Intuitionistic Dual-intuitionistic Nets

Olivier LAURENT
Laboratoire de l'Informatique du Parallélisme
ENS Lyon – Université de Lyon
UMR 5668 CNRS ENS-Lyon UCBL INRIA
46, allée d'Italie
69364 Lyon cedex 07 – FRANCE
Olivier.Laurent@ens-lyon.fr

June 24, 2009

Abstract

The intuitionistic sequent calculus (at most one formula on the right-hand side of sequents) comes with a natural dual system: the dual-intuitionistic sequent calculus (at most one formula on the left-hand side). We explain how the duality between these two systems exactly corresponds to the intensively studied duality between call-by-value systems and call-by-name systems for classical logic.

Relying on the uniqueness of the computational behaviour underlying these four logics (intuitionistic, dual-intuitionistic, call-by-value classical and call-by-name classical), we define a generic syntax of nets which can be used for any of these logics.

Starting from Griffin's work [Gri90], the question of finding logical foundations of control operators in functional programming languages has been intensively studied. We focus on one particular line of work dealing with classical logical systems with a deterministic cut elimination procedure, translations into intuitionistic logic, linear logic analysis, constructions of denotational models, ... [Gir91a, Par92, LRS93, DJS97, SR98, HS02, Sel01].

An important by-product of these developments is the understanding of a duality between the call-by-name and the call-by-value evaluation procedures for classical logic [Fil89, Sel01].

In the present work, we try to clarify the relations between Gentzen's sequent calculus for classical logic, intuitionistic logic, call-by-name systems for classical logic (such as LKT [DJS95]), call-by-value systems for classical logic (such as LKQ [DJS95]), dual-intuitionistic logic [Cze77], ... The material presented here is not particularly new, but we think putting all the ingredients together (in particular for dual-intuitionistic logic which is usually not present in the picture) helps to understand the deep relations they have. The heart of this being the existence of *only one* computational behaviour corresponding to the cut elimination procedures of *all* the considered systems.

Uniqueness of computation in this setting allows us to introduce a common graphical syntax for all the presented systems. Taking inspiration from the work on polarized proof-nets [Lau03] for linear logic, we introduce *id-nets* (*i.e.* intuitionistic/dual-intuitionistic nets) which have the nice properties of proof-nets (such as a simple confluent and strongly normalizing cut-elimination procedure, the quotient of the irrelevant permutations of rules of the sequent calculus, ...) and which can be read both as a syntax for intuitionistic logic and for dual-intuitionistic logic.

We base our presentation of classical logic on the systems LKQ and LKT [DJS95], but a lot of other possibilities would have done the job as well (see for example [Par92, Lev99, CH00]). It is clear that the systems studied here have strong relations with linear logic [Gir87] and with the linear logic analysis of classical logic. However we try to avoid linear logic in this paper in order to make it readable by people not completely aware of linear logic developments.

1 Intuitionistic and dual-intuitionistic logics

The two basic systems we are going to study in this paper are intuitionistic logic and dual-intuitionistic logic. The duality between these two systems is the heart of the present work.

1.1 Implication-free intuitionistic logic

There is no particular need to present the intuitionistic sequent calculus. We just focus on some particular choices we make in the system LJ_0.

First, at the level of formulas, we consider all the propositional connectives but implication! This is a key restriction. The possibility of recovering implication is discussed in Section 3.1, but this only gives restricted rules for this connective and there is no reasonable way to do more without breaking strong properties of the systems. Due to this restriction, the intuitionistic system we use is not as strongly related with the λ-calculus as usual. This explains the name LJ_0, to avoid possible confusions with LJ in which implication is usually the core connective.

The grammar of formulas is:

$$A \quad ::= \quad X \quad | \quad \neg A \quad | \quad A \wedge A \quad | \quad A \vee A \quad | \quad \top \quad | \quad \bot$$

Sequents are of the shape $\Gamma \vdash \Pi$, with a finite multiset of formulas Γ on the left-hand side and at most one formula on the right-hand side Π (Π is either empty or reduced to a unique formula).

Rules are then the usual rules for such formulas and such sequents in intuitionistic logic (see Table 1).

