In: *Formal Grammar*,
Barcelona, August 95.

Pomset logic as an alternative categorial grammar

Alain Lecomte* Christian Retoré
lecomte@shm.grenet.fr retore@loria.fr

INRIA-Lorraine & CRIN-C.N.R.S.
615, rue du jardin botanique
B.P. 101
F-54 602 Villers lès Nancy cedex FRANCE

Abstract: Lambek calculus may be viewed as a fragment of linear logic, namely intuitionistic non-commutative multiplicative linear logic. As it is too restrictive to describe numerous usual linguistic phenomena, instead of extending it we extend MLL with a non-commutative connective, thus dealing with partially ordered multisets of formulae. Relying on proof net technique, our study associates words with parts of proofs, modules, and parsing is described as proving by plugging modules. Apart from avoiding spurious ambiguities, our method succeeds in obtaining a logical description of relatively free word order, head-wrapping, clitics, and extraposition (these latest two constructions are unfortunately not included, for lack of space).

*and Université Pierre Mendès-France, Grenoble

0 Introduction

Classical systems for grammatical deduction (**NL**, **L**, **LP** and so on, see for instance [MO94, MK94] deal with sequents relating two sequences of formulae, each of them built by a constructor (the comma) which is associative or not, commutative or not. This gives raise to systems dealing with lists (**L**), multisets (**LP**), or binary trees (**NL**). Flexibility is added by the use of the so-called structural modalities [Moo94, Mor94].

In this paper we aim at introducing a formalism which directly processes partially ordered multisets. This formalism, which is called POMset Logic, was introduced in [Ret93], and was rather thought as a link between concurrency and linear logic [Ret93, Ret95, Asp91, AD94].

Such a calculus shares many properties with ordinary (commutative) Linear Logic [Gir87, Tro92, Gir95] and of course with Lambek Calculus [Lam58] (this latter is nothing else than Intuitionistic Multiplicative Non Commutative Linear Logic without negation [Abr91, Abr94]). Among these good logical properties, let us quote the cut-elimination theorem, strong normalisation and confluence, a proof-net syntax and a coherent semantics.

But it is not a conservative extension of the Lambek calculus: it does not make possible to define directed implications, as the order involved is rather a temporal order. Therefore we have to describe the rather spatial word order by other means.

This leads to a system where words and expressions behave like parts of proofs, i.e. modules, which are plugged together either directly or by using cuts. Such a technique, first developed in [Gir86] is very related to linear logic as a logic of communication.

When eliminating these cuts, a net is obtained, which exactly expresses what must be the correct word order. These properties are here applied to linguistic constructions which usually hardly fit in the categorial grammar setting.

As examples, we expose our description of relatively free word order, head-wrapping (see the example of the French negation) and but, due to lack of space, we skipped cliticisation and extra position, which, however, simply extends the solution for head wrapping represented by French negation.

1 Pomset Logic

Before getting into technicalities of proof net syntax, we give the flavour of pomset logic. For more details, one should have a look at [Ret93, Ret94, Ret95]

1.1 Overview

1.1.1 Origin: coherence spaces

As said above, the basic idea was to introduce non-commutative features in MLL. So we had a look at coherence semantics, the closest denotational semantics for linear logic.

In this setting we observed that there just exists one non-commutative multiplicative connective that we christened `before` . Within this semantics the generalisation of `before` led us to consider partially ordered product of coherence spaces.

Thus our goal was to investigate the possibility of a logical calculus involving POMsets of formulae. However without extra structure on the sequents, there are just the two usual multiplicative conjunction and disjunction, hence it is a fairly sensible solution. [Ret93, Ret94].

We decided to first look at proof net syntax, because of its close relation to coherence spaces, as explained in [Ret94].

1.1.2 Structure and properties of POMset Logic

The language we consider is defined from atomic formulae, with the following connectives: (unary) negation $(...)^{\perp}$, (binary) conjunction $... \otimes ...$, (binary) disjunction $...\wp...$, and (binary) `before` $... < ...$.

We consider formulae up to de Morgan laws, which extend to this language as follows:

$$\begin{array}{rcll} (A^{\perp})^{\perp} & \equiv & A & \\ (A \wp B)^{\perp} & \equiv & A^{\perp} \otimes B^{\perp} & \\ (A < B)^{\perp} & \equiv & A^{\perp} < B^{\perp} & \text{and not } B^{\perp} < A^{\perp} \\ (A \otimes B)^{\perp} & \equiv & A^{\perp} \wp B^{\perp} & \text{from } 1^{st} \text{ and } 2^{nd} \text{ lines} \end{array}$$

As said above, the conclusion of a proof is a partially ordered multiset of such formulae. We write

$$\vdash A_1, ..., A_n[\mathfrak{i}]$$

for a multiset $\{A_1, ...A_n\}$ partially ordered by $\mathfrak{i}$.

We did not yet say a word about cuts: as usual in linear logic, a cut may be viewed as a conclusion of the shape $A \otimes A^{\perp}$, which behaves like $\exists X. X \otimes X^{\perp} \equiv \perp$ (*the unit*). Here, we shall keep a track of the cuts with a symbol $\bullet$, so we actually deal with sequents:

$$\vdash A_1, ..., A_n, \bullet_1, ..., \bullet_p[\mathfrak{i}]$$

Why such a trick? it is very useful that the partial order $\mathfrak{i}$ involves the cuts, while this trick is known to be harmless.

Although the algebraic properties should be derived from the syntax, we gave them right now to fix the ideas.

