Proof-Nets, hybrid logics and minimalist representations

Alain Lecomte (alain.lecomte@upmf-grenoble.fr)
LORIA-Calligramme

Abstract. In this paper, we aim at giving a logical account of the representationalist view on minimalist grammars by refering to the notion of Proof-Net in Linear Logic. We propose at the same time a hybrid logic, which mixes one logic (Lambek calculus) for building up elementary proofs and another one for combining the proofs so obtained. Because the first logic is non commutative and the second one is commutative, this brings us a way to combine commutativity and non commutativity in the same framework. The dynamic of cut-elimination in proof-nets is used to formalise the *move*-operation. Otherwise, we advocate a proof-net formalism which allows us to consider formulae as nodes to which it is possible to assign weights which determine the final phonological interpretation.

Keywords: generative grammar, type-logical grammar, linear logic, proof-nets, hybrid logics

1. Introduction

The basic idea concerning the use of Proof-nets is to consider words and expressions as building blocks in the construction of proofs of sequents. These building blocks are called *modules*, they correspond to Proof-nets where some premisses are mere hypotheses. This conception has many relations with works on partial proof-trees (PPTs) in the context of *Tree Adjoining Grammars* (Joshi and Kullick, 1997). Like in the case of PPTs, we are led to *hybrid* logics in order to give a precise logical formulation of combining PNs: we need a logic for building up elementary proofs and then we need another one for combining these proofs. One of the particularities of our approach is that we shall not use some special rules for combining proofs like *stretching* in PPTs. Another particularity consists in using proof-nets, whereas in the Joshi-Kulick-Kurtonina approach, it is claimed that Natural Deduction trees exactly provide what is needed for linguistic purposes (Joshi and Kullick, 1997). Our motivation for it is that the proof-net machinery allows a better formalisation of *move*-operations by means of cut-elimination, where cut-formulae are complex formulae ($\otimes$-conjunctions and $\wp$ -disjunctions). The Minimalist Program (Chomsky, 1996) also makes reference to *features* which are either *weak* or *strong*: this suggests that if we treat features as atomic types (similarly to

(Cornell, 1998)), then these types, when considered nodes in a net, can receive inequal strengths which could explain how variants of the same sentence can be produced.

2. Two logics

The use of the Lambek calculus (**L**) with product, in the context of so-called Lambek grammars, requires that the hypotheses be totally ordered. Moreover, its absolute lack of structural rules makes it difficult to reuse a lexical type, even if it is what happens in some linguistic phenomena like cyclic movement. The operation *Move*, of frequent use in Minimalist Grammars (Stabler, 1997) cannot be conveniently represented in this framework. Let us imagine that for instance we want to describe an up- and left-ward movement of a constituant with regards to a verbal head. We would like then the $\mathtt{d}$-constituant be used twice : one time at the position where it is selected by the verbal head, and another time at the position where it receives $\mathtt{case}$. We could think of a product type associated with each determiner phrase, something like: $\mathtt{d} \otimes \mathtt{case}$ (or $\mathtt{d} \otimes \overline{\mathtt{k}}$ like it will be noted further) but even if so, the Lambek calculus fails because it cannot express any kind of *wrapping*. The solution we shall propose to this problem consists in adding an upper level to this rudimentary logic: a level in which it becomes easy to manipulate ready made proofs in **L**.
We call *module* a partial proof in a sequent calculus (and later on, a partial proof-net representing this partial proof). A proof is said to be partial if it uses (not discharged) hypotheses.
Let us see for instance what could be a "module" associated with a transitive verb, say *to like* (where $\mathtt{d}$ denotes the determiner category, which is a *categorial* feature, $\overline{\mathtt{k}}$ the requirement for a $\mathtt{case}$-feature, which is a *functional* feature, and $\mathtt{vp}$ the verbal phrase category) .

to like [1] :

$$\mathtt{d}^3 \otimes \overline{\mathtt{k}}^4, ((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})^1 \otimes ((((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})^1 \bullet \mathtt{d}^3 \multimap (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))^2) \otimes (\overline{\mathtt{k}}^4 \bullet (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))^2 \multimap (\mathtt{d}\backslash\mathtt{vp})) \vdash (\mathtt{d}\backslash\mathtt{vp})$$

This module uses proofs (or more precisely : conclusions of those proofs) and hypotheses.

