Denotational semantics for polarized (but non-constrained) LK by means of the additives

Lorenzo Tortora de Falco*

Équipe de Logique Mathématique, Université Paris VII

Abstract. Following the idea that a classical proof is a superimposition of 'constructive' proofs, we use the linear connective "&" to define an embedding P^a of polarized but non-constrained LK in linear logic, which induces a denotational semantics. A classical cut-elimination step is now a cut-elimination step in one of the 'constructive slices' superimposed by the classical proof.

1 Introduction

Much work has been done in the past 6-7 years to extract the computational content from classical proofs. On the proof-theoretical side, let's quote for example [Girard 91] (LC), [Parigot 91] (FD), [DJS 95] (LK^{tq} and its restrictions) and [BarBer 95] (the symmetric λ-calculus).
With the notable exception of the symmetric λ-calculus of [BarBer 95] which enjoys only strong normalization, all the systems previously mentioned enjoy the usual good computational properties (strong normalization and confluence) and have a denotational semantics. To obtain such good properties, it is necessary to cope with the "non-determinism" of classical cut-elimination: this is done by equipping classical formulas with an index, a colour or a polarity (depending on the personal taste of the authors), which will tell us how to perform cut-elimination in case of conflict.
But it turned out (perhaps a bit surprisingly) that adding this information is not enough to fix classical logic: there are rules that are "difficult" to handle, in the sense that if we don't restrict their application to the premises of a particular form, then the elimination of certain 'logical cuts' (Gentzen 's key-cases) yields *two* proofs. What is catastrophic is the fact that the identification of these two proofs leads to the collapse of the whole system: we have to identify all the proofs of a given sequent (see [Girard 91] and [DJS 95]).
The solution proposed by the works previously mentioned is either to avoid such difficult rules (this is possible keeping completeness w.r.t. classical provability, see [Parigot 91] and [BarBer 95]) or to restrict the application of such difficult rules to the premises which lead to key-cases the elimination of which is unproblematic (see [Girard 91] and [DJS 95]).
A trace of this restriction is the fact that the translations from LK (with polarized or coloured formulas) to LC or LK^{tq} (and its subsystems) are far from

* tortora@logique.jussieu.fr.

being deterministic: there are two possible ways to translate the difficult rules with certain premises. The same holds for the translations in linear logic (LL), which are the actual starting point of both the works [Girard 91] and [DJS 95]. However, these translations suggest that the computational object which lies 'behind' a classical proof *is not* a linear proof, but rather a superimposition of linear proofs. This work is a first attempt to turn this intuition into a mathematical statement: "there exists a denotational semantics of polarized (but non-constrained) LK in coherent spaces".

Trying to take seriously the idea of superimposition, we define inductively an embedding P^a of polarized LK-proofs in LL, which is an extension of the P-embedding defined by Girard in [Girard 91]: when we meet a "difficult" rule, then, instead of choosing one of the two possible linear translations, we superimpose them.

Technically speaking, the obvious candidate to superimpose proofs is the linear connective "&". Following one of the main points of [Tortora 96] (namely the "independence lemma"), to make sure that the components of the additive boxes introduced by P^a do not interact during cut-elimination, we keep these additive boxes at exponential depth zero. This is done by choosing as main doors for these additive boxes a "purely negative" formula (in the sense of [Girard 91]), namely $\bot \& \bot$: this makes the additive boxes "commute" (semantically speaking) with all the exponential ones.

Thus, we translate the classical proof α by the linear proof $P^a(\alpha)$, and then, by "delaying" as much as possible all the additive boxes introduced by P^a, we obtain a superimposition of linear proofs with the same denotational semantics of $P^a(\alpha)$.

This suggests to define a classical cut-elimination step as a cut-elimination step in *one* of the "slices" that the classical proof superimposes. We obtain in such a way a denotational semantics (induced by P^a) for polarized LK, which still satisfies all the computational isomorphisms of LC (see [Girard 91]).

2 Preliminaries: linear logic and classical logic, additives

2.1 Linear logic and classical logic

The use of LL as a tool to analize classical cut-elimination is at the core of [Girard 91] and [DJS 95].

Girard observes that if A and B are linear formulas of type "!" (resp. of type "?"), that is if $A \vdash !A$ (resp. $?A \vdash A$) is provable in LL, then so are the formulas $A \otimes B$ and $A \oplus B$ (resp. $A \wp B$ and $A \& B$), and that A is of type "!" (resp. "?") iff $A^\perp$ is of type "?" (resp. "!"). This suggests a translation P from classical to linear logic, which translates each classical formula by a linear formula of type "!" or of type "?". To apply structural operations directly to formulas of type "?" (and not only to formulas starting with "?"), Girard slightly modifies the coherent denotational semantics of LL and introduces the semantics of correlation spaces for the linear fragment necessary to interpret classical logic. He also proves that the main

properties of classical connectives can be recovered at the computational level (i.e. are isomorphisms in the semantics).
Danos, Joinet and Schellinx work directly with special translations in LL (called "decorations") which suggest a cut-elimination procedure (the tq-protocol) for different versions of coloured second order classical sequent calculus (LK^{tq}, LK^{η}, LK^{η}_{pol}). They show that Girard 's LC is a subsystem of LK^{η}_{pol} (up to the stoup/no stoup formulation, see subsection 2.3), that the P-embedding defined by Girard is suitable for the whole (first order) LK^{η}_{pol}, and that the cut-elimination procedure defined by Girard for LC is the tq-protocol restricted to LC.
In [QT 96a], we prove that the P-embedding induces a denotational semantics (to which we will also refer to as the P-semantics) in coherent spaces for first order LK^{η}_{pol} (then for LC): if π and π' are LK^{η}_{pol}-proofs, from $\pi \twoheadrightarrow_{tq} \pi'$ we deduce $[P(\pi)] = [P(\pi')]$ (where $[d]$ denotes the coherent semantics of the linear proof d). In [QT 96b], we clarify the links between first order LK^{η}_{pol} and LC: LC is obtained from the multiplicative fragment of LK^{η}_{pol} by decomposing the multiplicative conjunction between two negative formulas into an additive one and a multiplicative conjunction with "V" (the classical unit for the multiplicative conjunction).