Properties:[Ret93, Ret95]

- $\otimes, <, \wp$ are all associative,
- $\wp$ and $\otimes$ are commutative, but not $<$: no relation between $A < B$ and $B < A$ in general.
- $<$ is self-dual: $(A < B)^{\perp} \equiv A^{\perp} < B^{\perp}$
- $<$, with respect to linear implication defined by $A \multimap B \equiv A^{\perp} \wp B$, lies in between $\wp$ and $\otimes$:[1].

  $$A \otimes B \multimap A < B \qquad \text{and} \qquad A < B \multimap A \wp B$$
- one does not have, in general $A^{\perp} < A$, but whenever $A^{\perp} < B$ and $B^{\perp} < C$ then $A < C$.

1.1.3 Intuitive meaning of pomset logic

In the *proof-as-programs* paradigm (see e.g. [dG95, Abr93]), where the cuts are the computations to be performed, we have a partial order on cuts. This partial order may thus be viewed as a strategy for computing the proof net, which is described within the syntax itself. This is a concurrent strategy, which simply consists in first evaluating the cuts (i.e. the computations to be performed) according to this order. In this setting, `before` corresponds to sequential composition, a cut between $A \wp B$ and $A^{\perp} \otimes B^{\perp}$ reduces into a cut between A and $A^{\perp}$ and a cut between B and $B^{\perp}$ that can be done in *parallel*, while a cut between $A < B$ and $A^{\perp} < B^{\perp}$reduces into

[1] we are working with the MIX rule [FR94].

a cut between A and $A^\perp$ which is to be computed first and a cut between B and $B^\perp$ which is to be computed next.

When thinking of *proofs-as-processes*, as in [Abr93, Asp91, AD94], where $\otimes$ is internal choice and $\wp$ parallel composition, $<$ corresponds to sequential composition too.

Finally, when looking at *proof-search-as-computation*, the connective $<$ also corresponds to sequential composition, as noticed in [Gug94].

Even though the interpretation of $\otimes$ and $\wp$ is changing according to the considered setting, one can harmlessly think of $<$ as sequential composition, i.e. as expressing a partial temporal order.

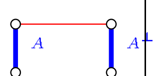
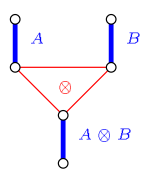
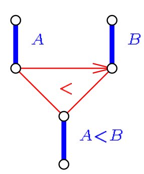
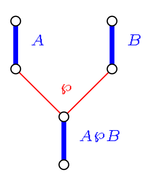
1.2 Pomset proof nets

1.2.1 Defining proof nets

We use standard vocabulary from graph theory, one can refer e.g. to [Ber73].

Definition 1 (BR-graphs) *We deal with edge coloured graphs, the colour of an edge being either blue (bold,* B*) or red (regular,* R*). The blue/bold edges, or* B*-edges are undirected. They correspond to formulae and define a perfect matching of the graph. The red/regular edges, or* R*-edges, may be undirected or directed in which case we call them* R*-arcs. They correspond to connections between formulae. Such graphs will be called* BR*-graphs.*

A link is one of the following BR-graph

Links			
Name	Graph	Conclusions	Premises
Axiom		A and $A^\perp$	none
Tensor		$A \otimes B$	A and B
Before		$A<B$	A and B
Par		$A\wp B$	A and B

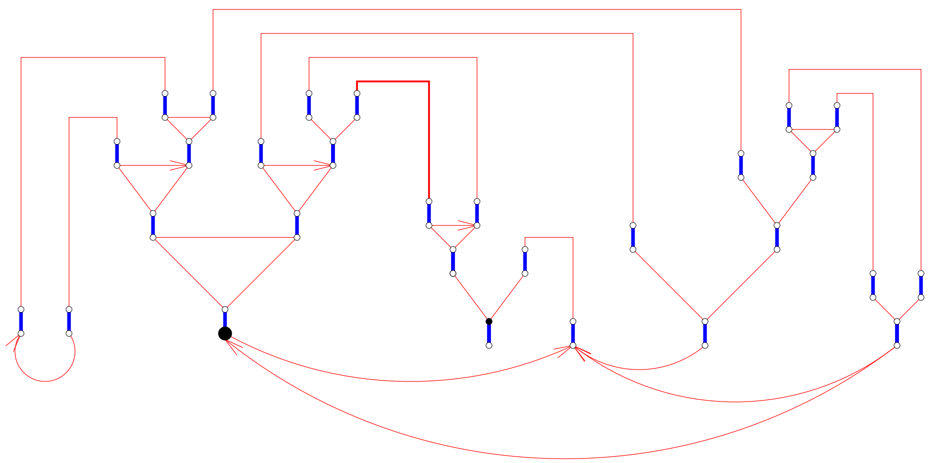
Definition 2 (proof structure) *Let BR be a* BR*-graph whose* B*-edges are labelled with formulae. A link of BR is a link which is a full subgraph of BR. An ordered proof structure consists in*

- *A* BR*-graph such that any* B*-edge is the conclusion of exactly one link, and the premise of at most one link, and the formulae (*B*-edges) which are not the the premise of any link are called conclusions of the proof structure.*

- *A set of* R*-arcs between (the bottom side) of the conclusions which defines a strict partial order (transitive and anti-reflexive relation).*

- *A subset of the conclusions which are the conclusion of a* `times` *link whose premises are dual: we mark them with a* $\bullet$*, and call them cuts.*

Notice that only the label of the axiom links'conclusions (or leaves) are needed. Here is an example, without the formulae, for readibility:



Now, as usual, not any proof structure correspond to proofs, but only the proof nets:

Definition 3 (proof net) *A proof net is a proof structure which contains no* alternate elementary *(Æ) circuit.*

Notice the Æ-paths do not compose.

Beware that the adjective "elementary" is necessary. "elementary" is equivalent to "simple" in the case of alternate paths because the B-edges are a matching.