- $\mathtt{d} \otimes \overline{\mathtt{k}}$ is a hypothesis,
- $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})^1$ is "proved" by the lexical item *to like*,
- $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d}) \bullet \mathtt{d} \multimap (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))$ is a correct deduction in **L**,

- the same for : $\overline{\mathtt{k}} \bullet (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))\multimap(\mathtt{d}\backslash\mathtt{vp})$,
- indices $(1, 2, 3, 4)$ relate formulae which will be linked by an axiom in the final proof

Finally, this deduction relation says that:

- if we have the hypothesis $\mathtt{d} \otimes \overline{\mathtt{k}}$
- and proofs of
  - $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})$,
  - $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d}) \bullet \mathtt{d}\multimap(\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))$
  - $\overline{\mathtt{k}} \bullet (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))\multimap(\mathtt{d}\backslash\mathtt{vp})$

then, by combining them, we can have a proof of $(\mathtt{d}\backslash\mathtt{vp})$, with this proof built with axiom links as indicated by the indices. We show this proof in Figure 1 where:

- $\mathtt{v}$ is an abbreviation for $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})$
- $\mathtt{v1}$ an abbreviation for $(\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))$
- $\mathtt{v2}$ an abbreviation for $(\mathtt{d}\backslash\mathtt{vp})$

It is important to notice about this proof that '$\bullet$' is treated like '$\otimes$' in the assembly logic. We nevertheless keep the connective '$\bullet$' for interpretation in the internal logic (the interpretation provides the correct labelling). When reading the proof from the top, the product '$\bullet$' and a deliberate order on conjuncts are introduced rather than the product '$\otimes$' simply to satisfy the requirements of the internal logic.

Of course, such a module can also be represented by a tree (because we remain in an Intuitionistic framework). This tree is given on figure 2 (with conclusions on the top, premisses and hypotheses at the bottom). Let us imagine now that we have a module associated with a $\mathtt{d}$-phrase:

$$mary : \overline{\mathtt{k}}, mary : \mathtt{d} \vdash \{mary, mary\} : \overline{\mathtt{k}} \otimes \mathtt{d} \qquad [2]$$

This module says that the item *mary* provides two informations, a categorial one ($\mathtt{d}$) and a functional one: it requires a case, something which is denoted by $\overline{\mathtt{k}}$. These two informations together give a $\otimes$-product, each component of which is labelled by the phonological form *mary*.

By applying the cut-rule between [1] and [2], we obtain:

$$\textit{to like} : ((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})^1 \otimes (((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})^1 \bullet mary : \mathtt{d}\multimap(\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))^2)\otimes$$
$$(mary : \overline{\mathtt{k}} \bullet (\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))^2\multimap(\mathtt{d}\backslash\mathtt{vp})) \vdash (\mathtt{d}\backslash\mathtt{vp})$$

$$
\dfrac{
  \dfrac{
    \dfrac{}{\mathtt{v} \vdash \mathtt{v}} \quad \dfrac{}{\mathtt{d} \vdash \mathtt{d}}
  }{
    \mathtt{v}, \mathtt{d} \vdash \mathtt{v} \bullet \mathtt{d}
  }
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{}{\overline{\mathtt{k}} \vdash \overline{\mathtt{k}}} \quad \dfrac{}{\mathtt{v1} \vdash \mathtt{v1}}
    }{
      \overline{\mathtt{k}}, \mathtt{v1} \vdash \overline{\mathtt{k}} \bullet \mathtt{v1}
    }
    \quad
    \dfrac{}{\mathtt{v2} \vdash \mathtt{v2}}
  }{
    \mathtt{v1}, (\overline{\mathtt{k}} \bullet \mathtt{v1} \multimap \mathtt{v2}), \overline{\mathtt{k}} \vdash \mathtt{v2}
  }
}{
  \dfrac{
    \mathtt{v}, (\mathtt{v} \bullet \mathtt{d} \multimap \mathtt{v1}), (\overline{\mathtt{k}} \bullet \mathtt{v1} \multimap \mathtt{v2}), \mathtt{d}, \overline{\mathtt{k}} \vdash \mathtt{v2}
  }{
    \dfrac{
      \mathtt{v}, ((\mathtt{v} \bullet \mathtt{d} \multimap \mathtt{v1}) \otimes (\overline{\mathtt{k}} \bullet \mathtt{v1} \multimap \mathtt{v2})), \mathtt{d}, \overline{\mathtt{k}} \vdash \mathtt{v2}
    }{
      \dfrac{
        \mathtt{v} \otimes ((\mathtt{v} \bullet \mathtt{d} \multimap \mathtt{v1}) \otimes (\overline{\mathtt{k}} \bullet \mathtt{v1} \multimap \mathtt{v2})), \mathtt{d}, \overline{\mathtt{k}} \vdash \mathtt{v2}
      }{
        \mathtt{v} \otimes ((\mathtt{v} \bullet \mathtt{d} \multimap \mathtt{v1}) \otimes (\overline{\mathtt{k}} \bullet \mathtt{v1} \multimap \mathtt{v2})), \mathtt{d} \otimes \overline{\mathtt{k}} \vdash \mathtt{v2}
      }
    }
  }
}
$$

Figure 1. A verbal module as a proof

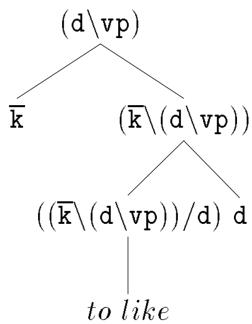

Figure 2. Partial proof-tree

Word order then follows by propagation of the labels. Labels (= words) are transmitted by axiom links, and new labels are built inside the internal logic, according to the usual conventions on labelling in Lambek grammars.
Here for instance, the label *to like* is transmitted by an axiom link to the left conjunct of the first $\bullet$-product, giving in [2] a concatenation of labels: *to like mary*, which is in its turn transmitted to the right conjunct of the second $\bullet$-product, thus finally giving a type labelled with *mary to like mary*. If we are in a SVO language, in fact the weak $\overline{\mathtt{k}}$ is *empty* thus producing *to like mary*, but if we are in a SOV language, $\overline{\mathtt{k}}$ is full and its second occurrence is deleted, according to the *Move* theory, thus resulting in *mary to like*: this labelling, depending on the parameter weak/strong associated with a feature, will be made more explicit in section 4 where the use of nets and paths defined in them will reveal more adapted to this problem.