2.2 About the additives

In this subsection, we briefly recall the independence lemma, proven in [Tortora 96]. Although we will not use this result (so that this subsection is not, strictly speaking, necessary to understand the content of the paper), it has been one of the main sources of inspiration for the present work. We assume here that the reader has some acquaintance with the proof-nets of LL defined in [Girard 87].
If T and T' are two proof-nets s.t. $T \twoheadrightarrow T'$ (i.e. T' is obtained from T by applying some steps of cut-elimination), then any additive box B of T' comes from exactly one additive box $\overleftarrow{B}$ of T: we say that B is a residue of $\overleftarrow{B}$ in T'.

Lemma 1 ***(Independence lemma).*** *Let T, T' be two proof-nets, s.t. $T \twoheadrightarrow T'$. Let $\mathbb{B}$ be an additive box with exponential depth 0 in T, and $\mathbb{B}'$ a residue of $\mathbb{B}$ in T'. If $\mathbb{B}'$ has additive depth p and $B_1, ..., B_p$ are the additive boxes containing $\mathbb{B}'$ in T', then for all residue $\mathbb{B}''$ of $\mathbb{B}$ in T' (different from $\mathbb{B}'$), there exists $i \in \{1, ..., p\}$ s.t. $\mathbb{B}''$ is a subproof-net of the component of B_i which does not contain $\mathbb{B}'$.*

A possible way of reading this lemma (connected to the present paper) is the following: "normalization preserves superimposition *at exponential depth* 0".

2.3 Notations and conventions

We shall use the conventions of [Girard 91], thus distinguish between *positive* and *negative* formulas and take a one-sided version of sequent calculus, rather than distinguish between formulas coloured q and t and take a two-sided version of sequent calculus like in [DJS 95].

For classical formulas we take the following conventions:

P, Q, R,... will denote positive formulas

L, M, N,... will denote negative formulas

A, B, C,...will denote formulas that may be either positive or negative.

We say that a formula (or, better said, an occurrence of formula) A in a proof π of any subsystem of LK is *main*, if A is either one of the (two) conclusions of an axiom, or the main conclusion of a logical rule of π.

In the sequel we will call LC the extension of Girard 's original system to first order LK^{η}_{pol}. (We don't call it LK^{η}_{pol} because we will use a version which makes use of "the stoup", and this is more in the spirit of LC, and also because we don't want to create confusions with the system LK_{pol} defined in the next section).

To be precise one should note that there are LC-rules that are not (strictly speaking) rules of LK^{η}_{pol}. This is because the constraint on proofs induced by Girard's stoup is (slitghly) less strong than [DJS 95]'s η-constraint: an attractive formula (i.e. necessary positive in the one-sided version) which is active in an irreversible rule has to be main following [DJS 95], and only main up to reversible, structural and (some kind of) cut rules on the context, following [Girard 91]. From the semantical point of view the two constraints are the same: to each LC-proof one can associate several LK^{η}_{pol}-proofs, but the denotational semantics induced by the P-embedding in LL equates all of them. But syntactically speaking, in LC there are for example two ways of performing certain logical steps: if $\gamma = cut(\alpha, \beta)$ is a proof of $\vdash \Gamma, \Delta; \Pi$ obtained from the proof α of $\vdash \Gamma; R \wedge_m Q$ and the proof β of $\vdash \Delta, \neg R \vee_m \neg Q; \Pi$ by applying a logical cut rule, and if we call γ_1 and γ_2 the two proofs obtained from γ by performing the two cuts with active formulas $R/\neg R$, $Q/\neg Q$ in the two possible ways, then γ_1 and γ_2 might have two syntactically different normal forms w.r.t. the tq-protocol of cut-elimination (of course the normal forms will anyway be equal in the semantics).

So, what we call LC in this paper, is the system LK^{η}_{pol} (with the stronger η-constraint on proofs), the only *notational* difference being that we will write sequents making use of the stoup (this is particularly helpful for our purposes, as it will appear clearly in the next section).

In the whole paper, we will apply directly the structural operations to any linear formula A of type "?", implicitely meaning (if A does not actually start with "?") that we first apply a dereliction to A, then the structural operation to $?A$, and finally we cut with the proof of $?A \vdash A$ (with $\vdash A, !A^{\perp}$ in the one-sided version). One can also think directly in terms of correlation spaces (which, by the way, is proven to be the same in [QT 96a]).

If B and C are the immediate subformulas of A, an η-proof of A is the proof of $\vdash \neg A, A$ consisting in the axioms $\vdash \neg B, B$ and $\vdash \neg C, C$, and precisely one instance of each of the logical rules introducing A (or $\neg A$)'s main connective. We will also call "piece(s) of η-proof" the subproof(s) of η, obtained from η by removing the last (reversible) rule.

3 LK_{pol} and the embedding P^a in linear logic

We consider here a version of the sequent calculus LK where each formula comes equipped with a polarity, in the style of [Girard 91].
The real difference between the approach we present here and the ones previously mentioned lies in the fact that we *do not* restrict the rules of the calculus LK. Both in [Girard 91] and in [DJS 95] there is a restriction in the application of certain rules ("the irreversible ones in which one of the active formulas of one of the premises is positive"): as stated, this is the reason for the introduction of "the stoup" in [Girard 91] and of the "η-constraint" in [DJS 95].

3.1 The sequent calculus LK_{pol}

LK_{pol} is nothing but the usual calculus LK (where we distinguish between additive and multiplicative rules) for first order classical logic. However, in LK_{pol}, each formula comes equipped with a polarity (it is positive or negative), which is completely determined by the polarity of the atoms: the polarity of compound formulas is obtained by looking at the main connective of the formula, if it is $\vee_m$ or $\wedge_a$ (resp. $\wedge_m$ or $\vee_a$) then the formula is negative (resp. positive). Of course, for any formula A, A and $\neg A$ have different polarities (see [Girard 91] and [DJS 95] for more details).
Moreover, we "decorate" (not in the technical sense of [DJS 95]!) each sequent of this polarized LK by means of the symbol ";", where on the right of ";" there is at most one *positive* formula. This is of course in the spirit of [Girard 91]. However, our calculus does not put any constraint on proofs. The following system (which will be called LK_{pol}) shows how it is possible to "decorate" by means of ";", in a unique way, *any* polarized proof α of LK, in such a way that all the sequents of α have the form $\vdash \Gamma; \Pi$, where Γ is a set of (polarized) formulas and Π is a set containing at most one formula, which has to be positive.

