# Words as Modules and Modules as Partial Proof-Nets in a Lexicalized Grammar

A. Lecomte & C. Retoré
INRIA-Lorraine (France)

September 6, 1996

## 1 Introduction

In this paper we describe a syntactic calculus based on partial proof-nets (or modules) as building blocks of the system. The main idea is to associate with each lexical item one or more modules as syntactic types. These modules are obtained by unfolding the components of formulae that would be associated as types with the lexical items in a type-logical grammar ([Mor 94]). Proof nets are obtained by combining these modules by a uniform set of "plugging" rules. There are other approaches, based on Partiel Proof-Trees as building blocks in a grammar ([Jos 95]), our approach differs from them mainly by the emphasis put on the geometric notion of Proof-Net (cf [Lec 95]).
Our main motivation is to obtain a very general and logical model in which it would be possible to embed other calculi like the Lambek grammars on one side and the Lexicalized Tree Adjoining Grammars on the other side. In this paper, we first give an overview of the whole system (called POMSET-logic) : it is based on Multiplicative Linear Logic enriched with the non-commutative "before" ([Ret 93]), and which consequently deals with Partially Ordered Multisets instead of ordinary multisets of formulae. Then, we give more restricted versions which can be used for linguistic purposes. We plan to give illustrative examples (concerning scrambling in German, clitics in French and wh-extraction) in a next article.

## 2 An overview of POMSET logic, from the semantic side

POMSET logic (or Ordered Calculus) comes from classical Linear Logic ([Gir 87], [Gir 95]) and was primilarly presented in ([Ret 93]). It is well known that linear logic has a nice semantics for proofs in terms of coherent spaces ([Gir 90]) [1] . A coherent space A is a pair $<\mid A \mid, \simeq>$ where $\mid A \mid$ is a set (called the *web*) and $\simeq$ is a reflexive and symetric relation (but not transitive) interpreted as "coherent with". We define:

- strict coherence : a $\frown$ b iff a $\neq$ b and a $\simeq$ b
- strict incoherence : a$\asymp$ b iff $\neg(a \frown b)$
- incoherence : a $\smile$ b iff $\neg(a \frown b) \wedge$ a $\neq$ b

[1]However, there is no completeness: some true results (like $\mathbf{1} \vdash \perp$) cannot be demonstrated in ordinary classical logic linear (where $\mathbf{1}$ denotes the neutral element of $\otimes$ and $\perp$ the neutral element of $\wp$ ).

A *clique* of A is a subset c of $\mid A \mid$ such that : $\forall x, y \in c, x \simeq y$
We denote Cl(A) the set of all cliques of A. This allows to define so called *stable functions* from a coherent space A to a coherent space B. They are functions F from Cl(A) to Cl(B) which are such that:

1. $a \subset b \Rightarrow F(a) \subset F(b)$
2. $F(limsupX) = limsupF(X)$
3. $a \cup a' \in Cl(A) \Rightarrow F(a \cap a') = F(a) \cap F(a')$

In categorical terms, Cl(A) is a category the objects of which are *cliques* and the morphisms of which are *inclusions.*(1) means that a stable function is a fonctor from Cl(A) to Cl(B), (2) that this functor preserves *directed colimits* and (3) that it preserves *pullbacks.* Moreover, if we add :

4 F($\emptyset$) = $\emptyset$

we get a subclass of stable functions which is exactly what is needed in order to interpret the linear implication: the so called class of linear functions. By means of this construction, we can interpret a linear formula like: A –o B as a linear function [2] from A to B, such an interpretation being the condition for regarding linear logic as a *constructive* logic. The connectives of linear logic are easily defined in this semantics. Every multiplicative formula A * B is interpreted as a coherent space the web of which is the cartesian product of the two webs $\mid A \mid and \mid B \mid : \mid A * B \mid = \mid A \mid \times \mid B \mid$. The coherence relation is defined in two ways, according whether "*" is the conjunction $\otimes$ or the disjunction $\wp$ . We have for A $\otimes$ B :

| A \ B | $\smile$ | = | $\frown$ |
|---|---|---|---|
| $\smile$ | $\smile$ | $\smile$ | $\smile$ |
| = | $\smile$ | = | $\frown$ |
| $\frown$ | $\smile$ | $\frown$ | $\frown$ |

and for A $\wp$ B :

| A \ B | $\smile$ | = | $\frown$ |
|---|---|---|---|
| $\smile$ | $\smile$ | $\smile$ | $\frown$ |
| = | $\smile$ | = | $\frown$ |
| $\frown$ | $\frown$ | $\frown$ | $\frown$ |

Negation is defined by :