The above example is a proof net but if the final link above the sixth conclusion (from left to right) was a `before` link instead of a `par` link, it would not be a proof net.

Although simple to prove, the next proposition shows that this proof net syntax is a sensible syntax by itself, even without the corresponding sequent calculus:

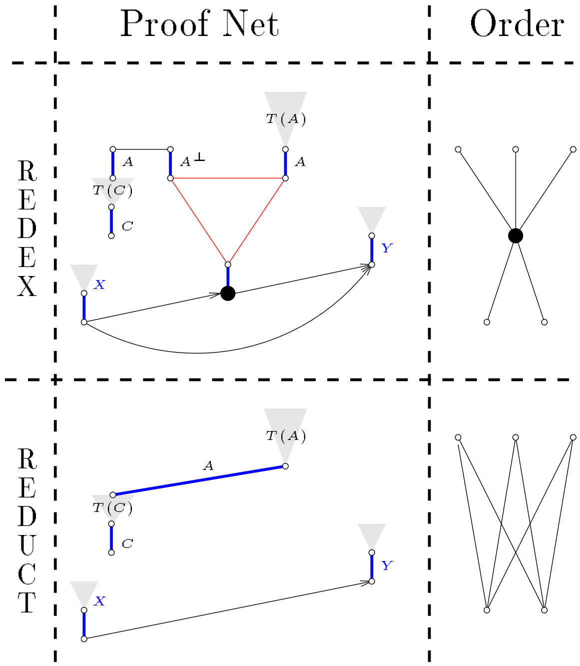
Proposition 4 *It is at most cubic to check whether a proof structure is a proof net.*

Proof: Assume we are given a proof structure, and two of its vertices X and Y. Checking whether there exist an Æ-path from X to Y starting with its unique incident B-edge is a standard breadth search algorithm. Each B-edge is visited once in each direction, and it is thus quadratic in twice the number of B-edges, i.e. in the number of vertices (the B-edges are a perfect matching of the graph). If we repeat this for any vertex X and $Y = X$ we get a cubic algorithm which checks the absence or presence of Æ-circuit. $\diamond$

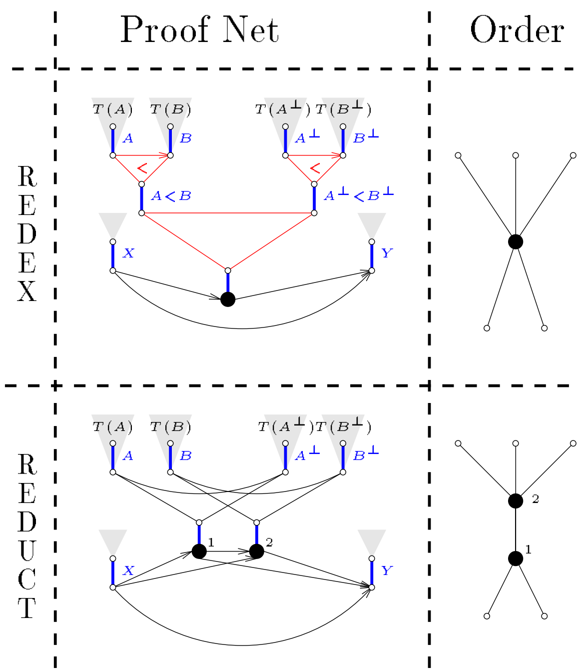
1.2.2 Cut elimination

Let us now turn our attention to cut elimination, which is a local graph rewriting turning a proof net into a proof net, in such a way that *the restriction of the order to the conclusions is preserved under cut elimination.*

There are three elementary steps of cut elimination to be described:

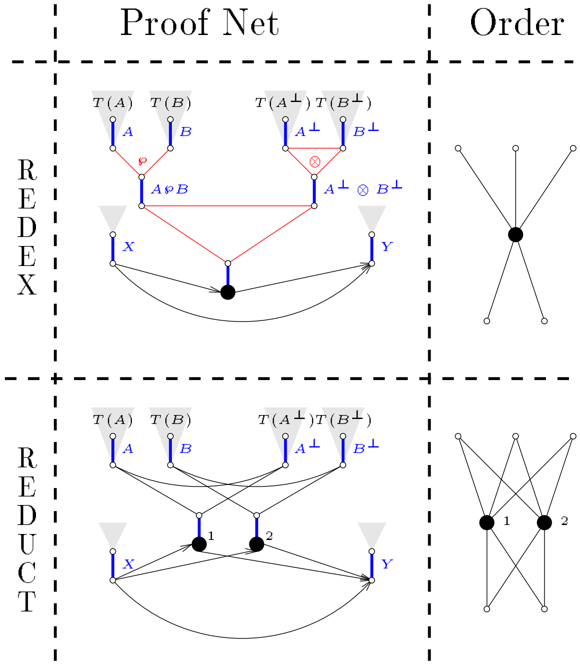


The order on the remaining cuts and conclusions barely is the restriction of the original order.



The order on the remaining conclusions and cuts is *the* order satisfying:

- on conclusions and cuts different from $\bullet_1$ (resp. $\bullet_2$) it is exactly the original order replacing $\bullet_1$ (resp. $\bullet_2$) with $\bullet$.
- $\bullet_1 < \bullet_2$



The order on the remaining conclusions and cuts is *the* order satisfying:

- on conclusions and cuts different from $\bullet_1$ (resp. $\bullet_2$) it is exactly the original order replacing $\bullet_1$ (resp. $\bullet_2$) with $\bullet$.