Axiom:

$$\vdash \neg P; P$$

Cut-rules:

$$\dfrac{\begin{matrix}\gamma\\ \vdots\\ \vdash \Gamma; P\end{matrix} \qquad \begin{matrix}\delta\\ \vdots\\ \vdash \Delta, \neg P; \Pi\end{matrix}}{\vdash \Gamma, \Delta; \Pi} \qquad \dfrac{\begin{matrix}\gamma\\ \vdots\\ \vdash \Gamma, P; \Pi\end{matrix} \qquad \begin{matrix}\delta\\ \vdots\\ \vdash \Delta, \neg P; \Pi'\end{matrix}}{\vdash \Gamma, \Delta, \Pi'; \Pi}$$

Structural rules:

$$\dfrac{\begin{matrix}\beta\\ \vdots\\ \vdash \Gamma; \Pi\end{matrix}}{\vdash \Gamma, A; \Pi} \qquad \dfrac{\begin{matrix}\beta\\ \vdots\\ \vdash \Gamma, A, A; \Pi\end{matrix}}{\vdash \Gamma, A; \Pi}$$

Logical rules:

1) constants:

$$\vdash ; V \qquad \vdash \neg F; \Pi$$

2) unary logical rules:

$$\frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, A, B; \Pi\end{array}}{\vdash \Gamma, A \vee_m B; \Pi} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, A; P\end{array}}{\vdash \Gamma, A \vee_m P;}$$

$$\frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma; P\end{array}}{\vdash \Gamma; P \vee_a B} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, P; \Pi\end{array}}{\vdash \Gamma, P \vee_a B; \Pi} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, N; \Pi\end{array}}{\vdash \Gamma, \Pi; N \vee_a B}$$

3) binary logical rules:
For the connective $\wedge_a$ we have

$$\frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, A, \Pi'; \Pi & \vdash \Gamma, B, \Pi; \Pi'\end{array}}{\vdash \Gamma, A \wedge_a B, \Pi, \Pi';} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, A; \Pi & \vdash \Gamma, B; \Pi\end{array}}{\vdash \Gamma, A \wedge_a B; \Pi}$$

where we obviously suppose $\Pi \neq \Pi'$, and

$$\frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, A; \Pi & \vdash \Gamma, \Pi; P\end{array}}{\vdash \Gamma, A \wedge_a P, \Pi;} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma; P & \vdash \Gamma; Q\end{array}}{\vdash \Gamma, P \wedge_a Q;}$$

The case of the connective $\wedge_m$ is the more complicated, and we have to distinguish all the possible cases for the polarity of the active formulas:

$$\frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma; P & \vdash \Delta; Q\end{array}}{\vdash \Gamma, \Delta; P \wedge_m Q} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, P; \Pi & \vdash \Delta, Q; \Pi'\end{array}}{\vdash \Gamma, \Delta, P \wedge_m Q, \Pi, \Pi';} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma; P & \vdash \Delta, Q; \Pi\end{array}}{\vdash \Gamma, \Delta, P \wedge_m Q; \Pi}$$

$$\frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, M; \Pi & \vdash \Delta, N; \Pi'\end{array}}{\vdash \Gamma, \Delta, \Pi, \Pi'; M \wedge_m N} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma, P; \Pi & \vdash \Delta, N; \Pi'\end{array}}{\vdash \Gamma, \Delta, P \wedge_m N, \Pi'; \Pi} \qquad \frac{\begin{array}{cc}\gamma & \delta\\ \vdots & \vdots\\ \vdash \Gamma; P & \vdash \Delta, N; \Pi'\end{array}}{\vdash \Gamma, \Delta, \Pi'; P \wedge_m N}$$

First order quantifiers:

$$\frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, A; \Pi\end{array}}{\vdash \Gamma, \forall x A; \Pi} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma; P\end{array}}{\vdash \Gamma, \forall x P;}$$

with the usual restriction "x is not free in Γ, Π", and

$$\frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, N[t/x]; \Pi\end{array}}{\vdash \Gamma, \Pi; \exists x N} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma; P[t/x]\end{array}}{\vdash \Gamma; \exists x P} \qquad \frac{\begin{array}{c}\beta\\ \vdots\\ \vdash \Gamma, P[t/x]; \Pi\end{array}}{\vdash \Gamma, \exists x P; \Pi}$$

We can add to these rules the "classical dereliction", which allows to deduce $\vdash \Gamma, \Pi;$ from a proof of $\vdash \Gamma; \Pi$.

3.2 The embedding P^a of LK_{pol} in linear logic

We are going to define P^a, an extension of the P-embedding defined by Girard in [Girard 91], which allows to translate in LL also all the rules of LK_{pol} which are not rules of LC: the irreversible rules ($\vee_a$, $\wedge_m$ and $\exists$) with a positive active formula which is not main.
We translate formulas in the usual P-way (see [Girard 91] and [DJS 95]), and the sequent $\vdash \Gamma_{neg}, \Gamma_{pos}; \Pi$ by the sequent $\vdash P(\Gamma_{neg}), ?P(\Gamma_{pos}), P(\Pi), \vec{\perp}$, where $\vec{\perp}$ indicates a certain number of occurrences of the formula $\perp \& \perp$, and Γ_{neg} (resp. Γ_{pos}) is the subset of Γ containing all the negative (resp. positive) formulas of Γ.

Remark: Observe that $\perp$ and $\perp\&\perp$ are linear formulas of type "?", so that (following our conventions) they can be active in a structural rule, or context in a promotion rule.