1. $\mid A \mid = \mid A^{\perp} \mid$
2. $x \neq y \Rightarrow (x \frown y[A^{\perp}] \Leftrightarrow \neg(x \frown y)\ [A])$

We omit here the semantics of the additives which will play no role, for the time being, in our system. As it can be seen, this logic is commutative and associative (for instance, for commutativity, we can easily exhibit an isomorphism from A $\otimes$ B to B $\otimes$ A). But Retoré ([Ret 93]) noticed that all the potentialities offered for a definition by this semantics were not used. There is in fact room for two other connectives if we abandon the commutativity: A < B :

| A \ B | $\smile$ | = | $\frown$ |
|---|---|---|---|
| $\smile$ | $\smile$ | $\smile$ | $\frown$ |
| = | $\smile$ | = | $\frown$ |
| $\frown$ | $\smile$ | $\frown$ | $\frown$ |

A > B :

[2]More precisely as the trace of a linear function, a concept we do not develop here

| A \ B | $\smile$ | $=$ | $\frown$ |
|---|---|---|---|
| $\smile$ | $\smile$ | $\smile$ | $\smile$ |
| $=$ | $\smile$ | $=$ | $\frown$ |
| $\frown$ | $\frown$ | $\frown$ | $\frown$ |

These definitions are the only ones which can preserve some coherence properties like :

1. $(x, y) \frown (x', y)[A * B] \Leftrightarrow x \frown x'$[A]
2. $(x, y) \frown (x, y')[A * B] \Leftrightarrow y \frown y'$[B]
3. $x \frown x'[A] \wedge y \frown y'[B] \Rightarrow (x, y) \frown (x', y')$[A*B]
4. $x \smile x'[A] \wedge y \smile y'[B] \Rightarrow (x, y) \smile (x', y')$ [A*B]

It is then easy to show the following properties :

1. "<" is selfdual : $(A < B)^{\perp} \equiv A^{\perp} < B^{\perp}$
2. it is in between $\otimes$ and $\wp$ [3]
   (A $\otimes$ B) –o (A < B) –o (A $\wp$ B)
3. it is associative:
   (A < B) < C $\equiv$ A < (B < C)
4. and it is obviously non commutative.

This non-commutative connective is intuitively interpreted as *before*. A computational interpretation is provided by Asperti ([Asp 91]) by saying that if $\otimes$ represents sequential composition of processes, their order being undetermined, and $\wp$ parallelisation of processes, < represents a sequential composition where the order is constrained. Thus, in terms of processes, A < B is interpreted as: "A must be performed before B". It is important to notice that A < B or $A^{\perp}$ < B are entirely different from $A^{\perp}$ –o B or A –o B because < is in fact a product, and –o is obtained by residuation from the product $\otimes$ .
Because we have a mean to express an order between formulae, it becomes possible to deal not only with multisets of formulae, like it is the case in classical (commutative) linear logic, but with partially ordered multisets.

## 3 Proof-nets for POMSET logic

### 3.1 BR-graphs

The familiar conception for proof-nets in linear logic ([Gir 87], [Dan 89], [Fleu 94], [Abr 95]) can be naturally extended to the ordered calculus. We present here the conception of proof-nets advocated by Retoré ([Ret 96]). It is based on the notion of BR-graph. Implicitely, a proof-net has conclusions which are parts of *one-sided sequents* (this is always possible since we have an involutive negation).
**Definition 3-1** (**BR-graphs**): A BR-graph is a graph with coloured edges (blue and red). B-edges are undirected. They correspond to formulae and define a perfect matching of the graph. The R-edges may be undirected or directed, in which case we call them R-arcs. They correspond to connections between formulae (axioms or connective-links).
**Definition 3-2** (**links**): there are 5 sorts of links:

- **Axiom-link** : an *axiom-link* has two conclusions, A and $A^{\perp}$ . These conclusions are represented by B-edges, and there is an undirected R-edge between the two.

[3]if we accept the MIX rule in the system, that is the rule which derives $\Gamma, \Gamma' \vdash \Delta, \Delta' from \Gamma \vdash \Gamma' and \Delta \vdash \Delta'$

- **Tensor-link** : a *tensor-link* has one conclusion : A $\otimes$ B, and two premises: A and B. The conclusion and the premises are represented by blue edges. There are three undirected R-edges : one with as ends: A $\otimes$ B and A, one with A $\otimes$ B and B and one with A and B.
- **Before-link** : a *before-link* has one conclusion : A $<$ B, and two premises: A and B. The conclusion and the premises are represented by blue edges. There are two undirected R-edges : one with as ends: A $<$ B and A and one with A $<$ B and B and one R-arc, from A to B.
- **Par-link** : a *par-link* has one conclusion : A $\wp$ B, and two premises: A and B. The conclusion and the premises are represented by blue edges. There are two undirected R-edges : one with as ends: A $\wp$ B and A and one with A $\wp$ B and B.
- **Cut-link** : a *cut-link* is similar to a tensor-link, but the conclusion is a B-edge marked by a $\bullet$ and the premises are dual formulae $F$ and $F^{\perp}$ (In fact, we can interpret this point as a formula like : $(\exists X)(X \otimes X^{\perp})$.

