Polarized and Focalized Linear and Classical Proofs

Olivier Laurent
IML–CNRS Marseille
Olivier.Laurent@pps.jussieu.fr

Myriam Quatrini
IML–CNRS Marseille
quatrini@iml.univ-mrs.fr

Lorenzo Tortora de Falco
Roma III
tortora@uniroma3.it

Abstract

We give the precise correspondence between polarized linear logic and polarized classical logic. The properties of focalization and reversion of linear proofs are at the heart of our analysis: we show that the tq-protocol of normalization for the classical systems LK^{η}_{pol} and $\mathrm{LK}^{\eta,\rho}_{pol}$ perfectly fits normalization of polarized proof-nets. Some more semantical considerations allow to recover LC as a refinement of multiplicative LK^{η}_{pol}.

MSC: 03F05; 03F07; 03F52

Keywords: Classical Logic, Linear Logic, Cut-Elimination, Proof-Nets, Denotational Semantics, Polarization, Focalization, Reversion.

Introduction

The extension of the Curry-Howard paradigm to classical logic turned out to be possible and interesting. The last years have seen lots of attempts to extract such a computational content from classical proofs. A possible proof-theoretical approach is to start by defining classical systems (*i.e.* complete w.r.t. classical provability) with a deterministic normalization procedure. This means that cut-elimination enjoys some particular properties: it has a denotational semantics, and it enjoys strong normalization and (usually) confluence. Let us quote, among the works following this approach, Parigot's FD and $\lambda\mu$-calculus [Par91, Par92], Girard's LC [Gir91a], LK^{tq} and its subsystems [DJS97].

Although all this work is not entirely based on it, linear logic (LL), introduced in [Gir87], seems to play a pre-eminent role, because it can be used as a looking-glass. On the one hand LL suggests a reasonable way to make choices in the cut-elimination procedure of classical logic, and on the other hand it organizes known solutions, thus acting as a unifying and clarifying tool (see [DJS97]).

Our approach follows [Gir91a] and [DJS97]: we go one step further in the *linear analysis* of *classical computations*.

Right from the start LL was seen as a way to analyze classical logic, but it took some years to find the "good" translations from classical into linear logic, and the fact that they were eventually found (see [Gir91a] and [DJS97]) was due to a better understanding of LL itself. Most notably, the works [And90, AP91, Gir91a, DJS97] pointed out two crucial properties of LL: focalization and reversion.

In [QTdF96], these two properties gave birth to the classical system $LK^{\eta,\rho}_{pol}$, for which some results were stated (but not proven). In [Lau99], the same two properties were studied in the framework of linear logic proof-nets. The starting point of the present work was to realize that these two papers are in fact tightly linked. We put together our knowledge on proof-nets [Lau99, Lau03, TdF03b, TdF00], on denotational semantics [Qua96], on the linear logic approach to classical logic [JSTdF02, TdF97, QTdF96], and focused on reversion and focalization. In this paper, we study these two properties both in a classical and in a linear framework. This leads us to consider two couples of classical/linear systems (LK^{η}_{pol}/LLP and $LK^{\eta,\rho}_{pol}/LL_{pol}$) which can be considered as two "twins", at least if we restrict to first order quantifiers. Reversion, which is a *semantically invisible* operation, is shown to be the *bridge* from LK^{η}_{pol}/LLP to $LK^{\eta,\rho}_{pol}/LL_{pol}$. The two twins actually correspond to two possible choices: either reduce the classical derivation's space by means of reversion ($LK^{\eta,\rho}_{pol}/LL_{pol}$) or extend the linear derivation's space by means of *admissible* rules (LK^{η}_{pol}/LLP).

Let us now be more precise on the contents of this paper. In section 1, we recall the main definitions and results of [Gir91a] and [DJS97], among which the notion of decoration (a translation from classical to linear logic preserving the *skeleton* of the proof). The *Polaro-embedding* (denoted by P) defined in [DJS97] (already implicit in [Gir91a]) is a translation of the classical system LK^{η}_{pol} into LL, which makes use of a very few amount of exponentials. This is rather a quality of the translation, because it allows to represent classical proofs by proof-nets with very few boxes, thus in a more geometrical way. However, the P-embedding is not a decoration (it introduces cuts). It is then quite natural to wonder whether there exists a restriction on classical derivations allowing to turn this embedding into a decoration. In section 2, we define the ρ-constraint on LK^{η}_{pol}-proofs, and we show that it is complete w.r.t. classical provability and stable w.r.t. the tq-protocol (the normalization procedure for classical logic defined in [DJS97]). This yields the system $LK^{\eta,\rho}_{pol}$, for which the P-embedding of section 1 is shown to be a decoration. We then define (section 3) the polarized fragment LL_{pol} of LL which contains the P-image of $LK^{\eta,\rho}_{pol}$, and its natural extension LLP. In section 4, we show that polarized proof-nets have good properties w.r.t. normalization. The main result of section 5 is theorem 8: for the two twins, the classical tq-normalization perfectly fits the normalization of the corresponding polarized proof-nets. This endows classical logic with all the denotational semantics of LL. Section 6 is devoted to reformulate Girard's request for denotational isomorphisms [Gir91a] with a more syntactical flavour: we give a definition of *syntactical isomorphism* in classical logic, which allows to recover LC as a refinement of multiplicative LK^{η}_{pol} (proposition 10). Finally, section 7 shows that second order quantification drastically separates the two twins: it only makes sense for LK^{η}_{pol}/LLP.

1 Preliminaries: linear logic and classical logic

This section is devoted to briefly recall some of the main notions introduced in [Gir91a] and in [DJS97]. We also introduce some notations and conventions: we essentially follow [DJS97], except for the definition of the P-embedding (see section 1.7) which is slightly different.

1.1 Conventions

We use the (more or less) standard terminology for the sequent calculus rules, assuming that the reader has some acquaintance with the notions of sequent, structural and logical rules, premise and conclusion of a rule, active and context formula in a rule, cut-formula, ...

A(n occurrence of) formula is said to be *main* in a logical rule (resp. in an axiom) when it is the conclusion of the rule containing the logical connective introduced by the rule (resp. it is one of the two conclusions of the axiom).

In case of structural rules, if necessary, we speak of the *weakened*, respectively *contracted* formula.

The reader can find, in appendix A, the rules of second order classical (Gentzen) sequent calculus, to which we will refer as LK^2, or simply LK. Notice that our presentation of LK is not standard, and is very much inspired from the works on linear logic: we stressed in appendix A the difference between the *styles* of the rules. This is of course a notion coming from LL, which is meaningless if one sticks to mere provability, but becomes crucial if one wishes to analyze proofs and their normalization (see [Gir91a, DJS97]). We denote by $\vee_m$ (resp. $\wedge_m$) the multiplicative disjunction (resp. conjunction), and $\vee_a$ (resp. $\wedge_a$) the additive disjunction (resp. conjunction).

All the sequent calculi for classical logic we deal with in the paper are one-sided. This is not a simple notational convention, but rather a consequence of our approach: we decided to bet on LL to analyze classical logic, and since linear negation is indeed involutive (one has $A = A^{\perp\perp}$), classical negation is also involutive. It is not completely clear whether or not classical negation should be involutive in such a strong sense, but this discussion is out of the scope of the present work.

A notable consequence of the one-sided choice, is that negation has to be defined by De Morgan laws (respecting the style of the connectives): the dual of $\vee_m$ (resp. $\vee_a$) is $\wedge_m$ (resp. $\wedge_a$). There is nothing to say concerning the quantifiers, but...beware the units! One has in fact four (different) units in

LK: V, $\neg V$, F, $\neg F$ and two kinds of atoms X and $\neg X$. Every unit A satisfies $\neg\neg A = A$, but one *does not have* $\neg V = F$ (and then one does not have $\neg F = V$). This is again a distinction coming from LL, where one has four units.

Except for section 7, we will only consider the propositional fragment of LK and of LL. *Not* all the results we will prove can be extended to second order quantifiers (no problem for first order ones), and this is precisely what motivates section 7.

A *fragment* of a logical system is a subsystem obtained by restricting the language (*i.e.* by forbidding the use of some formulae). We will consider here various subsystems of LK obtained by restricting the use of some rules which does not imply a restriction of the language. Suppose that LK is endowed with a normalization procedure (see section 1.3), we will say that a subsystem S of LK is a *computational subsystem* when it is obtained by some restriction on the rules, in such a way that S is complete w.r.t. provability in LK (*i.e.* A is provable in LK iff it is provable in S) and stable w.r.t. the given normalization procedure (*i.e.* if the proof π' is obtained by a normalization step from the proof π of S, then π' is still a proof of S).

In this paper, we will only consider computational subsystems, because they allow to study dynamical properties.

Definition 1 ((Ir)reversible rules, formulae)
The reversible (resp. irreversible) rules of LK are the ones introducing the connectives $\vee_m$, $\wedge_a$ (resp. $\wedge_m$, $\vee_a$), the units $\neg V$ and $\neg F$ (resp. V) and the quantifier $\forall$ (resp. $\exists$).

The main formula of a reversible (resp. an irreversible) rule is called reversible (resp. irreversible). The formula F is irreversible.

Remark: The notion of reversible/irreversible formula is well-known in logic. A possible definition is "a formula A is reversible iff for every proof π of the sequent $\vdash \Gamma, A$, there exists a proof π^r of $\vdash \Gamma, A$ whose last rule is a rule introducing A". We shall see in the sequel that to the notion of reversible formula a much stronger meaning can be given: not only π^r does exist, but it also has *the same denotational semantics as* π (see proposition 6).

1.2 Decoration

Let us now be more precise on the "bet on LL to analyze classical logic" we previously mentioned. Our approach is based on the notion of *decoration*, introduced by V. Danos, J.-B. Joinet and H. Schellinx.

Let us start with the following (very) simple remark: if π is any LL-proof, we can easily get a proof of LK, by simply erasing all the exponential rules and substituting the linear connectives by the corresponding classical ones (one might also have to erase some redundant sequents). We obtain in such a way a proof $sk(\pi)$ of LK (the classical *skeleton* of π). We can also proceed the other way round: a decoration of the LK-proof π' is a proof $D(\pi')$ of LL preserving the classical skeleton of π', *i.e.* such that $sk(D(\pi')) = \pi'$. One can then try to define a normalization procedure for LK-proofs suggested by the embeddings in LL *which are decorations*: the skeleton preservation is supposed to guarantee that we *did not move away from the starting proof.*

This idea led to the system LK^{tq} defined in [DJS97]: each formula comes equipped with a mapping of the set of its subformulae into a *colour space* $\{t, q\}$. (When necessary, we will make explicit the colour ε of the formula itself by means of a superscript: A^ε, with the convention $\neg A^\varepsilon = \neg(A^\varepsilon)$.) The rules are now supposed to preserve colours, *i.e.* colours should respect identity classes of formulae in a proof (cf. [DJS93]).
We shall not define formally what the identity class of a(n occurrence of) formula in an LK-proof is (for which the reader can refer to the just mentioned paper), but we now give an explanation. The two (occurrences of) formulae conclusions of an axiom rule belong to the same identity class, so as the two active formulae which are premises of a cut rule: in both cases if A^ε and $(\neg A^\varepsilon)^{\varepsilon'}$ are these two (occurrences of) formulae, one has $\varepsilon \neq \varepsilon'$. There is no constraint on the colour of the main formula of a logical rule, and in the sequent conclusion of the rule, the colours of the subformulae of the main formula are those of the premises: one has for example

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \vdash A^\varepsilon, \Gamma\end{array} \qquad \begin{array}{c}\pi_2\\ \vdots\\ \vdash B^{\varepsilon'}, \Delta\end{array}}{\vdash (A^\varepsilon \wedge_m B^{\varepsilon'})^{\varepsilon''}, \Gamma, \Delta}$$

with an arbitrary ε''. Finally, the two active (occurrences of) formulae in a contraction rule belong to the same identity class and must then have the same colour.

The reader should notice that (at least for the moment) there is no connection between the main connective of a formula and its colour. In particular, a formula and its negation can occur in the same proof with the same colour (this will no more be possible in the system LK^{η}_{pol} of section 1.5).

Colours do not interfere with the notion of provability of a formula, and are relevant only from a *dynamical* point of view: they allow to define the *tq*-protocol.

1.3 Classical normalization: the *tq*-protocol

We now recall the definition of *tq*-normalization (see [DJS97]).

An occurrence of coloured formula in a sequent is said to be *attractive* (resp. *non-attractive*) if its colour is q (resp. t): the terminology is introduced to remind us that the subproof of the sequent containing the non-attractive active cut-formula *has to move first.*

A *logical cut* is an occurrence of the cut-rule where both active formulae are main in logical rules. A *structural cut* (*i.e.* not logical) is either of *kind* S_1 or of *kind* S_2, depending on whether the attractive occurrence of the cut-formula is main in a logical rule. If it is, then the cut (and the associated reduction step) is of kind S_2. If it is not, then the cut (and the associated reduction step) is of kind S_1. Intuitively, the number (1 or 2), after the letter "S", indicates the "status" of a cut or the "stage" of the reduction of a cut: at stage 1 nothing has been done to reduce the cut, while at stage 2 (the intuition is that) a first structural step has already been performed, and we are closer to the "logical" step defined below (Gentzen's key-case): $S_1 > S_2 > L$.

Let us call the subderivation containing the *attractive* occurrence of the cut-formula the *attracting* subderivation.

Definition 2 (*tq*-protocol)
Reduction according to *tq*-protocol proceeds via two possible kinds of steps, *structural* ones, S_1 and S_2, and *logical* ones, L (*key-steps*):

- An L-step applies when both cut-formulae are main in a logical rule. L-steps have to be specified for each connective. We obtain as descendants one or two cuts on the immediate subformula(e) of the cut-formula. In case of two descendants, the order in which these cuts are applied is irrelevant, thanks to the η-constraint, see [DJS97] and section 1.4.

- In case no L-step is applicable, necessarily an S-step applies, which consists in *transporting* one of the cut's subderivations up the tree of the cut-formula's ancestors[1] in the other one, duplicating it and contracting the context whenever passing an instance of contraction (or via the context of a $\wedge_a$-rule); this process ends when reaching instances of introduction in an axiom, in which case the resulting *axiom-cuts* are reduced *immediately*, when reaching instances of introduction by weakening, which are replaced by weakenings on the context formulae, or when reaching instances of introduction of the main connective of the cut-formula (see figure 1).

Of course, now one needs to know *which* of the two subderivations has to move. This is decided by asking whether or not the attractive cut-formula is main *in a logical rule*. If the answer is *"yes!"*, we transport the attracting subderivation (S_2); if it is *"no!"*, we transport the other one (S_1).

And that is it.

The main difference between *tq*-reduction and the standard definitions of cut-elimination steps in sequent calculus is in the definition of the *tq-structural steps*, where the *complete* tree of ancestors of one of the occurrences of the cut-formula A is involved: we raise (a copy of) the transported subderivation right up to the leaves, where an ancestor of the cut-formula was introduced. Then we push the context-formulae of the final sequent of the transported subderivation

[1] The tree of ancestors of the occurrence A_1 of the formula A in the sequent calculus proof π is the tree of the occurrences of A *above* A_1 in π. See [DJS97] for a more precise definition.

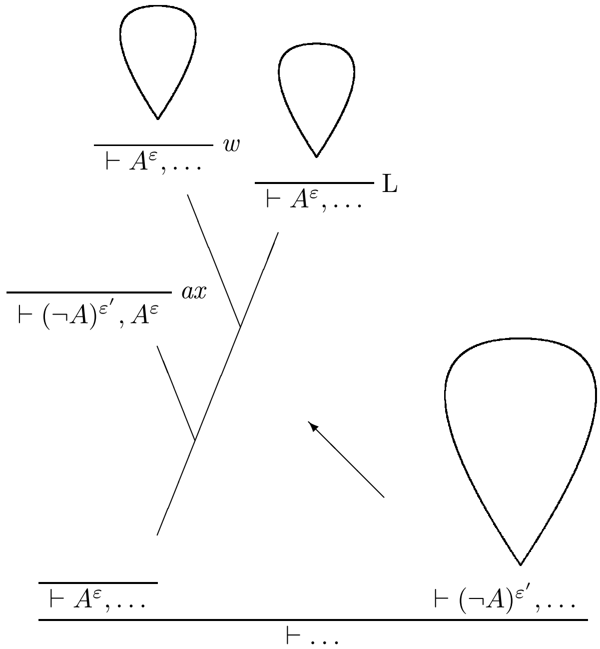


Figure 1: A structural cut, either of kind S_1 or of kind S_2, depending on whether the attractive occurrence of the cut-formula is main in a logical rule. If it is, then the cut (and the associated reduction step) is of kind S_2. If it is not, then the cut (and the associated reduction step) is of kind S_1.

back down along the tree. At branchings originally due to explicit contractions on A, the contractions now are inherited by these context-formulae. (Cf. the *complete exponential reduction* of proof nets, as described in [Reg92].)