In the sequel we will write directly A instead of $P(A)$ and $?\Gamma$ instead of $P(\Gamma_{neg}), ?P(\Gamma_{pos})$.
The embedding P^a from the set of LK_{pol}-proofs to the set of linear proofs is defined inductively in the following way:
- for the rules of LK_{pol} which are also rules of LC we translate in the obvious way, and the previous remark says precisely that the set of formulas $\vec{\perp}$ does not introduce any problem.
However, even among these rules, there is a rule which has to be carefully translated: it is the $\wedge_a$. We proceed in the following way (if the main conclusion of the rule $\wedge_a$ is $A \wedge_a B$, suppose for example that A is positive and not in the stoup, B is negative, and the two formulas in the two stoups are different):

$$
\begin{prooftree}
\AxiomC{$P^a(\gamma)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma, ?P, ?\Pi', \Pi, \overrightarrow{\perp}_1$}
\UnaryInfC{$\vdash ?\Gamma, ?P, ?\Pi', ?\Pi, \overrightarrow{\perp}_1$}
\doubleLine
\RightLabel{W}
\UnaryInfC{$\vdash ?\Gamma, ?P, ?\Pi', ?\Pi, \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\AxiomC{$P^a(\delta)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma, N, ?\Pi, \Pi', \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Gamma, N, ?\Pi, ?\Pi', \overrightarrow{\perp}_2$}
\doubleLine
\RightLabel{W}
\UnaryInfC{$\vdash ?\Gamma, N, ?\Pi', ?\Pi, \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\BinaryInfC{$\vdash ?\Gamma, ?P \& N, ?\Pi', ?\Pi, \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\end{prooftree}
$$

The important point here is that we do not "mix" formulas coming from $\overrightarrow{\perp}_1$ and formulas coming from $\overrightarrow{\perp}_2$, even if they are the same (see also the remark following definition 5).

- for the rules of LK_{pol} that are not rules of LC, let's consider some of the rules introducing the connective $\wedge_m$ which contain all the difficult cases:

• suppose that in the premisses the active formula P is out of the stoup and the main conclusion of the rule is $P \wedge_m N$, then we translate by

$$
\begin{prooftree}
\AxiomC{$P^a(\delta)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Delta, N, \Pi', \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Delta, N, ?\Pi', \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Delta, !N, ?\Pi', \overrightarrow{\perp}_2$}
\AxiomC{$P^a(\gamma)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma, ?P, \Pi, \overrightarrow{\perp}_1$}
\AxiomC{$\vdash P, P^\perp$}
\AxiomC{$\vdash N^\perp, N$}
\UnaryInfC{$\vdash ?N^\perp, N$}
\UnaryInfC{$\vdash ?N^\perp, !N$}
\BinaryInfC{$\vdash P \otimes !N, P^\perp, ?N^\perp$}
\UnaryInfC{$\vdash ?(P \otimes !N), P^\perp, ?N^\perp$}
\UnaryInfC{$\vdash ?(P \otimes !N), !P^\perp, ?N^\perp$}
\BinaryInfC{$\vdash ?\Gamma, ?(P \otimes !N), ?N^\perp, \Pi, \overrightarrow{\perp}_1$}
\BinaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes !N), ?\Pi', \Pi, \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\end{prooftree}
$$

By inspection of cases, one can check that there is a unique (semantically speaking) canonical way to translate all the rules introducing $\wedge_m$, except in case both the active formulas are positive and out of the stoup.

• suppose now that in the premisses both the active formulas P and Q are out of the stoup (this is *the* crucial case), then we translate by:

$$
\begin{prooftree}
\AxiomC{$P^a(\delta)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Delta, ?Q, \Pi', \overrightarrow{\perp}_2$}
\AxiomC{$P^a(\gamma)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma, ?P, \Pi, \overrightarrow{\perp}_1$}
\AxiomC{$\vdash P, P^\perp$}
\AxiomC{$\vdash Q, Q^\perp$}
\BinaryInfC{$\vdash P \otimes Q, P^\perp, Q^\perp$}
\UnaryInfC{$\vdash ?(P \otimes Q), P^\perp, Q^\perp$}
\UnaryInfC{$\vdash ?(P \otimes Q), !P^\perp, Q^\perp$}
\BinaryInfC{$\vdash ?\Gamma, ?(P \otimes Q), Q^\perp, \Pi, \overrightarrow{\perp}_1$}
\UnaryInfC{$\vdash ?\Gamma, ?(P \otimes Q), Q^\perp, ?\Pi, \overrightarrow{\perp}_1$}
\UnaryInfC{$\vdash ?\Gamma, ?(P \otimes Q), !Q^\perp, ?\Pi, \overrightarrow{\perp}_1$}
\BinaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, \Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2, \perp$}
\AxiomC{$P^a(\gamma)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma, ?P, \Pi, \overrightarrow{\perp}_1$}
\AxiomC{$P^a(\delta)$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Delta, ?Q, \Pi', \overrightarrow{\perp}_2$}
\AxiomC{$\vdash P, P^\perp$}
\AxiomC{$\vdash Q, Q^\perp$}
\BinaryInfC{$\vdash P \otimes Q, P^\perp, Q^\perp$}
\UnaryInfC{$\vdash ?(P \otimes Q), P^\perp, Q^\perp$}
\UnaryInfC{$\vdash ?(P \otimes Q), P^\perp, !Q^\perp$}
\BinaryInfC{$\vdash ?\Delta, ?(P \otimes Q), P^\perp, \Pi', \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Delta, ?(P \otimes Q), P^\perp, ?\Pi', \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Delta, ?(P \otimes Q), !P^\perp, ?\Pi', \overrightarrow{\perp}_2$}
\BinaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), \Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2$}
\UnaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2, \perp$}
\BinaryInfC{$\vdash ?\Gamma, ?\Delta, ?(P \otimes Q), ?\Pi, ?\Pi', \overrightarrow{\perp}_1, \overrightarrow{\perp}_2, \perp \& \perp$}
\end{prooftree}
$$

Remark: All the (proof-theoretical) solutions that we know to the problem of extracting the computational content from the proofs of classical logic *choose* as the translation of the rule $\wedge_m$ with two positive active formulas that are in the stoup *one* of the two linear subproofs premisses of the last rule introducing $\perp\&\perp$. The upshot of this work is precisely *not to choose* but rather to superimpose. This corresponds to the basic idea that a proof in LK is a superimposition of proofs of linear logic (or of LC).