**Definition 3-3 (proof-structure)** : a proof-structure is a BR-graph such that any B-edge is the conclusion of exactly one link and the premise of at most one link (the B-edges which are not a premise of any link are called conclusions of the proof-structure, they contain all the cuts), provided with a set of R-arcs between conclusions which defines a strict partial order.
**Definition 3-4 (proof-net)** : an ordered proof-net is a proof-structure which contains no alternate elementary circuit.

### 3.2 Properties

**Theorem 3-2-1 (Cut-elimination)** : cuts can be eliminated. More precisely: let $\Pi$ be a proof net whose conclusions are $F_1, F_2, ..., F_k, \bullet_1, ..., \bullet_p$, ordered by $\Re$, (where $F_1, ..., F_k$ are formulae and all the $\bullet_i$ are cuts) it is possible to rewrite $\Pi$ as $\Pi'$ with conclusions $F_1, F_2, ..., F_k$ ordered by the restriction of $\Re$ to these formulae. Moreover, this rewriting enjoys strong normalisation and confluence.

## 4 Proof-structures for word order

### 4.1 Labeling the B-edges with pomsets

In the system that we present here (henceforth called PPNL, for "Partial Proof-Nets for Language"), each word is associated with a partial proof-net. This partial proof-net displays the parts of a formula which expresses the type of the word, that means : *what is its category and how it must be processed in any context where it occurs.* See below for more explanations. B-edges of the corresponding partial proof-nets are themselves typed with a subformula of the whole formula and labeled with an ordering on strings. Let us explain here to what intuitions corresponds this labeling. If we take the Lambek calculus as an example, types are associated with words and the correctness of a sequence of words (or its belonging to some category a) is demonstrated by the rules of the calculus. Lexicalisation allows to express *strict locality* (like subcategorisation), but unfortunately in Lambek grammars, we are limited to the strict string adjacency, which makes difficult to express structural adjacency and impossible to express some movements like non-peripheral extraction or right extraction. Moreover, it is frequently the case that no ordering is strictly imposed. Examples are often given from German, where "there is no syntactic constraint on the ordering of the nominal arguments of a verb, as long as the verb remains in final position" ([Ram 94]). Haider ([Hai 91]) notes that in the following sentence : ... *dass [eine hiesige Firma] [meinem Onkel] [die Möbel] [vor drei Tager] [ohne Voranmeldung] zugestellt hat*

(= that a local company delivered the furniture to my uncle three days ago without advance warning), "any permutation of these five elements is grammatically well-ordered". We are then led to conceive a framework where such cases can be dealt with. We can inspire us from the Lambek calculus, where it is possible to label coherently the types with strings and operations on strings (in fact only concatenation). In the labeled Lambek calculus, the directionality of the slash (= the oriented linear implication) is mirrored in the order of concatenation of the strings. We have for instance for the left rule for / :

$$\frac{s : \Theta \vdash B \qquad \Gamma, w + s : A, \Gamma' \vdash C}{\Gamma, w : A/B, B, \Theta, \Gamma' \vdash C}$$

In the same way, we will mirror the order of (sub)formulae occurring in a proof-structure in the order of lexical items. We need then to enrich our definition of links (cf definition 3-2) by inserting orderings on labels into it. Each B-edge is enriched by a partially ordered multiset of strings. Let us define now : (we keep definition 3-2 just adding conditions on labeling orderings)
**Definition 4-1 (labeled links)** :

- **Labeled axiom-link** : it has two conclusions, (A, $\Re$) and (A$^\perp$ $\Re'$) where $\Re$ and $\Re'$ are partially ordered multisets of strings such that $\Re = \Re'$.
- **Labeled tensor-link** : if the conclusion is labeled by the pomset $\Re$, one of the two premises (we do not know which one) of the tensor-link is labeled with the union of $\Re$ with the pomset $\Re'$ that labels the other premise (thus, if $\Re = \emptyset$, then the two premises are labeled by the same pomset)
- **Labeled before-link** : the conclusion A < B is labeled by the pomset $\Re < \Re'$ (where "<" means that all the elements of $\Re$ are before all the elements of $\Re'$).
- **Labeled par-link** : the conclusion A $\wp$ B is labeled by the pomset $\Re \cup \Re'$ (no order between elements of $\Re$ and elements of $\Re'$).
- **Labeled cut-link** : the cut is labeled by $\emptyset$ and the pomsets of the two premises are equated (particular case of the tensor-link).