- neither $\bullet_1 < \bullet_2$ nor $\bullet_2 < \bullet_1$

In [Ret93, Ret95] it is shown that these local transformations turn a proof net into a proof net (i.e. no Æ-circuit appear), and that this calculus enjoys the following

Theorem 5 (strong normalisation and confluence) *The calculus of ordered proof nets enjoys strong normalisation and confluence: a proof net with conclusions and cuts $F_1, \ldots, F_n, \bullet_1, \ldots, \bullet_p$ ordered by* i *reduces to a cut free proof net with conclusions $F_1, \ldots, F_n$ ordered by* $\text{i}|_{F_1,\ldots,F_n}$.

Let us also say that this proof net syntax is sound and in some sense complete with respect to coherence semantics [Ret94].

1.3 Linguistic remarks on POMset logic

1.3.1 Relation with the Lambek calculus

After introducing a somewhat complicated calculus, it is right to ask whether $A^\perp < B$ is any different from Lambek's $A\backslash B$, i.e. from Abrusci's $A \multimap B \equiv A^\perp \wp B$.

Firstly, we do not have in general $A^\perp < A$, which is very coherent with the temporal interpretation: $A^\perp$ and A can not match, because they live in a different instant.

Secondly, define $A\dot{/}B \equiv A{<}B^\perp$, and $A\dot{\backslash}B \equiv A^\perp < B$. Then we do not have $(A\dot{/}B)\dot{/}C = A\dot{/}(C{\otimes}B)$ like in the Lambek calculus, but, because of associativity and self duality:
$$(A\dot{/}B)\dot{/}C = A\dot{/}(B < C)$$
This formula and many others are hardly understandable from Lambek calculus viewpoint.

This already stresses the difference from the Lambek calculus, and next comes a similar remark stated on linguistic ground.

1.3.2 The fail of a too naïve linguistic model

The guidelines provided by the Lambek calculus viewed as a syntactic theory leads to the following idea. Take the basic sentence *"Peter loves Mary"*. Following the

Lambek calculus, one could assume $Peter{:}np$, $Mary{:}np$ and $loves{:}np^\perp{<}s{<}np^\perp$ and try to derive s. Unfortunately we are thus unable to express the word order as an order between the conclusions, and to prove a sequent
$\vdash\ Peter{:}np^\perp, Mary{:}np^\perp, loves{:}np{<}s^\perp{<}np, s[\mathtt{i}]$
with $\mathtt{i} \supset \{(Peter{:}np^\perp, loves{:}np{<}s^\perp{<}np)(loves{:}np{<}s^\perp{<}np, Mary{:}np^\perp)\}$. A proof net with such conclusions always contains Æ-circuit, e.g. the one containing the axiom between the $np^\perp$ coming from *Peter* and the first np in *loves*, and the arrow $(Peter{:}np^\perp, loves{:}np{<}s^\perp{<}np)$ of the order.

But if we do not write any order, we are able to derive, as well *Peter Mary loves*.

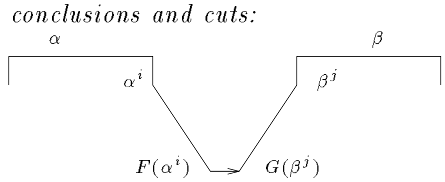
So if the spatial order has to do with the temporal order expressed by `before`, it is in a more subtle way. Thus, looking forward a richer setting we thought of associating an incomplete proof to each word. Then we noticed that a proof induces a partial order on its axioms, and that having an axiom in the part of a proof, could solve our problems. But the existence of such an order must first be established.

1.3.3 The solution: the order on axioms

Notation 6 *Let α be an axiom link of a proof net. We write $\{\alpha^1, \alpha^2\}$ for its two* B*-edges or formulae; thus $(\alpha^1)^\perp = \alpha^2$.*

Upper indices $i, j, k, l...$ range over $\{1, 2\}$.

Definition 7 *Given two axiom links α and β of an ordered proof net we say that $\alpha \lhd \beta$ iff there exists $i, j \in \{1, 2\}$ and an* R*-arc from a formula $F(\alpha^i)$ to a formula $G(\beta^j)$ in the proof net, this* R*-arc either belonging to a* `before` *-link or to the order on conclusions and cuts:*



We write $\blacktriangleleft$ for the transitive closure of $\lhd$.

Definition 8 *An* Æ*-path of a proof net Π is said to be a* **before-only** Æ*-path whenever it remains an* Æ*-path in the proof net Π' obtained by replacing $\otimes$-links with $\wp$-links.*

Remark 9 *If $\alpha \lhd \beta$ because of a formula $F(\alpha^i) < G(\beta^j)$ then its defines a before-only* Æ*-path from α^i to β^j.*

We have the following lemma, whose proof is in appendix:

Lemma 10 *Let α and β be axioms such that $\alpha \blacktriangleleft \beta$. Let n be the minimal integer such that there exists axioms $\alpha_p, 1 \leq p \leq n$ with:*

$$\alpha = \alpha_0 \lhd ....\alpha_{n-1} \lhd \alpha_n = \beta$$

Then we have:

1. $\alpha_p^{i(p+1)} = (\alpha_p^{j(p)})^\perp$
2. *the union of the before-only paths corresponding to $\alpha_p \lhd \alpha_{p+1}$ is itself a before-only path.*
3. $\alpha^i \neq \alpha^j$ *when* $i \neq j$

Proposition 11 *The relation $\blacktriangleleft$ is a partial order on the axioms of a proof net.*

Proof: As it is transitive by definition, we just need to check that it is anti-reflexive. If we had $\alpha \blacktriangleleft \alpha$ for some axiom α, then, by previous lemma, there would exists a before only path from α to itself, not containing twice the same axiom: this is impossible, as the axiom α appears twice in such a path. $\diamond$

2 The linguistic model based on POMset logic and an example of relatively free word order

For linguistic purposes the language will include as atomic propositions: lexical categories, and a special constant S. In this section the material is minimal, but gives the flavour of the next one on simpler constructions.