4 Canonical form

Definition 2 *We say that the rule R of LK_{pol} is difficult iff $R = \wedge_m$ and both the active formulas in the premises of R are positive and are not in the stoup.*

Remark: It is proven in [Girard 91] and in [DJS 95] that each rule R of LK_{pol} can be translated by "some" rules of LC:
(i) if R is already an LC-rule then the translation is straightforward
(ii) if R is not a rule of LC (and if it is not difficult), then one cuts the premise(s) of the rule with a piece of η-proof of the conclusion of R. This procedure (which is also called a way of "constraining" R) might involve a choice in the order of performance of two cut rules; but this choice has no computational content: the two proofs corresponding to the two possible choices are interpreted by the same clique in the P-semantics
(iii) if R is a difficult rule, then one translates it like in case (ii), but now the order of performance of the two cuts *does* matter: not only the two proofs corresponding to the two possible choices are interpreted by different cliques in the P-semantics, but it might very well be the case that they cannot be identified by *any* (non degenerated) denotational semantics.
One has two different ways of constraining each difficult rule, which correspond exactly to the two different ways of translating R in LL through the P-embedding.
This means (as expected) that there exists a classical version of P^a (which will be called C^a) that translates proofs of LK_{pol} into proofs of LC. It is the identity on formulas, and acts on sequents and proofs like P^a: just substitute "$\neg V$" for "$\perp$", "$\wedge_a$" for "$\&$", and erase the exponential rules (through C^a we are constraining the proofs of LK_{pol} in "all the possible different ways"). For any LK_{pol}-proof α, $P^a(\alpha)$ and $P(C^a(\alpha))$ have the same denotational semantics.
We decided to focus on P^a (rather than on C^a) simply to stress the relation with LL.

Remark: Let α be a proof of LK_{pol} with k difficult rules. Let's call α^c the set of LC-proofs obtained from α by constraining all its rules in all the possible *different* ways. From the previous remark we deduce that there are 2^k such LC-proofs (because there are two different ways of constraining each difficult rule of α).
The reader should note that α^c is uniquely determined only from the *semantical* point of view: to each LK_{pol}-proof α is associated a unique set of 2^k cliques of

the coherent space interpreting the conclusion of α. However, as stated in (ii) of the previous remark, there might be different syntactical representations of "the same" proof of α^c. From now on, α^c will denote any such set of LC-proofs.

Definition 3 *Let α be an LK_{pol}-proof with k difficult rules and $\alpha^c = \{\alpha_1, \ldots, \alpha_{2^k}\}$. We'll call slice of α any LC-proof belonging to α^c.*

Definition 4 *Let α be an LK_{pol}-proof with k difficult rules and $\alpha^c = \{\alpha_1, \ldots, \alpha_{2^k}\}$. For all $i \in \{1, \ldots, 2^k\}$, we define $P^+(\alpha_i)$ as the proof $P(\alpha_i)$, to which we added k occurrences of the rule introducing $\bot$.*

Definition 5 *Let α be an LK_{pol}-proof with conclusion $\vdash \Gamma; \Pi$ and k difficult rules $R_1, \ldots, R_k$, and let $\alpha^c = \{\alpha_1, \ldots, \alpha_{2^k}\}$.*
To each $i \in \{1, \ldots, 2^k\}$ we can associate a sequence $\bot^1_{i_1}, \ldots, \bot^k_{i_k}$, where for all $j \in \{1, \ldots, k\}$ $i_j \in \{1, 2\}$, in such a way that R_j is constrained in the same way in α_l and α_m iff $l_j = m_j$.
This gives a bijection between $\{P^+(\alpha_i) : i \in \{1, \ldots, 2^k\}\}$ and $\{\bot^1_{i_1}, \ldots, \bot^k_{i_k} : i_j \in \{1, 2\}\}$.
We define the proof $\&(P^+(\alpha_i))$ of LL by building successively:
- the 2^{k-1} proofs with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_{i_2}, \ldots, \bot^k_{i_k}$, where $1 \leq i_j \leq 2$
- the 2^{k-2} proofs with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_1 \& \bot^2_2, \bot^3_{i_3}, \ldots, \bot^k_{i_k}$, where $1 \leq i_j \leq 2$

$$\vdots$$

- the proof $\&(P^+(\alpha_i))$ with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_1 \& \bot^2_2, \ldots, \bot^{k-1}_1 \& \bot^{k-1}_2, \bot^k_1 \& \bot^k_2$.
(The following example, in case $k = 2$, should clarify the construction).

Example: Suppose that in the previous definition $k = 2$. Then we have $\alpha^c = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, and for $i \in \{1, 2, 3, 4\}$ we have that $P^+(\alpha_i)$ is an LL-proof with conclusion $\vdash ?\Gamma, \Pi, \bot^1_{i_1}, \bot^2_{i_2}$ where $i_1, i_2 \in \{1, 2\}$. Suppose for example that the conclusion of $P^+(\alpha_1)$ is $\vdash ?\Gamma, \Pi, \bot^1_1, \bot^2_1$, the conclusion of $P^+(\alpha_2)$ is $\vdash ?\Gamma, \Pi, \bot^1_1, \bot^2_2$, the conclusion of $P^+(\alpha_3)$ is $\vdash ?\Gamma, \Pi, \bot^1_2, \bot^2_1$, the conclusion of $P^+(\alpha_4)$ is $\vdash ?\Gamma, \Pi, \bot^1_2, \bot^2_2$.
Then, to obtain the proof $\&(P^+(\alpha_i))$, we first build the 2 ($= 2^{2-1}$) proofs $P^+(\alpha_1) \& P^+(\alpha_3)$ with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_1$ and $P^+(\alpha_2) \& P^+(\alpha_4)$ with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_2$, and then the proof $\&(P^+(\alpha_i))$ with conclusion $\vdash ?\Gamma, \Pi, \bot^1_1 \& \bot^1_2, \bot^2_1 \& \bot^2_2$.