To sum up, we have used two logics. The first (*internal*) logic is a simple base logic (Moortgat, 1997). We can choose **L** but because it seems (for the time being) that we don't need right rules, we can content ourselves with:

Functional application:

$$\alpha : A, \beta : A\backslash B \vdash \alpha\beta : B$$

$$\beta : B/A, \alpha : A \vdash \beta\alpha : B$$

Left introduction of $\bullet$:

$$\frac{\Gamma, \alpha : A, \beta : B, \Delta \vdash C}{\Gamma, \alpha\beta : A \bullet B, \Delta \vdash C}$$

Of course, this logic has *no weakening*, *no contraction* and *no permutation* rules, and therefore $\bullet$ is non commutative.

The second logic (the *external* one) combines the conclusions of proofs in the first one (some of which being simple extra-logical axioms, like those directly associated with lexical entries[1] or simple hypotheses) and considers them blocks to asembly. We can take the Multiplicative fragment of Intuitionistic Linear Logic for this task, with the following rules:

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C}[\otimes L] \qquad \frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}[\otimes R]$$

$$\frac{\Gamma \vdash A \qquad \Gamma', B, \Delta \vdash C}{\Gamma', \Gamma, A \multimap B, \Delta \vdash C}[\multimap L] \qquad \frac{\Gamma, A \vdash C}{\Gamma \vdash A \multimap C}[\multimap R]$$

$$A \vdash A \qquad [axiom]$$

$$\frac{\Gamma \vdash A \qquad \Gamma', A, \Delta \vdash C}{\Gamma', \Gamma, \Delta \vdash C} \qquad [Cut]$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} \qquad [exchange]$$

where A, B... are hypotheses, extra-logical axioms or valid sequents of the first logic, translated into linear implications, and Γ and Δ are sequences of such formulae. We assume that these formulae and their

subformulae may bear indices which indicate where axiom links have to be put (in order to make labels propagate).
In fact we shall use only a small subset of these rules: [$\otimes$ L], [axiom], [CUT] and [exchange] because modules associated with lexical entries will provide sequents with connectives *already introduced*.

3. Proof-nets

Because of the complexity of formulae and sequents, it is temptating to represent proofs by proof-nets. Moreover, proof-nets have an advantage on proof-trees (even if we have often proof-trees rather than nets for sake of simplicity): natural operations on trees are limited to substituting a tree for one leaf at the same time, whereas in proof-nets, as we shall see, the natural operation consists in linking arbitrarily complex conclusions by a cut-link, thus allowing several substitutions at the same time, something which is precisely what we want for the formalisation of *Move*. This natural operation on proof-nets prevents us from defining complex operations like adjunction or stretching when using trees.
Proof-nets are generally conceived for *one-sided* sequents: that enforces us to translate our deductions into a one-sided calculus. We shall use MLL (Multiplicative Classical Linear Logic), the rules of which are:

$$\vdash \alpha, \alpha^{\perp} \qquad [axiom]$$

$$\frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Gamma'}{\vdash \Gamma, \Gamma'}[Cut]$$

$$\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \wp B, \Delta}[\wp] \qquad\qquad \frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}[\otimes]$$

$$\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}[exchange]$$

Among all the proofs in MLL, intuitionistic proofs are distinguished by means of polarities. There are two polarities, $\bullet$, called *Input* (*negative* polarity) and $\circ$, called *Output* (*positive* polarity). The two following tables show how to give recursively its polarity to any formula from the polarities of its subformulae:

$\otimes$	$\bullet$	$\circ$
$\bullet$		$\bullet$
$\circ$	$\bullet$	$\circ$

$\wp$	$\bullet$	$\circ$
$\bullet$	$\bullet$	$\circ$
$\circ$	$\circ$	

Intuitionistic proofs are those proofs which can be polarized by means of these tables.

3.1. Proof-nets for MLL

Retoré(1996) gives a criterion for correct nets in MLL. It is based on the notion of perfect matching. In what follows, we shall present simplified forms of proof-nets: we will not have in fact to check the correctness of our PNs, just because we will start with *modules*, which are proof-nets, and because we shall connect them only by operations (cut-plugging and cut-elimination) of which we know that **they preserve any correctness criterion**. Moreover, we shall represent formulae of the internal logic with arrows in order to express non commutativity. The links with arrows are considered *black boxes* for the upper level logic: they recall the ordering convention in the internal logic (something needed for the labelling but only for it in fact), but they must be replaced by $\otimes$ in the external one, which ignores the non-commutative product.