1.4 The η-constraint

In [And90, AP91], a crucial property of proofs was pointed out by the authors: focalization. In [Gir91a], this property (the *stoup*) found a semantical counterpart: central cliques of Girard's correlation semantics. In [DJS97], the authors show that focalization (the *η-constraint*) is preserved by cut-elimination.

Definition 3 (The η-constraint)
A proof of LK^{tq} is η-constrained when every attractive (occurrence of) formula active in an irreversible rule is main.
LK^{η} is the subsystem of LK^{tq} s.t. all the proofs are η-constrained.

The two following theorems are among the main results of [DJS97]: they state that the η-constraint is both complete and stable w.r.t. the tq-procedure, that is LK^{η} is a computational subsystem of LK^{tq}.

Theorem 1 (Stability)
If the proof π' is obtained by performing some steps of tq-reduction from the proof π of LK^{η}, then π' is a proof of LK^{η}.

Theorem 2 (Completeness)
Let π be an LK^{tq} derivation. There exists in LK^{tq} an η-constrained derivation π^{η} with the same sequent conclusion as π.

We now briefly sketch the proof of completeness of LK^{η}, introducing the notion of *η-proof* (this will also explain the terminology).

An η-proof is just the obvious expansion of an axiom (this reminds of course η-expansion of typed λ-calculus: $\lambda y^A x^{A\to B} y$ is the η-expansion of $x^{A\to B}$): if A_i are the main subformulae of A, an η-proof of A (notation η_A) consists in the axioms $\vdash A_i, \neg A_i$ and precisely one instance of each logical rule introducing A's main connective and one instance of each logical rule introducing $\neg A$'s main connective (see appendix C for more details). Notice that, for every formula A, $\eta_A = \eta_{\neg A}$.

To prove completeness of LK^{η}, one just has to *plug* pieces of η-proofs to unconstrained LK^{tq}-rules: every unconstrained LK^{tq}-rule is derivable in LK^{η} (see [DJS97]).

Remark: To be precise, let us mention that the η-constraint is a bit stronger than LC's stoup, defined in [Gir91a]: in LC, attractive formulae active in irreversible rules are main "up to some particular rules" (mainly reversible and structural rules on the context). This is simply due to the fact that these rules are "invisible" in correlation semantics. The stoup is a way to express a "linearity" condition on formulae (in particular a formula in the stoup has not been active in any structural rule). More precisely, we will say that a classical formula A is linear in a proof π when its P-translation $P(A)$ in the linear proof $P(\pi)$ (see paragraph 1.7) has not been active in any structural rule nor context in any promotion rule.

1.5 Reversion

We now introduce the computational subsystem LK^{η}_{pol} of LK^{η}. In LK^{η}_{pol}-proofs the reversible/irreversible nature of a formula is linked to its tq-colour:

Definition 4 ($\mathbf{LK}^{\eta}_{pol}$)
An LK^{η}-proof is an LK^{η}_{pol}-proof when every reversible (resp. irreversible) formula is coloured t (resp. q).

The reader should notice that in LK^{η}_{pol} negation flips colours (which was not the case up to now): if ε (resp. ε') is the colour of the non-atomic formula A (resp. $\neg A$), then $\varepsilon \neq \varepsilon'$.

We will often use in the sequel the following operation of "reverting a (reversible) formula in a proof".

Definition 5 (Reversion)
Let $A = B \vee_m C$ (resp. $A = B \wedge_a C$) be an LK^{η}_{pol}-formula. Let π be an LK^{η}_{pol}-proof of $\vdash \Gamma, A$. The *reversion of A in π* yields a proof π^r, which is obtained by:

1. substituting, in π, the axioms $\vdash A, \neg A$ by η_A;
2. erasing the (logical) rules with main conclusion A;
3. transferring structural rules on A to the formulae B and C;
4. adding as a last rule the rule $\vee_m$ (resp. $\wedge_a$) with main conclusion A.

If $A = \neg V$ and π is an LK^{η}_{pol}-proof of $\vdash \Gamma, A$, then the reversion of A in π yields the proof π^r, which is obtained by erasing from π the whole tree of A's ancestors and by adding to the thus obtained proof a rule with conclusion $A = \neg V$.
If $A = \neg F$ and π is an LK^{η}_{pol}-proof of $\vdash \Gamma, A$, then the reversion of A in π yields the proof π^r, which consists in the unique rule $(\neg F)$ with conclusion $\vdash \Gamma, \neg F$.

Remark: One can also obtain the proof π^r previously defined by eliminating the cut between π and η_A (following the tq-protocol) (see [DJS97]). This shows, by the way, that π^r is indeed an LK^{η}_{pol}-proof.

1.6 Semantics

In [Gir91a], Girard sums up the main features of what he considers as a "good" denotational semantics:

"... The kind of semantics we are interested in is *concrete*, *i.e.* to each proof π we associate a set π^*. This map can be seen as a way to define an equivalence $\approx$ between proofs ($\pi \approx \pi'$ iff $\pi^* = \pi'^*$) of the same formulae (or sequents), which should enjoy the following:

(i) if π normalizes to π', then $\pi \approx \pi'$

(ii) $\approx$ is non-degenerated, *i.e.* one can find a formula with at least two non-equivalent proofs

(iii) $\approx$ is a congruence: this means that if π and π' have been obtained from π_0 and π'_0 by applying the same logical rule, and if $\pi_0 \approx \pi'_0$, then $\pi \approx \pi'$

(iv) certain canonical *isomorphisms* are satisfied ..."

The decoration method will allow to endow classical logic with some denotational semantics: thanks to the simulation theorem (theorem 8) of section 5, every "good" denotational semantics of linear logic proof-nets yields a "good" denotational semantics for classical logic. In the sequel, we will often consider the "denotational semantics of a linear proof π". Depending on the context, we will refer to coherent set-based or multiset-based semantics, relational semantics, correlation semantics, ... If we do not specify the kind of semantics we are considering, this means that we are speaking of any denotational semantics of the proof-net associated with π. We denote by $[\pi]$ *the interpretation* or *the semantics* of the LL proof(-net) π.

Notice that we did not discuss point (iv), which is indeed one of the main ingredients of [Gir91a]: this is because it motivates our section 6.

1.7 The P-embedding

We finally come to an (explicit) relation with linear logic, by means of the so-called *Polaro-embedding* (P-embedding for short). The definition of the linear translation $P(\pi)$ of the LK^{η}_{pol}-proof π is delicate: on the one hand we want to give an inductive definition of $P(\pi)$, and on the other hand we have to make sure that each time we perform a &-rule or a $\top$-rule in $P(\pi)$ the context formulae are all negative in the sense of definition 11 (this condition is essential to prove theorem 8).

We are first going to split the set of S_1-cuts in two disjoint subsets: S_1^{ax} and $S_1^{\neg ax}$. This is because we are going to translate them differently (see definition 8). The true motivation for this splitting will be clear in section 2.

Definition 6 (Kinds of cuts)
We have already seen the notion of cut of *kind* S_1, S_2 and L in section 1.3. We refine it in the following way: an S_1-cut is of *kind* S_1^{ax} when the attractive

occurrence of the cut formula is main in an axiom, and it is of kind $S_1^{\neg ax}$ otherwise.

We will consider the following order on kinds corresponding to their possible evolution during reduction: $S_1^{\neg ax} > S_1^{ax} > S_2 > L$.

Say that a linear formula A is !-fix (resp. ?-fix) when the sequent $\vdash A^\perp, !A$ (resp. $\vdash A, !A^\perp$) is provable in LL. The following proposition (which is lemma 46 of [DJS97]) is a way to express the splitting of linear connectives into *positive connectives* and *negative connectives*.

Proposition 1
The units $1, 0$ *(resp.* $\perp, \top$*) are !-fix (resp. ?-fix). The formula* $!A$ *(resp.* $?A$*) is !-fix (resp. ?-fix), where* A *is any formula.*
If A *and* B *are !-fix (resp. ?-fix), then* $A \otimes B$*,* $A \oplus B$*,* $\exists X A$ *(resp.* $A \parr B$*,* $A \& B$*,* $\forall X A$*) are !-fix (resp. ?-fix).*

Definition 7 (P-translation)
We define $P(V) = 1$, $P(F) = 0$, and if X is a classical atomic formula, then $P(X^t) = ?X$, $P(X^q) = !X^\perp$, $P((\neg X)^t) = ?X$ and $P((\neg X)^q) = !X^\perp$. For compound formulae, we have:

A	B	$A \wedge_m B$
q	q	$P(A) \otimes P(B)$
q	t	$P(A) \otimes !P(B)$
t	q	$!P(A) \otimes P(B)$
t	t	$!P(A) \otimes !P(B)$

A	B	$A \vee_a B$
q	q	$P(A) \oplus P(B)$
q	t	$P(A) \oplus !P(B)$
t	q	$!P(A) \oplus P(B)$
t	t	$!P(A) \oplus !P(B)$

The translations of $\neg V$, $\neg F$, $A \wedge_a B$ and $A \vee_m B$ are obtained by duality, in such a way that $P(\neg A) = P(A)^\perp$.

Notice that every q-formula (resp. t-formula) of LK^η_{pol} is translated by an !-fix (resp. ?-fix) linear formula. The definition of the P-embedding is based on this remark.

We can summarize the different characteristics of LK^η_{pol}-formulae with the following table:

A			$P(A)$
t	non-attractive	reversible	?-fix
q	attractive	irreversible	!-fix

Definition 8 (P-embedding)
Let π be an LK^η_{pol}-proof with conclusion $\vdash \Gamma^t, \Delta^q, \Pi^q$ where Π^q contains the main formula of π's last rule, provided this formula exists and is q-coloured (otherwise Π^q is empty).

The P-embedding of LK^η_{pol}-proofs into linear proofs associates with π a linear proof $P(\pi)$, and translates the sequent $\vdash \Gamma^t, \Delta^q, \Pi^q$ by the sequent $\vdash P(\Gamma^t), ?P(\Delta^q), ?P(\Pi^q)$.

We give a mutually recursive definition of the P-embedding and of the P_0-embedding which associates with π a linear proof $P_0(\pi)$ of the sequent $\vdash P(\Gamma^t), ?P(\Delta^q), P(\Pi^q)$. With the property that either Π^q is empty and $P(\pi) = P_0(\pi)$ or Π^q is not empty and $P(\pi)$ is obtained by adding a $?d$-rule on $P(\Pi^q)$ to $P_0(\pi)$. We proceed by induction on π:

- If π is an axiom $\vdash A^q, (\neg A)^t$, then $\Pi^q = A^q$ and $P_0(\pi)$ is the LL axiom $\vdash P(A), P(A)^\perp$.
- If π is the logical rule with conclusion $\vdash V$ (in this case Π^q is V), then $P_0(\pi)$ is the logical rule with conclusion $\vdash 1$. If π is the logical rule with conclusion $\vdash \neg F, \Gamma^t, \Delta^q$ (in this case Π^q is empty), then $P_0(\pi)$ is the logical rule with conclusion $\vdash \top, P(\Gamma^t), ?P(\Delta^q)$.
- If π's last rule is a cut rule c between π_1 with sequent conclusion $\vdash \Gamma_1^t, \Delta_1^q, \Pi_1^q, A^t$ and π_2 with sequent conclusion $\vdash \Gamma_2^t, \Delta_2^q, \Pi_2^q, (\neg A)^q$, then we have four possibilities (in fact, one should say three plus one possibilities):
  - if c is logical, then $P_0(\pi) = P(\pi)$ is obtained by performing a cut rule of LL between $P_0(\pi_1)$ and $P_0(\pi_2)$: one thus gets a proof of $\vdash P(\Gamma_1^t), ?P(\Delta_1^q), P(\Gamma_2^t), ?P(\Delta_2^q)$ (in this case Π_1^q is empty and Π_2^q is $(\neg A)^q$).
  - if c is structural of kind S_2, π is translated by a cut rule between $P(\pi_1)$ and $P_0(\pi_2)$.
  - if c is structural of kind S_1^{ax}, then we proceed exactly like in the case of an S_2-cut: π is translated by a cut rule between $P(\pi_1)$ and $P_0(\pi_2)$.
  - if c is structural of kind $S_1^{\neg ax}$, then one applies to $P(\pi_1)$ a promotion rule on $P(A)$, and finally a cut rule between the thus obtained proof and $P(\pi_2)$. Of course, the promotion rule can be applied only if every conclusion of $P(\pi_1)$ is ?-prefixed, and this *is not* the case (in general). But we can use the ?-fix property of the P-translation of the t-coloured formulae, and by adding some cuts we can obtain a linear proof of $\vdash P(\Gamma_1^t), ?P(\Delta_1^q), P(\Gamma_2^t), ?P(\Delta_2^q), ?P(\Pi_1^q)$. See also the following remark for a more detailed description.
- If π's last rule is a reversible rule (including the rule $(\neg V)$), then one applies, to the P-translation (not the P_0-translation) of the subproof(s) premise of the rule, the corresponding LL-rule.
- If π's last rule is an irreversible rule, then one applies to the P_0-translation (not the P-translation) of the subproof(s) premise of the rule, the corresponding LL-rule: a promotion might be necessary (it is not necessary when the rule is η-constrained that is Π_i^q is not empty), and if the context is not ?-prefixed we have to apply some derelictions and some cuts (see the following remark).
- If π's last rule is a structural rule, we just apply the corresponding structural linear rule to the P-translation of the premise of the rule, *except*

when the active formula is reversible: in this case we have (again) to use the ?-fix property of the P-translation of the t-coloured formulae (see the following remark).

Remarks:

(i) In the language of [Gir91a], the formula contained in Π^q (if any) is the formula contained in the stoup.

(ii) Another peculiarity of our definition of the P-embedding is the splitting of S_1-cuts. The necessity of such a splitting will appear clearly in the next section 2, see remark 2 after definition 9.

(iii) The P-embedding is not a decoration. It is easy to be convinced by the two following examples:

1. Let us consider the following derivation in LK^η_{pol} (with $\Gamma = \Gamma_1, \Gamma_2$ and $\Delta = \Delta_1, \Delta_2$)

$$\dfrac{\begin{matrix}\pi_1 \\ \vdots \\ \vdash A^t, \Gamma_1^t, \Delta_1^q, \Pi_1^q\end{matrix} \qquad \begin{matrix}\pi_2 \\ \vdots \\ \vdash B^q, \Gamma_2^t, \Delta_2^q\end{matrix}}{\vdash (A \wedge_m B)^q, \Gamma^t, \Delta^q, \Pi_1^q}$$

We have to deduce from the two linear proofs obtained by decorating π_1 and π_2 (which are, respectively, $P(\pi_1)$ with conclusion $\vdash P(A), P(\Gamma_1), ?P(\Delta_1), ?P(\Pi_1)$ and $P_0(\pi_2)$ with conclusion $\vdash P(B), P(\Gamma_2), ?P(\Delta_2)$) a proof of conclusion $\vdash\, !P(A) \otimes P(B), P(\Gamma), ?P(\Delta), ?P(\Pi_1)$. To be able to perform a promotion on $P(A)$ as the P-translation of formulae requires, we have to make some derelictions on the context $P(\Gamma_1)$ (as soon as Γ_1 contains reversible and non-atomic formulae).

Then we have to perform cuts with the proofs of $\vdash\, !P(G)^\perp, P(G)$ (for every formula G of Γ_1 that we have derelicted) using the ?-fix property of the translation of t-formulae.

2. Let us look at the following derivation in LK^η_{pol}, in which the formula A^t is non-atomic:

$$\dfrac{\begin{matrix}\pi \\ \vdots \\ \vdash A^t, A^t, \Gamma^t, \Delta^q, \Pi^q\end{matrix}}{\vdash A^t, \Gamma^t, \Delta^q, \Pi^q}$$

We can obtain a proof of $\vdash ?P(A), P(\Gamma), ?P(\Delta), ?P(\Pi)$ from the sequent $\vdash P(A), P(A), P(\Gamma), ?P(\Delta), ?P(\Pi)$ conclusion of the LL derivation $P(\pi)$ by means of two derelictions and one contraction. But we want a proof of $\vdash P(A), P(\Gamma), ?P(\Delta), ?P(\Pi)$. We then have to perform a cut with the proof of $\vdash P(A), !P(A)^\perp$.