2.1 Words as modules, composition as plugging

Definition 12 *A module is defined as a proof net, in particular they contain no* Æ*-circuit. The only difference lies in the definition of a proof structure: in a module some formulae or* B*-edges are the conclusion of no link. These formulae are called the hypotheses of the module.*

The lexicon provide each word or expression with a module: this module possesses a main formula, which is the underlined one in pictures. The type of the word, which is closed to his Lambek type, is the dual of this main formula. The module contains axioms, whose two conclusions are labelled either by the word or by a word appearing in the expression, or by a variable.

The type of the word is of the shape $g \otimes (\ldots g^{\perp} \ldots.)$, where g is lexical category, like verb, adjective ... as soon as the word has a functional behaviour.

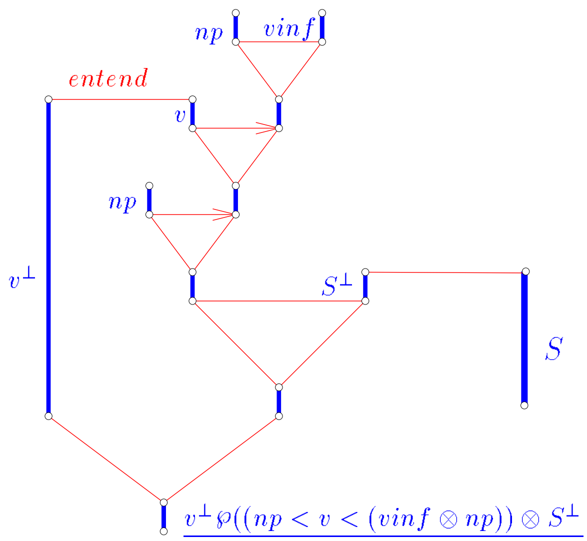
Here are word examples:

$Pierre : np$

$Marie : np$

$chanter : vinf$

$entend : v \otimes ((np < v < (np \otimes vinf)) \multimap S) \equiv v \otimes ((np^{\perp} < v^{\perp} < (np^{\perp} \wp vinf^{\perp})) \wp S)$



Remark 13 *This example is here to be a simple one, but it thus suffers from too particular properties:*

- *we do not really need the $vinf$ atomic proposition, this is just for simplicity, the solution for incorporating morphological information will be describe below.*
- *although modules can play totally symmetric rôle, this example gives the impression that functionality is leaded by one of the constituents while it is not always the case, see below.*

But this example is convenient to illustrate the following properties, and to compute all possible pluggings.

Definition 14 *The plugging of two modules M_1 and M_2 is a module obtained by identifying pairs $(b_1, b_2), b_1 \in M_1, b_2 \in M_2$ of* B*-edges either satisfying:*

- *b_1 is a conclusion of M_1 and b_2 an hypothesis of M_2*
- *b_1 is an hypothesis of M_1 and b_2 a conclusion of M_2*

This is rather a communication by substitution than by cut, as opposed to Lambek calculus style. Also notice the perfect symmetry between M_1 and M_2, quite different from, e.g. substitution in λ-calculus: an hypothesis of M_1 may be identified with a conclusion of M_2, *and simultaneously* an hyptohesis of M_2 with a conclusion of M_1. When parsing the example, there will be examples of pluggings.

2.2 The parsing algorithm

Assume we are given a sentence $w_1 ... w_p$, each word w_i having the associated type t_i and the associated module M_i.

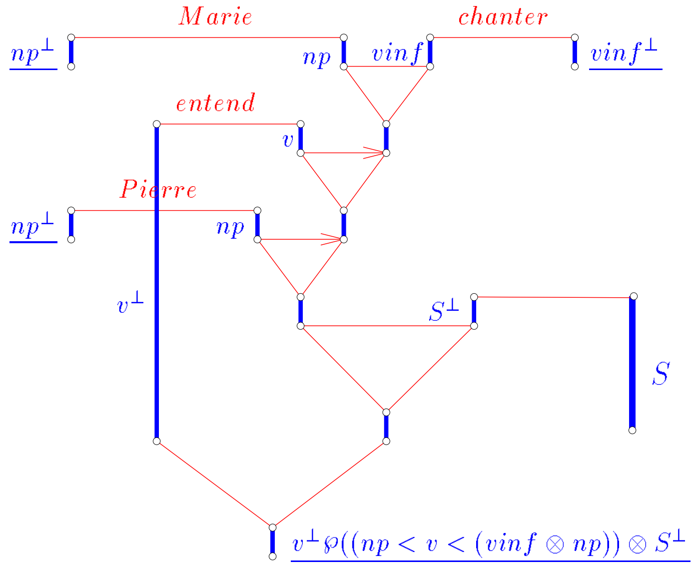
1. Try all possible pluggings of the M_i, in order to obtain a proof structure with conclusions $t_1^{\perp}, ..., t_p^{\perp}, S$.
2. Check whether the proof structure is a proof net.

3. Compute the order on axioms.

4. Check whether it is included in the linear word order of the actual sentence $w_1 \ldots w_p$.

Before reviewing the advantages of such a method, let us try the example.

There are only two solutions, namely the following and the one obtained by exchanging *Marie* and *Pierre*.



This is easily checked to be correct, because of proposition 4. What is the order on axioms, i.e. the minimal order required by this analysis? It is $Pierre < entend, entend < chanter, entend < Marie$ but there is no requirement on the relative position of *chanter* and *Marie*.

This means that the sentences *Pierre entend Marie chanter* and *Pierre entend chanter Marie* are both recognised, *and by the same syntactic analysis*. If wondering what correct sentences may be constructed out of these four words, one gets the two obtained by exchanging *Pierre* with *Marie*, and that is all... and that is exactly what the correct constructions of a French perception verb are.