We can comment these definitions in the following way : only the "before" creates an order on strings. $\otimes$ is preservative : let us remember that, in a one-sided presentation of the sequent calculus, with only right sequents, a $\otimes$ -formula is the dual of an implicational formula. In this case, nothing is added, there is only a transformation : a sequence of types gives a new type, but the string support remains the same[4]. A $\wp$ -formula corresponds to a $\otimes$ -formula on the left. It is then natural to retrieve the orderings which occur on the two components, but without creating any new ordering.

## 4.2 Partial Labelled Ordered Proof-Net

**Definition 4-2 (PPN)** : A *Partial Proof-Net* (PPN) is like a proof-net except that some B-edges are the conclusion of no link (they are called hypothesis).
**Definition 4-3 (LPPN)** : A *Labelled Partial Proof-Net* (LPPN) is a Partial Proof-Net enriched by labels according to the definition of labeled links.
**Proposition** : *In a Labelled Partial Proof-Net, the structures on strings built up by means of the labeled links are pomsets.*
*Proof*: this results by induction and from the absence of elementary alternate circuits.

[4]and all cases of parallelized formulae can be viewed as implicational ones, because of the duality and the De Morgan laws. For instance : A $\wp$ B = A $\multimap$ B$^\perp$ .

## 4.3 A restricted case : "intuitionnistic" proof-structures

In a first time, we shall restrict ourselves to particular PPNs, those which have one and only one distinguished B-edge that is :

- labeled by a negative atom,
- a premise of a tensor-link,
- the conclusion of no link

Such a B-edge will be called an *output*. And these LPPN have the particularity to have one and only one output, it is the reason why we call them Intuitionnistic Labeled Partial Proof-Nets (ILPPN).
**Definition 4-4** (**Intuitionnistic LPPN**) : an intuitionnistic LPPN is a LPPN which contains one and only one output.

## 4.4 Lexicalized Intuitionnistic Labeled Partial Proof-Nets

**Definition 4-5** (**Lexicalized Intuitionnistic LPPN**) : an ILPPN is said to be lexicalized if and only if :

1. its conclusion is of the form : $a^{\perp} \wp (X_1 \otimes Y_1) \wp ... \wp (X_{\mathrm{m}} \otimes Y_{\mathrm{m}})$, where $a^{\perp}$ is labeled by a pomset consisting in one string w (which belongs to the lexicon), and the formulae $X_{\mathrm{i}} \otimes Y_{\mathrm{i}}$ by $\emptyset$,
2. $a^{\perp}$ is linked by an axiom-link to an atom a which occurs in one of the $X_{\mathrm{i}}$,
3. every $X_{\mathrm{i}}$ is a par/before formula (ie : containing only the connectives par and before[5])

**Remarks** : 1) if we remember that connectives on the right correspond to their duals on the left, we see that such a conclusion, when moved to the left, leads to a formula : $a \otimes (X_1 \multimap Y_1{}^{\perp}) \otimes ... \otimes (X_{\mathrm{m}} \multimap Y_{\mathrm{m}}{}^{\perp})$.
Otherwise, the second condition in the definition makes an occurrence of $a^{\perp}$ necessary in one of the $X_{\mathrm{i}}$. This can be interpreted as : the word w is an a and (multiplicative conjonction) when supplied with additional material in $X_{\mathrm{i}}$, it will give an $Y_{\mathrm{i}}{}^{\perp}$.
2) let us notice that this definition is coherent with our definition of labeled links: only the "declarative" part (w "is" an a) provides an initial pomset. The "procedural" one contains initially no pomset, but by means of the labeled links, one at least of the $X_{\mathrm{i}}$ will receive a pomset (which contains at least w, because of the axiom link between $a^{\perp}$ and a, and because $X_{\mathrm{i}}$ is a par/before formula), and this pomset will be transfered to $Y_{\mathrm{i}}{}^{\perp}$, and maybe to the output (if there are axiom links between the $Y_{\mathrm{j}}$ and the $X_{\mathrm{k}}$).
**Example** :

$$(\{\text{loves}\}\text{: v}) \otimes ((\text{v} < \text{np}) \multimap \text{vp}) \otimes ((\text{np} < \text{vp}) \multimap \text{s})$$