Remark: In the previous definition, it is crucial to observe that, in the construction of the proof $\&(P^+(\alpha_i))$, the formulas active in the additive contractions of the rules $\&$ (that is the context formulas) which are different from $?\Gamma, \Pi$ are $\bot^i_1 \& \bot^i_2$, or $\bot^i_1$, or $\bot^i_2$.
The fact that there are no additive contractions between $\bot^i_l$ and $\bot^i_m$ with $l \neq m$ means that there is no "mixing" between two "$\bot$" which stem for two different ways of constraining the rule R_i.
Observe also that all the linear proofs $\&(P^+(\alpha_i))$ associated to the LK_{pol}-proof α are semantically equal.

Proposition 6 *(canonical form)*
Let α be an LK_{pol}-proof, let $R_1 \dots R_k$ be its difficult rules, and let $\alpha^c = \{\alpha_1, \dots, \alpha_{2^k}\}$. Then $[P^a(\alpha)] = [\&(P^+(\alpha_i))]$, where if d is a linear proof then $[d]$ is its denotational semantics.

proof: The idea is that, because the formulas $\bot \& \bot$ introduced by the embedding P^a are all formulas of the sequent conclusion of $P^a(\alpha)$ and because they are reversible and purely negative, it is possible to apply the rules introducing them "at the end" of the proof. The main point is of course the commutation of the rules introducing $\bot \& \bot$ with the promotion rule. But remember that the rule $\bot$ commutes to the promotion, and that $\&$ is reversible in a very strong sense: the denotational semantics of the proof obtained by applying a structural operation to the formula $A \& B$ (where A and B are linear formulas of type "?") *is* the denotational semantics of the proof obtained by applying first the same structural operation to A and to B and then by applying the rule introducing $\&$ (see [Girard 91] and [QT 96a] for more details).
A detailed proof can be given by induction on the proof α, i.e. by considering all the possible cases for the last rule R of α. □

Definition 7 *Let α be a proof of LK_{pol} with k difficult rules and let $\alpha^c = \{\alpha_1, \dots, \alpha_{2^k}\}$.*
We shall call canonical form of α, and we shall denote by $\bigwedge_a(\alpha_1, \dots, \alpha_{2^k})$ the proof of LC obtained by applying to $\{\alpha_1, \dots, \alpha_{2^k}\}$ exactly the same construction applied in definitions 4 and 5 to $\{P(\alpha_1), \dots, P(\alpha_{2^k})\}$: just substitute $\neg V$ for $\bot$ and $\wedge_a$ for $\&$.

Corollary 8 *Let α be a proof of LK_{pol} with k difficult rules and let $\alpha^c = \{\alpha_1, \dots, \alpha_{2^k}\}$. Then $[P^a(\alpha)] = [P(\bigwedge_a(\alpha_1, \dots, \alpha_{2^k}))]$*
(or, otherwise stated, $[P(C^a(\alpha))] = [P(\bigwedge_a(\alpha_1, \dots, \alpha_{2^k}))]$).

proof: It is the LC-counterpart of the canonical form theorem (proposition 6). □

Remark: The corollary says (in particular) that all the canonical forms of an LK_{pol}-proof α have the same denotational semantics.

5 Cut-elimination in LK_{pol}

Definition 9 *Let α and β be two proofs in LK_{pol}. We will say that β is obtained from α by applying one cut-elimination step, and we will write $\alpha \to \beta$, iff there exist a canonical form $\bigwedge_a(\alpha_1, \dots, \alpha_{2^k})$ of α and a canonical form $\bigwedge_a(\beta_1, \dots, \beta_{2^h})$ of β s.t.:*
- $k = h$
- there exists $i \in \{1, \dots, 2^k\}$ such that in LC $\alpha_i \to \beta_i$ (that is β_i is obtained from α_i by applying one cut-elimination step of LC)
- $\forall j \in \{1, \dots, 2^k\}$, $j \neq i$, we have $\alpha_j = \beta_j$.
$\twoheadrightarrow$ will denote, as usual, the reflexive and transitive closure of $\to$.
An LK_{pol}-proof is said to be normal when it is an LC-proof without cuts.

Remark: (i) All the reducts of a proof α of LK_{pol} can be "materialized" only through their canonical forms.
(ii) If α is an LK_{pol}-proof without cuts, then its canonical forms are not necessary cut-free (take any proof without cuts containing a rule which is not an LC-rule). Intuitively, this means that such a proof is not (yet) completely explicit.

Theorem 10 *If in LK_{pol} $\alpha \twoheadrightarrow \beta$, then $[P^a(\alpha)] = [P^a(\beta)]$, i.e. P^a is a denotational semantics for LK_{pol}.*

proof: To prove that P^a induces a denotational semantics for LK_{pol}, it is clearly enough to prove that if $\alpha \rightarrow \beta$, then $[P^a(\alpha)] = [P^a(\beta)]$. But remember that it is proven in [QT 96a] that P is a denotational semantics for LC, so that from corollary 8 we deduce: $[P^a(\alpha)] = [P(\bigwedge_a(\alpha_1, \ldots, \alpha_{2^k}))] = [P(\bigwedge_a(\beta_1, \ldots, \beta_{2^k}))] = [P^a(\beta)]$. □

Remark: Observe that, because to an LK_{pol}-proof α we can associate (syntactically) different sets α^c, the cut-elimination procedure previously defined does not enjoy the Church-Rosser property. For the same reason, although cut-elimination in LK_{pol} seems to be strongly normalizing, this does not follow immediately from the same result for LC proven in [DJS 95] and [JST 96].
This seems to indicate that (following a suggestion of [Girard 91]), one should perhaps try to find a better syntax (which would identify two different ways of representing "the same" proof of α^c) for classical logic.