2.3 Immediate outcomes

Even without sophisticated features needed to deal with more complicated phenomena, we get several nice properties that the Lambek calculus lacks.

2.3.1 Getting rid of spurious ambiguities

A major drawback of the Lambek calculus is known as spurious ambiguities. Lambek calculus allows any analysis, even the one pictured below by the brackets.

$$(Peters\ (eats\ an))\ apple$$

Here the situation is somewhat different. We could get a bracketing corresponding to the order in which modules, i.e. words, are pairwise plugged, but it does not make any sense.

Plugging operation needs not be binary, and, moreover, once the proof net is built, the syntactical analysis it provides does not reveal this information — this is because proof nets are the parallel syntax for logic, [Gir95].

What remains, the subformula trees appearing in the proof net, indicates how lexical categories are composed into syntactical constituents. But this bracketing, like (*Pierre* (*entend* (*Marie chanter*))) is the one we fixed in the lexicon when associating modules with words, and is precisely the one we want.

So, if within the Lambek calculus and its usual linguistic use, via provability, there is a need of non-associativity, as developed by [MO94], when looking at the proofs, there is no associativity: the associativity morphism is a proof interacting with other to moves the brackets, but this proof does not have an empty content.

2.3.2 Relatively free word order

Notice how easily the previous example deals with relatively free word order, while it is so difficult, in the Lmabek calculus to obtain the same syntactical analysis for two constructions, where two contiguous constituents are permuted. In fact we are able to do so at least for each such phenomena involving serie-parallel orders, see e.g. [LTV82] — which precisely correspond to orders which may be described with $\wp$ and $<$, see [Ret93, Ret95]).

2.3.3 Incremental strategy

Modules can be plug in any order, but it is natural to first try the order in which they appear, preserving this nice property of categorial grammar. Nevertheless it is satisfying not to need to modify the system for dealing with distant interaction, like in right or left extra position.

3 Extending the scope of categorial grammar

Because there is little room left, we shall just give the general principles, and an example for some of the phenomena we're able to deal with.

3.1 Principles

To describe more sophisticated phenomena, we also plug module through cuts the analysis being the normal form of the proof net. We thus provide a linguistic meaning of the most famous proof theoretical process.

We also use strings to label the atomic propositions occurring in the formulae of the proof net.

When we plug two modules, either directly or via a cut, these variables get instantiated.

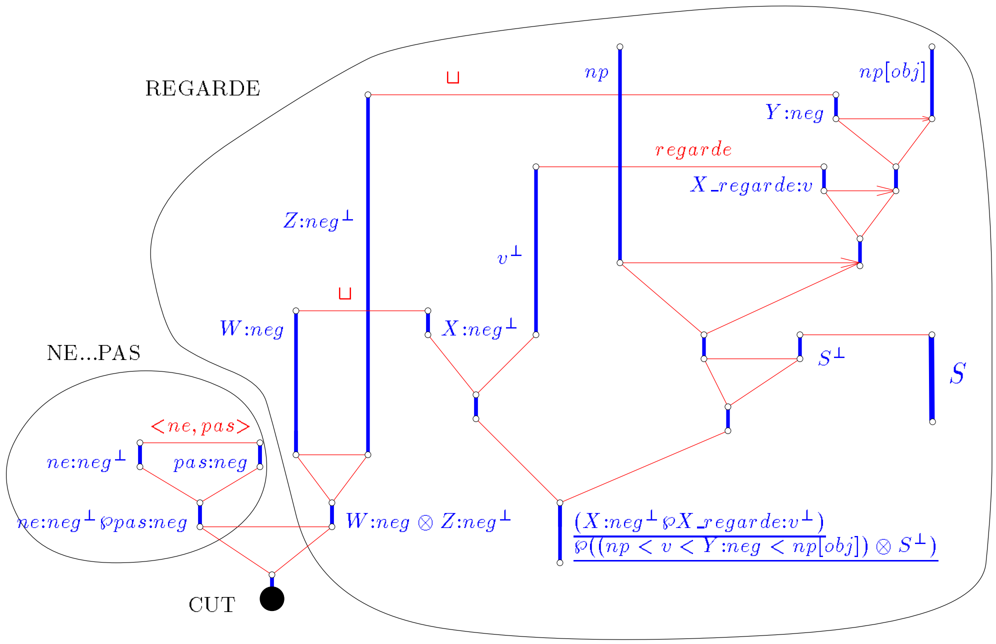
When we take discontinuous constituents into account, the convention expressing the relation between the temporal order and the word order is changed into the following one.

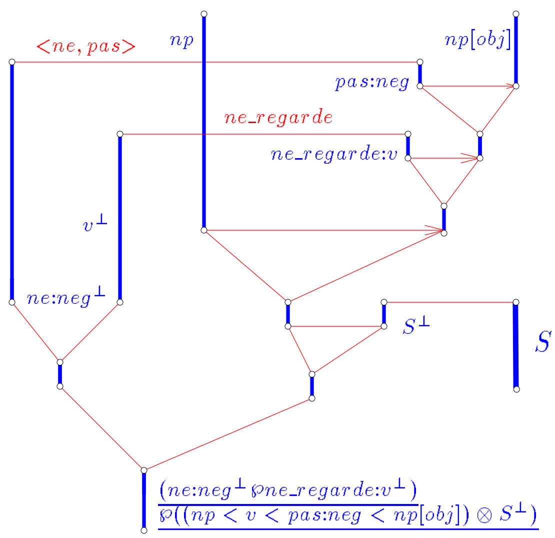
The order on axioms induces a partial ordering over their extremities: if two axioms α and β are such that $\alpha < \beta$, if $\alpha^1, \alpha^2, \beta^1, \beta^2$ are the end vertices of these axioms, given $u, v \in \{\alpha^1, \alpha^2, \beta^1, \beta^2\}$, we say that $u \blacktriangleleft v$ whenever there is a *directed* Æ-path from u to v neither passing through α nor β.