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Gamma \vdash A \qquad A, \Delta \vdash \Pi}{\Gamma, \Delta \vdash \Pi}\, cut \qquad \frac{\Gamma \vdash \Pi}{\Gamma, A \vdash \Pi}\, wkL \qquad \frac{\Gamma, A, A \vdash \Pi}{\Gamma, A \vdash \Pi}\, ctrL$$

$$\frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}\, \neg R \qquad \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash}\, \neg L$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}\, \wedge R \qquad \frac{\Gamma, A, B \vdash \Pi}{\Gamma, A \wedge B \vdash \Pi}\, \wedge L \qquad \frac{}{\vdash \top}\, \top R \qquad \frac{\Gamma \vdash \Pi}{\Gamma, \top \vdash \Pi}\, \top L$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\, \vee R \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\, \vee R \qquad \frac{\Gamma, A \vdash \Pi \qquad \Gamma, B \vdash \Pi}{\Gamma, A \vee B \vdash \Pi}\, \vee L$$

$$\frac{}{\Gamma, \bot \vdash \Pi}\, \bot L$$

Table 1: Intuitionistic sequent calculus LJ_0

1.2 Dual-intuitionistic logic

While the intuitionistic restriction of the classical sequent calculus with at most one formula on the right-hand side of sequents is very common, the naive symmetric restriction (which is obviously well behaved with respect to cut elimination since the intuitionistic restriction is) is not so popular. One can find an explicit study of the induced system in [Cze77]. We use the name LD_0 for the slight variation we present here.

Formulas are the same as for LJ_0:

$$A \quad ::= \quad X \quad | \quad \neg A \quad | \quad A \wedge A \quad | \quad A \vee A \quad | \quad \top \quad | \quad \bot$$

Sequents now have at most one formula on the left-hand side: $\Pi \vdash \Gamma$ (Π is either empty or restricted to one formula, and Γ is a finite multiset of formulas).

The rules of LD_0 are presented in Table 2.

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Pi \vdash \Gamma, A \qquad A \vdash \Delta}{\Pi \vdash \Gamma, \Delta}\, cut \qquad \frac{\Pi \vdash \Gamma}{\Pi \vdash \Gamma, A}\, wkR \qquad \frac{\Pi \vdash \Gamma, A, A}{\Pi \vdash \Gamma, A}\, ctrR$$

$$\frac{A \vdash \Gamma}{\vdash \Gamma, \neg A}\, \neg R \qquad \frac{\vdash \Gamma, A}{\neg A \vdash \Gamma}\, \neg L$$

$$\frac{\Pi \vdash \Gamma, A, B}{\Pi \vdash \Gamma, A \vee B}\, \vee R \qquad \frac{A \vdash \Gamma \qquad B \vdash \Delta}{A \vee B \vdash \Gamma, \Delta}\, \vee L \qquad \frac{\Pi \vdash \Gamma}{\Pi \vdash \Gamma, \bot}\, \bot R \qquad \frac{}{\bot \vdash}\, \bot L$$

$$\frac{\Pi \vdash \Gamma, A \qquad \Pi \vdash \Gamma, B}{\Pi \vdash \Gamma, A \wedge B}\, \wedge R \qquad \frac{A \vdash \Gamma}{A \wedge B \vdash \Gamma}\, \wedge L \qquad \frac{B \vdash \Gamma}{A \wedge B \vdash \Gamma}\, \wedge L$$

$$\frac{}{\Pi \vdash \Gamma, \top}\, \top R$$

Table 2: Dual-intuitionistic sequent calculus LD_0

1.3 Duality

The perfect symmetry between the systems LJ_0 and LD_0 leads to a duality between them expressed through the following translation:

$$\begin{array}{rcl} \overline{X} & = & X \\ \overline{A \wedge B} & = & \overline{A} \vee \overline{B} \\ \overline{\top} & = & \bot \end{array} \qquad\qquad \begin{array}{rcl} \overline{\neg A} & = & \neg \overline{A} \\ \overline{A \vee B} & = & \overline{A} \wedge \overline{B} \\ \overline{\bot} & = & \top \end{array}$$

Proposition 1 (Duality)

- $\overline{\overline{A}} = A$
- $\Gamma \vdash_{\mathsf{LJ}_0} \Pi$ *if and only if* $\overline{\Pi} \vdash_{\mathsf{LD}_0} \overline{\Gamma}$

A kind of summary of this duality is presented in Table 3.

LJ_0		LD_0
$\wedge$	$\leftrightarrow$	$\vee$
$\top$	$\leftrightarrow$	$\bot$
$\neg$	$\leftrightarrow$	$\neg$
$\vee$	$\leftrightarrow$	$\wedge$
$\bot$	$\leftrightarrow$	$\top$
$\Gamma \vdash \Pi$	$\leftrightarrow$	$\Pi \vdash \Gamma$

Table 3: Duality summarized

2 Translations of classical logic

We now move to classical logic and to the question of the expressiveness of LJ_0 and LD_0.

In order to show that there are direct translations of call-by-value classical logic into LJ_0 and of call-by-name classical logic into LD_0, we rely on the systems LKQ and LKT presented in [DJS95]. These two systems provide us with a logical presentation of these two evaluation procedures (call-by-value with LKQ and call-by-name with LKT, see [Oga98]). This is one particular choice, a lot of other "equivalent" possibilities would work as well (see for example [Par92, Lev99, CH00] which describe pairs of dual systems for call-by-value and call-by-name classical logics).