is the type of a transitive verb. It gives a string labeling the verb, and two other informations: if followed by a np, it will give a vp and if this vp is preceded by a np it will give a s. Of course, we need to express the identity of the two occurrences of v and the identity of the two vp. There, the proof representation is needed. It is obtained by taking the dual formula of the type and decomposing it into a tree of subformulae with axiom-links. (cf fig 1.) In the following figure, we note the labels by using integers. By the conditions on labeled links, (1) is identified with {loves}. When the np (2) is identified with a real np, thus giving : (2) = {a < woman} for instance, the conclusion of the before-link will be labeled by : {loves < a < woman}. This pomset is transmitted to the output $v^{\perp}$ (because of the tensor-link). It is equated with (3), and when the second np (4) is instanciated (for instance by {a < man}), we shall get as a new pomset labeling the conclusion of the second before-link : {a

[5]and of course negation

Figure 1:

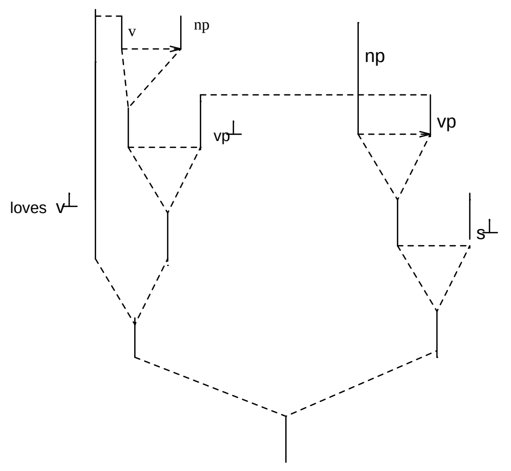


Figure 2:

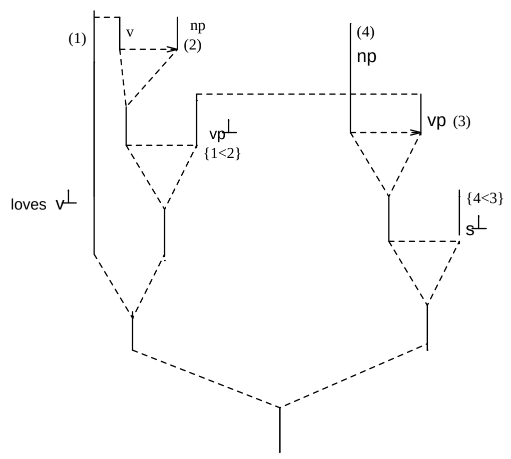
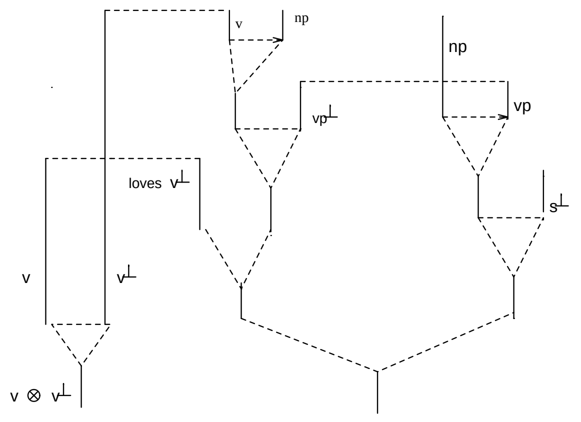


Figure 3:



< man < loves < a < woman}, and this pomset is transmitted to the output $s^\perp$. Let us notice that the conclusions in $\otimes$ remain empty, and thus the conclusion of the lexicalized PPN associated with the verbal form *loves* is also labeled by {loves}.

## 5 Plugging PPNs

There are obviously two ways of plugging PPNs together :