The word order is correct whenever this order on axiom end vertices is included within word order. For instance, in the following figure, dealing with French negation, we have: $v^\perp < neg$ but $neg^\perp$ and v are unordered.

After cut-elimination in such a case, we assume that the first component of the label (here: $<ne, pas>$) instantiates the left end vertex of the axiom and the second component instantiates the right one. This results in saying that the order: $ne\_regarde < pas$ must be satisfied in the sentence.

3.2 Head wrapping: French negation in two pictures





4 Conclusion

These techniques are already extended to clitics, and extraposition, and their combination, and we are thinking of improving and simplifying our description while pursuing the description of other linguistic phenomena, e.g. idiomatic expressions.

On a computational perspective, Denis Bechet (INRIA-Lorraine), started implementing these techniques, which contribute to automated deduction for linear logic. Before reaching the linguistic parsing, one needs to implement proof nets, modules, the correction algorithm and the « art of plugging modules ».

5 Appendix: Proof of lemma 10

Lemma 1 *Let α and β be axioms such that $\alpha \blacktriangleleft \beta$. Let n be the minimal integer such that there exists axioms $\alpha_p, 1 \leq p \leq n$ with:*

$$\alpha = \alpha_0 \triangleleft \ldots . \alpha_{n-1} \triangleleft \alpha_n = \beta$$

Then we have:

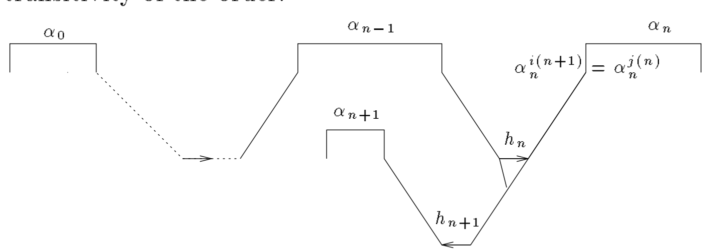
1. $\alpha_p^{i(p+1)} = (\alpha_p^{j(p)})^{\perp}$
2. *the union of the before-only paths corresponding to $\alpha_p \triangleleft \alpha_{p+1}$ is itself a before-only path.*
3. $\alpha^i \neq \alpha^j$ *when* $i \neq j$

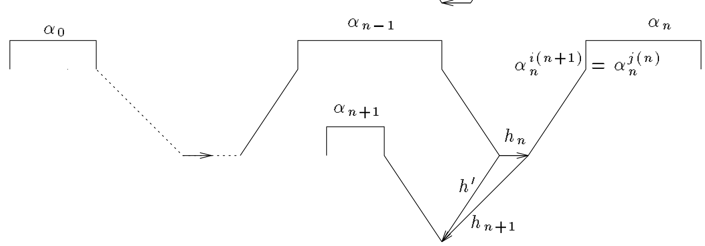
Proof: We proceed by induction on n and for $n = 1$ it is clear — in particular, we may not have $\alpha^0 \triangleleft \alpha^0$.

Assume we have $\alpha_0 \triangleleft \ldots . \alpha_{n-1} \triangleleft \alpha_n \triangleleft \alpha_{n+1}$ with $n+1$ being the minimal number of intermediate axioms. We a fortiori know that n is the minimal number of intermediate axioms for having $\alpha_0 \triangleleft \ldots . \alpha_{n-1} \triangleleft \alpha_n$, and we can apply the induction hypothesis to it.

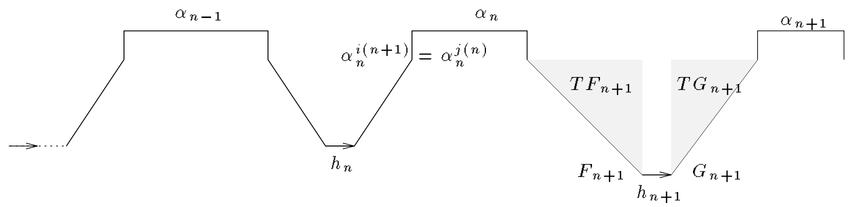
Therefore proving the two following points is enough:
1 $\alpha_n^{i(n+1)} = (\alpha_n^{j(n)})^{\perp}$
2 the union of the two before-only paths $\mathcal{P}$ and $\mathcal{P}'$ corresponding respectively to $\alpha_0 \triangleleft \ldots . \alpha_{n-1} \triangleleft \alpha_n$ (which is a before-only path because of the induction hypothesis) and to $\alpha_n \triangleleft \alpha_{n+1}$ is a before-only path.
3 for all $i < n+1$, $\alpha^i \neq \alpha^{n+1}$

Let $h_p = F(\alpha_{p-1}^{i(p)}) \longrightarrow G(\alpha_p^{j(p)})$ be the R-arcs corresponding to $\alpha_{p-1} \triangleleft \alpha_p$ — $\{i(p), j(p)\} = \{1, 2\}$.
Proof of 1 If we had $\alpha_n^{i(n+1)} = \alpha_n^{j(n)}$ the two R-arcs h_{n+1} and h_n may not be equal because of the direction of their arrows. Therefore is below another in the subformula tree, or the target of h_n is the source of h_{n+1} and there a R-arc h' due to transitivity of the order.