2 The ρ-constraint

The P-image of LK^{η}_{pol} is a subsystem of LL. We just saw that the P-embedding introduces some cuts. These cuts arise between the P-translation of the initial derivations (inductively built) and some very peculiar derivations: the derivations proving the preservation of the type ! or ?. Also the following questions arose: what happens when we eliminate exactly the cuts introduced by P? Is it possible to characterize the set of proofs thus obtained as some subsystem LL_S of LL? Is it possible to isolate as some LK^{η}_{pol}-computational subsystem (tq-stable and complete) the part of LK^{η}_{pol} which contains exactly the skeletons of LL_S derivations? (this last question is very natural according to the decoration method). Answering to these questions lead us to the present study.

In the previous section we recalled the definition of reversion of a reversible formula A in an LK^{η}_{pol} derivation π. This operation enables us to put off some reversible rules until the end of the derivation and can be useful to avoid the problematic cases in the P-embedding. Moreover, we will see in section 5 (proposition 6) that reversion does not adulterate the semantics of the derivation.

In this section, we introduce a supplementary requirement on the LK^{η}_{pol} derivations. This new constraint (which will be denoted by ρ) yields a new polarized system $\mathrm{LK}^{\eta,\rho}_{pol}$ for which the P-embedding in LL is a decoration. We then show that $\mathrm{LK}^{\eta,\rho}_{pol}$ is a computational subsystem of LK^{η}_{pol}, *i.e.* it is closed by tq-normalization and complete with respect to classical provability.

We will see in the next section that the P-image of this system in LL is included in LL_{pol}.

We introduce, for LK^{η}_{pol}-proofs, a new constraint ρ which requires that the derivations are reversed each time it is necessary.

Definition 9 (The ρ-constraint)

- *Reversible context:* Let $\vdash \Gamma$ be an LK^{η}_{pol}-sequent and let $A \in \Gamma$. Suppose that the sequent $\vdash \Gamma$ is a premise of a rule R in which A is active. We will say that A is in a reversible context (resp. non-reversible context) if there is at least one (resp. no) reversible formula in $\Gamma \backslash A$.

- *The ρ-forbidden rules:* The following rules are ρ-forbidden: the structural rules on reversible formulae; the irreversible rules or the $S_1^{\neg ax}$-cuts in which the context of the active and non-attractive occurrence is reversible.

- *A ρ-constrained proof:* A proof π of LK^{η}_{pol} is called ρ-constrained when it contains no ρ-forbidden rule.

Remarks:

1. The subsystem $\mathrm{LK}^{\eta,\rho}_{pol}$ of LK^{η}_{pol} contains exactly the classical skeletons of the LL derivations obtained by eliminating the cuts introduced by the P-embedding.

2. Notice that S_1^{ax}-cuts (so as S_2-cuts) are not ρ-constrained. In the general case, the P-embedding of S_1-cuts requires to add a dereliction on the formula Π^q (if any) and then a promotion on the t-occurrence of the cut formula. Two essential properties of the ρ-constraint will be stability w.r.t. cut-elimination (theorem 4) and the fact that it allows the P-embedding to be a decoration of $\mathrm{LK}_{pol}^{\eta,\rho}$ into LL_{pol} (proposition 2). In order to obtain the first one, S_1^{ax}-cut cannot be ρ-constrained (unconstrained S_1^{ax}-cut might be created during the reduction of other cuts). As a consequence we asked the P-embedding not to introduce a promotion rule in the case of an S_1^{ax}-cut, for example:

$$\dfrac{\dfrac{}{\vdash (\neg A)^t, A^q} \qquad \vdash (\neg A)^t, B^t \vee_m C^t, \Pi^q}{\vdash (\neg A)^t, B^t \vee_m C^t, \Pi^q}$$

$$\downarrow$$

$$\dfrac{\dfrac{}{\vdash P(A)^\perp, P(A)} \qquad \dfrac{\vdash P(A)^\perp, P(B) \parr P(C), P(\Pi)}{\vdash P(A)^\perp, P(B) \parr P(C), ?P(\Pi)}}{\vdash P(A)^\perp, P(B) \parr P(C), ?P(\Pi)}$$

and it would not be possible to promote $P(A)^\perp$ in $\vdash P(A)^\perp, P(B) \parr P(C), P(\Pi)$, as we do for an $S_1^{\neg ax}$-cut.

From this point of view, the distinction between S_1^{ax}-cuts and S_2-cuts looks rather artificial, and one would be tempted to split differently the structural cuts: S_2-cuts and S_1^{ax}-cuts on the one hand and $S_1^{\neg ax}$-cuts on the other hand. This is what we did in definition 8, when we translated the cut rule.

3. The ρ-constraint on cuts is not as strong as it seems: one cannot perform a cut between two arbitrary proofs but this becomes possible after one step of reversion of one of the proofs (see theorem 3).

4. Let us notice that the ρ-constraint and the η-constraint have not exactly the same nature. Indeed, the ρ-constraint regards only the sequents that are premises of some particular rules, and does not ask anything to the rules of the subderivations premises. In this sense ρ is more local than η.

Definition 10 (Reversibility complexity)
Let A be an LK_{pol}^{η}-formula. The reversibility complexity of A, that we will denote by c_A, is defined as follows:

- if A is an atomic formula or a q-coloured formula, then $c_A = 0$;
- if $A = \neg V$ or $A = \neg F$, $c_A = 1$;
- if $A = B \wedge_a C$ or $B \vee_m C$ then $c_A = c_B + c_C + 1$.

The reversibility complexity of a sequent $\Gamma = A_1, \ldots, A_n$ is the sum of the reversibility complexity of the (occurrences of) formulae contained in Γ: $c_\Gamma = c_{A_1} + \cdots + c_{A_n}$.

From now on, we will use the following language convention: if π is an LK^{η}_{pol} derivation containing the rules R and R', we will say that "the rule R' follows the rule R in π" when there exists a subderivation of π, containing R, and that is a premise of R'. We will say that "the sequent Γ follows R in π" when Γ is the sequent conclusion of a subderivation of π, containing R. We will say that "(the occurrence of) the formula A follows the rule R in π" when $A \in \Gamma$ and Γ follows R in π.

Theorem 3 (Completeness)
Let π be an LK^{η}_{pol} derivation. There exists in LK^{η}_{pol} a ρ-constrained derivation π^{ρ} with the same sequent conclusion as π.

Proof: The proof is by induction on the number of ρ-forbidden rules.

Let $n \geq 1$ be the number of ρ-forbidden rules in π. We prove that there exists a derivation π_1^{ρ} of the same sequent containing at most $n-1$ ρ-forbidden rules. Let us consider a ρ-forbidden rule R such that the subderivation(s) of π premise(s) of R is (are) ρ-constrained. We distinguish three cases:

- if R is a structural rule on a reversible formula A, then we proceed by induction on the reversibility complexity of A: we reverse A in the subderivation terminating by the rule R and we apply (if necessary) the induction hypothesis. By means of this transformation we have ρ-constrained R without introducing new ρ-forbidden rules.

- if R is an $S_1^{\neg ax}$-cut then the context Γ of the t-coloured cut formula contains at least a reversible formula A. We proceed by induction on the reversibility complexity of Γ. We reverse A in the subderivation terminating by the rule R and we apply (if necessary) the induction hypothesis. By means of this transformation we have ρ-constrained R without introducing new ρ-forbidden rules.

  Notice that one of the effects of these reversions is that some reversible rules have now a different position: they follow the $S_1^{\neg ax}$-cut(s) generated by R. It is important to convince oneself that this cannot violate the η-constraint.

- if R is an irreversible rule, then the context Γ (resp. the union of the contexts Γ and Γ') of the formula (resp. the formulae) t-coloured and active in R contains at least a reversible formula A. As in the foregoing case we proceed by induction on the reversibility complexity of Γ (resp. $\Gamma \cup \Gamma'$). In order to reverse A we need to look for the place where it is possible to perform a cut on A without breaking the η-constraint and without adding a ρ-forbidden rule: the natural place to do so is the subproof premise of the rule of π in which A is active. If such a rule does not exist then A is among the conclusions of π and we reverse A in π.

During this transformation the reversibility complexity of the sequents premises of R decreases, preserving the ρ-constraint and without introducing ρ-forbidden rules. □

Remark: For some LK^{η}_{pol} derivation, the derivation obtained by the foregoing procedure is not unique. However this procedure gives derivations which are equal up to reversions. Also, we will see in section 5 that when they are P-embedded into LL all these derivations have the same interpretation (by proposition 6).

In order to prove the stability of the ρ-constraint, we use the notion of *residue* of a logical (resp. cut) rule of an LK^{tq}-proof π in a proof π', where π' is obtained from π applying some steps of tq-reduction. We do not give here the definition of such a notion (rather clear intuitively), which can be found for example in [JSTdF02]. Notice that a logical cut of π to which the tq-step is applied has no residue in π'.

The following lemma is a straightforward consequence of the stability lemma (lemma 3) of [JSTdF02]. It basically says that the kind of a cut cannot "increase" during tq-reduction.

Lemma 1
Let π be an LK^{tq}-proof, let π' be a tq-reduct of π and let c be a cut rule of π. Then:

- *If c is a logical cut, then all of c's residues in π' are logical cuts.*
- *If c is a structural cut of kind S_2, then none of c's residues in π' is a structural cut of kind S_1.*
- *If c is a structural cut of kind S_1^{ax}, then none of c's residues in π' is a structural cut of kind $S_1^{\neg ax}$.*

Otherwise stated, the kind of a residue of a cut cannot be bigger than the kind of the original cut (according to the order of definition 6).

Theorem 4 (Stability)
The ρ-constraint in LK^{η}_{pol} is stable with respect to the cut elimination.

Proof: We outlined the *locality* of the ρ-constraint: the fact that, in an LK^{η}_{pol} derivation π, some rule satisfies the ρ-constraint does not depend on the previous rules. It is also clear that if R is a rule of the LK^{η}_{pol} derivation π following a cut c and if π' is obtained from π by eliminating the cut c, then R still satisfies the ρ-constraint in π' (using lemma 1 in the case of a cut rule R immediately following c). We will therefore focus on the rules preceding the cut c.

We will suppose that π is obtained by performing a cut between π_1 and π_2:

$$\dfrac{\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\vdash \Gamma, A^t}\,R_1 \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\vdash \Delta, (\neg A)^q}\,R_2}{\vdash \Gamma, \Delta}\,c$$

Let π' be the derivation obtained by eliminating the cut c. We have to check that π' still satisfies the ρ-constraint. We have to consider all the possible kinds of cut.

- Let us suppose that c is a logical cut. We denote by B and C the A's main sub-formulae.

  We deal with the case of the logical cut $\wedge_m/\vee_m$ and leave the other cases to the reader.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}1\\ \vdots\end{matrix}}{\vdash B, C, \dots}\,T}{\vdash A^t, \dots}\,R_1 \qquad \dfrac{\dfrac{\begin{matrix}2\\ \vdots\end{matrix}}{\vdash \neg B, \dots}\,T' \quad \dfrac{\begin{matrix}3\\ \vdots\end{matrix}}{\vdash \neg C, \dots}\,T''}{\vdash (\neg A)^q, \dots}\,R_2}{\vdash \dots}\,c$$

  The rule R_1 is reversible, the rule R_2 is irreversible. The derivation π' is:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}1\\ \vdots\end{matrix}}{\vdash B, C, \dots}\,T \quad \dfrac{\begin{matrix}2\\ \vdots\end{matrix}}{\vdash \neg B, \dots}\,T'}{\vdash C, \dots}\,c_1 \qquad \dfrac{\begin{matrix}3\\ \vdots\end{matrix}}{\vdash \neg C, \dots}\,T''}{\vdash \dots}\,c_2$$

  Only the rules c_1 and c_2 are in π' and not in π. We have to check they are ρ-constrained. Since the ρ-constrained cuts are necessarily S_1-cuts, we will suppose that c_1 is an S_1-cut.

  In this case, if $\neg B$ is q-coloured the η-constraint (in π) forces $\neg B$ to be main in T'. Two cases may arise:

  1. $\neg B$ is main in a logical rule: c_1 is therefore S_2 not S_1.
  2. $\neg B$ is main in an axiom: therefore we are in the case of a S_1^{ax}-cut without ρ-constraint.

  If c_1 is an S_1-cut, we can then suppose that B is q-coloured. The formula $\neg B$ (t-coloured and active in c_1) is in the sequent conclusion of T'. By hypothesis π is ρ-constrained. It follows that in this sequent $\neg B$ is in a non-reversible context. The cut c_1 satisfies the ρ-constraint.

  The case of an S_1-cut c_2 can be dealt with in the same way.

In the following cases, c is a structural cut and we will show that all the rules for which a sequent premise changed are still ρ-constrained.

It will not be necessary to deal with the residues of c in π', since they will necessarily be S_2-cuts or logical cuts (without ρ-requirements). Moreover, lemma 1 guarantees that every $S_1^{\neg ax}$-cut in π' is a residue of an $S_1^{\neg ax}$-cut of π. This implies, more generally, that every π'-rule which has to fulfill the ρ-constraint is a residue of a π-rule which has itself to fulfill the ρ-constraint.

- Suppose that c is an S_2-cut. The rule R_2 is logical and introduces $(\neg A)^q$. This means that A^t is reversible and because of the ρ-constraint it has not been contracted nor weakened in π_1; the only leaf of the tree of ancestors of A^t is an axiom or a reversible rule T introducing A^t. The derivation π' is then as follows (in the case of an axiom you have to replace by π_2 the subderivation of π' with last rule c'):

$$\dfrac{\dfrac{\vdots}{\vdash A^t, \Gamma'}\,T \qquad \dfrac{\overset{\pi_2}{\vdots}}{\vdash (\neg A)^q, \Delta}\,R_2}{\vdash \Gamma', \Delta}\,c'$$

$$\dfrac{\vdots}{\vdash \Gamma, \Delta}\,R'_1$$

  Note that R'_1 is the rule R_1 of π (but with different premises).

  We have to check that all the rules which follow c' are ρ-constrained. This is easy. Indeed no sequent following T (until R'_1) can be liable to the ρ-constraint since in π the sequents which follow T contain the reversible and non-active formula A^t.

- Suppose that c is S_1.

  If R_2 is an axiom, π' is the subderivation π_1 of π (and we know by hypothesis it is ρ-constrained).

  Otherwise c is an $S_1^{\neg ax}$-cut, and because of the ρ-constraint in π there are no reversible formulae in Γ, which implies that π' is ρ-constrained. □

Proposition 2
The P-embedding is a decoration of $LK_{pol}^{\eta;\rho}$.

Proof: Thanks to the ρ-constraint, each time we have to perform a promotion in LL on the P-translation of a t-coloured formula, active in an irreversible rule or an $S_1^{\neg ax}$-cut, the context is correctly modalized (the formulae are q-coloured or atomic and then ?-prefixed). Moreover, structural rules in $\mathrm{LK}_{pol}^{\eta;\rho}$ are performed only on q-coloured or t-atomic formulae, which are ?-prefixed in the P-translation of formulae.

The reader should notice that in the case of an S_1^{ax}-cut (which *is not* a ρ-constrained rule), the P-embedding *does not* require any promotion rule: this is indeed the reason why we distinguished this case in definition 8. Notice also that S_1^{ax}-cuts *cannot* be ρ-constrained, because they naturally arise when reducing a logical cut (remember the proof of theorem 4). □

3 LLP

We now turn our attention to linear logic. We already pointed out in subsection 1.7 that every q-formula (resp. t-formula) of LK^{η}_{pol} is translated by a !-fix (resp. ?-fix) linear formula. The definitions of subsection 1.7 show that a *very small* subset of the set of !-fix and ?-fix linear formulae (which is itself a *very small* subset of linear formulae) is enough to represent classical logic. This suggests to precisely define and to study such a set of formulae.

We define the *polarized* subsystem LL_{pol} of LL and its natural extension LLP. We will prove in the next section some important properties of such systems w.r.t. normalization.

Definition 11 (Polarized formulae and LL_{pol})
If X is any atomic (linear) formula, a *polarized formula* is defined by

$$\begin{array}{lclcccccccccc} P & ::= & !X^{\perp} & \mid & P \otimes P & \mid & P \oplus P & \mid & 1 & \mid & 0 & \mid & !N \\ N & ::= & ?X & \mid & N \parr N & \mid & N \,\&\, N & \mid & \bot & \mid & \top & \mid & ?P \end{array}$$

without any succession of exponentials !? or ?! (except for $?!X^{\perp}$ and $!?X$). A formula of the shape P (resp. N) is *positive* (resp. *negative*). In the sequel P, Q, ... (resp. N, M, ...) will denote positive (resp. negative) formulae.