3.2. Proof-nets associated with modules

Of course, because we represent proofs in a one-sided calculus, external formulae are transformed into their dual forms. Let us start for instance from a valid sequent for this mixed logic:

$$\mathtt{v}\otimes(((\mathtt{v}\bullet\mathtt{d})\multimap\mathtt{v1})\otimes((\overline{\mathtt{k}}\bullet\mathtt{v1})\multimap\mathtt{v2})),\overline{\mathtt{k}},\mathtt{d}\vdash\mathtt{v2}$$

It translates into:

$$\vdash \mathtt{v}^{\perp}\wp\ (((\mathtt{v}\bullet\mathtt{d})\otimes\mathtt{v1}^{\perp})\wp\ ((\overline{\mathtt{k}}\bullet\mathtt{v1})\otimes\mathtt{v2}^{\perp})),\mathtt{d}^{\perp},\overline{\mathtt{k}}^{\perp},\mathtt{v2}$$

In Figure 3, we give two correct partial proof-nets, one associated with the dualization of the verbal module [1], for the infinitive *to like*, and the other with the dualization of [2]. It is important to see how they are obtained: first, we build the tree of subformulae of each formula in each sequent, second, we link by *axiom links* (horizontal upper links) pairs of nodes labelled by dual atoms, (which communicate by axiom-sequents in the sequential proof), third we connect undischarged hypotheses (here $\mathtt{d}$ and $\overline{\mathtt{k}}$) by a $\wp$-link. Observe that $\otimes$-links and $\wp$-links are

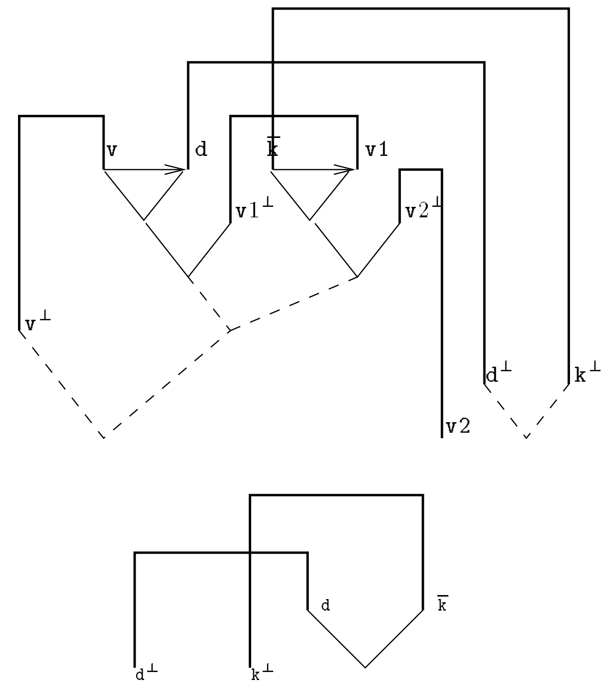

Figure 3. PN associated with a transitive verb and with a det phrase

distinct: plain lines for the first ones and dashed lines for the second ones. In Figure 4, we show how these two PNs can be plugged in order to get a new correct PN, where cut-elimination can be performed.

3.3. Proof-nets and (Pseudo-) Natural Deduction trees

Because we are in Intuitionistic Logic, the proof-nets we build up in this system have in fact a tree representation which corresponds to proofs in Natural Deduction format.
In order to transform a proof-net into a Natural Deduction tree, we perform the following operations.

- a negative tensor with positive premise A is replaced by a single formula A,
- nodes of opposite polarities related by an axiom link are identified, that means transformed into single positive nodes,
- negative $\wp$ -links may be ignored, except if they may be associated by a cut with a conclusion of another module,