- **by axiom links**
- **by cut links**

Plugging by axiom-links essentially corresponds to complementation (according to the subcategorisation frames of the words that PPNs are associated with), and plugging by cut-links to modification, or adjonction. *Plugging by axiom-link* amounts to identifying pairs ($b_1$, $b_2$) where $b_1$ and $b_2$ are B-edges, $b_1 \in M_1, b_2 \in M_2$ ($M_1$ and $M_2$ are two independant PPNs) either satisfying: $b_1$ conclusion of $M_1$ and $b_2$ hypothesis of $M_2$, or $b_1$ hypothesis of $M_1$ and $b_2$ conclusion of $M_2$. *Plugging by cut-link* amounts to putting a cut between two dual conclusions and then removing it by cut-elimination.
The operation of plugging by cut-links leads us to a slight change in our PPNs, because for the time being, there is no obvious possibility of finding interesting cut formulae. This leads us to adding new conclusions in our LPPNs, but they are empty (in contrast with the main conclusion which is always labeled, like we previously saw, by the same pomset as the "declarative" B-edge). These conclusions are in fact instances of cuts : we know that a cut is equivalent to a conclusion ($\exists$ X)(X $\otimes$ X$^\perp$ ), and we add conclusions which have exactly the form X $\otimes$ X$^\perp$ . For instance, if we want to introduce adverbials, like *passionately*, we must open such a "gate" in every LPPN associated with a verb. This will give raise to new LPPNs like shown in figure 3.
The only modification consists in breaking an axiom link, thus creating two new axiom-links which link previously linked dual nodes in the PPN respectively to two premises of a tensor-link. It is easy to see that such a transformation preserves the property of being an ILPPN.
This LPPN can be then used together with a module which introduces a verb modifier or the negation on v (in French for instance). This second LPPN contains a dual conclusion $v^\perp \wp$ v (cf figure 4). By cut-elimination, the v node, and then the vp$^\perp$ node are transformed in order to provide a correct sentence. Let us see for instance the main conclusion for the adverbial : adv$^\perp$ $\wp$ ((adv < v) $\otimes$ v$^\perp$) (where v and v$^\perp$ are not linked, only adv$^\perp$ and adv

Figure 4:

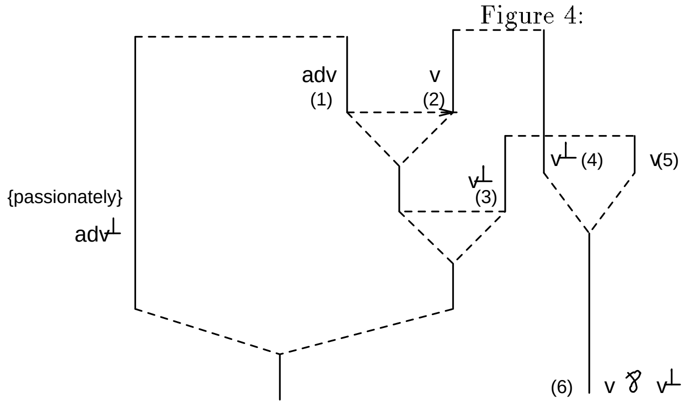
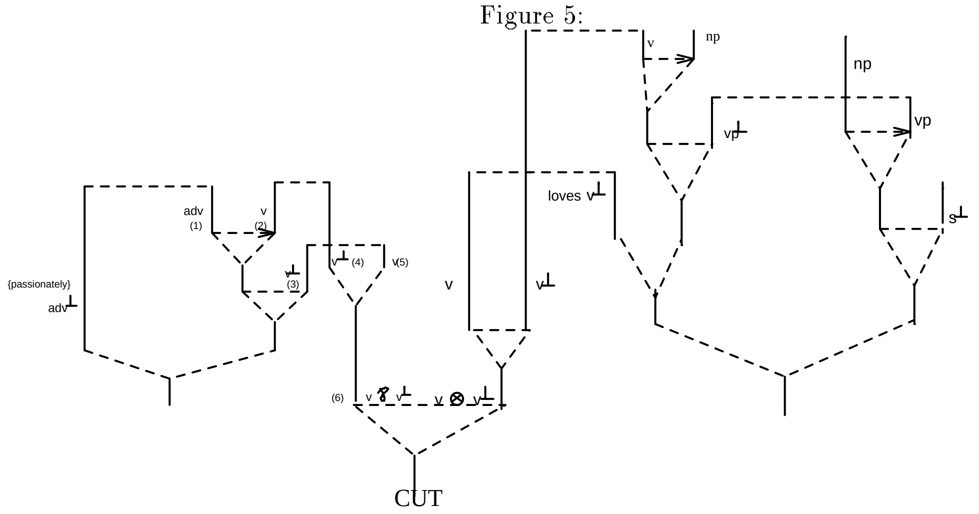


Figure 5:



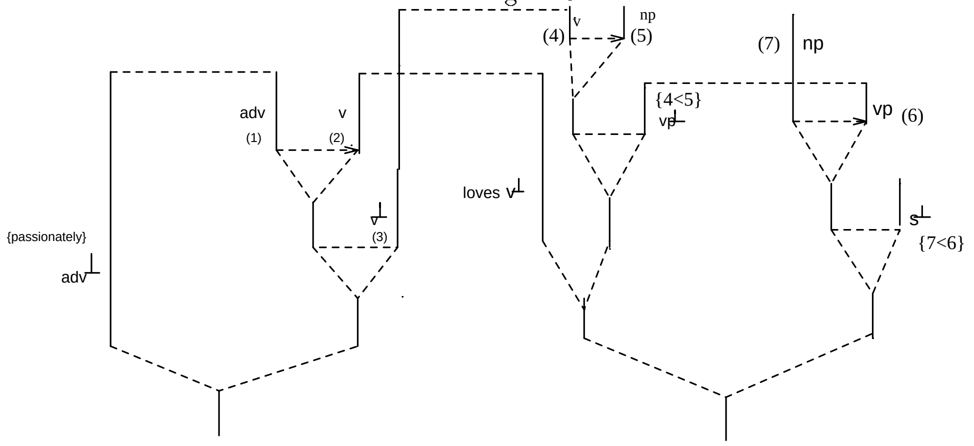
are). The ILPPN associated with it has an axiom-link between adv$^\perp$ and adv.(fig 4)
This PPN can be extended into a PPN with an additional conclusion v$^\perp$ $\wp$ v which will be not empty. The transformation here only consists in linking the two dual atoms v and v$^\perp$ (which are not linked in the initial PPN) respectively with the two premises of a par-link (that only expresses the identity principle, actually). We get a new kind of LPPN, it will be in fact characteristic of modifiers. We shall call them MLPPNs.

Comment on fig4 : (1) is equated to {passionately}. (2) and (4) are identical, (3) and (5) also. (3) is the pomset {passionately}. Then (6) is also labeled by {passionately}. When linked by a cut link to an LPPN associated with a verbal form (cf fig. 5), the corresponding tensor formula is labeled by the same pomset (from the definition of labeled cut-links) and so, the entry (1) in the verbal PPN will be labeled by {passionately, loves}, but without any ordering on it. It is the operation of cut-elimination which will provide the correct word order. This operation (which is an intrinsec part of the process of plugging by cut) gives us finally the PPN of fig. 6.

In this new PPN, (3) is identified with {passionately < loves} (order is created), and also (4), then (6) is equated with a pomset which includes the direct object on the right and finally the ouput $s^\perp$ is labeled with a correct order. This new PPN is still an intuitionnistic one (having a sentence as an output).

If there is no modifier, the PPN of fig. 3 is still valid. In this case, cut-elimination amounts to suppressing the tensor-link v $\otimes$ $v^\perp$, which is, as already said, a mere instance of a cut. Let us notice that without cut-elimination, the word order would remain free. It is a question whether we could represent in this way dislocated sentences, sentences that result from

Figure 6:



different orderings of words with particular intonations.

## 6 Conclusion

The system we have presented here is specifically a *resource-conscious system.* Duality must be interpreted in terms of exchange and communication: the axiom link between two atomic formulae expresses that some demand is satisfied, a cut-link makes a communication between two processes which can be synchronized: by removing the cut, we actually observe that one cut-formula provides what the other formula (the dual one) needs. This characteristic allows to express many phenomena. For instance, in wh-extraction like in cliticization, a word (the wh-word or the clitic word) is supposed to "kill" a demand for a np of such and such form (for instance an accusative np for what, and a np of the same case of the clitic in the clitic-case). But to kill a demand is... to satisfy it, just because of the involutive negation ($(\mathrm{np}^{\perp})^{\perp} = \mathrm{np}$ : a demand for a demand for a np is equivalent to a np). This kind of mechanism is proper to every system based on resource consomption (see for instance categorial grammar and its operation of type-raising) and it provides here an important expressive power. Adding the connective "before" allows to manage questions of word order. Labelled links are needed in order to record the (partial) word orders resulting from the before-formulae. We insist here on the fact that labels play no active role in the process: "unification" of labels does not "drive" the construction of the PPNs, they only play a passive role, by building growing pomsets according to the very general conventions on labeled links.
Reduction (in the sense used in $\lambda$-calculus) plays here an important role under the form of cut-elimination. We can consider that sentences obtained by cut-plugging without cut-elimination are "non normal sentences" (for instance sentences uttered with particular intonations). Normalisation provides the right word-order.
Finally, let us notice the important advantage of this system, being based on logic. We have not to formulate algebrically some *ad hoc* and very particular operations (for instance on trees): *all the machinery can be summarized in the following instruction:given the initial LPPNs as they are, build a correct proof-net!* And let us recall that the criterion of correctness is very simple : *no elementary alternate circuit!*

**Acknowledgements**
Many ideas in this paper come from discussions in the Calligramme project, especially with Philippe de Groote.

## References

[Abr 95] Abrusci, M.: 1995, Non commutative proof-nets in Girard, Lafont, Régnier (eds), *Advances in Linear Logic*, Cambridge University Press, London Mathematical Society Lecture Notes.