LL_{pol} is the subsystem of LL which uses only polarized formulae with the constraint: *only negative formulae can appear in the context of a &-rule and of a $\top$-rule.*

Remark: We can then reformulate proposition 2 in a more precise way: the P-embedding is a decoration of $\mathrm{LK}^{\eta,\rho}_{pol}$ whose image is contained in LL_{pol}. One just has to check that the P-image of every $\mathrm{LK}^{\eta;\rho}_{pol}$-proof is a proof of LL_{pol}, and this is immediate.

3.1 From structural rules to LLP

The study of the properties of polarized formulae will allow us to extend the system LL_{pol} to a richer one, LLP.

Lemma 2
*Let N be a negative formula, N is ?-fix (*i.e. $\vdash_{LL} !N^{\perp}, N$*).*

Proof: By induction on N. □

Lemma 3
In LL, if N is negative,

- *from* $\vdash \Gamma$ *we can deduce* $\vdash \Gamma, N$.
- *from* $\vdash \Gamma, N, N$ *we can deduce* $\vdash \Gamma, N$.
- *from* $\vdash N, N_1, ..., N_k$ *we can deduce* $\vdash !N, N_1, ..., N_k$.

Proof:

$$\dfrac{\dfrac{\vdash \Gamma}{\vdash \Gamma, ?N} \qquad \vdash !N^{\perp}, N}{\vdash \Gamma, N}$$

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma, N, N}{\vdash \Gamma, ?N, ?N}}{\vdash \Gamma, ?N} \qquad \vdash !N^{\perp}, N}{\vdash \Gamma, N}$$

$$\begin{array}{c}\dfrac{\dfrac{\dfrac{\vdash N, N_1, ..., N_k}{\vdash N, ?N_1, ..., ?N_k}}{\vdash !N, ?N_1, ..., ?N_k} \qquad \vdash !N_1^{\perp}, N_1}{\vdash !N, N_1, ?N_2, ..., ?N_k} \\ \vdots \\ \dfrac{\vdash !N, N_1, ..., N_{k-1}, ?N_k \qquad \vdash !N_k^{\perp}, N_k}{\vdash !N, N_1, ..., N_k}\end{array}$$

□

Definition 12 (LLP)
The system LLP is obtained by extending LL_{pol} with the three rules of lemma 3, and by defining the polarity of atomic (linear) formulae as follows: X is a positive formula and dually $X^{\perp}$ is negative.

The previous definition of polarity for the atoms allows us to define a second order extension of LLP (see section 7).

3.2 The P-embedding is a decoration

The P-embedding translates any t-formula as a negative formula and any proof of LK^{η}_{pol} as a proof of LL_{pol}. It is not a decoration, due to the structural rules of LK^{η}_{pol} on t-formulae and not only on formulae translated as $?A$ (for some A).

Observe that the LLP-rules which are not LL-rules are exactly the ones needed to turn the P-embedding into a decoration of LK^{η}_{pol} in LLP.

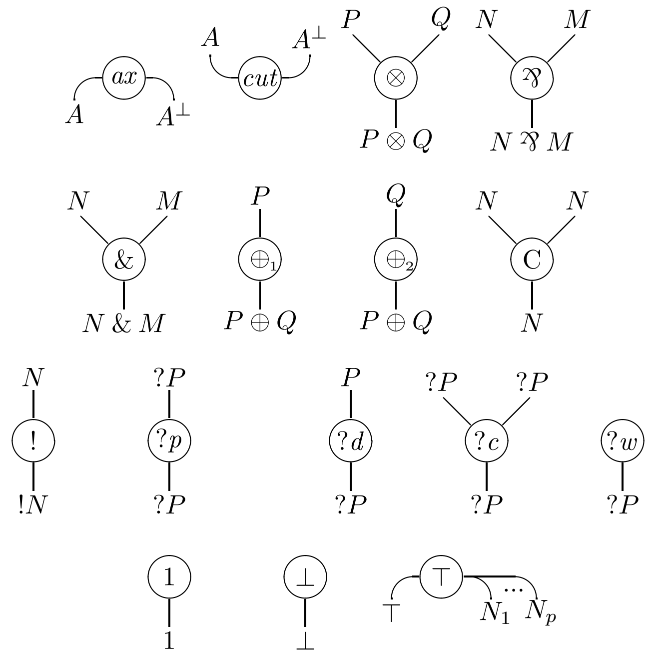


Figure 2: Proof-nets nodes

4 Polarized proof-nets

Proof-nets have been introduced for LL in [Gir87], but the cut elimination steps are difficult to define for the additive connectives [Gir96, TdF03b, TdF03a]. The main point of this section is precisely to show that the constraints of LL_{pol} are sufficient to give a very nice definition of these cut elimination steps, thanks to the notion of *positive tree* (see [Lau99]). We then move to a more general system of proof-nets (proof-nets for LLP), for which we prove confluence and strong normalization. The reader should notice that we are dealing with all the linear connectives, and that in a non-polarized framework confluence is wrong for full LL, and strong normalization has not been completely proven (see [TdF00]).

4.1 Definition

Definition 13 (Proof-structure)
A *proof-structure* is a finite directed acyclic graph built over the alphabet of nodes represented in figure 2 (where the orientation is the top-down one). Edges are typed with polarized formulae. The incident (top) edges of a node are its *premises* and the emergent (bottom) edges are its *conclusions*. Each edge is conclusion of exactly one node and premise of at most one node. The edges which are not premise of a node are the conclusions of the proof-structure.
Moreover:

- with each !-node is associated a !-box, that is a sub-proof-structure with conclusions Γ, N where N is the premise of the !-node and the formulae of Γ are premises of $?p$-nodes (these $?p$-nodes are *associated* with the !-node);
- with each &-node is associated a &-box, that is two disjoint sub-proof-structures with conclusions Γ, N and Γ, M where N and M are the two premises of the &-node and the pairs of corresponding formulae of the two Γ are premises of the same C-nodes (these C-nodes are *associated* with the &-node).
- Every $?p$-node (resp. C-node) is associated with exactly one !-node (resp. &-node).

The !-node (resp. the &-node) of a !-box (resp. &-box) is called its *main door* and the corresponding $?p$-nodes (resp. C-nodes) are called the *auxiliary doors*. Two boxes are either disjoint or included one in the other. The number of boxes in which a node is included is called its *depth*.

Definition 14 (Polarized edges and nodes)
Edges with a positive (resp. negative) type are *positive* (resp. *negative*) edges. The following nodes are called *positive* (resp. *negative*): $\otimes$, $\oplus_i$ and 1 (resp. ⅋, &, $?c$, $?w$, $\bot$ and $\top$), they have only positive (resp. negative) edges.

Definition 15 (Structural tree)
The *structural tree* of an edge a of type N is the tree (possibly empty) containing the edges of type N above a.

We will now define a new orientation of the edges of the proof-structures and in the sequel we will only talk about this new orientation and never about the orientation appearing in the definition of proof-structure (except through the notions of premise and conclusion, up and down).

Definition 16 (Correction graph)
The *correction graph* of a proof-structure is obtained by:

1. erasing the conclusion edges,
2. replacing any !-box (resp. &-box) at depth 0 with its !-node and the associated $?p$-nodes (resp. &-node and the associated C-nodes) by a generalized axiom node (*i.e.* without premises) with the same conclusions in the resulting graph,
3. orienting negative (resp. positive) edges downwardly (resp. upwardly).

Definition 17 (Correctness and Proof-nets)
A proof-structure is *correct* or is a *proof-net* if its correction graph is acyclic and has exactly one non-weakening (and non-$\bot$) initial node (*i.e.* without incident edge) and recursively each box contains proof-nets (one for !-boxes and two for &-boxes).

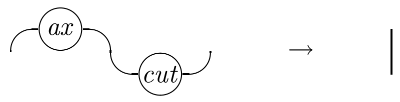


Figure 3: Axiom step

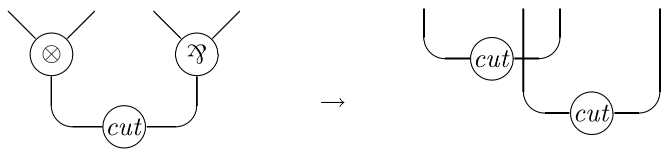


Figure 4: Multiplicative step

Remark: A positive conclusion of a proof-net is conclusion of an initial node of the correction graph thus, by the correctness criterion, a proof-net has at most one positive conclusion.

Remark: By definitions 16 and 17, the orientation of the correction graphs defines a partial order on the nodes of a proof-net.

We will now look at the different properties required for a good correctness criterion for proof-nets. The first point is the preservation of the correctness criterion w.r.t. the cut elimination steps (otherwise computation is impossible). The second point is the sequentialization property: correct proof-structures must correspond to proofs. A third crucial point is to study the properties of cut elimination: termination and confluence.

4.2 Cut elimination

These proof-nets correspond to a fragment of usual proof-nets, this is why we can use usual reduction steps for the following cuts (as described in [Dan90] for example):

- axiom (or *ax*-) step: figure 3
- multiplicative (or $\otimes$-$⅋$-) step: figure 4
- neutral multiplicative (or 1-$\bot$-) step: figure 5
- additive (or $\&$-$\oplus_i$-) step: figure 6
- contraction (or $?c$-$!$-) step: figure 7
- weakening (or $?w$-$!$-) step: figure 8
- dereliction (or $?d$-$!$-) step: figure 9

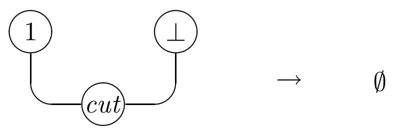


Figure 5: Neutral multiplicative step

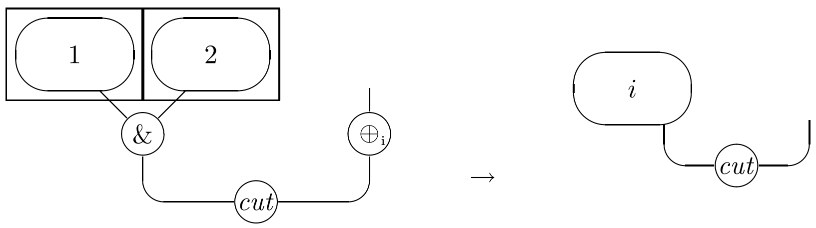


Figure 6: Additive step

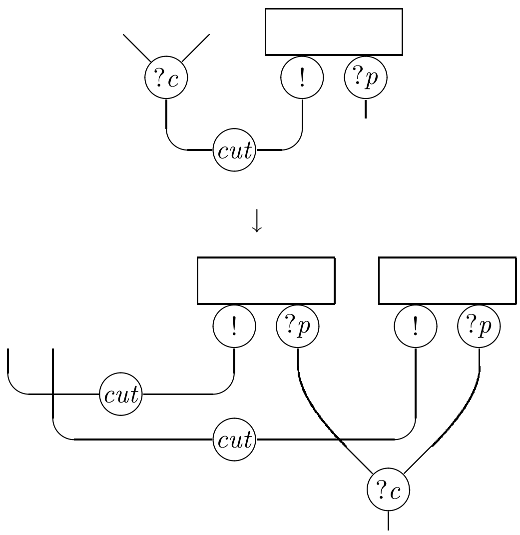


Figure 7: Contraction step

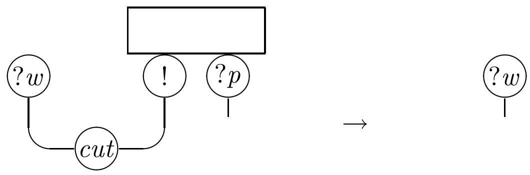


Figure 8: Weakening step

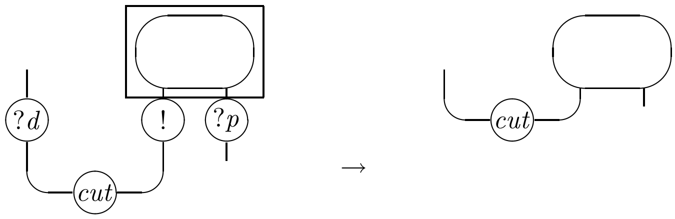


Figure 9: Dereliction step

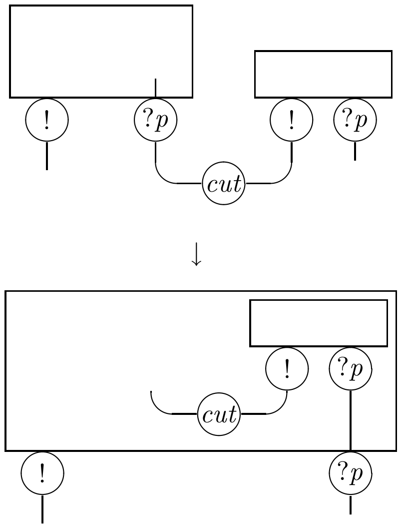


Figure 10: Commutative exponential step

- commutative exponential (or !-!-) step: figure 10

To define the C- and ⊤-steps, we need the notion of positive tree.

Definition 18 (Positive tree)
A *positive tree* is a particular proof-structure with exactly one positive conclusion called its *root*, the other conclusions (if any) are called the *leaves*. We define it by induction:

- a 1-node, a !-box or an axiom is a positive tree;
- adding a ⊗-node between the roots of two positive trees gives a positive tree;
- adding a ⊕-node on the root of a positive tree gives a positive tree.

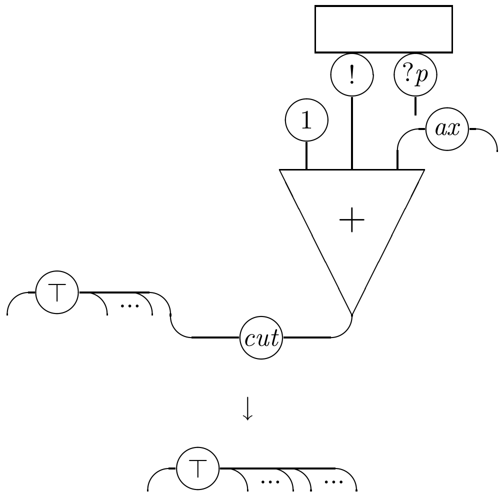


Figure 11: Neutral additive step in polarized proof-nets

Lemma 4
*A positive tree is a proof-net, more precisely it is the kingdom of its root (*i.e.* the smallest sub-proof-net having the root among its conclusions).*

Proof: By induction on the definition of a positive tree. □

The main property of these polarized proof-nets is that we can easily complete the set of steps of the cut elimination procedure to obtain a full system: any cut can be eliminated (as done for usual MELL proof-nets). This strongly uses the fact that the leaves of a positive tree are negative formulae:

- neutral additive (or $\top$-) step: figure 11

  If c is a cut on a conclusion N of a $\top$-node, the other premise $N^\perp$ of c is positive and we eliminate the cut by erasing the positive tree of $N^\perp$ and by adding its leaves as conclusions of the $\top$-node.

- commutative additive (or C-) step: figure 12

  If c is a cut on an auxiliary door N of a &-box, the other premise $N^\perp$ of c is positive and we eliminate the cut by duplicating the positive tree of $N^\perp$ (putting a copy of the tree in each side of the &-box and contracting the leaves with C-nodes).

Remark: For this last step, the additive critical pair of LL [TdF00] is obviously avoided by the polarization constraint: the two premises of a cut have distinct polarities.

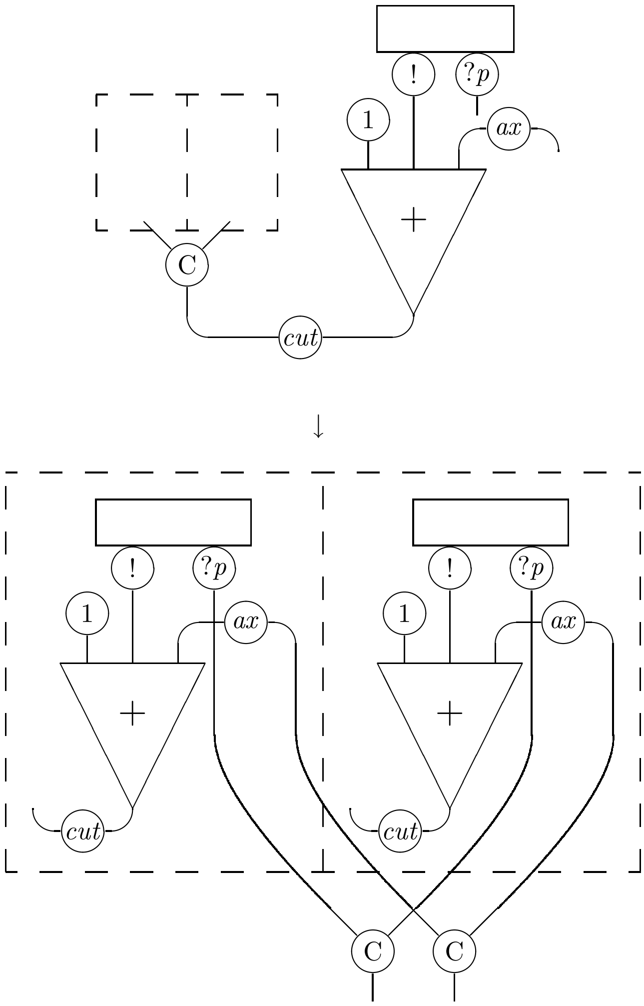


Figure 12: Commutative additive step in polarized proof-nets

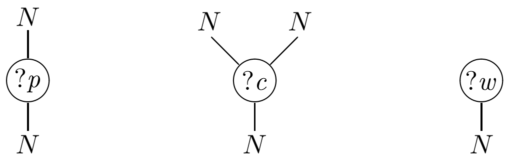
Proposition 3
The correctness of a proof-structure is preserved by any cut elimination step.

