupy the proper respective position required by the elimination rule.

Let us check that $\otimes_e$ may swing over any other rule R':

- R' is $\backslash_e$:
  - if the hypotheses to be cancelled are in the left premise of $\backslash_e$:

$$\dfrac{\dfrac{\Delta \vdash A \otimes B \quad \dfrac{\Gamma[(A,B)] \vdash D \quad \Phi \vdash D \setminus C}{\langle \Gamma[(A,B)]; \Phi\rangle \vdash C}\,[\backslash_e]}{\langle \Gamma[\Delta]; \Phi\rangle \vdash C}\,[\otimes_e]$$

$$\Rightarrow \quad \dfrac{\dfrac{\Delta \vdash A \otimes B \quad \Gamma[(A,B)] \vdash D}{\Gamma[\Delta] \vdash D}\,[\otimes_e] \quad \Phi \vdash D \setminus C}{\langle \Gamma[\Delta]; \Phi\rangle \vdash C}\,[\backslash_e]$$

- if the hypotheses to be cancelled are in the right premise of $\backslash_e$:

$$\dfrac{\Delta \vdash A \otimes B \qquad \dfrac{\Gamma \vdash D \quad \Phi[(A,B)] \vdash D \setminus C}{\langle \Gamma; \Phi[(A,B)] \rangle \vdash C}\,[\backslash_e]}{\langle \Gamma; \Phi[\Delta] \rangle \vdash C}\,[\otimes_e]$$

$$\Rightarrow \quad \dfrac{\Gamma \vdash D \qquad \dfrac{\Delta \vdash A \otimes B \quad \Phi[(A,B)] \vdash D \setminus C}{\Phi[\Delta] \vdash D \setminus C}\,[\otimes_e]}{\langle \Gamma; \Phi[\Delta] \rangle \vdash C}\,[\backslash_e]$$

- R' is $/_e$ — symmetrical to the previous case.
- R' is $\multimap_e$:

  - if the hypotheses to be cancelled are in the left premise of $\multimap_e$:

$$\dfrac{\Delta \vdash A \otimes B \qquad \dfrac{\Gamma[(A,B)] \vdash D \quad \Phi \vdash D \multimap C}{(\Gamma[(A,B)], \Phi) \vdash C}\,[\multimap_e]}{(\Gamma[\Delta]; \Phi) \vdash C}\,[\otimes_e]$$

$$\Rightarrow \quad \dfrac{\dfrac{\Delta \vdash A \otimes B \quad \Gamma[(A,B)] \vdash D}{\Gamma[\Delta] \vdash D}\,[\otimes_e] \qquad \Phi \vdash D \multimap C}{(\Gamma[\Delta], \Phi) \vdash C}\,[\multimap_e]$$

  - if the hypotheses to be cancelled are in the right premise of $\multimap_e$:

$$\dfrac{\Delta \vdash A \otimes B \qquad \dfrac{\Gamma \vdash D \quad \Phi[(A,B)] \vdash D \multimap C}{(\Gamma, \Phi[(A,B)]) \vdash C}\,[\multimap_e]}{(\Gamma, \Phi[\Delta]) \vdash C}\,[\otimes_e]$$

$$\Rightarrow \quad \dfrac{\Gamma \vdash D \qquad \dfrac{\Delta \vdash A \otimes B \quad \Phi[(A,B)] \vdash D \multimap C}{\Phi[\Delta] \vdash D \multimap C}\,[\otimes_e]}{(\Gamma, \Phi[\Delta]) \vdash C}\,[\multimap_e]$$

- R' is $/_i$:

$$\dfrac{\Delta \vdash A \otimes B \quad \dfrac{\langle \Gamma[(A,B)]; D\rangle \vdash C}{\Gamma[(A,B)] \vdash C \,/\, D}\,[/_i]}{\Gamma[\Delta] \vdash C \,/\, D}\,[\otimes_e] \Rightarrow \dfrac{\dfrac{\Delta \vdash A \otimes B \quad \langle \Gamma[(A,B)]; D\rangle \vdash C}{\langle \Gamma[\Delta]; D\rangle \vdash C}\,[\otimes_e]}{\Gamma[\Delta] \vdash C \,/\, D}\,[/_i]$$

- R' is $\backslash_i$ — symmetrical to the previous case.