[Asp 91] Asperti, A.: 1991, 'A Linguistic Approach to Deadlock'. Rapport de recherches, LIENS-91-15, LIENS, 1991, Paris.

[Bec 55] Bech, G.: 1955, Studien über das deutsche Verbum infinitum, Det Kongelike Danske videnskabernes selskab. Historik-Filosofiske Meddelelser, bd. 35, 2.

[Bec 91] Becker, T., Joshi, A., Rambow, O.:1991, Long distance scrambling and tree adjoining grammars, in *Fifth Conference of the European Chapter of ACL*, 21-26.

[Bes 88] Bès, G.: 1988, Clitiques et constructions topicalisées dans une grammaire GPSG du français,*Lexique***55-81**

[Bha 94] Bhatt, R.: 1994, *Word order and case in Kashmiri*PhD thesis, Univ. Illinois.

[Dan 89] Danos, V. & Regnier, L.:1989, 'The structure of multiplicatives' *Arch. Math. Logic* **28**181-203.

[Fleu 94] Fleury, A. & Retoré, C.: 1994, The mix rule, *Mathematical Structures in Computer Science*vol. **4**, 2.

[Gir 87] Girard, J-Y.:1987, 'Linear logic' *Theoretical Computer Science*, **50**, 1987, 1-102.

[Gir 95] Girard, J-Y.: 1995, Linear Logic: its syntax and semantics, in Girard, Lafont, Régnier (eds) *Advances in Linear Logic* Cambridge University Press, London Mathematical Society Lecture Notes.

[Gir 90] Girard, J-Y., Lafont Y., Taylor P.: 1990, *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science, **7**, Cambridge University Press, Cambridge.

[Hai 91] Haider,H.:1991, Argument structure: Semantic basis and syntactic effects. Class notes from the Third European Summer School in Language, Logic and Information, Saarebrücken.

[Jos 75] Joshi, A., Levy, L. & Takahashi, M.: 1975, Tree Adjunct Grammars, *Journal of Computer and System Sciences*, **10**, 136-163

[Jos 87] Joshi, A.:1987, An introduction to tree adjoining grammars, in Manaster-Ramer (ed) *Mathematics of Language*, 87-114, John Benjamins

[Jos 95] Joshi, A. & Kulick, S.: 1995, Partial Proof Trees as Building Blocks for a Categorial Grammar, in Morrill & Oehrle (eds) *Formal Grammar, Proceedings of the Conference of the European Summer School in Logic, Language and Information*, Barcelona.

[Kra 95] Kraak, E.:1995, French object clitics: a multimodal analysis, in Morrill & Oehrle (eds) *Formal Grammar, Proceedings of the Conference of the European Summer School in Logic, Language and Information*, Barcelona.

[Lec 95] Lecomte, A. & Retoré, C.: 1995, POMSET-logic as an alternative categorial grammar, in Morrill & Oehrle (eds) *Formal Grammar, Proceedings of the Conference of the European Summer School in Logic, Language and Information*, Barcelona.

[Mil 92] Miller, P.:1992, *Clitics and Constituents in Phrase Structure Grammar*, New York, Garland.

[Mor 94] Morrill, G.:1994, *Type-Logical Grammar*, Kluwer.

[Ram 95] Rambow, O.: 1995, Coherent Constructions in German: Lexicon or Syntax? in Morrill & Oehrle (eds) *Formal Grammar, Proceedings of the Conference of the European Summer School in Logic, Language and Information*, Barcelona.

[Ram 95] Rambow, O., Vijay-Shanker, K., Weir, D.: 1995, D-Tree Grammars in *Proceedings of the 33rd Meeting of the ACL*,

[Ram 94] Rambow, O. & Joshi, A.:1994, A Processing Model for Free Word Order Languages, in *Perspectives on Sentence Processing*, Clifton, Frazier and Rainer (eds) Lawrence Erlbaum.

[Ret 93] Retoré, C.:1993, *Calcul des séquents ordonnés* , Thèse Paris 7.

[Ret 96] Retoré, C.:1996, Perfect matchings and SYMPA-graphs: multiplicative proof structures and nets as RB-graphs, Congrès de Logique Linéaire, Tokyo.

[Ver 95] Versmissen, K.:1995, Word order domains in categorial grammar in Morrill & Oehrle (eds) *Formal Grammar, Proceedings of the Conference of the European Summer School in Logic, Language and Information*, Barcelona.

[Vij 1992] Vijay-Shanker, K.:1992, Using descriptions of trees in a tree adjoining grammar, *Comput. Ling*, **18**,(4), 481-517.