Proof: We only prove it is the case for the commutative additive step:

If the cut node c is at depth d, we have to look at the correction graph at depth d and at the two correction graphs inside the &-box at depth $d+1$. At depth d, we erase nodes and edges preserving the proof-structure properties, moreover we cannot create cycles and none of the erased nodes are initial ones. At depth $d+1$, we replace a proof-net by this proof-net cut against another proof-net (by lemma 4) thus we still have a proof-net. □

4.3 Extension to LLP

As we have seen, a positive tree is duplicable and erasable, just like a !-box. Using these properties we can generalize the structural nodes $?c$, $?w$ and $?p$ to any negative formula:



A cut on such a node has a positive tree above its positive premise, and this allows to extend to LLP the reduction steps (see figure 13 for the $?c$ case).

From now on, by *polarized proof-nets* we will mean proof-nets of LLP (remember that LLP contains LL_{pol}).

4.4 Sequentialization

We are going to prove that our notion of polarized proof-net (proof-structures satisfying the correctness criterion) corresponds to sequent calculus proofs by decomposing any proof-net into such a proof: the sequentialization theorem.

Definition 19 (Final negative node)
A negative node is *final* if it is at depth 0 and maximal for the partial order induced by the orientation of the correction graph except for a &-node which also requires that all the associated C-nodes are maximal for this order.

Theorem 5 (Sequentialization)
A proof-structure is correct iff it can be deconstructed inductively (or sequentialized) into a sequent calculus proof.

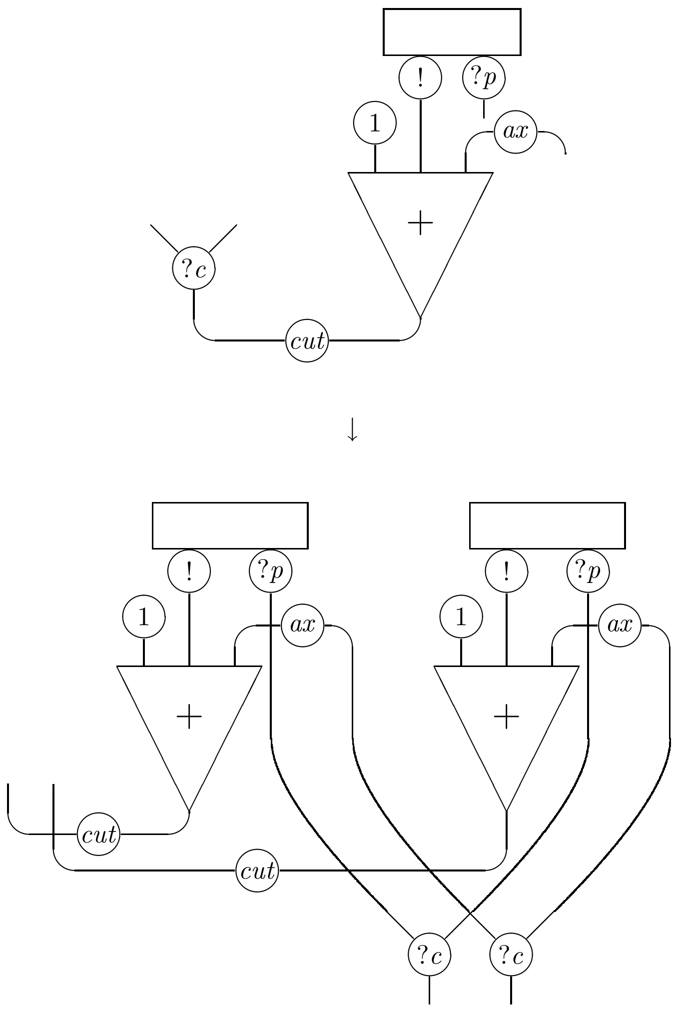


Figure 13: Contraction step in polarized proof-nets

Proof: We consider three main steps for the sequentialization. At depth 0:

- If there is a final negative node, we can erase it and we still have a proof-net (or two proof-nets for a &-node).
- If there is no final negative node but a cut (at depth 0), we consider a *cut*-node maximal with respect to the order induced by the orientation. This cut splits the proof-net into two proof-nets, more precisely the proof-net above its positive premise is exactly a positive tree. Otherwise we would have a positive tree with a node under one of its leaves and following the orientation from this node would eventually lead to a final negative node or to a cut (contradicting the maximality of the starting one).
- If there is no final negative node and no cut at depth 0, the proof-net is a positive tree which may have a $?d$-node under its root. This can be easily sequentialized from the bottom to the top because a positive tree always has a sequential structure by definition. □

A more detailed proof of sequentialization of polarized proof-nets without additives can be found in [Lau03].

4.5 Translation into taLL and properties of normalization

The translation of MELL polarized proof-nets as MELL proof-nets described in [Lau03] can be extended to a translation of polarized proof-nets as taLL proof-nets. taLL is the fragment of linear logic in which the context of the &- and $\top$-rules contains only ?-formulae [DJS97].

We decorate proofs with ? in such a way that the encoding of each negative formula begins with a ?:

$$\begin{array}{rcl} \overline{X^\perp} & = & ?X^\perp \\ \overline{N \parr M} & = & ?(\overline{N} \parr \overline{M}) \\ \overline{N \& M} & = & ?(\overline{N} \& \overline{M}) \\ \overline{\perp} & = & ?\perp \\ \overline{\top} & = & ?\top \\ \overline{?P} & = & ?\overline{P} \end{array}$$

The translation of positive formulae is obtained by duality. We translate proof-nets as follows:

- we replace each formula by the translated one,
- we add a $?d$-node under each logical negative node: $\parr$, &, $\perp$ and $\top$ (only under the main conclusion $\top$),
- we put a !-box around the positive tree of each positive node.

This translation emphasizes the fact that positive trees behave like boxes.

Remark: We use here a slight extension of the taLL system of [DJS97] with a $\top$-rule (with only ?-formulae in the context). The properties proved in [DJS97] are easy to extend to this setting (with the same proofs).

Proposition 4 (Simulation of reduction)
If R reduces to R' by one step of cut elimination then $\overline{R}$ reduces to $\overline{R'}$ by at least one step of cut elimination.

$$\begin{array}{ccc} R & \longrightarrow & R' \\ \big\downarrow & & \big\downarrow \\ \overline{R} & \xrightarrow{+} & \overline{R'} \end{array}$$

Proof: We only consider the case of the commutative additive step, leaving the others to the reader.

Let c be a commutative additive cut in R, if its premises are exponential formulae the simulation is done applying to $\overline{R}$ the same step applied to R. Otherwise the cut elimination step applied to R duplicates the positive tree of the positive premise of c and puts it (with the cut) inside the &-box (see figure 12), then by translation the two copies of this positive tree are put inside two !-boxes. In $\overline{R}$ the positive tree is already inside a !-box and the additive commutative step duplicates this box and puts it with the cut inside the &-box (see figure 14). □

Lemma 5 (Injectivity of the translation)
If R_1 and R_2 are two polarized proof-nets such that $\overline{R_1} = \overline{R_2}$ then $R_1 = R_2$.

Proof: We can define a reverse translation $\widetilde{R}$:

- in formulae, we erase each ! (resp. ?) applied on a positive (resp. negative) formula,
- we erase $?d$-nodes under negative nodes,
- we erase !-nodes under positive nodes, and we erase the associated $?p$-nodes.

It is clear that these exponentials are exactly those added by the $\overline{(.)}$ translation (because they are the only ones which do not respect polarities) thus $\widetilde{\overline{R}} = R$. □

Theorem 6 (Strong normalization)
There is no infinite sequence of cut elimination steps in polarized proof-nets.

Proof: A consequence of strong normalization for taLL [DJS97] by proposition 4. □

Theorem 7 (Confluence)
The normal form of a polarized proof-net is unique.

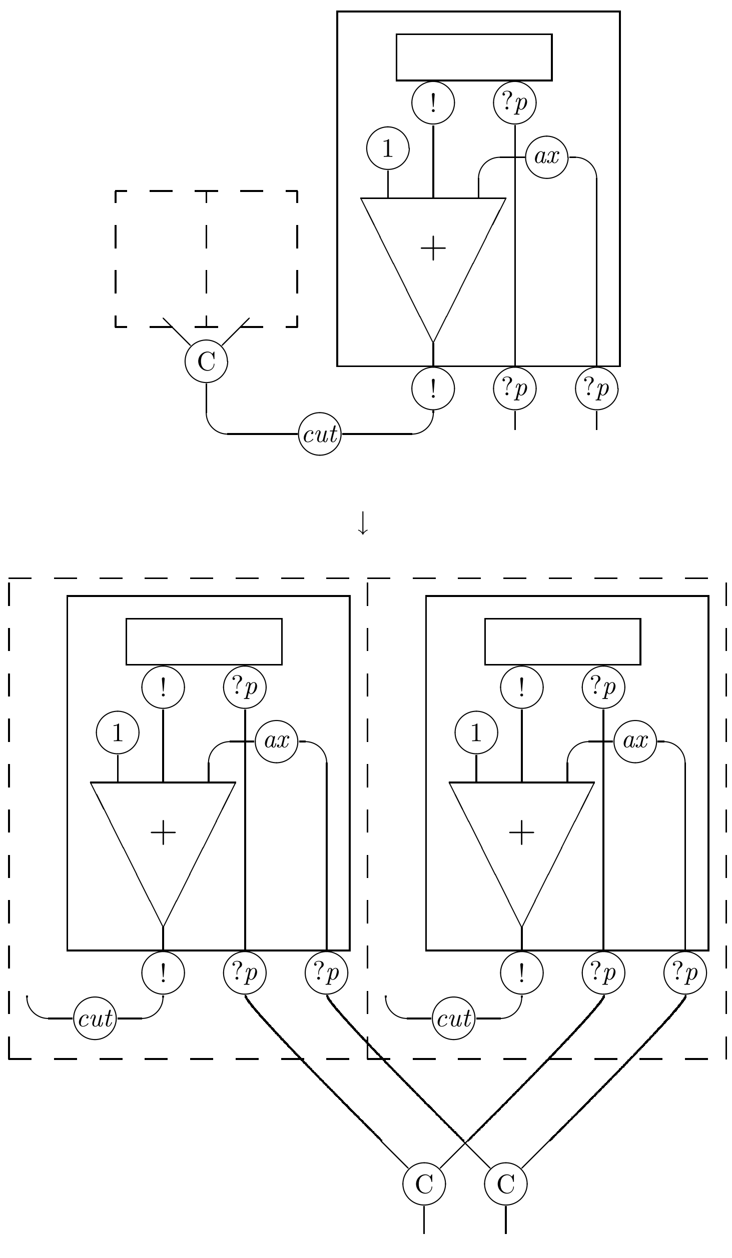


Figure 14: Commutative additive step applied to the $\overline{(.)}$ translation of polarized proof-nets

Proof: Let R_1 and R_2 be two normal forms of R, by proposition 4, $\overline{R_1}$ and $\overline{R_2}$ are two reducts of $\overline{R}$ which are obviously normal. Thus $\overline{R_1} = \overline{R_2}$ by confluence of taLL [DJS97] and by lemma 5, we have $R_1 = R_2$. □

4.6 Reversion

We are going to show how polarized proof-nets (that is proof-nets of LLP *without quantifiers*) can be translated as LL_{pol} polarized proof-nets. This means that we have to replace $?c$, $?w$ and $?p$ nodes acting on negative formulae by the corresponding nodes acting only on ?-formulae.

We consider the following transformation of a given LLP proof-net R:

- if it contains an atomic formula X (resp. $X^\perp$), we replace it by $!X^\perp$ (resp. $?X$);
- for each $?c$, $?w$ or $?p$ node acting on a negative formula which is not a ?-formula, we introduce a cut between the conclusion N of the node and the proof-net associated with the expansion of $\dfrac{}{\vdash N^\perp, N}\,ax$ (see appendix D), and we eliminate this cut, the new cuts this reduction generates, ... (we will call this sequence of reductions the *complete reduction* of the cut).

Proposition 5 (Linear reversion)
If, starting with an LLP proof-net, we apply the transformation described just above as many times as we can, we finally end with an LL_{pol} proof-net.

Proof: It is clear that an LLP proof-net on which the transformation is the identity is an LL_{pol} proof-net.

We now have to prove that the repetition of the application of the transformation always terminates. If R is a polarized proof-net, we consider the multiset containing, for each $?c$, $?w$ or $?p$ node, the associated negative formula. The result is obtained by induction on the sum of the sizes of the formulae of this multiset. □

5 Diagrams

In this section, we show three aspects of the relation between classical and linear logic, expressed by three commuting diagrams. We first prove (theorem 8) that tq-normalization in $\mathrm{LK}^{\eta}_{pol}/\mathrm{LK}^{\eta,\rho}_{pol}$ corresponds exactly to normalization of polarized proof-nets (of $\mathrm{LLP}/\mathrm{LL}_{pol}$). We then observe that the P-translation of classical reversion (the bridge between LK^{η}_{pol} and $\mathrm{LK}^{\eta,\rho}_{pol}$) is linear reversion (the bridge between LLP and LL_{pol}). Finally, we turn to denotational semantics: we notice (proposition 6) that reversion is "semantically invisible", and show that it is actually the "invisible bridge" between usual coherent semantics and Girard's correlation semantics introduced in [Gir91a] (in the polarized framework).

5.1 The syntactical diagram (simulation)

The P-embedding (resp. P_0-embedding) can be extended to a translation of LK^{η}_{pol} derivations into polarized proof-nets by using the natural translation of LLP sequent calculus into polarized proof-nets.

We will show in this subsection that this P-embedding from LK^{η}_{pol} into LLP is not only a static decoration but is setting also some dynamical correspondence: the P-embedding yields a simulation of classical reduction steps by proof-net reduction steps.

In order to prove one of the main results of the paper (theorem 8), we (again) use the notion of residue of a logical (resp. a cut) rule of an LK^{η}_{pol}-proof π in a proof π', where π' is obtained from π applying some steps of tq-reduction (see section 2). We also use lemma 1 of section 2.

Lemma 6
If π is a proof of $\vdash A^q, \Gamma$ with A^q main in the last rule of π, the proof-net $P_0(\pi)$ is a positive tree with conclusion $P(A)$.

Proof: By induction on π using the η-constraint. $\square$

Theorem 8
If the proof π' in LK^{η}_{pol} is obtained from π by one step of tq-reduction, the P-image R in polarized proof-nets of π reduces to the P-image R' of π' by some reduction steps.

$$\begin{array}{ccc} \pi & \xrightarrow{P} & R \\ \downarrow & & \downarrow * \\ \pi' & \xrightarrow{P} & R' \end{array}$$

Proof: We will use the same notations as in definition 8. Let c be a cut rule in the LK^{η}_{pol} derivation π between two π subderivations and suppose that the active formulae of c are A^t and $(\neg A)^q$. We denote by π_1 the sub-proof containing A^t and by π_2 the sub-proof containing $(\neg A)^q$.

In order to stress the main points of the proof, we are going to describe step by step two easy cases: c is an L-cut or an S_1^{ax}-cut. Part of the following discussion will be repeated later (when we will describe the more complicated $S_1^{\neg ax}$ and S_2 steps), but we hope that this way to argue makes the proof easier to grasp.

In both cases (L-cut and S_1^{ax}-cut) the reduction step does not involve a global move (it is *local*). If c is an L-cut, the classical reduction is simulated by a linear logical step (multiplicative or additive), which might be followed by a dereliction step (depending on the *kind* of the cut(s) created by the logical reduction step). If c is an S_1^{ax}-cut, the classical reduction is simulated by a linear axiom step.