- R' is $\multimap_i$:

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

- R' is $\otimes_e$ (as R):

  - if the hypotheses to be cancelled are in the right premise of $\otimes_e$:

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

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

  - if the hypotheses to be cancelled are in the left premise of $\otimes_e$:

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

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

- R' is $\odot_e$:
  - if the hypotheses to be cancelled are in the right premise of $\odot_e$:

$$\dfrac{\Gamma \vdash A \otimes B \quad \dfrac{\Delta \vdash C \odot D \quad (\Phi, (A, B), \Psi, \langle C; D\rangle, \Psi', \Phi') \vdash E}{(\Phi, (A, B), \Psi, \Delta, \Psi', \Phi') \vdash E}\,[\odot_e]}{(\Phi, \Gamma, \Psi, \Delta, \Psi', \Phi') \vdash E}\,[\otimes_e]$$

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

  - if the hypotheses to be cancelled are in the left premise of $\odot_e$:

$$\dfrac{\Gamma \vdash A \otimes B \quad \dfrac{(\Delta, (A, B), \Delta') \vdash C \odot D \quad (\Phi, \Psi, \langle C; D\rangle, \Psi', \Phi') \vdash E}{(\Phi, \Psi, \Delta, (A, B), \Delta', \Psi', \Phi') \vdash E}\,[\odot_e]}{(\Phi, \Psi, \Delta, \Gamma, \Delta', \Psi', \Phi') \vdash E}\,[\otimes_e]$$

$$\Rightarrow$$

$$\dfrac{\dfrac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C \odot D}{(\Delta, \Gamma, \Delta') \vdash C \odot D}\,[\otimes_e] \quad (\Phi, \Psi, \langle C; D\rangle, \Psi', \Phi') \vdash E}{(\Phi, \Psi, \Delta, \Gamma, \Delta', \Psi', \Phi') \vdash E}\,[\odot_e]$$

- R' is $\otimes_i$:
  - if the hypotheses to be cancelled are in the left premise of $\otimes_i$:

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

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

- if the hypotheses to be cancelled are in the right premise of $\otimes_i$:

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

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

- R' is $\odot_i$:

  - if the hypotheses to be cancelled are in the left premise of $\odot_i$:

$$\dfrac{\dfrac{(\Delta, (A, B), \Delta') \vdash C \quad \Phi \vdash D}{\langle (\Delta, (A, B), \Delta'); \Phi \rangle \vdash C \odot D}\,[\odot_i] \quad \Gamma \vdash A \otimes B}{\langle (\Delta, \Gamma, \Delta'); \Phi \rangle \vdash C \odot D}\,[\otimes_e]$$

$$\Rightarrow \dfrac{\dfrac{\Gamma \vdash A \otimes B \quad (\Delta, (A, B), \Delta') \vdash C}{(\Delta, \Gamma, \Delta') \vdash C}\,[\otimes_e] \qquad \Phi \vdash D}{\langle (\Delta, \Gamma, \Delta'); \Phi \rangle \vdash C \odot D}\,[\odot_i]$$

  - if the hypotheses to be cancelled are in the right premise of $\odot_i$:

$$\dfrac{\Gamma \vdash A \otimes B \quad \dfrac{\Delta \vdash C \quad (\Phi, (A, B), \Phi') \vdash D}{\langle \Delta; (\Phi, (A, B), \Phi') \rangle \vdash C \odot D}\,[\odot_i]}{\langle \Delta; (\Phi, \Gamma, \Phi') \rangle \vdash C \odot D}\,[\otimes_e]$$

$$\Rightarrow \dfrac{\Delta \vdash C \qquad \dfrac{\Gamma \vdash A \otimes B \quad (\Phi, (A, B), \Phi') \vdash D}{(\Phi, \Gamma, \Phi') \vdash D}\,[\otimes_e]}{\langle \Delta; (\Phi, \Gamma, \Phi') \rangle \vdash C \odot D}\,[\odot_i]$$

- R' is entropy $\sqsubset$:

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

The elimination of the non-commutative product may swing over any rule. Most cases are easy adaptation of the cases in the proof of the property 1 (and almost similar to the $\otimes_e$ that we exhaustively presented). Let us nevertheless show how $R = \odot_e$ can swings over R' when R' is entropy ($\sqsubset$), since this case did not occur in the proof of property 1.

$$\dfrac{\dfrac{\Gamma \vdash A \otimes B}{\Gamma' \vdash A \otimes B}\,[\sqsubset] \qquad \Delta[\langle A;B\rangle] \vdash D}{\Delta[\Gamma'] \vdash D}\,[\odot_e] \quad \Rightarrow \quad \dfrac{\Gamma \vdash A \otimes B \quad \Delta[\langle A;B\rangle] \vdash D}{\dfrac{\Delta[\Gamma] \vdash D}{\Delta[\Gamma'] \vdash D}\,[\sqsubset]}\,[\odot_e]$$

□

The procedure for turning a proof into a normal proof is analogous to the one for $\mathrm{L}_{\odot}$. To do so, we firstly introduce the **redexes** of this calculus and generalise the notion of k-extended redex.