2.1 Call-by-value

The system LKQ was originally introduced with the connectives $\rightarrow$ and $\forall$. It might be surprising that none of them appears in the system we present here. The point is that LKQ was designed from an analysis in linear logic which also gives the rules below when applied to the other propositional connectives. This justifies the name LKQ here. We discuss how the rules for implication can be recovered in Section 3.1 and the same for universal quantification in Section 5.2.

As for LJ_0 and LD_0, formulas are obtained from all the propositional connectives (but implication):

$$A \quad ::= \quad X \quad | \quad \neg A \quad | \quad A \wedge A \quad | \quad A \vee A \quad | \quad \top \quad | \quad \bot$$

Sequents have the particular shape $\Gamma \vdash \Delta \,;\, \Pi$, where Γ and Δ are finite multisets of formulas and Π contains at most one formula.

The rules of LKQ are given in Table 4.

We recall a key result of [DJS95] showing that LKQ is as expressive as LK.

Proposition 2 (Classical provability of LKQ)

- $\Gamma \vdash_{\mathsf{LKQ}} \Delta \,;\, \Pi \Longrightarrow \Gamma \vdash_{\mathsf{LK}} \Delta, \Pi$
- $\Gamma \vdash_{\mathsf{LK}} \Delta \Longrightarrow \Gamma \vdash_{\mathsf{LKQ}} \Delta \,;$

Proof: The first implication is immediate.

The second one is proved in [DJS95] for the connectives $\rightarrow$ and $\forall$ by means of linear logic. A direct translation in the spirit of Girard's LC [Gir91a] is also possible. A key case is given by the rule:

$$\frac{}{A \vdash \ ; A}\ ax \qquad \frac{\Gamma \vdash \Delta \ ; A \qquad A, \Gamma' \vdash \Delta' \ ; \Pi}{\Gamma, \Gamma' \vdash \Delta, \Delta' \ ; \Pi}\ cut \qquad \frac{\Gamma \vdash \Delta, A \ ; \Pi \qquad A, \Gamma' \vdash \Delta' \ ;}{\Gamma, \Gamma' \vdash \Delta, \Delta' \ ; \Pi}\ cut$$

$$\frac{\Gamma \vdash \Delta \ ; A}{\Gamma \vdash \Delta, A \ ;}\ der$$

$$\frac{\Gamma \vdash \Delta \ ; \Pi}{\Gamma, A \vdash \Delta \ ; \Pi}\ wkL \qquad \frac{\Gamma \vdash \Delta \ ; \Pi}{\Gamma \vdash \Delta, A \ ; \Pi}\ wkR \qquad \frac{\Gamma, A, A \vdash \Delta \ ; \Pi}{\Gamma, A \vdash \Delta \ ; \Pi}\ ctrL \qquad \frac{\Gamma \vdash \Delta, A, A \ ; \Pi}{\Gamma \vdash \Delta, A \ ; \Pi}\ ctrR$$

$$\frac{\Gamma, A \vdash \Delta \ ;}{\Gamma \vdash \Delta \ ; \neg A}\ \neg R \qquad \frac{\Gamma \vdash A, \Delta \ ; \Pi}{\Gamma, \neg A \vdash \Delta \ ; \Pi}\ \neg L$$

$$\frac{\Gamma \vdash \Delta \ ; A}{\Gamma \vdash \Delta \ ; A \vee B}\ \vee R \qquad \frac{\Gamma \vdash \Delta \ ; B}{\Gamma \vdash \Delta \ ; A \vee B}\ \vee R \qquad \frac{\Gamma, A \vdash \Delta \ ; \Pi \qquad \Gamma, B \vdash \Delta \ ; \Pi}{\Gamma, A \vee B \vdash \Delta \ ; \Pi}\ \vee L$$

$$\frac{\Gamma \vdash \Delta \ ; A \qquad \Gamma' \vdash \Delta' \ ; B}{\Gamma, \Gamma' \vdash \Delta, \Delta' \ ; A \wedge B}\ \wedge R \qquad \frac{\Gamma, A, B \vdash \Delta \ ; \Pi}{\Gamma, A \wedge B \vdash \Delta \ ; \Pi}\ \wedge L$$

$$\frac{}{\Gamma, \bot \vdash \Delta \ ; \Pi}\ \bot L \qquad \frac{}{\vdash \ ; \top}\ \top R \qquad \frac{\Gamma \vdash \Delta \ ; \Pi}{\Gamma, \top \vdash \Delta \ ; \Pi}\ \top L$$

Table 4: The sequent calculus LKQ

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\ \wedge R$$

which is translated as:

$$\dfrac{\Gamma \vdash A, \Delta \ ; \qquad \dfrac{\Gamma' \vdash B, \Delta' \ ; \qquad \dfrac{\dfrac{\dfrac{}{A \vdash \ ; A}\ ax \quad \dfrac{}{B \vdash \ ; B}\ ax}{A, B \vdash \ ; A \wedge B}\ \wedge R}{A, B \vdash A \wedge B \ ;}\ der}{\Gamma', A \vdash A \wedge B, \Delta' \ ;}\ cut}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta' \ ;}\ cut$$

An important point is the possibility to put the two cut rules in the opposite order [Gir91a].