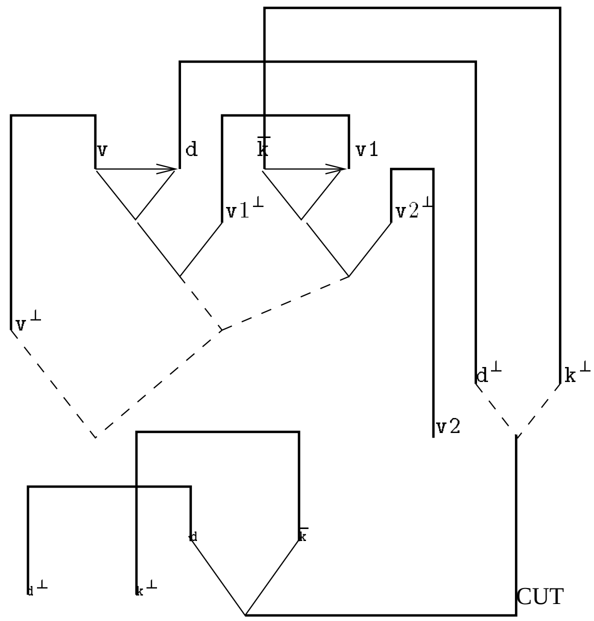

Figure 4. Plugging two PNs

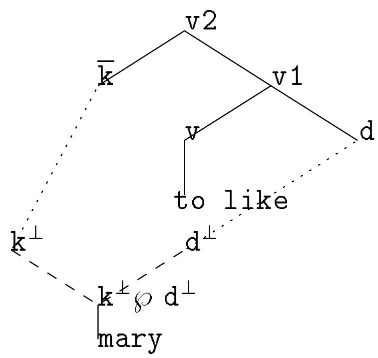

Figure 5. Pseudo-Natural Deduction tree associated with Figure 4

- in case a negative ℘ -link has to be associated by a cut with a conclusion of another module, its conclusion is directly connected to its components in the tree. When the cut is eliminated, this connection is suppressed (with the ℘ -link) and replaced by a coindexation between the components in question.

For instance, figure 5 shows the translation of the proof-net obtained by plugging [1] and [2].

These trees will be called *pseudo ND-trees* of course because they are

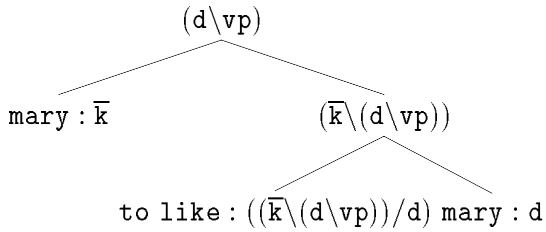

Figure 6. Tree equivalent to the PN after cut-elimination

not trees properly speaking, but after cut-elimination, we get back to ordinary trees, like the one given in Figure 6.

4. Paths

4.1. General definition

Because of their polarization, proof-nets allow the definition of paths, which are very similar to those paths used by F. Lamarche (Lamarche, 1995) in finding a correctness criterion for Proof-Nets for Intuitionistic Linear Logic (*Essential Nets*). Lamarche's paths have the following definition:
Let us assume that x, y, z... denote nodes, and u, v, w ... denote sequences of nodes representing paths. Let y be the unique positive root of an essential net. Let Node($\mathbf{A}$) the set of nodes of $\mathbf{A}$. Path($\mathbf{A}$) is the smallest nonempty set Path($\mathbf{A}$) $\subset Node(\mathbf{A})^*$ closed under the conditions below:

- **Root** y$\in$Path($\mathbf{A}$).
- **Up** If u.z$\in$Path($\mathbf{A}$) and if z' is positive such that its predecessor is z (in the tree-order from the root to the leaves), then:

  $$\text{u.z.z'} \in \text{Path}(\mathbf{A}).$$

- **Down** If u.z$\in$Path($\mathbf{A}$) and z is negative and its predecessor z' is also negative, then u.z.z'$\in$Path($\mathbf{A}$).
- **DnTurn** If u.z$\in$Path($\mathbf{A}$) and z is positive and z' is linked to z by an axiom link then u.z.z'$\in$Path($\mathbf{A}$).

Let us call Path'($\mathbf{A}$) the set of *reverse* paths w.r.t. paths belonging to Path($\mathbf{A}$), and starting from terminal input-nodes and directed towards the final output.

These paths are used to produce *interpretations*. We shall restrict our attention to *phonological* interpretations.
For that, we imagine several *tokens* firing at the same time and starting from terminal input nodes. These phonological tokens meet at $\bullet$ nodes and they merge at these nodes according to the labelling of functional application rules.
A consequence of the convergence of reverse paths to the final output is that a phonological token will always reach this output, and that it will be made of the totally ordered set of phonologies.

4.2. Travels along paths determined by weak and strong features

Tokens starting from $\wp$ -conclusions may have different trips according to the relative strengths of their premisses. Let us preliminarilly define a notion of heighth for (sub)formulae, in a given proof.

DEFINITION 1. *An occurrence of a (sub)formula a is said to be immediately higher than an occurrence of a (sub)formula b (a>b) in a proof-net π if and only if:*

- *these two occurrences belong to a formula $p(b)\multimap a$, where $p(b)$ denotes a $\bullet$-product,*
- *or a is linked by an axiom link to a formula a' which is such that a'>b,*
- *or a is a sister (= premisse of the same conclusion) of a (sub)-formula a' such that a'>b.*

The relation $a \geq^* b$ is the transitive closure of the relation a>b. This allows us to define the following trips for tokens:

Trips for phonological tokens:
Let $a_1{}^\perp \wp\ a_2{}^\perp \wp \ldots a_{n-1}{}^\perp \wp\ a_n{}^\perp$a $\wp$ -conclusion (or *chain*), where a_n is the only categorial feature, and all the other ones represent functional features (like $\overline{\texttt{k}}$, $\overline{\texttt{wh}}$ etc.) such that their duals a_1, a_2, ..., a_n are totally ordered for the relation $\geq^*$ (a_1 being the highest and a_n the lowest in some proof-net π)

- if among the a_i (i<n) there are strong features, the *full* phonology associated with the chain travels through the highest strong feature, and empty phonologies travel through all the other features,
- if there is no strong feature, the *full* phonology travels through the categorial feature a_n[2].