The 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):

- Redex$_{/}$: $/_i$ immediately followed by $/_e$.

$$\dfrac{\dfrac{\vdots}{\langle \Gamma; D\rangle \vdash C}\,[/_i] \qquad \vdots\ \delta_1}{\dfrac{\Gamma \vdash C \,/\, D \qquad \Delta \vdash D}{\langle \Gamma;\Delta\rangle \vdash C}\,[/_e]} \quad \Rightarrow \quad \begin{array}{c} \vdots \\ \delta_1 \\ \vdots \\ \Delta \vdash D \\ \vdots \\ \langle \Gamma;\Delta\rangle \vdash C \end{array}$$

- Redex$_{\backslash}$: $\backslash_i$ immediately followed by $\backslash_e$.

$$\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \vdots \\ \Delta \vdash D & \dfrac{\langle D;\Gamma\rangle \vdash C}{\Gamma \vdash D \setminus C}\,[\setminus_i]\end{array}}{\langle \Delta;\Gamma\rangle \vdash C}\,[\setminus_e] \quad \Rightarrow \quad \begin{array}{c}\vdots\,\delta_1 \\ \Delta \vdash D \\ \vdots \\ \langle \Delta;\Gamma\rangle \vdash C\end{array}$$

- Redex$_{\multimap}$: $\multimap_i$ immediately followed by $\multimap_e$.

$$\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \vdots \\ \Delta \vdash D & \dfrac{(D,\Gamma) \vdash C}{\Gamma \vdash D \multimap C}\,[\multimap_i]\end{array}}{(\Delta,\Gamma) \vdash C}\,[\multimap_e] \quad \Rightarrow \quad \begin{array}{c}\vdots\,\delta_1 \\ \Delta \vdash D \\ \vdots \\ (\Delta,\Gamma) \vdash C\end{array}$$

- Redex$_{\odot}$: $\odot_i$ immediately followed by $\odot_e$ on the left.

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \vdots\,\delta_2 \\ \Delta_1 \vdash A & \Delta_2 \vdash B\end{array}}{\langle \Delta_1;\Delta_2\rangle \vdash A \odot B}\,[\odot_i] \qquad \vdots \\ \Gamma[\langle A;B\rangle] \vdash D}{\Gamma[\langle \Delta_1;\Delta_2\rangle] \vdash D}\,[\odot_e] \quad \Rightarrow \quad \begin{array}{c}\vdots\,\delta_1 \quad \vdots\,\delta_2 \\ \Gamma[\langle\ A\ ;\ B\ \rangle] \vdash D\end{array}$$

- Redex$_{\odot}$: $\odot_i$ immediately followed by $\odot_e$ on the right.

$$\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \dfrac{A \vdash A \quad B \vdash B}{\langle A;B\rangle \vdash A \odot B}\,[\odot_i] \\ \Gamma \vdash A \odot B & \end{array}}{\Gamma \vdash A \odot B}\,[\odot_e] \Rightarrow \quad \begin{array}{c}\vdots\,\delta_1 \\ \Gamma \vdash A \odot B\end{array}$$

- Redex$_{\otimes}$: $\otimes_i$ immediately followed by $\otimes_e$ on the left.

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \vdots\,\delta_2 \\ A & B\end{array}}{A \otimes B}\,[\otimes_i] \qquad \begin{array}{c}A\ \ B \\ \vdots \\ D\end{array}}{D}\,[\otimes_e] \quad \Rightarrow \quad \begin{array}{c}\vdots\,\delta_1 \quad \vdots\,\delta_2 \\ \Gamma[(\ A\ ,\ B\ )] \vdash D\end{array}$$

- Redex$_{\otimes}$: $\otimes_i$ immediately followed by $\otimes_e$ on the right.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \delta_1\\ \vdots\\ \Gamma \vdash A \otimes B\end{array} \qquad \dfrac{A \vdash A \quad B \vdash B}{(A,B) \vdash A \otimes B}\,[\otimes_i]}{\Gamma \vdash A \otimes B}\,[\otimes_e]}{} \quad \Rightarrow \quad \begin{array}{c}\vdots\\ \delta_1\\ \vdots\\ \Gamma \vdash A \otimes B\end{array}$$

Here as well, we consider k-extended-redexes, defined as in definition 2:

Definition 4. *Every path of a principal branch $B(S_0)$ of length k — counting every rule, including entropy rules if any — from S_0 to S_n with $|S_0|^r = |S_n|^r$ (i.e. the conclusions of those two sequents are the same), such that $|S_0|$ is the conclusion of an elimination rule R_e and S_n is the conclusion of an introduction rule R_i of the same connective, is called a* ***k-extended-redex****. Note that* 0-extended-redexes *are* redexes.