Notice that this description is not very precise: it explains why the linear and classical movements correspond to each other, but it gives no details on the fact that the decoration of π' can be obtained by cut-elimination from the decoration of π. Let us consider the S_1^{ax} case: suppose that $\vdash \Gamma_1^t, \Delta_1^q, \Pi_1^q, A^t$ (with Π_1^q not empty) is the sequent conclusion of π_1, and that U is the rule following c in π. By definition 8, we know that in LL we will have $\vdash P(\Gamma_1^t), ?P(\Delta_1^q), ?P(\Pi_1^q), P(A)$ as the premise (and the conclusion) of the cut rule in $P(\pi)$. We need to check that the decoration of U in $P(\pi')$ requires a dereliction on $P(\Pi_1^q)$. We can exclude (because π is η-constrained) that U is an irreversible rule having as active formula a formula of Δ_1^q or Π_1^q, but we must look closely at the case in which U is a cut (in any other case the decoration of U in $P(\pi')$ requires a dereliction on $P(\Pi_1^q)$). The delicate case is when U is a cut of kind $S_1^{\neg ax}$ on Π_1^q in π and U (more precisely U's residue) is a cut of kind S_2 on Π_1^q in π': in this case definition 8 requires a dereliction on $P(\Pi_1^q)$ in $P(\pi)$ but no dereliction on $P(\Pi_1^q)$ in $P(\pi')$. Actually, what happens is that this dereliction disappears applying a dereliction step, transforming the decoration of an $S_1^{\neg ax}$-cut into the decoration of an S_2-cut. (We shall notice later that this is a general phenomenon: when the kind of a cut decreases, the decoration of the *new* cut can be obtained from the decoration of the *old* one by zero or one step of linear cut-elimination).

Similarly, in the case of the L-cut, one needs to check that the newly created cut rule(s) are correctly decorated, which is the case.

Let us now come to the more complicated cases of the structural $S_1^{\neg ax}$ and S_2 steps. We are going to follow the same pattern as in the previous cases: we first show that the linear and classical movements correspond to each other, and then that the decoration of π' can be obtained by cut-elimination from the decoration of π.

We look at the different cases of cut-elimination steps, and we describe the cut-elimination process both on the sequent calculus and on the proof-nets side. Concerning proof-nets, three stages can be distinguished: the first one is the reduction of the cut corresponding to c, the second one consists in applying some axiom, weakening and dereliction steps following (on the proof-nets side) the tq-reduction of c, finally the last stage consists in some more dereliction steps on other cuts (different from the one corresponding to c and its residues) necessary to obtain $P(\pi') = R'$. We call in the sequel $\tilde{R}$ the proof-net obtained after the first two stages.

The general pattern of a structural reduction step is described by figures 15 and 16:

1. Suppose that c is an S_2-cut. By η-constraint, we are exactly in the case where π_2 is translated by a positive tree with root $P(A)^\perp$ (by lemma 6).

   On the sequent calculus side, the proof π' is obtained by transporting the sub-proof π_2 upwards the sub-proof π_1: each time that some contraction rule on A^t or some $\wedge_a$-rule (in which A^t appears in the context) is gone through, π_2 is duplicated, and its conclusions are contracted. This process stops when we arrive at:

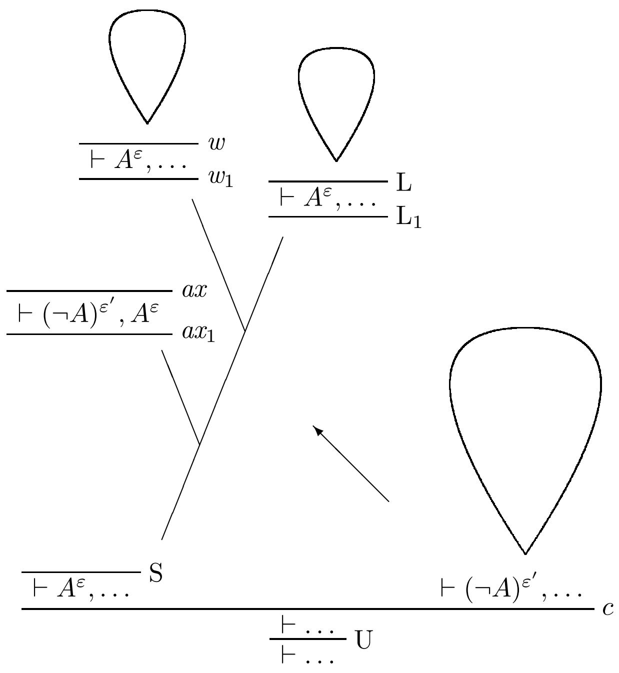


Figure 15: The structural cut $S_1^{\neg ax}$ or S_2 of the proof π.

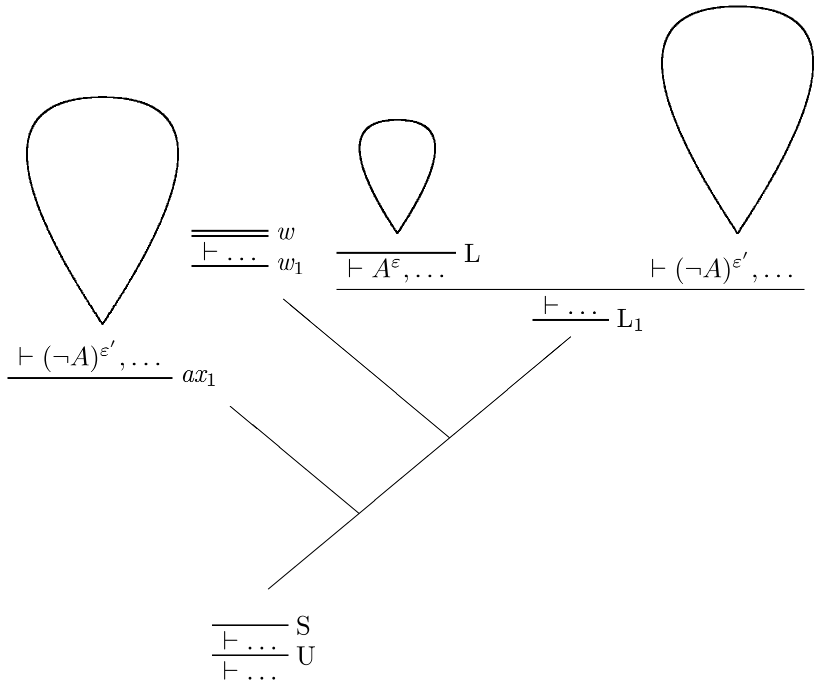


Figure 16: The proof π' obtained by an $S_1^{\neg ax}$ step or an S_2 step from the proof π.

- a weakening on A^t, in this case π_2 is erased and we replace, in π_1, the weakening with conclusion A^t by weakenings on the conclusions of π_2 (different from $(\neg A)^q$);
- an axiom introducing A^t, in this case the resulting axiom-cut is immediately reduced;
- a logical rule introducing A^t.

On the proof-nets side the corresponding process consists in at least one reduction step and leads us from R to $\tilde{R}$: in R the positive tree with root $P(A)^\perp$ is duplicated each time that $P(A)$ is conclusion of a $?c$-node or of a C-node, and new $?c$-nodes or C-nodes are performed on the leaves of the positive tree (which, remember this crucial point, are all negative formulae!). If there are in the translation of π_1 some $?w$-nodes with conclusion $P(A)$, some new $?w$-nodes with conclusions the leaves of the positive tree are added. Some supplementary steps might be necessary to reduce some axiom-cuts. And, in $\tilde{R}$ some logical cut nodes between $P(A)^\perp$ and $P(A)$ may arise.

2. Suppose now that c is an $S_1^{\neg ax}$-cut: $(\neg A)^q$ is not main in the previous rule and so it is translated in LLP by a ?-prefixed formula. So in order to be able to perform the cut in LLP it is necessary to use a promotion rule (!-node) on the formula $P(A)$.

   On the sequent calculus side we are in the symmetrical situation of the previous one: to obtain π' from π, we have to duplicate or erase π_1 moving up along π_2. And of course we have the same correspondence on the proof-nets side (except that now in a more traditional way we deal with a !-box instead of a positive tree) as long as we have to duplicate or erase the !-box. The simulation is slightly less immediate here. Indeed, each time that we arrive in π' to a logical rule or to an axiom rule there is a $?d$-node under the logical node or the axiom node on the proof-nets side. Supplementary steps are necessary to eliminate some $?d$-!-cut before recovering $\tilde{R}$.

Let us prove now that the decoration of π' can be obtained by cut-elimination from the decoration of π. We start with some general remarks:

1. the decoration of a rule T depends on the sequent(s) premise(s) of T and sometimes on the rule(s) immediately preceding T (when there is a main q-coloured formula in the sequent(s) premise(s) of T). More precisely, if a sequent premise of T contains a main q-coloured formula B^q, then:

   (a) B^q is active in T and T is irreversible or a cut (S_2 or logical), in which case there is no dereliction on $P(B)$ in the decoration of T

   (b) otherwise there is a dereliction on $P(B)$ in the decoration of T.

   Also, if T is the last rule of π, then its decoration might require one more dereliction

2. when the kind of a cut decreases (following the order of definition 6), the decoration of the *new* cut can be obtained from the decoration of the *old* one by zero or one step of linear cut-elimination.

It is then rather clear that the rules which might require a different decoration in π and π' are ax_1, w_1, L_1, S and U. We are going to check that this is never the case. We assume that the previous rules are five different occurrences of rules, leaving it to the reader to deal with the particular cases (for example S = w_1, S = ax_1, etc...). Notice that if any of the previous rules is a cut rule, then by lemma 1 its kind cannot increase during cut-elimination, and the previous remark allows to settle this case.

We now distinguish all the possible cases, in π, for the five previously mentioned rules (excluding the case of the cut-rule just settled). For the rule ax_1:

- if ax_1 is an irreversible rule having $(\neg A)^q$ among its active formulae, then c is necessarily an S_2-cut, $P_0(\pi_2)$ is duplicated and both in π and π' the linear translation of the rule ax_1 does not require any exponential rule on the premise containing $(\neg A)^q$;
- otherwise, if c is an S_2-cut, $P_0(\pi_2)$ is duplicated and both in π and π' a dereliction is added on $P(A)^\perp$ before applying the linear version of ax_1;
- otherwise, if c is an S_1-cut, let $\vdash \Gamma_1^t, \Delta_1^q, \Pi_1^q, A^t$ (with Π_1^q not empty) be the sequent conclusion of π_1. The dereliction on $P(A)^\perp$ preceding the linear version of ax_1 in $P(\pi)$ disappears during the cut-elimination process previously described (in a dereliction reduction step), and the dereliction on $P(\Pi_1^q)$ preceding (the promotion and) the linear cut in $P(\pi)$ is still present after cut-elimination, as required by the translation of the rule ax_1 of π'.

For the rule w_1, simply notice that one weakening is replaced by several weakenings and this cannot change anything to the decoration of w_1. Only a very particular case should be mentioned: when a weakening is replaced by *zero* weakenings (the context is empty in the duplicated subproof). Here we might be in trouble if w_1 could be an irreversible rule having an attractive premise among the formulae of the sequent conclusion of w: but this cannot be the case by η-constraint.

For the rule L_1:

- if c is an S_2-cut, $P_0(\pi_2)$ is duplicated and in $P(\pi)$ no dereliction is needed on any of the formulae of the sequent conclusion of L before applying the linear version of L_1, so as in $P(\pi')$ no dereliction is needed on any of the formulae of the sequent conclusion of c before applying the linear version of L_1. This is because in such sequents there is no main q-coloured formula. A promotion is required before applying the linear version of L_1 in $P(\pi)$ iff it is required before applying the linear version of L_1 in $P(\pi')$.

- if c is an S_1-cut, let $\vdash \Gamma_1^t, \Delta_1^q, \Pi_1^q, A^t$ (with Π_1^q not empty) be the sequent conclusion of π_1. Notice that L_1 cannot be η-constrained, so there has to be a dereliction on $P(A)^\perp$ preceding the linear version of L_1 in $P(\pi)$. Like in the case of ax_1, this dereliction disappears during the cut-elimination process previously described (in a dereliction reduction step), and the dereliction on $P(\Pi_1^q)$ preceding (the promotion and) the linear cut in $P(\pi)$ is still present after cut-elimination, as required by the translation of the rule (residue of) c of π'. A promotion is required before applying the linear version of L_1 in $P(\pi)$ iff it is required before applying the linear version of L_1 in $P(\pi')$.

For the rule S, there is nothing special to say, except if S becomes the last rule of π' (that is U does not exist): in this case we have to check that if there is a main q-coloured formula in the conclusion of S, there is a dereliction on it in $P(\pi')$. This is the case, because such a dereliction is also required by the decoration of c in $P(\pi)$.

For the rule U, notice that (again by η-constraint) U cannot be an irreversible rule having an attractive premise among the formulae of the sequent conclusion of c in π and of S in π'. If the conclusion of S contains an attractive formula B^q, there is a dereliction on $P(B)$ both in $P(\pi)$ and in $P(\pi')$. And again, a promotion is required before applying the linear version of U in $P(\pi)$ iff it is required before applying the linear version of U in $P(\pi')$. □

5.2 Reversions

Notice that the P-image of an LK^η_{pol} (resp. $LK^{\eta,\rho}_{pol}$) proof is an LLP (resp. LL_{pol}) proof. Both in the classical (section 2) and in the linear (section 4.6) case, the subsystem is obtained by a *reversion* procedure. It turns out that these two procedures coincide.

Theorem 9
If the proof π' in $LK^{\eta;\rho}_{pol}$ is obtained from the proof π in LK^η_{pol} by reversion, the P-image R' in polarized proof-nets of π' is obtained by reversion of the P-image R of π.

$$\begin{array}{ccc} LK^\eta_{pol} & \xrightarrow{P} & LLP \\ {\scriptstyle\rho}\big\downarrow & & \big\downarrow{\scriptstyle\rho} \\ LK^{\eta,\rho}_{pol} & \xrightarrow{P} & LL_{pol} \end{array}$$

Proof: Indeed let us recall that to reverse some LK^η_{pol} derivation π in order to obtain a ρ-constrained proof π^ρ we introduce and eliminate some cuts with the adequate η-proof each time it is necessary. What happens, is that if one applies the same treatment to the LLP proof-net R associated with $P(\pi)$, then one obtains the LL_{pol} proof-net R^ρ associated with $P(\pi^\rho)$.

We just have to remark that, refining theorem 8, if we apply a complete reduction of a cut in π this is simulated by the complete reduction of the corresponding cut in (the proof-net associated with) $P(\pi)$. $\square$

5.3 The semantical diagram

We have considered from all angles the syntactical translation of polarized classical logic into LL. In order to ensure that the translation is also dynamical, *i.e.* suitable with respect to the normalization process (basically, it is a decoration), we observed that there were two possible choices: either reduce the space of classical derivations by reversion ($\mathrm{LK}^{\eta;\rho}_{pol}/\mathrm{LL}_{pol}$) or extend the space of linear derivations ($\mathrm{LK}^{\eta}_{pol}/\mathrm{LLP}$). The previous sections convinced us that these two choices were perfectly equivalent in the following sense: it is the same thing to translate some LK^{η}_{pol} derivation π into LLP and then come back into LL (in fact LL_{pol}) to obtain a linear derivation or to reverse π and then to decorate it to obtain a linear derivation. We now turn our attention to the semantical features of this analysis.

As for section 1.6, we first consider $[R]$ to be the interpretation of the polarized proof-net R in a given (arbitrary) denotational model of LLP.

We start with some immediate consequences of theorem 8.

Corollary 1
The P-embedding of LK^{η}_{pol} in LLP induces a denotational semantics for LK^{η}_{pol}: if $\pi \to_{tq} \pi'$ in LK^{η}_{pol} then $[P(\pi)] = [P(\pi')]$.

Proof: If $\pi \to_{tq} \pi'$ then $P(\pi) \to^{*}_{LLP} P(\pi')$ thus $[P(\pi)] = [P(\pi')]$. $\square$

Proposition 6
Let π be an LK^{η}_{pol} derivation with conclusion $\vdash \Gamma$, and let A be a reversible formula of Γ. If π^r is the (LK^{η}_{pol}) derivation obtained by reverting A in π, then $[P(\pi)] = [P(\pi^r)]$.

Proof: We use here the content of subsection 1.5. Let η_A be the η-proof of $\vdash (\neg A)^q, A^t$ and let $\eta_{P(A)}$ be the η-proof of $\vdash P(A)^{\perp}, P(A)$. We will denote by $cut(\eta_A, \pi)$ (resp. $cut(\eta_{P(A)}, P(\pi))$) the LK^{η}_{pol} derivation (resp. the LL derivation) obtained by cutting the derivations η_A and π (resp. $\eta_{P(A)}$ and $P(\pi)$).