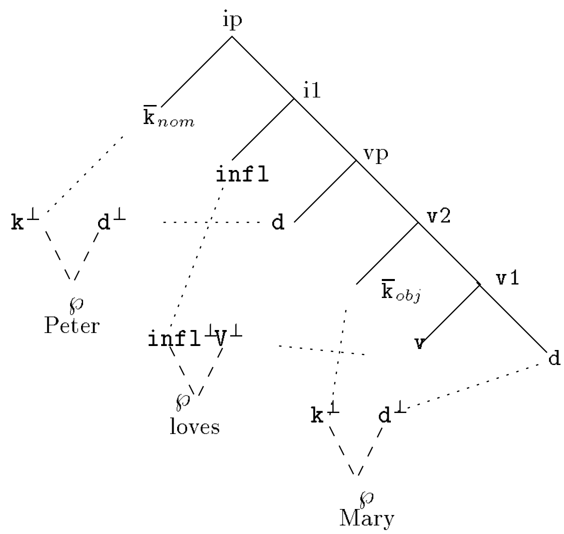

Figure 7. Peter loves Mary

Figure 7 shows the (simplified) complete module associated with the sentence

Peter loves Mary

The phonology *Peter loves Mary* results from the fact that in English, the nominative case is supposed to be strong, inflection weak and accusative also weak.
A strong inflection with weak nominative and accusative cases would results in a VSO order and a weak inflection with a strong accusative case in a SOV order.

5. More examples

5.1. Nominative case assignment

The previous example showed how to raise a `d` with regards to an infinitival verbal head in order for it to receive an accusative case. By doing so, we get an object of type $(\mathtt{d}\backslash\mathtt{vp})$ which is still waiting for another `d` and this other `d` also needs case but, according to the case-theory, it must come from an inflection head. Finally, in order to select the `d`-subject, the verbal module must be completed by an *inflection*-module, in order to obtain a *module for an inflected verb*. Let us make explicit this machinery for constructing *intermediate modules* by means of more elementary ones.

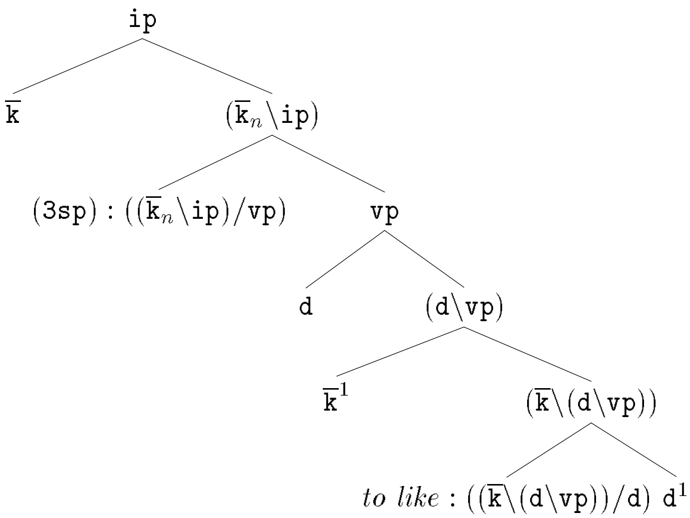

Figure 8. Module for an inflected form

The proper *inflection module* is associated with the following sequent, completed with indices for showing axiom-links.

$$\begin{array}{c} inflection \\ (\mathtt{d}\backslash\mathtt{vp})^1, \mathtt{d}^2 \otimes \overline{\mathtt{k}}^3, (\mathtt{d}^2 \bullet (\mathtt{d}\backslash\mathtt{vp})^1 {-\!\circ} \mathtt{vp}^4) \otimes ((\overline{\mathtt{k}}_n\backslash\mathtt{ip})/\mathtt{vp})^5 \\ \otimes(((\overline{\mathtt{k}}_n\backslash\mathtt{ip})/\mathtt{vp})^5 \bullet \mathtt{vp}^4 {-\!\circ} (\overline{\mathtt{k}}_n\backslash\mathtt{ip})^6) \otimes (\overline{\mathtt{k}}^3 \bullet (\overline{\mathtt{k}}_n\backslash\mathtt{ip})^6 {-\!\circ} \mathtt{ip}) \vdash \mathtt{ip} \end{array}$$

where:

- $(\mathtt{d}\backslash\mathtt{vp})$, $\mathtt{d} \otimes \overline{\mathtt{k}}$ are hypotheses,
- $((\overline{\mathtt{k}}_n\backslash\mathtt{ip})/\mathtt{vp})$ is proved by an auxiliary (*may*, *will*, ...) or by a terminal inflection (*-s*), or rather by an "abstract" morphological feature like $(\mathtt{3sp})$ (for third person - singular - present).