Definition 5. *A proof is said to be in* ***normal form*** *whenever it does not contain any k-extended-redexes, $\forall k \in \mathbb{N}$.*

5.2 Normalisation of pcIMLL

A proof is in **normal form** if it does not contain any k-extended-redex.

As we did for $L_{\odot}$, we define the three components of the measure to be used for proving normalisation.

1. Given an *implication elimination* rule R ($\backslash_e$, $/_e$ or $\multimap_e$) with conclusion S_0 the integer $e(R)$ is k if there is a k-extended-redex in $B(S_0)$ (called an implication k-extended-redex over R), and 0 otherwise — the k-rules may include entropy rules.

2. Given a *product elimination* rule R ($\odot_e$ or $\otimes_e$) with S_0 as conclusion, the integer $g(R)$ is k if there is a k-extended-redex in $B(S_0)$ (called a product k-extended-redex over R) and 0 otherwise — the k-rules may include entropy rules.

We introduce the size $|\delta|$ of a proof δ as a triple of integers, with the lexicographic order:

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

where:

- $r(\delta)$ is simply the number of rules in δ — the number of rules decreases when one reduces a redex, in pcIMLL as in any linear calculus.

- $e(\delta) = \min_{R \in IER(\delta)}(e(R))$ or 0 when there is no implication k-extended-redex, where IER is the set of Implication Elimination Rules.
- $g(\delta) = \min_{R \in PER(\delta)}(g(R))$ or 0 when there is no product k-extended-redex where PER is the set of Product Elimination Rules.

Property 4. *A proof δ is normal if and only if it does not contain* 0*-extended-redex, and $e(\delta) = g(\delta) = 0$.*

Proof. Let δ a proof of pcIMLL.

- $e(\delta)$ is the minimal distance between the introduction rule and the elimination rule of an implication k-extended-redex ($\backslash$, / or $\multimap$). If its value is zero, while there is no redex that can be reduced, then there is no implication k-extended-redex.
- $g(\delta)$ is the distance between the introduction rule and the elimination rule of a product k-extended-redex ($\odot$ or $\otimes$). If its value is zero, while there is no redex that can be reduced, then there is no product k-extended-redex.

Note that the two other redexes could only be 0-extended-redexes. Thus a proof without 0-extended redex and such that $e(\delta) = g(\delta) = 0$ is in normal form. The figure 7 shows example of proofs which are not in normal form. □

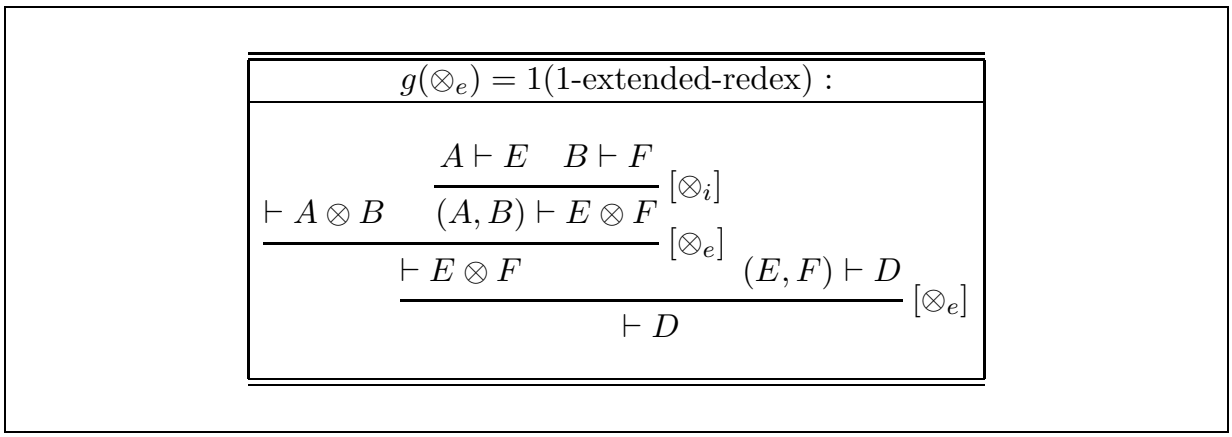


Figure 7: Yet another example of a non normal proof in pcIMLL — see figure 4 for other examples

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

Proof. Let δ a proof in normal form, and let us consider the principal branch starting with the conclusion.