Remember now that $cut(\eta_A, \pi)$ tq-reduces to π^r (see the remark following definition 5), so that the previous corollary gives $[P(\pi^r)] = [P(cut(\eta_A, \pi))]$. To conclude it is enough to note that the derivations $P(cut(\eta_A, \pi))$ and $cut(\eta_{P(A)}, P(\pi))$ have the same interpretation (as proof-nets, the first one reduces to the second). From this we can deduce that $P(\pi^r)$ and $P(\pi)$ have the same interpretation since $\eta_{P(A)}$ is interpreted by the identity map on the space $P(A)$. $\square$

We can now strengthen the completeness of the ρ-constraint (expressed by theorem 3 of section 2) in the following way:

Theorem 10 (Strong completeness)
Let π be an LK^{η}_{pol} derivation. There exists in LK^{η}_{pol} a ρ-constrained derivation π^{ρ} with the same sequent conclusion as π. Moreover, one has $[P(\pi)] = [P(\pi^{\rho})]$.

Proof: To prove that $[P(\pi)] = [P(\pi^{\rho})]$, one simply has to notice that the proof π^{ρ} (defined in the proof of theorem 3) is obtained from π by performing some steps of reversion. Then one applies proposition 6. □

Notation. We consider in the sequel of the paper the usual denotational semantics for linear logic [Gir87]: formulae are interpreted by coherent spaces and derivations are interpreted by cliques. We use the multiset formulation [Gir91a] for the web of the spaces $!\mathcal{C}$ and $?\mathcal{C}$. We denote by $\mathcal{A}$ the coherent space associated with some formula A and we denote by $[\pi]$ the clique associated with the LL derivation π.

The correlation semantics of LLP. The correlation spaces are the main ingredient used by J.-Y. Girard in order to provide its classical logic system LC with a denotational semantics [Gir91a].

Polarized formulae can be interpreted by correlation spaces of the corresponding polarity because the structure of positive (resp. negative) correlation space is preserved by the constructions associated with the positive (resp. negative) connectives.

The structure of correlation space enables us to generalize to polarized formulae the interpretation of contraction, weakening, promotion, defined in LL only for spaces of the form $!\mathcal{A}$ or $?\mathcal{A}$ (which are indeed some particular cases of correlation spaces): let π be an LLP derivation, we associate with it some clique that we will denote $[\![\pi]\!]$ by using the usual inductive definition except that we use the generalized operation for the contraction rules, the weakening rules performed on negative formulae and the promotion rules performed with negative contextual formulae.

Remark: Any formula of LL_{pol} (and thus the P-translation of any LK^{η}_{pol}-formula) is naturally equipped with a correlation space structure induced by the coherent interpretation. Every atom X is sent on $!X$ and provided $\mathcal{X}$ is a coherent space, $!\mathcal{X}$ is a positive correlation space. Then the constructions used on the spaces associated with the classical polarized formulae preserve the correlation structure. Thus if A is a classical polarized formula, the space associated with $P(A)$ is of course a coherent space but can also be seen as a correlation space.

A denotational semantics for LK^{η}_{pol}. In order to provide a denotational semantics to classical derivations using the P-embedding, two choices are possible. If π is an LK^{η}_{pol}-proof, we can:

- reverse π to obtain the $LK^{\eta;\rho}_{pol}$-proof π^{ρ}, translate into LL_{pol} and compute the coherent interpretation $[P(\pi^{\rho})]$;

- translate π into LLP and compute the correlation interpretation $[\![P(\pi)]\!]$ (which can be seen as a coherent interpretation).

Proposition 7
Let π be an LK^{η}_{pol} derivation, we have that $[P(\pi^{\rho})] = [\![P(\pi)]\!]$.

Proof: It is clear that the two semantics (coherent spaces versus correlation spaces) coincide on the subsystem $\mathrm{LK}^{\eta;\rho}_{pol}$: if π' is some $\mathrm{LK}^{\eta;\rho}_{pol}$ derivation then $[P(\pi')] = [\![P(\pi')]\!]$. Moreover, by theorem 10, $[\![P(\pi)]\!] = [\![P(\pi^{\rho})]\!]$. $\square$

Remark: Reversion is the syntactical counterpart of correlation semantics. To ρ-constrain a derivation means to substitute the structural rules on reversible formulae by structural rules on their main sub-formulae. This is equivalent to consider that the coherent spaces associated with the reversible formulae are provided with semantical operations of contraction, weakening and promotion. Otherwise stated, the reversible classical polarized formulae are interpreted by negative correlation spaces.

6 Isomorphisms

In this section, we come back to "Girard's request of denotational isomorphisms" (point (iv) of subsection 1.6): we define a syntactical notion of *classical isomorphism*, which is not trivial, due to the fact that the cut rule of LK^{tq} *is not* an elementary operation (see section 6.2). Our notion of syntactical isomorphism allows to recover LC as a refinement of multiplicative LK^{η}_{pol} (proposition 10).

6.1 Some recalls on LC

In [Gir91a] the problem of extracting a computational content from classical derivations was considered from a mathematical point of view: the point is to provide classical logic with a denotational semantics. Following this approach, the question of the properties of such a semantics arises: is it enough to obtain a cut-elimination invariant?

Girard's answer is that we have to maximize the isomorphisms. It means that the structures associated with the classical formulae (in a given model) will have to satisfy isomorphisms such as commutativity, associativity, ... In other words we would like to recover some boolean equivalences as semantical isomorphisms. For example, if A, B and C are formulae respectively interpreted by the structures $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, we will require the existence of an isomorphism between $(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}))$ and $((\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C})$.

Girard showed that there exists a classical denotational semantics which satisfies "lots of" isomorphisms: it is the correlation spaces semantics. By means of this semantics he built the classical system LC and its cut-elimination procedure.

The P-translation of LC. Let us recall that LC is the classical polarized system with the two following main specificities:

- the LC-formulae are obtained by choosing the additive formulation for the conjunction between two t-coloured formulae and the multiplicative one otherwise. The P-translation of disjunction is deduced by duality;
- instead of being η-constrained the derivations are set into a peculiar sequent presentation: the *stoup* is a place in a sequent containing at most one positive formula which is "linear" (see section 1.4).

In order to study the P-embedding of LC it is more convenient to consider its η-constrained computational subsystem (thus stable and complete), in which the stoup-formulation is no more useful. It is then possible to give the scheme of the inductive P-translation of the LC-formulae in LLP:

$$\begin{array}{ccccc}
A^+ \wedge B^+ & \leftrightarrow & A^q \wedge_m B^q & \rightsquigarrow & P(A) \otimes P(B) \\
A^+ \wedge B^- & \leftrightarrow & A^q \wedge_m B^t & \rightsquigarrow & P(A) \otimes !P(B) \\
A^- \wedge B^+ & \leftrightarrow & A^t \wedge_m B^q & \rightsquigarrow & !P(A) \otimes P(B) \\
A^- \wedge B^- & \leftrightarrow & A^t \wedge_a B^t & \rightsquigarrow & P(A) \,\&\, P(B)
\end{array}$$

The P-translation of derivations is the one of LK^{η}_{pol} into LLP.

Notice that from an LK^{tq} point of view the system LC is rather surprising: it could be seen as an LK^{η}_{pol} subsystem but with only one conjunction (and of course one disjunction) which is now multiplicative now additive. In fact the P-image of LC is an LLP-subsystem, the fragment containing strongly polarized formulae.

Definition 20 (Strongly polarized linear formulae)
If X is any atomic (linear) formula, a linear formula P (resp. N) is *strongly positive* (resp. *strongly negative*) when it is built in the following way:

$$\begin{array}{lllllllllllllll}
P & ::= & !X^\perp & | & P \otimes P & | & P \otimes !N & | & !N \otimes P & | & P \oplus P & | & 1 & | & 0 \\
N & ::= & ?X & | & N \parr N & | & N \parr ?P & | & ?P \parr N & | & N \,\&\, N & | & \perp & | & \top
\end{array}$$

Remark: For example $!N$ or $?P \parr ?P$ are polarized formulae which are not strongly polarized.

6.2 About a syntactical notion of isomorphism

We can observe that the isomorphisms between correlation spaces previously mentioned are still satisfied by the P-semantics. In order to understand what this means for the LK^{η}_{pol} derivations, we define the notion of *syntactical isomorphism* for this system. This will give a new lighting on the connective-style choice of LC.

Notations: Let A be an LK^{η}_{pol}-formula, we will denote by ax_A the LK^{η}_{pol} derivation consisting in only one rule: the axiom rule $\vdash A, \neg A$. Recall that we denote by η_A the η-proof with the same conclusion.

Let π (resp. π') be an LK^{η}_{pol} derivation or an LL derivation containing A (resp. $\neg A$ or $A^{\perp}$) among its conclusions. We will denote by $cut_A(\pi, \pi')$ the derivation obtained by applying a cut rule on A between π and π'.

In the spirit of our analysis, let us first have a look on the syntactical notion of isomorphism in LL.

Definition 21 (LL-isomorphisms)
Let A and B be two LL-formulae and let ϕ (resp. ψ) be a cut-free proof of $\vdash A^{\perp}, B$ (resp. $\vdash B^{\perp}, A$). We will write $A \simeq_{\phi,\psi} B$ when $[cut_A(\psi, \phi)] = [ax_B]$ and $[cut_B(\phi, \psi)] = [ax_A]$.

Remark: In the definitions of isomorphisms, we will only deal with cut-free proofs, just because the extension to the general case does not seem to be of any interest and would surely make the presentation heavier.

Remark: Let us recall that the interpretation of an LL derivation is a linear map and the cut is interpreted by the composition between two maps. It is also interesting to outline that $A \simeq_{\phi,\psi} B$ can be read as $[cut_B(\phi, \psi)] = [\psi] \circ [\phi] = Id_{\mathcal{A}}$ and $[cut_A(\psi, \phi)] = [\phi] \circ [\psi] = Id_{\mathcal{B}}$ where $\mathcal{A}$ and $\mathcal{B}$ are the structures respectively associated with A and B. That is the structures associated with A and B are isomorphic: $\mathcal{A} \simeq \mathcal{B}$.

Let us come back to the classical case. The most natural definition with respect to the previous definition should be:

Definition 22 (LK^{η}_{pol}-isomorphisms: "pseudo" proposition)
Let A and B be two LK^{η}_{pol}-formulae and let ϕ (resp. ψ) be a normal proof of $\vdash \neg A, B$ (resp. $\vdash \neg B, A$). We will write $A \leftrightarrow_{\phi,\psi} B$ when $[P(cut_A(\psi, \phi))] = [P(ax_B)]$ and $[P(cut_B(\phi, \psi))] = [P(ax_A)]$.

Remark: One could give a purely syntactical definition of the previous isomorphism notion (without using the P-embedding neither the semantics) like in [DJS03].

Remark: **(Key example)** The foregoing isomorphism definition leads to isomorphisms between opposite-coloured formulae, for example: $A^t \wedge_a B^t \leftrightarrow A^t \wedge_m B^t$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg A^t, A^t}}{\vdash \neg A^t \vee_a \neg B^t, A^t} \qquad \dfrac{\dfrac{}{\vdash \neg B^t, B^t}}{\vdash \neg A^t \vee_a \neg B^t, B^t}}{\vdash \neg A^t \vee_a \neg B^t, \neg A^t \vee_a \neg B^t, A^t \wedge_m B^t}}{\vdash \neg A^t \vee_a \neg B^t, A^t \wedge_m B^t}$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \neg A^t, A^t}}{\vdash \neg A^t, \neg B^t, A^t} \qquad \dfrac{\dfrac{}{\vdash \neg B^t, B^t}}{\vdash \neg A^t, \neg B^t, B^t}}{\dfrac{\vdash \neg A^t, \neg B^t, A^t \wedge_a B^t}{\vdash \neg A^t \vee_m \neg B^t, A^t \wedge_a B^t}}$$

It is also clear that the structures associated with the formulae $A^t \wedge_m B^t$ and $A^t \wedge_a B^t$ will not be isomorphic. This is in fact our request in the search of a syntactical definition of isomorphism: if for some LK^{η}_{pol} derivations ϕ and ψ we have $A \simeq_{\phi,\psi} B$, then the spaces associated with $P(A)$ and $P(B)$ should be isomorphic.

We are going to see that the problem with the previous definition is due to the presence of certain structural rules.

The fact that the previous definition of syntactical isomorphism does not guarantee the existence of a semantical isomorphism is due to the cut rule interpretation. Indeed, the semantical operation associated with such a rule is not the composition of functions: the translation of a cut rule of LK^{η}_{pol} might involve a promotion rule (and not only a cut rule).

It seems that in classical logic the cut rule cannot be represented by an elementary mathematical operation. This is indeed true if we consider the functional approach (a derivation is a function) on which the Curry-Howard correspondence is based:

Remark: An LK^{η}_{pol} denotational semantics in which derivations are interpreted by functions and the cut by the composition of functions does not exist.

Proof: It is just a reformulation of the *critical pair* of Y. Lafont [GLT89]. Let us consider the following LK^{η}_{pol} derivation π (for simplicity the contexts are omitted):

$$\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \vdash A^q, \ldots\end{array} \quad \begin{array}{c}\pi_2\\ \vdots\\ \vdash (\neg A)^t, B^t, \ldots\end{array}}{\vdash B^t, \ldots}\,cut_1 \qquad \begin{array}{c}\pi_3\\ \vdots\\ \vdash (\neg B)^q, \ldots\end{array}}{\vdash \cdots}\,cut_2$$

In some semantics interpreting the cut as the composition, the derivation π would be interpreted by $[\pi_3]\circ([\pi_2]\circ[\pi_1]) = ([\pi_3]\circ[\pi_2])\circ[\pi_1]$. But $([\pi_3]\circ[\pi_2])\circ[\pi_1]$ is still associated with the following derivation π':

$$\dfrac{\begin{array}{c}\pi_1\\ \vdots\\ \vdash A^q, \ldots\end{array} \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdots\\ \vdash (\neg A)^t, B^t, \ldots\end{array} \quad \begin{array}{c}\pi_3\\ \vdots\\ \vdash (\neg B)^q, \ldots\end{array}}{\vdash (\neg A)^t, \ldots}\,cut_2}{\vdash \cdots}\,cut_1$$

This is very problematic: if for example A^q (resp. $(\neg B)^q$) is weakened in π_1 (resp. π_3) the identification of π and π' leads to an algorithmic inconsistency (all the derivations of the same conclusion would be identified by the denotational semantics). □

So, in order to be able to deduce $\mathcal{A} \simeq \mathcal{B}$ from $A \simeq_{\phi,\psi} B$, we will require that a cut between the LK^{η}_{pol} derivations ϕ and ψ is translated into LL by exactly a cut. In other words (following the definition of the P-embedding) we ask that the exponential which may prefix the cut-formulae in $P(\phi)$ and $P(\psi)$ are "superfluous".

The notion of superfluous exponential is intuitive enough. An exponential is superfluous when it is ... superfluous! It means that this exponential is useless in the derivation that we are considering.

Definition 23 (P-main formula)
Let A be a q-formula of the sequent conclusion of some LK^{η}_{pol} derivation π. Let R be the polarized proof-net associated with π. We denote by a the R-conclusion edge of type $?P(A)$. We will say that A is P-main in π when the structural tree of a (see definition 15) contains only C-nodes.

Remark: The fact that a formula A^q is P-main means that the $?d$-nodes with conclusion $?P(A)$ are "useless": $?P(A)$ is not active in a structural rule nor part of the context of a promotion rule. The introduced ? are only used for C-nodes.

Lemma 7
Let π be an LK^{η}_{pol} derivation with conclusion $\vdash \Gamma$. There is at most one P-main formula A in Γ.

Proof: By induction on the derivation π. □

Remark: To say that a formula A^q is P-main in the LK^{η}_{pol} derivation π is equivalent to say that it is in the stoup. In the language of Girard [Gir91a], A^q is P-main in π when the interpretation of $P(\pi)$ is a *central clique* in the correlation space associated with the sequent conclusion of π.

Terminology: For this reason we will speak of the *centrality* of an LK^{η}_{pol} derivation π when there exists a P-main formula among the conclusions of π.

6.3 The syntactical isomorphisms in LK^{η}_{pol}

We are now able to give the isomorphism definition in a classical setting:

Definition 24 (LK^{η}_{pol}-isomorphism)
Let A and B be two LK^{η}_{pol}-formulae. We will say that A and B are (ϕ, ψ)-isomorphic and we will write $A \simeq_{\phi,\psi} B$, when ϕ and ψ are central cut-free LK^{η}_{pol} derivations of respectively $\vdash \neg A, B$ and $\vdash \neg B, A$ and $A \leftrightarrow_{\phi,\psi} B$.