This module can be plugged to the verbal one by means of the cut rule, using the cut-formula $(\mathtt{d}\backslash\mathtt{vp})$. Cut elimination and representation in pseudo ND trees lead to Figure 8. where the index indicates that the two nodes are linked to the same $\wp$ -formula.
We link the nodes $\overline{\mathtt{k}}$ and $\mathtt{d}$ which are still free to a same $\wp$ -formula $\overline{\mathtt{k}}^{\perp} \wp\ \mathtt{d}^{\perp}$, in such a way that any nominal phrase, like *Peter*, of type $\mathtt{d}\otimes\overline{\mathtt{k}}$ can be plugged into that tree.
If $((\overline{\mathtt{k}}_n\backslash\mathtt{ip})/\mathtt{vp})$ is proved by an auxiliary like *will*, the construction ends up, but if it is proved by some "abstract" feature like $(\mathtt{3sp})$, such an abstract feature must be erased, it will be made by connecting $\mathtt{infl}^{\perp}$ and $\mathtt{v}^{\perp}$ by a specific $\wp$ -link, thus producing the conclusion $\mathtt{infl}^{\perp} \wp\ \mathtt{v}^{\perp}$

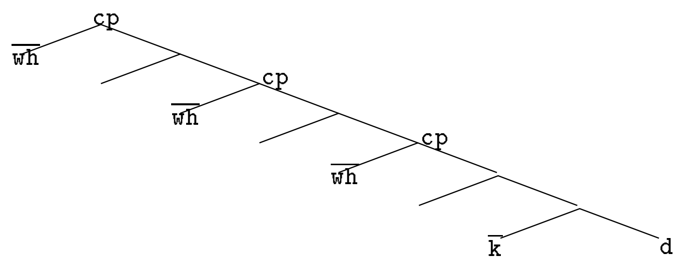

Figure 9. A big $\mathtt{cp}$-module

(where $((\overline{\mathtt{k}}_n\backslash\mathtt{ip})/\mathtt{vp})$ is abbreviated by $\mathtt{infl}$ and $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})$ by $\mathtt{v}$), labelled by the *inflected verbal form*. We thus obtain the module associated with a given inflected form. We may obviously assume this module is already given in the lexicon[3].
Again, the weak/strong opposition can be marked: categorial features like here the inflection head may also be strong, thus resulting in a raising move of the phonological content of the attracted verbal head to the inflection head.

5.2. Cyclic movements

Cyclic movement supposes that a same feature is consumed several times. Linear logic admits such situations by allowing us to use so-called *exponentials*. Formulae which are !-marked can be contracted and weakened when on the left-hand side of an intuitionistic sequent. There is also a transition to non-marked formulae by the *Dereliction*-rule:

$$\frac{\Gamma, A \vdash B}{\Gamma, !A, \vdash B} \qquad [D]$$

Let us suppose for instance we have a *wh*-expression which moves up to the *Spec*-$\mathtt{cp}$ position of an embedding sentence (a phenomenon also known as unbounded dependency), this can be represented by a module associated with a sequent which proves $!\overline{\mathtt{wh}} \otimes \overline{\mathtt{k}} \otimes \mathtt{d}$, like for instance:

$$whom : !\overline{\mathtt{wh}}, whom : \overline{\mathtt{k}}_a, whom : \mathtt{d} \vdash$$

$$\{whom^*, whom, whom\} : !\overline{\mathtt{wh}} \otimes \overline{\mathtt{k}} \otimes \mathtt{d}$$

In order to plug this module at its proper place, a big $\mathtt{cp}$-module has to be built by combining smaller ones, which can have several $\overline{\mathtt{wh}}$, like in Figure 9. The combination of the two modules results in linking the components of:

$$?\overline{\mathtt{wh}}^{\perp} \wp\ \overline{\mathtt{k}}^{\perp} \wp\ \mathtt{d}^{\perp} \text{to the features } \overline{\mathtt{wh}}, \overline{\mathtt{wh}}, \overline{\mathtt{wh}}, \overline{\mathtt{k}}, \mathtt{d}$$

and if $\overline{\mathtt{wh}}$ is a strong feature, according to the previous convention, the phonological content raises up to the highest $\overline{\mathtt{wh}}$-node after having filled all the other $\overline{\mathtt{wh}}$-nodes, and of course the lower $\overline{\mathtt{k}}$ and $\mathtt{d}$.
The requirement "after having filled all the other $\overline{\mathtt{wh}}$-nodes" is nevertheless problematic because nothing prevents in pure logic that only some of the nodes be connected, (and even none!).
At this point, we are enforced to delimitate some particular proofs in the space of all proofs: the proofs which will be admissible for our purpose.

DEFINITION 2. *A proof will be said Move-admissible iff every cut-tensor formula* $\overline{\mathtt{f}}^m \otimes \mathtt{a}_1 \otimes ...\mathtt{a}_n$ *(where* $\overline{\mathtt{f}}^m$ *means the* $\otimes$*-product of* m *occurrences of* $\overline{\mathtt{f}}$*) which is built up in order to correspond to a given chain* $?\overline{\mathtt{f}}^\perp \wp\ \mathtt{a}_1{}^\perp \wp\ ...\wp\ \mathtt{a}_n{}^\perp$ *is such that if the items* $\overline{\mathtt{f}}$, $\mathtt{a}_1$, $\ldots$, $\mathtt{a}_n$ *are in the total order given by the relation* $\geq^*$*, there is no occurrence of* $\overline{\mathtt{f}}$ *strictly in between the highest one and* $\mathtt{a}_1$ **which is not inside the product or which belongs to a subformula which is not inside the product**.