Any minimal k-extended-redex has the following structure:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{X}\,[introduction]}{\vdots\,\delta_3}}{U}}{U \,/\, A}\,[/_i]}{\vdots\,\delta_2}$$
$$\dfrac{U \,/\, A \qquad\qquad A}{U}\,[/_e]$$
$$\dfrac{\vdots\,\delta_1}{X}$$
$$\dfrac{}{}\,[elimination]$$

Then, we define:

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

The formula U is the conclusion of the highest implication elimination rule. For this derivation, the number of symbols in U is greater than the number of symbols in X.

Then, still following the principal branch, δ_2 is a sequence of product eliminations and entropy.

The formula $U \,/\, A$ can only result from an introduction rule.

On the example, 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, $k > k'$. □

Here is a consequence of the previous property:

Lemma 1. *In the proof δ, a minimal k-extended-redex (whose size is $e(\delta)$) only contains product elimination rules and entropy.*

Proof. If the k-extended-redex is minimal, with the property 5, it does not contain any elimination rule. Moreover, if we do not use elimination rules, the number of symbols in the formula could not decrease. Hence it must be constant in the k-extended-redex. In this case, the rules that we could use are rules whose conclusion is one of the premises. Therefore the sequence of rules may only contain product elimination rules and entropy rules: $\odot_e, \otimes_e$ or $\sqsubset$. □

Property 6. *Product eliminations and entropy rules may swing under implication elimination rules:*

Let R a product elimination rule $\otimes_e$ (resp. $\odot_e$) yielding $\Gamma[\Delta] \vdash C$ from a proof δ_0 of $\Delta \vdash A \otimes B$ and a proof δ_1 of $\Gamma[(A, B)] \vdash C$ (resp. $\Gamma[\langle A; B\rangle] \vdash C$). Assume that the result is merged with a proof δ_2 of $\Theta \vdash U$ via an implication elimination rule $R' \setminus_e$ (resp. $/_e$, $\multimap_e$) yielding $\langle \Theta; \Gamma[\Delta]\rangle \vdash V$ if R' is $\setminus_e$ (resp. $\langle \Gamma[\Delta]; \Theta\rangle \vdash V$ if R', $(\Gamma[\Delta], \Theta) \vdash V$). Figure 8 presents the case where R' is $\setminus_e$.

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 of $\Theta \vdash U$ and the proof δ_1 of $\Gamma[(A, B)] \vdash X$ (resp. $\Gamma[\langle A; B\rangle] \vdash X$) yielding $\langle \Theta; \Gamma[(A, B)]\rangle \vdash V$ (resp. $\langle \Gamma[\langle A; B\rangle]; \Theta\rangle \vdash V$ 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 $(\Theta, \Gamma[\Delta]) \vdash V$).

$$\dfrac{\dfrac{\vdots\,\delta_2}{\Theta \vdash U} \quad \dfrac{\dfrac{\vdots\,\delta_0}{\Delta \vdash A \otimes B} \quad \dfrac{\vdots\,\delta_1}{\Gamma[(A;B)] \vdash C}\,R}{\Gamma[\Delta] \vdash C}\,[\setminus_e]}{\langle \Theta; \Gamma[\Delta]\rangle \vdash V} \quad \Rightarrow \quad \dfrac{\dfrac{\vdots\,\delta_0}{\Delta \vdash A \otimes B} \quad \dfrac{\dfrac{\vdots\,\delta_2}{\Theta \vdash U} \quad \dfrac{\vdots\,\delta_1}{\Gamma[(A;B)] \vdash C}\,[\setminus_e]}{\langle \Theta; \Gamma[(A;B)]\rangle \vdash V}\,R}{\langle \Theta; \Gamma[\Delta]\rangle \vdash V}$$

Figure 8: The product elimination rule R swings under the rule $\setminus_e$ in pcIMLL.

Proof. Implication eliminations do not modify the order between formulae of the same premise and do modify hypotheses of the premises. Product elimination and entropy rules do not modify the conclusions of the premises but only their hypotheses. Consequently those rule do not interact and may commute. □

Theorem 3. *Every proof δ of pc*IMLL *has a normal form.*

Proof. We proceed by induction on the size of the proof. Let δ be a proof with $|\delta| = \langle n, e, g\rangle$. By induction, we may assume that every proof δ' of measure $|\delta'| < \langle n, e, g\rangle$ has a normal form.

If δ contains a redex, the reduction of 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 hypothesis δ' has a normal form, hence δ has one too.