Remark: Observe that if $A \simeq_{\phi,\psi} B$ then $P(A)$ and $P(B)$ are LL-isomorphic (this is not the case for the example page 46).

Lemma 8
Let A and B be two LK^{η}_{pol}-formulae. If $A^{\varepsilon} \simeq B^{\mu}$ then $\varepsilon = \mu$.

Proof: Otherwise, we consider the case $\varepsilon = t \neq q = \mu$ and there is no central derivation of $\vdash \neg B, A$. □

This result was not true with the previous isomorphism definition (as the key example shows).

Proposition 8
Let A, B and C be LK^{η}_{pol}-formulae and let $\odot$ be an LK^{η}_{pol}-connective. If $A \simeq_{\phi,\psi} B$ then $A \odot C \simeq_{\phi',\psi'} B \odot C$.

Proof: We check it is the case for every connective. The important point is the following: since the q-coloured conclusion of ϕ or ψ has to be P-main, we are able to extend the derivations ϕ and ψ preserving the η-constraint. We then obtain two LK^{η}_{pol} derivations ϕ' and ψ' whose q-coloured conclusions are P-main. □

Proposition 9
Let A and B be two LK^{η}_{pol}-formulae. If $A \simeq_{\phi,\psi} B$ then the P-semantics identifies the following two derivations:

$$\dfrac{\begin{matrix}\pi_1 \\ \vdots \\ \vdash A, \ldots\end{matrix} \qquad \begin{matrix}\pi_2 \\ \vdots \\ \vdash \neg A, \ldots\end{matrix}}{\vdash \ldots}$$

and

$$\dfrac{\dfrac{\begin{matrix}\pi_1 \\ \vdots \\ \vdash A, \ldots\end{matrix} \qquad \begin{matrix}\phi \\ \vdots \\ \vdash \neg A, B\end{matrix}}{\vdash B, \ldots} \qquad \dfrac{\begin{matrix}\psi \\ \vdots \\ \vdash \neg B, A\end{matrix} \qquad \begin{matrix}\pi_2 \\ \vdots \\ \vdash \neg A, \ldots\end{matrix}}{\vdash \neg B, \ldots}}{\vdash \ldots}$$

Proof: Once more it is a consequence of centrality! □

Proposition 10
Let A and B be two t-coloured LK^{η}_{pol}-formulae. We have that $A^t \wedge_m B^t \simeq (A^t \wedge_a B^t) \wedge_m V$.

Proof: The point is to extend the two derivations of the example page 46 ($A^t \wedge_m B^t \leftrightarrow A^t \wedge_a B^t$) in such a way to respect centrality. $\square$

Remarks:

1. The foregoing proposition is the same thing as the well known isomorphism $!(A \mathbin{\&} B) \simeq\ !A \otimes !B$, but it was not so easy to express it in the classical language.

2. From an LK^{η}_{pol} point of view, the meaning of this proposition is that the multiplicative conjunction between two t-coloured formulae is not "primitive": it can be decomposed into an additive conjunction and a colour exchange.

   The system LC can also be seen as (an improvement of) the multiplicative fragment of LK^{η}_{pol}, and this explains also that LC's connectives are *associative*, for example:

$$\begin{aligned}(A^t \wedge_a B^t) \wedge_m C^q &\simeq (A^t \wedge_a B^t) \wedge_m (V \wedge_m C^q)\\ &\simeq ((A^t \wedge_a B^t) \wedge_m V) \wedge_m C^q\\ &\simeq (A^t \wedge_m B^t) \wedge_m C^q\\ &\simeq A^t \wedge_m B^t \wedge_m C^q\\ &\simeq A^t \wedge_m (B^t \wedge_m C^q)\end{aligned}$$

3. The fundamental isomorphism $!(A \mathbin{\&} B) \simeq\ !A \otimes !B$ is the unique bridge between the additive and multiplicative LL's fragments. Therefore the two fragments are not symmetrical. It is the reason why, in terms of isomorphisms, there is a unique optimal solution: LC.

7 Second order

All what we have shown up to now is true for two twins:

the couple $\mathrm{LK}^{\eta}_{pol}/\mathrm{LLP}$ on one side and the couple $\mathrm{LK}^{\eta,\rho}_{pol}/\mathrm{LL}_{pol}$ on the other side. We are going to see that one (and only one) of the two has to be ruled out as soon as one wishes to deal with second order quantification: $\mathrm{LK}^{\eta}_{pol}/\mathrm{LLP}$ survives, $\mathrm{LK}^{\eta,\rho}_{pol}/\mathrm{LL}_{pol}$ does not.

7.1 About the ρ-constraint

The ρ-constraint, defined for the propositional fragment, is not preserved by reduction in a second order framework. For example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash A^q} \quad \dfrac{\vdots}{\vdash B^t, X^t}}{\vdash A^q \wedge_m B^t, X^t}}{\vdash A^q \wedge_m B^t, \forall X X^t} \quad \dfrac{\dfrac{\vdots}{\vdash C^q}}{\vdash \exists X \neg X^t}}{\vdash A^q \wedge_m B^t}$$

$$\downarrow$$

$$\dfrac{\dfrac{\begin{matrix}\vdots \\ \vdash A^q\end{matrix} \qquad \begin{matrix}\vdots \\ \vdash B^t, (\neg C)^t\end{matrix}}{\vdash A^q \wedge_m B^t, (\neg C)^t} \qquad \begin{matrix}\vdots \\ \vdash C^q\end{matrix}}{\vdash A^q \wedge_m B^t}$$

The attentive reader surely noticed that the problem is due to the replacement of a t-coloured atom by an arbitrary t-coloured formula. On the linear side (after P-translation), this yields to the replacement of an occurrence of $?X$ by an arbitrary negative formula possibly contradicting LL-substitution.

This suggests to slightly modify the P-embedding at the atomic level, which cannot be done in a too narrow $\mathrm{LK}^{\eta,\rho}_{pol}/\mathrm{LL}_{pol}$ framework. We then move to $\mathrm{LK}^{\eta}_{pol}/\mathrm{LLP}$ where this modification is sound.

7.2 Second order polarization

Definition 25 (LLP^2)
Formulae of LLP^2 are obtained as follows:

$$\begin{array}{lcccccccccccccc} P & ::= & X & \mid & P \otimes P & \mid & P \oplus P & \mid & 1 & \mid & 0 & \mid & !N & \mid & \exists X P \\ N & ::= & X^\perp & \mid & N \parr N & \mid & N \,\&\, N & \mid & \bot & \mid & \top & \mid & ?P & \mid & \forall X N \end{array}$$

We add the two following rules for second order quantifiers:

$$\frac{\vdash \Gamma, N[{}^Y/_X]}{\vdash \Gamma, \forall X N} Y \notin \Gamma, N \qquad \frac{\vdash \Gamma, P[{}^Q/_X]}{\vdash \Gamma, \exists X P}$$

Polarized proof-nets can be extended with second order quantifiers as done in [Gir91b, Lau99]. All the results of section 4 (preservation of correction by reduction, sequentialization, confluence, strong normalization, ...) are preserved. In particular the translation in taLL can be straightforwardly extended to second order.

Definition 26 (P-translation and P-embedding)
We modify the definition 7 for the atoms and we add the quantifiers cases:

$$\begin{array}{lcl} X^q & \to & X \\ X^t & \to & X^\perp \\ (\neg X)^q & \to & X \\ (\neg X)^t & \to & X^\perp \\ & \vdots & \\ \exists X A^q & \to & \exists X P(A) \\ \exists X A^t & \to & \exists X !P(A) \\ \forall X A^q & \to & \forall X ?P(A) \\ \forall X A^t & \to & \forall X P(A) \end{array}$$

The simulation result (theorem 8) is still true, thus the P-embedding provides second order LK^{η}_{pol} with any denotational semantics of LLP^2 (see for example [Qua96]).

In fact, all the analysis of the couple LK^{η}_{pol}/LLP done before is extensible without difficulty to the second order.

A The second order classical system LK^2

Axiom and cut:

$$\frac{}{\vdash A, \neg A}\,ax \qquad \frac{\vdash \Gamma, A \qquad \vdash \neg A, \Delta}{\vdash \Gamma, \Delta}\,cut$$

Multiplicative logical rules:

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \wedge_m B}\,\wedge_m \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee_m B}\,\vee_m$$

Additive logical rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \vee_a B}\,\vee_a^1 \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \vee_a B}\,\vee_a^2 \qquad \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \wedge_a B}\,\wedge_a$$

Structural rules:

$$\frac{\vdash \Gamma}{\vdash \Gamma, A}\,\mathrm{W} \qquad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\,\mathrm{C}$$

Rules for the units:

$$\frac{}{\vdash V}\,V \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \neg V}\,\neg V \qquad \frac{}{\vdash \Gamma, \neg F}\,\neg F$$

Rules for the second order quantifiers (Y fresh):

$$\frac{\vdash \Gamma, A[{}^{Y}/_{X}]}{\vdash \Gamma, \forall X A}\,\forall \qquad \frac{\vdash \Gamma, A[{}^{B}/_{X}]}{\vdash \Gamma, \exists X A}\,\exists$$

B The second order linear system LL^2

Axiom and cut:

$$\frac{}{\vdash A, A^{\perp}}\,ax \qquad \frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\,cut$$

Multiplicative logical rules:

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\parr$$

Additive logical rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\oplus^1 \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\oplus^2 \qquad \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\&$$

Exponential logical rules:

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}! \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}?\mathrm{d}$$

Structural rules:

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}?\mathrm{w} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}?\mathrm{c}$$

Rules for the units:

$$\frac{}{\vdash 1}1 \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\bot \qquad \frac{}{\vdash \Gamma, \top}\top$$

Rules for the second order quantifiers (Y fresh):

$$\frac{\vdash \Gamma, A[{}^{Y}/_{X}]}{\vdash \Gamma, \forall X A}\forall \qquad \frac{\vdash \Gamma, A[{}^{B}/_{X}]}{\vdash \Gamma, \exists X A}\exists$$

C Classical expansion of axioms

$$\frac{}{\vdash A \wedge_m B, \neg A \vee_m \neg B}ax \quad \rightsquigarrow \quad \frac{\dfrac{\dfrac{}{\vdash A, \neg A}ax \qquad \dfrac{}{\vdash B, \neg B}ax}{\vdash A \wedge_m B, \neg A, \neg B}\wedge_m}{\vdash A \wedge_m B, \neg A \vee_m \neg B}\vee_m$$

$$\frac{}{\vdash A \vee_a B, \neg A \wedge_a \neg B}ax \quad \rightsquigarrow \quad \frac{\dfrac{\dfrac{}{\vdash A, \neg A}ax}{\vdash A \vee_a B, \neg A}\vee_a^1 \qquad \dfrac{\dfrac{}{\vdash B, \neg B}ax}{\vdash A \vee_a B, \neg B}\vee_a^2}{\vdash A \vee_a B, \neg A \wedge_a \neg B}\wedge_a$$

$$\frac{}{\vdash V, \neg V}\,ax \quad\leadsto\quad \frac{\dfrac{}{\vdash V}\,V}{\vdash V, \neg V}\,\neg V$$

$$\frac{}{\vdash F, \neg F}\,ax \quad\leadsto\quad \frac{}{\vdash F, \neg F}\,\neg F$$

$$\frac{}{\vdash \exists X A, \forall X \neg A}\,ax \quad\leadsto\quad \frac{\dfrac{\dfrac{}{\vdash A, \neg A}\,ax}{\vdash \exists X A, \neg A}\,\exists}{\vdash \exists X A, \forall X \neg A}\,\forall$$

D Linear expansion of axioms

$$\frac{}{\vdash A \otimes B, A^\perp \parr B^\perp}\,ax \quad\leadsto\quad \frac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,ax \qquad \dfrac{}{\vdash B, B^\perp}\,ax}{\vdash A \otimes B, A^\perp, B^\perp}\,\otimes}{\vdash A \otimes B, A^\perp \parr B^\perp}\,\parr$$

$$\frac{}{\vdash A \oplus B, A^\perp \,\&\, B^\perp}\,ax \quad\leadsto\quad \frac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,ax}{\vdash A \oplus B, A^\perp}\,\oplus^1 \qquad \dfrac{\dfrac{}{\vdash B, B^\perp}\,ax}{\vdash A \oplus B, B^\perp}\,\oplus^2}{\vdash A \oplus B, A^\perp \,\&\, B^\perp}\,\&$$

$$\frac{}{\vdash !A, ?A^\perp}\,ax \quad\leadsto\quad \frac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,ax}{\vdash A, ?A^\perp}\,?}{\vdash !A, ?A^\perp}\,!$$

$$\frac{}{\vdash 1, \perp}\,ax \quad\leadsto\quad \frac{\dfrac{}{\vdash 1}\,1}{\vdash 1, \perp}\,\perp$$

$$\frac{}{\vdash 0, \top}\,ax \quad\leadsto\quad \frac{}{\vdash 0, \top}\,\top$$

$$\frac{}{\vdash \exists X A, \forall X A^\perp}\,ax \quad\leadsto\quad \frac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,ax}{\vdash \exists X A, A^\perp}\,\exists}{\vdash \exists X A, \forall X A^\perp}\,\forall$$

References

[And90] Jean-Marc Andreoli. *Proposition pour une synthèse des paradigmes de la programmation logique et de la programmation par objets.* Thèse de doctorat, Université Paris VI, June 1990.

[AP91] Jean-Marc Andreoli and Remo Pareschi. Linear objects: logical processes with built-in inheritance. *New Generation Computing*, 9(3–4):445–473, 1991.

[Dan90] Vincent Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul).* Thèse de doctorat, Université Paris VII, 1990.

[DJS93] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: uncovering the dynamics of linear logic proofs. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Computational Logic and Proof Theory, KGC '93*, volume 713 of *Lecture Notes in Computer Science*, pages 159–171. Springer, 1993.

[DJS97] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A new deconstructive logic: linear logic. *Journal of Symbolic Logic*, 62(3):755–807, September 1997.

[DJS03] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. Computational isomorphisms in classical logic. *Theoretical Computer Science*, 294(3):353–378, February 2003.

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

[Gir91a] Jean-Yves Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, 1(3):255–296, 1991.

[Gir91b] Jean-Yves Girard. Quantifiers in linear logic II. In Corsi and Sambin, editors, *Nuovi problemi della logica e della filosofia della scienza*, pages 79–90, Bologna, 1991. CLUEB.

[Gir96] Jean-Yves Girard. Proof-nets: the parallel syntax for proof-theory. In Ursini and Agliano, editors, *Logic and Algebra*, New York, 1996. Marcel Dekker.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*, volume 7 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 1989.

[JSTdF02] Jean-Baptiste Joinet, Harold Schellinx, and Lorenzo Tortora de Falco. SN and CR for free-style LK^{tq}: linear decorations and simulation of normalization. *Journal of Symbolic Logic*, 67(1):162–196, March 2002.

[Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications '99*, volume 1581 of *Lecture Notes in Computer Science*, pages 213–227. Springer, April 1999.

[Lau03] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, January 2003.

[Par91] Michel Parigot. Free deduction: an analysis of computation in classical logic. In *Russian Conference on Logic Programming*, volume 592 of *Lecture Notes in Artificial Intelligence*, pages 361–380. Springer, 1991.

[Par92] Michel Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proceedings of Internationnal Conference on Logic Programming and Automated Deduction*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[QTdF96] Myriam Quatrini and Lorenzo Tortora de Falco. Polarisation des preuves classiques et renversement. *Compte-Rendu de l'Académie des Sciences de Paris*, 323:113–116, 1996.

[Qua96] Myriam Quatrini. A denotational semantics of LC2. *Archive for Mathematical Logic*, 35:1–32, 1996.

[Reg92] Laurent Regnier. *Lambda-Calcul et Réseaux*. Thèse de doctorat, Université Paris VII, 1992.

[TdF97] Lorenzo Tortora de Falco. Denotational semantics for polarized (but non-constrained) LK by means of the additives. In 5^{th} *Kurt Goedel Colloquium KGC'97, computational logic and proof theory*, volume 1289 of *Lecture Notes in Computer Science*, pages 290–304. Springer, 1997.

[TdF00] Lorenzo Tortora de Falco. *Réseaux, cohérence et expériences obsessionnelles*. Thèse de doctorat, Université Paris VII, January 2000.

[TdF03a] Lorenzo Tortora de Falco. The additive multiboxes. *Annals of Pure and Applied Logic*, 120(1–3):65–102, January 2003.

[TdF03b] Lorenzo Tortora de Falco. Additives of linear logic and normalization - Part I: a (restricted) Church-Rosser property. *Theoretical Computer Science*, 294(3):489–524, February 2003.