6. Conclusion and further generalization

One could think equally possible to develop a proof system based on formulae in which correct sentences would result from all the correct proofs in that system. Many attempts to do that have shown that we must always add special constraints in order to admit only some proofs, those which correspond to some mysterious *economy principle*. The advantage of our system is that (except when using exponentials) we don't have to express such general extra-logical conditions. This is so because we are working with ready made proofs where the problematic issues are already solved. Because our "modules" are combined by means of complex cut-formulae, we can directly have correct associations (cf. which $\overline{\mathtt{k}}$ goes with which $\mathtt{d}$ for instance?). It is only when we have to introduce a $\wp$ -link or generally when we are obliged to *make* a cut-formula (that means a cut-formula not already given from the lexical entry or the intermediate modules) that difficulties come and compel us to restrict the space of all proofs.
This conception of minimalist grammars is similar to Cornell's in (Cornell, 1998), but it makes use of logical concepts already existing in order to make a link with resource logics. Such a link would probably help in implementing grammars of this kind by means of Programming Languages based on Linear Logic (like Lolli).
Dealing with questions like *island* constraints is still missing. We plan

to treat such questions by means of more values for the strength parameter. If besides `w` (for *weak*) and `s` (for *strong*), a third value `b` is admitted, for *blocked*, we are able to give an account of examples like:

**John is likely that will leave*

simply by assuming that *John*'s phonology is blocked at a `b`-feature and that nominative $\overline{\mathtt{k}}$ associated with the inflectional head has the value `b`.

Moreover, this particular view on minimalist grammars seems to offer a new opportunity to see grammatical representations as networks, where *nodes replace features*. This may be an important turn just because features can be viewed as connecting nodes and their parameters weak or strong as the *strength of these connections*. This strength determines the way in which phonological tokens are travelling. We have no room here to explain the "semantic trips", but that would be similar. Actually, the semantic information ignores the weak/strong distinction and always goes up to the highest node of the chain it is associated with.

We can see this as an opposition between the stability of the information system and the unstability of the phonetic-perceptive one. If a feature is wrongly assigned a weak value instead of a strong one, that will result in a so-called "mistake" but most of the time, the sentence will be still correctly semantically interpreted. Let us imagine for instance an *in situ* question like: **Mary reads which book?*.

We thus can assume that in learning language, children only learn to specify the nodes strength and how sometimes to change it for reasons of topicalization, yes-no questions and so on. Of course there can be variations from an individual to another one, such variations could be represented if we affected weights to nodes instead of the rough opposition weak/strong.

Acknowledgements

I thank Christian Retoré and François Lamarche for their fruitful suggestions. I also thank the three referees who helped me to correct this paper and gave me suggestions to improve it. If I failed to do it, it is obviously only my own responsability.

Notes

[1] It is the case of: $((\overline{\mathtt{k}}\backslash(\mathtt{d}\backslash\mathtt{vp}))/\mathtt{d})$ which is "proved" by the lexical item *to like*

[2] We also think to introduce a third modality for the strength parameter: *blocking* - see section 6 - which would behave like a feature which captures a full phonology,

without allowing it to climb towards a higher position: this seems to be legitimate with regards to islands phenomena.

[3] We thus let open the question whether the lexicon must include only lemmas and morphemes or all the forms.

References

Chomsky, N.: 1996, *The Minimalist Program.* The MIT Press.

Cornell, T.: 1998, 'Derivational and Representational Views of Minimalist Transformational Grammar'. In: A. Lecomte, F. Lamarche, and G. Perrier (eds.): *Logical Aspects of Computational Linguistics.*

Joshi, A. and S. Kullick: 1997, 'Partial Proof Trees, resource sensitive logics and syntactic constraints'. in (Retoré, 1997), pp. 21–42.

Lamarche, F.: 1995, 'Proof-nets for Intuitionistic Linear logic: Essential nets'. Technical report, Imperial College, London.

Moortgat, M.: 1997, 'Categorial Type Logics'. In: J. van Benthem and A. ter Meulen (eds.): *Handbook of Logic and Language.* Elsevier, Chapt. 2, pp. 93–178.

Retoré, C.: 1996, 'Perfect matchings and series-parallel graphs: multiplicative proof-nets as BR-graphs'. In: J. Girard, M. Okada, and A. Scedrov (eds.): *Linear'96*, Vol. 3 of *Electronic Notes in Theoretical Computer Sciences.*

Retoré, C. (ed.): 1997, 'Logical Aspects of Computational Linguistics', Vol. 1328 of *LNCS/LNAI.* Springer.

Stabler, E.: 1997, 'Derivational minimalism'. in (Retoré, 1997), pp. 68–95.

Address for Offprints:
615, Rue du Jardin Botanique
BP101-54602-Villers-les-Nancy
France