From now on, we can assume without loss of generality that δ has no redex.

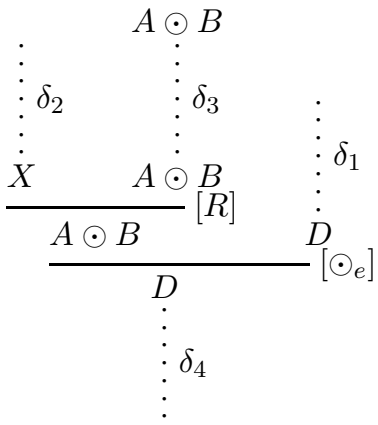
If $e(\delta) \neq 0$, then there is an implicative elimination rule S such that S is in a $e(\delta)$-extended-redex. As an $e(\delta)$-extended-redex is minimal, the property 1 implies that it contains only product elimination and entropy rules. Moreover, the property 6 allows to swing S over the rule above it (a product elimination rule or an entropy rules swings under an implication elimination rule). The proof obtained δ' is such that $n(\delta') = n(\delta)$ and $e(\delta') = e(\delta) - 1 < e(\delta)$. By induction hypothesis δ' has a normal form and hence δ has one too.

From now on, we can assume without loss of generality that $e(\delta) = 0$

If $g(\delta) \neq 0$: then there exists a product elimination rule R such that R is in a $g(\delta)$-extended-redex. In this case, δ does not contain any implication extended-redex and $g(R)$ is minimal, hence the extended redex only contains product elimination rule and entropy rules. So R can swing over its left premise (where the rule conjoined $\otimes_i$ is) and over product elimination rule and entropy rule (as property 3 shows). Thus we turned δ into a proof δ' with $n(\delta') = n(\delta)$.

To apply the induction hypothesis we need the size of δ' to be less than the size of δ, and the only thing to check is that no k-extended-redex based on a rule of $IER(\delta)$ appears:

If δ looks like:

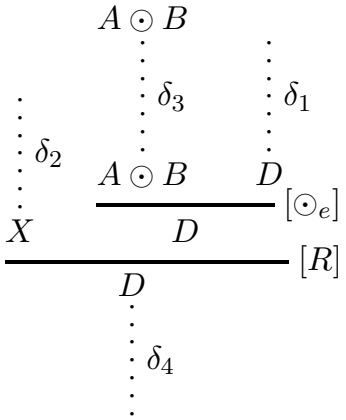


- every principal branch in δ_3 followed by δ_4 does not contain extended-redex because δ_3 is in the left part of $\odot_e$;

- every principal branch in δ_1 followed by δ_4 may contain extended-redexes;

- every principal branch in δ_2 followed by 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':



In this new proof:

- every principal branch in δ_3 followed by δ_4 does not contain extended-redexes because δ_3 is in the left part of $\odot_e$;

- every principal branch in δ_1 followed by δ_4 does not contain new extended-redexes, and the measure of these extended-redexes is decremented by 1;

- every principal branch in δ_2 followed by 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 any new k-extended-redex ; in particular the proof does not contain any new implication k-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 therefore δ also has one.

If none of the previous transformations applies we have $e(\delta) = g(\delta) = 0$, and therefore, because of property 4, δ is in normal form.

□

Figure 9 is the normal form of the example from figure 7, obtained by following the procedure described in the proof above.

Now, let us establish that proofs in normal form enjoy the subformula property.

$$\begin{array}{|c|}\hline g(\otimes_e) = 0(\text{no more } k\text{-extended-redex}) : \\ \hline \\ \dfrac{\vdash A \otimes B \qquad \dfrac{\dfrac{A \vdash E \quad B \vdash F}{(A,B) \vdash E \otimes F}\,[\otimes_i] \qquad (E,F) \vdash D}{(A,B) \vdash D}\,[\otimes_e]}{\vdash D}\,[\otimes_e] \\ \\ \hline\end{array}$$

Figure 9: Normal proof for the proof in figure 7

5.3 Subformula property for pcIMLL

Theorem 4. *The subformula property holds for pc*IMLL*: in a normal proof δ of a sequent $\Gamma \vdash C$, every formula of a sequent is a subformula of some hypothesis (Γ) or of the conclusion (C).*

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

1. every formula in a normal proof is subformula of some hypotheses or of the conclusion of the proof;

2. if the last rule used is an implicative elimination $\backslash_e$, $/_e$ or $\multimap_e$ every subformula is a subformula of some hypothesis.

Axioms enjoy the subformula property.

When the last rule is an entropy rule, the subformula property holds simply because of the induction hypothesis. Indeed, the formulae of a sequent are preserved under entropy rule which only affects the order on the formulae.