In both cases we thus have $\alpha_{n-1} \triangleleft \alpha_{n+1}$; this conflicts with $n+1$ being the minimal number of intermediate axioms.
Proof of 2 From previous point we know that: $\alpha_n^{i(n+1)} = (\alpha_n^{j(n)})^{\perp}$ Let TF_{n+1} and TG_{n+1} be the subformula trees of F_{n+1} and G_{n+1}.



The before only path $\mathcal{P}$ may not pass through TF_{n+1}.

Consider the first B-edge of $\mathcal{P}$ in TF_{n+1}. If it were an axiom $\alpha_i^k, i < n$ then we would have $\alpha_i \triangleleft \alpha_{n+1}$, and this would conflict with $n+1$ being the minimal number of intermediate axioms. If it were F_{n+1}, letting $\alpha_i^k, i < n$ be the axiom that $\mathcal{P}$ encountered just before, we would have $\alpha_i \triangleleft \alpha_{n+1}$ and this would conflict with $n+1$ being the minimal number of intermediate axioms.

The before only path $\mathcal{P}$ may neither pass through TG_{n+1}.

Consider the last vertex of $\mathcal{P}$ in TG_{n+1}. If it were an axiom, we would have an Æ-path from the tip of the arrow $F_{n+1} \to G_{n+1}$ to the leaf $\alpha_n^{i(n+1)}$ of TF_{n+1}, and thus an Æ-circuit. If it were G_{n+1}, we would have an Æ-path from $F_{n+1} < G_{n+1}$ to the leaf $\alpha_n^{i(n+1)}$ of TF_{n+1}, and thus an Æ-circuit.

Therefore the before-only path $\mathcal{P}$ does neither pass through TF_{n+1} nor through TG_{n+1}.

As the vertices of $\mathcal{P}'$ all belong to TF_{n+1} or to TG_{n+1} the union of $\mathcal{P}$ and $\mathcal{P}'$ is a before only path.

Proof of 3 Since the before only path $\mathcal{P}$ may not pass through TF_{n+1}, as established in **2**, $\alpha^{n+1} \neq \alpha^i$ when $i < n+1$. $\diamond$

Acknowledgements: Thanks to Philippe de Groote (INRIA-Lorraine) for helpful discussions.

References

[Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[Abr91] V. Michele Abrusci. Phase semantics and sequent calculus for pure noncommutative classical linear logic. *Journal of Symbolic Logic*, 56(4):1403–1451, December 1991.

[Abr94] Michele Abrusci. Ecxhange connectives for non-commutative intuitionistic propositional calculus. In M. Abrusci, C. Casadio, and M. Moortgat, editors, *Linear logic and Lambek calculus*, Roma, 1994. DYANA.

[AD94] Andrea Asperti and Giovanna Dore. Yet another correctness criterion for multiplicative linear logic with mix. In A. Nerode and Yu. Matiyasevich, editors, *Logical Fundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 34–46, St. Petersburg, July 1994. Springer Verlag.

[Asp91] Andrea Asperti. A linguistic approach to dead-lock. Technical Report LIENS 91-15, Dép. Maths et Info, Ecole Normale Supérieure, Paris, October 1991.

[Ber73] Claude Berge. *Graphs and Hypergraphs*. North-Holland, 1973.

[FR94] Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2), June 1994.

[dG95] Philippe de Groote, editor. *The Curry-Howard isomorphism*, volume 8 of *Cahiers du centre de logique, Université catholique de Louvain*. Academia, 1995.

[Gir86] Jean-Yves Girard. Multiplicatives. *Rendiconti del Seminario dell'Universitá é Politecnico Torino*, October 1986. Special issue on Logic and Computer Science.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Gir95] Jean-Yves Girard. Linear logic: its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Adavances in linear logic*, London Mathematical Society Lecture Notes. Cambridge university press, 1995. Also in [dG95].

[Gug94] Alessio Guglielmi. Concurrency and plan generation in a logic programming language with a sequential operator. In Pascal van Hentenryck, editor, *International Conference on Logic Programming*, pages 240–254, Genova, 1994. M.I.T. Press.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, pages 154–170, 1958.

[LTV82] E.L. Lawler, R.E. Tarjan, and J. Valdes. The recognition of Series-Parallel digraphs. *SIAM Journal of Computing*, 11(2):298–313, May 1982.

[MK94] Michael Moortgat and Natasha Kurtonina. Controlling resource management. In M. Abrusci, C. Casadio, and M. Moortgat, editors, *Linear Logic and Lambek calculus*, Roma, 1994. DYANA.

[MO94] Michael Moortgat and R Oehrle. Adjacency, dependency and order. In *Ninth Amsterdam Colloquium*, pages 447–466. ILLC, 1994.

[Moo94] Michael Moortgat. Multimodal linguistic inference. Bulletin of the IGPL, 3(2,3):215-246, 1994. Special issue on Deduction and Language, guest editor Ruth Kempson. (to appear under the same title in Journal of Logic, Language and Information)

[Mor94] Glyn Morrill. Structural facilitation and structural inhibition. In M. Abrusci, C. Casadio, and M. Moortgat, editors, *Linear Logic and Lambek calculus*, Roma, 1994. DYANA.

[Ret93] Christian Retoré. *Réseaux et Séquents Ordonnés.* Thèse de Doctorat, spécialité Mathématiques, Université Paris 7, février 1993.

[Ret94] Christian Retoré. On the relation between coherence semantics and multiplicative proof nets. Rapport de Recherche 2430, INRIA, décembre 1994. (and in LDPL'95).

[Ret95] Christian Retoré. Pomset logic: a non-commutative extension of linear logic. Rapport de recherche, INRIA, September 1995.

[Tro92] Anne Sjerp Troelstra. *Lectures on Linear Logic*, volume 29 of *Center for the Study of Language and Information (CSLI) Lecture Notes*. University of Chicago Press, 1992.