5.1 Gentzen 's cut-elimination is not explicitation

This is an informal subsection, where we give two examples, the aim of which is simply to suggest a possible retrospective analysis of Gentzen 's procedure of cut-elimination in the framework of LK_{pol}. What turns out is that this procedure is not (in general) compatible with the P^a-denotational semantics. Intuitively, if we assume that $P^a(\alpha)$ is the implicit content of the LK_{pol}-proof α, then Gentzen's cut-elimination does something different from simply making explicit what is already implicit in the proof. Sometimes it interacts with explicitation by making choices (like in example 1: it "does too much"), and sometimes it adds some "new" implicit content (like in example 2: "it doesn't do enough"): in both cases this procedure deeply modify the computational content of the proof.

example 1: The proof α (where R is supposed to be the unique difficult rule)

$$\dfrac{\dfrac{\dfrac{\beta}{\vdots}\atop{\vdash P; \Pi} \qquad \dfrac{\gamma}{\vdots}\atop{\vdash Q; \Pi'}}{\vdash P \wedge_m Q, \Pi, \Pi';}\,R \qquad \dfrac{\dfrac{\delta}{\vdots}\atop{\vdash \neg P, \neg Q; \Pi''}}{\vdash \neg P \vee_m \neg Q; \Pi''}}{\vdash \Pi, \Pi', \Pi'';}$$

is substituted by one of the two following proofs α_1 and α_2:

$$\dfrac{\dfrac{\begin{matrix}\beta\\ \vdots\\ \vdash P;\Pi\end{matrix} \qquad \dfrac{\begin{matrix}\gamma\\ \vdots\\ \vdash Q;\Pi'\end{matrix}\qquad \begin{matrix}\delta\\ \vdots\\ \vdash \neg P,\neg Q;\Pi''\end{matrix}}{\vdash \neg P,\Pi'';\Pi'}}{\vdash \Pi',\Pi'';\Pi}}{\vdash \Pi,\Pi',\Pi'';} \qquad \dfrac{\dfrac{\begin{matrix}\gamma\\ \vdots\\ \vdash Q;\Pi'\end{matrix} \qquad \dfrac{\begin{matrix}\beta\\ \vdots\\ \vdash P;\Pi\end{matrix}\qquad \begin{matrix}\delta\\ \vdots\\ \vdash \neg P,\neg Q;\Pi''\end{matrix}}{\vdash \neg Q,\Pi'';\Pi}}{\vdash \Pi,\Pi'';\Pi'}}{\vdash \Pi,\Pi',\Pi'';}$$

It is possible to check that each of these two proofs is a reduct (w.r.t. the cut-elimination of LC) of one of the two slices of α; choosing one of them means interacting with the process of cut-elimination of LK_{pol}. In LL this interaction is a cut between $\&(P^+(\alpha_1), P^+(\alpha_2))$ and one of the (two) proofs of $1 \oplus 1$ (which "selects" one of the two slices of the reduct of α).

example 2: The proof α

$$\dfrac{\dfrac{\dfrac{\vdash \neg R; R \qquad \vdash \neg R; R}{\vdash R, R; \neg R \wedge_m \neg R}}{\vdash R; \neg R \wedge_m \neg R} \qquad \begin{matrix}\beta\\ \vdots\\ \vdash \neg R, P \wedge_m Q;\end{matrix}}{\vdash P \wedge_m Q; \neg R \wedge_m \neg R}$$

is substituted by the proof α'

$$\dfrac{\dfrac{\begin{matrix}\beta\\ \vdots\\ \vdash \neg R, P \wedge_m Q;\end{matrix} \qquad \begin{matrix}\beta\\ \vdots\\ \vdash \neg R, P \wedge_m Q;\end{matrix}}{\vdash P \wedge_m Q, P \wedge_m Q; \neg R \wedge_m \neg R}}{\vdash P \wedge_m Q; \neg R \wedge_m \neg R}$$

Suppose that $P \wedge_m Q$ in the subproof β is the conclusion of the unique difficult rule of α; then α has two slices while α' has four (because it has *two* difficult rules). Observe also that the two slices of α reduce to two (of the four) slices of α', and that the slices of α' which are not reducts of the slices of α cannot (in general) be identified with the reducts of the two slices of α without producing (algorithmic) inconsistency: to identify them means to identify all the proofs of a given sequent.

The point is that for β we have two possible choices for the translation of the difficult rule (let 's call them $P(\beta_{PQ})$ and $P(\beta_{QP})$) and in $P^a(\alpha)$ when we reduce the cut following the contraction on $?P(R)$, we have to contract all the context formulas ... including $\bot \& \bot$ introduced by the translation!

This means that the reduct of $P^a(\alpha)$ choose either $P(\beta_{PQ})$ or $P(\beta_{QP})$ for *both* the copies of β, while the P-image of one of the slices of α' will contain $P(\beta_{PQ})$ and $P(\beta_{QP})$ as subproofs.

Actually, for α' to have the same implicit content of α, one has to introduce the notion of "synchronization" of two difficult rules, and to synchronize the two difficult rules of α'. Technically speaking, to synchronize two difficult rules (which of course don't need to be equal, i.e. to have the same active formulas and conclusions) means precisely to contract the two "$\bot \& \bot$" introduced by the P^a-translation of the two rules.

We see that in this case Gentzen 's procedure "doesn't do enough" to keep the same implicit content.

6 Computational isomorphisms in LK_{pol}

Definition 11 *We say that R and Q (resp. M and N) are (ϕ,ψ)-isomorphic in LK_{pol}, and we write $R \overset{\phi,\psi}{\simeq} Q$, (resp. $M \overset{\phi,\psi}{\simeq} N$) if there exist a proof ϕ without difficult rules of $\vdash \neg R; Q$ (resp. a proof ϕ without difficult rules of $\vdash M; \neg N$) and a proof ψ without difficult rules of $\vdash \neg Q; R$ (resp. a proof ψ without difficult rules of $\vdash N; \neg M$) s. t.*
- the coherent semantics of the proof $cut(\phi,\psi)$ of $\vdash \neg R; R$ (resp. $\vdash M; \neg M$) is the identity of $P(R)$ (resp. of $P(M)$)
- the coherent semantics of the proof $cut(\psi,\phi)$ of $\vdash \neg Q; Q$ (resp. $\vdash N; \neg N$) is the identity of $P(Q)$ (resp. of $P(N)$).