Let us call R^* the last rule of the proof.

1. R^* is $\backslash_e$:

$$\dfrac{\begin{array}{cc}\Delta_1 & \Gamma_2 \\ \vdots\,\delta_1 & \vdots\,\delta_2 \\ \Delta \vdash C & \Gamma \vdash C \backslash D\end{array}}{\langle \Delta ; \Gamma \rangle \vdash D}\,[\backslash_e]$$

By induction hypothesis, every formula in δ_1 is a subformula of some hypothesis in Δ_1 or of the conclusion C. In addition every formula in δ_2 is a subformula of Γ_2 or of the conclusion $C \setminus D$. However C is a subformula of $C \setminus D$ and D too. Let us show that $C \setminus D$ is subformula of some hypothesis in δ_2.

Let us look at the rule R' that yields $C \setminus D$:

- if R' is $\setminus_i$: this may not happen because it would be a 0-extended-redex while the proof is in normal form.
- if R' is $/_i$, $\multimap_i$, $\otimes_i$ or $\odot_i$: these cases are structurally impossible because they can not produce $C \setminus D$.
- if R' is $\setminus_e$, $/_e$ or $\multimap_e$: we use the induction hypothesis and $C \setminus D$ is a subformula of Γ_2.
- if R' is $\otimes_e$, $\odot_e$ or entropy: they preserve the conclusion, thus we have to investigate what the rule above can be:

  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. Those rules preserve the conclusion and therefore $C \setminus D$ is one of the hypothesis in Γ_2.

2. R^* is $/_e$ or $\multimap_e$: similar to $\setminus_e$ above.

3. R^* is $\multimap_i$, let δ be the following proof:

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\,\delta_1\\ \langle \Gamma; C\rangle \vdash D\end{array}}{\Gamma \vdash C \multimap D}\,[\multimap_i]$$

   By induction hypothesis, every formula in δ_1 is a subformula of some hypothesis in Γ or of the conclusion D. The formula D is a subformula of $D \multimap C$, hence every formula of δ is a subformula of some hypothesis Γ or of the conclusion $D \multimap C$. The property holds for δ.

4. R^* is $\setminus_i$ or $/_i$ — similar to the previous case.

5. R^* is $\otimes_i$: let δ be the following proof:

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

- every formula in δ_1 is a subformula of some hypothesis in Δ_1 or of the conclusion C.
- every formula in δ_2 is a subformula of some hypothesis in Γ_2 or of the conclusion D.
- however C and D are themselves subformulae of $C \otimes D$, then in Γ, every formula is a subformula of some hypotheses in Δ and Γ or of the conclusion $C \otimes D$.

6. R^* is $\odot_i$: similar to the previous case.

7. R^* is $\otimes_e$:

$$\dfrac{\begin{array}{cc}\Delta_1 & \Gamma_2 \\ \vdots\,\delta_1 & \vdots\,\delta_2 \\ \Delta \vdash A \otimes B & \Gamma[A,B] \vdash D\end{array}}{\Gamma[\Delta] \vdash D}\,[\otimes_e]$$

- every formula of δ_1 is a subformula of some hypothesis in Δ_1 or of the conclusion $A \otimes B$.
- every formula of δ_2 is a subformula of some hypothesis in Γ_2 or of the conclusion D.
- moreover, D is the conclusion of δ. Thus, every formula of δ_2 is a subformula of some hypothesis in Γ_2 or of the conclusion of the proof δ: D.

In order to prove that the property holds for the other part of the proof, we must prove that $A \otimes B$ is a subformula of some hypothesis in δ_1. Let us look at the rule R' above:

- if R' is $\backslash_e$, $/_e$ or $\multimap_e$, using the induction hypothesis $A \otimes B$ is subformula of hypotheses Δ_1.
- if R' is $\otimes_i$: this case is impossible because there cannot be any 0-extended-redex in a normal proof.
- if R' is $\backslash_i$, $/_i$, $\multimap_i$ or $\odot_i$: these cases are structurally impossible because these rules cannot produce $A \otimes B$.
- if R' is $\otimes_e$, $\odot_e$ or entropy which preserve the conclusion of the proof, let us analyse the rule above:

- either it is one of the previous rules, thus, using the same arguments we conclude.
- either, given that a proof only contains a finite number of rules, the sequence of such rules is finite. Givven that 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 subformula of hypotheses.

8. R^* is $\odot_i$: similar to the previous case.

In pcIMLL, all proofs have a normal form which enjoys the subformula property. □

6 Decidability

An immediate but interesting consequence of normalisation with a subformula property is the following:

Theorem 5. *The provability of a sequent in pc*IMLL *is decidable, and in $L_\odot$ as well.*

Proof. Because of normalisation, one only has to look for normal proofs. Given that normal proofs enjoy the subformula property it is enough to try the *finite* number of rules that are possible. There are finitely many rules, and each of them may only lead to try to prove a finite number of sequents because of the subformula property. By considering principal branches, premises of these rules are sequents that have less connectives. Therefore an easy induction shows that the calculus is decidable. For more details, see the proof of decidability for product free Lambek calculus based on natural deduction in [17]. □

7 Conclusion

With concurrency and linguistics motivations, we defined pcIMLL in natural deduction and proved normalisation. For Lambek calculus with product, a subcalculus of pcIMLL, we also characterised the unique normal form.

As a perspective, we look forward a proof net syntax for pcIMLL. This would also allow to easily compute lambda terms (that are semantic reading in linguistic applications). Although related systems do have proof nets (MLL, Lambek calculus, NL of Abrusci and Ruet) there is not yet any proof net calculus for pcIMLL. The present work on natural deduction can be viewed as a first step in this direction.

We avoided tricky details and discussions about the uniqueness of the normal form for pcIMLL. Let us say it can be achieved if one consider as equivalent proofs

that only differ because of the relative order of several commutative product elimination rules in a sequence of product eliminations that are just below the rule which gathers the cancelled hypotheses.

With respect to computational linguistic application, we look forward a simpler translation from pcIMLL formulae to arrow types on e and t and thus from parse structures that are pcIMLL deduction to intuitionistic deductions, which are semantic readings. This is related to the interpretation of noun phrase and generalised quantifiers as the combination of the categories k (case) and d (entities).

References

[1] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[2] 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.

[3] V. Michele Abrusci and Paul Ruet. Non-commutative logic I: The multiplicative fragment. *Annals of pure and applied logic*, 101(1):29–64, 1999.

[4] Maxime Amblard. *Calcul de représentations sémantiques et suntaxe générative: les grammaires minimalistes catégorielles*. PhD thesis, université de Bordeaux 1, 2007.

[5] Maxime Amblard. Encoding Phases using Commutativity and Non-commutativity in a Logical Framework. In Sylvain Pogodalla and Jean-Philippe Prost, editors, *Logical Aspect of Computational Linguistic*, volume 6736 of *Lecture Notes in Computer Science*, pages 1–16, Montpellier, France, June 2011. Springer.

[6] Maxime Amblard, Alain Lecomte, and Christian Retoré. Categorial minimalist grammars: From generative grammar to logical form. *Linguistic Analysis*, 36(1–4):273–306, 2010. Festschrift on the occasion of Jim Lambek's 85th birthday.

[7] 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.

[8] 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.

[9] Samuel Epstein and Robert Berwick. On the convergence of 'minimalist' syntax and categorial grammar. In A. Nijholt, G. Scollo, and R. Steetkamp, editors, *Algebraic Methods in Language Processing*. Universiteit Twente, 1995.

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

[11] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Number 7 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1988.

[12] Alessio Guglielmi. A system of interaction and structure. *ACM Trans. Comput. Logic*, 8(1), January 2007.

[13] J.. S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and computation*, pages 327–365, 1994.

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

[15] 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.

[16] Marcel Masseron, Christophe Tollu, and Jacqueline Vauzeilles. Generating plans in linear logic: I. actions as proofs. *Theoretical Computer Science*, 113:349–370, 1993.

[17] Richard Moot and Christian Retoré. *The logic of categorial grammars: a deductive account of natural language syntax and semantics*, volume 6850 of *LNCS*. Springer, 2012. `http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-31554-1`.

[18] Sara Negri. A normalizing system of natural deduction for intuitionistic linear logic. *Archive for Mathematical Logic*, 2002.

[19] Christian Retoré. *Réseaux et Séquents Ordonnés*. Thèse de Doctorat, spécialité Mathématiques, Université Paris 7, février 1993.

[20] 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.

[21] 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.

[22] Christian Retoré and Edward Stabler. Generative grammar in resource logics. *Research on Language and Computation*, 2(1):3–25, 2004. Introductory survey for the special issue on *Resource logics and minimalist grammars*.

[23] Paul Ruet. *Logique non-commutative et programmation concurrente*. Thèse de doctorat, spécialité logique et fondements de l'informatique, Université Paris 7, 1997.

[24] Edward Stabler. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics, LACL'96*, volume 1328 of *LNCS/LNAI*, pages 68–95. Springer-Verlag, 1997.

 Received January 2014