Remark: We ask positive formulas in the conclusions of ϕ and ψ to be in the stoup, so that the cuts between ϕ and ψ are interpreted as composition of functions in the coherent semantics. This ensures us that when $A \overset{\phi,\psi}{\simeq} B$, then $P(A)$ and $P(B)$ are isomorphic as coherent spaces.

Theorem 12 *In LK_{pol} we have the same computational isomorphisms than in LC.*

proof: Girard proves (in [Girard 91]) that the correlation semantics of LC satisfies a certain number of isomorphisms. In [QT 96b], we prove that these isomorphisms are also satisfied if one takes the (slightly more syntactical) previous definition of isomorphism, and simply by considering the usual coherent semantics induced by the P-embedding (see [QT 96a]). The fact that this still holds for LK_{pol} (which contains LC as a subsystem) is straightforward: the definition of isomorphism is the same! □

Remark: (i) Observe however that, in spite of the previous theorem, the situation in LK_{pol} is different from the one we have in LC. Consider for example the associativity of conjuction.
If we take three proofs α_1 of $\vdash P;$, α_2 of $\vdash Q;$ and α_3 of $\vdash R;$, then the two proofs β of $\vdash (P \wedge_m Q) \wedge_m R;$ and γ of $\vdash P \wedge_m (Q \wedge_m R);$ that one can get simply by performing the two conjuctions in a different order, have both four slices. Let 's distinguish them by the order of application of the cut-rules of LK_{pol} between the pieces of η-proofs and the proofs α_1, α_2 and α_3.
The slice QPR (for example) of β is not a slice of γ. But this is only apparently contradictory with the isomorphism $(P \wedge_m Q) \wedge_m R \overset{\phi,\psi}{\simeq} P \wedge_m (Q \wedge_m R)$. Actually, roughly speaking, there exists a proof of $\vdash P \wedge_m (Q \wedge_m R);$ containing "the same" slices of β. To convince himself, the reader may observe that what one eventually does to get the proof β of $\vdash (P \wedge_m Q) \wedge_m R;$ from α_1, α_2 and α_3 is to cut α_1, α_2 and α_3 with the canonical proof of $\vdash \neg P, \neg Q, \neg R; (P \wedge_m Q) \wedge_m R$ in four of the six possible ways. Then, one obtains a proof β' of $\vdash P \wedge_m (Q \wedge_m R);$ with "the same" slices of β by cuting α_1, α_2 and α_3 with the canonical proof of $\vdash \neg P, \neg Q, \neg R; P \wedge_m (Q \wedge_m R)$ in *the same* four ways (which *are not* the four ways used to obtain the four slices of γ).

This last proof β' is precisely a reduct of the proof obtained by cuting β with the (canonical) proof ϕ of $\vdash \neg(P \wedge_m Q) \vee_m \neg R; P \wedge_m (Q \wedge_m R)$.
(ii) If one really wants to have simoultaneously, for the proofs of $\vdash P \wedge_m (Q \wedge_m R)$; and of $\vdash (P \wedge_m Q) \wedge_m R$; obtained from α_1, α_2 and α_3 of (i), all the six possible slices, then one has to take a version of LK_{pol} with difficult rules of arbitrary arity. But observe that this is not necessary to have a computational isomorphism between $A \wedge_m (B \wedge_m C)$ and $(A \wedge_m B) \wedge_m C$, whatever the polarities of A, B and C are.

7 Further developments

From the technical point of view, the next step (which looks difficult) would be to find a better syntax for LK_{pol}, identifying all the canonical forms of a given proof.
What seems interesting (in the framework of LK_{pol}) is to define some kind of relationship between the slices superimposed by an LK_{pol}-proof.
The notion of synchronization of two (or more) difficult rules might also play an important role. It is a very natural notion, which is technically translated by one (or more) contraction(s) on the $\bot \& \bot$ introduced by P^a (as stated in example 2 of 5.1). To take difficult rules of arbitrary arity might also increase the flexibility of the calculus.
Finally, following an old idea of [Girard 91], one might also try to "understand" Gentzen 's cross-cut procedure in the framework of LK_{pol}.

References

[BarBer 95] Barbanera F., Berardi S., A Symmetric Lambda Calculus for "classical" Program extraction, to appear in $I\&C$, 1995
[DJS 95] Danos V., Joinet J.-B., Schellinx H., A new deconstructive logic: linear logic, to appear in the Journal of Symbolic Logic, 1995
[Girard 87] Girard J.Y., Linear logic, Theoretical Computer Science, 50:1-102, 1987.
[Girard 91] Girard J.Y., A new constructive logic: classical logic. Mathematical Strucures in Computer Science, 1(3):255-296, 1991.
[JST 96] Joinet J.-B., Schellinx H., Tortora de Falco L., Strong Normalization for "all-style" LK^{tq}, extended abstract, Lecture Notes in Artificial Intelligence 1071, p. 226-243, Springer-Verlag 1996
[Parigot 91] Parigot M., Free Deduction: an analysis of computation in classical logic. In: Voronkov, A, editor, Russian Conference on Logic Programming, pp. 361-380. Springer Verlag. LNAI 592. 1991.
[QT 96a] Quatrini M., Tortora de Falco L., Polarisation des preuves classiques et renversement, Compte-Rendu de l'Académie des Sciences Paris, t. 323, Série I, p. 113-116, 1996
[QT 96b] Quatrini M., Tortora de Falco L., Reversion and isomorphisms in classical logic: from LK^{η}_{pol} to LC, manuscript, 1996
[Tortora 96] Tortora de Falco L., Generalized Standardization lemma for the additives, vol. 3 Electronic Notes in Theoretical Computer Science, 1996

This article was processed using the LaTeX macro package with LLNCS style