The other cases are left to the reader. □

We now give a translation of LKQ into LJ_0. Together with the previous proposition, this shows how LJ_0 can represent LK proofs.

Proposition 3 (From LKQ to LJ_0)
If $\Gamma \vdash_{\mathsf{LKQ}} \Delta \ ; \Pi$ *then* $\Gamma, \neg\Delta \vdash_{\mathsf{LJ}_0} \Pi$.

Proof: We give the translation of each rule of LKQ:

$$\frac{}{A \vdash A}\ ax$$

$$\frac{\Gamma, \neg\Delta \vdash A \qquad A, \Gamma', \neg\Delta' \vdash \Pi}{\Gamma, \Gamma', \neg\Delta, \neg\Delta' \vdash \Pi}\, cut \qquad \frac{\dfrac{A, \Gamma', \neg\Delta' \vdash}{\Gamma', \neg\Delta' \vdash \neg A}\,\neg R \qquad \Gamma, \neg\Delta, \neg A \vdash \Pi}{\Gamma, \Gamma', \neg\Delta, \neg\Delta' \vdash \Pi}\, cut$$

$$\frac{\Gamma, \neg\Delta \vdash A}{\Gamma, \neg\Delta, \neg A \vdash}\,\neg L$$

$$\frac{\Gamma, \neg\Delta \vdash \Pi}{A, \Gamma, \neg\Delta \vdash \Pi}\, wkL \qquad \frac{\Gamma, \neg\Delta \vdash \Pi}{\Gamma, \neg\Delta, \neg A \vdash \Pi}\, wkL$$

$$\frac{A, A, \Gamma, \neg\Delta \vdash \Pi}{A, \Gamma, \neg\Delta \vdash \Pi}\, ctrL \qquad \frac{\Gamma, \neg\Delta, \neg A, \neg A \vdash \Pi}{\Gamma, \neg\Delta, \neg A \vdash \Pi}\, ctrL$$

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

$$\frac{\Gamma, \neg\Delta \vdash A}{\Gamma, \neg\Delta \vdash A \vee B}\,\vee R \qquad \frac{\Gamma, \neg\Delta \vdash B}{\Gamma, \neg\Delta \vdash A \vee B}\,\vee R \qquad \frac{A, \Gamma, \neg\Delta \vdash \Pi \qquad B, \Gamma, \neg\Delta \vdash \Pi}{A \vee B, \Gamma, \neg\Delta \vdash \Pi}\,\vee L$$

$$\frac{\Gamma, \neg\Delta \vdash A \qquad \Gamma', \neg\Delta' \vdash B}{\Gamma, \Gamma', \neg\Delta, \neg\Delta' \vdash A \wedge B}\,\wedge R \qquad \frac{A, B, \Gamma, \neg\Delta \vdash \Pi}{A \wedge B, \Gamma, \neg\Delta \vdash \Pi}\,\wedge L$$

$$\frac{}{\bot, \Gamma, \neg\Delta \vdash \Pi}\,\bot L \qquad \frac{}{\vdash \top}\,\top R \qquad \frac{\Gamma, \neg\Delta \vdash \Pi}{\top, \Gamma, \neg\Delta \vdash \Pi}\,\top L$$

□

Example 1
By composing Proposition 2 and Proposition 3, the rule:

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\,\wedge R$$

is translated as:

$$\dfrac{\neg L,\ \neg A, \neg\Delta \vdash \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\, ax \qquad \dfrac{}{B \vdash B}\, ax}{A, B \vdash A \wedge B}\,\wedge R}{A, B, \neg(A \wedge B) \vdash}\,\neg L}{A, \neg(A \wedge B) \vdash \neg B}\,\neg R \qquad \dfrac{\Gamma', \neg B, \neg\Delta' \vdash}{\Gamma', A, \neg(A \wedge B), \neg\Delta' \vdash}\,\neg R}{\Gamma, \Gamma', \neg(A \wedge B), \neg\Delta, \neg\Delta' \vdash}\, cut \qquad cut$$

2.2 Call-by-name

We now look at LKT as a logical system for call-by-name classical logic.

Formulas are the same as for LKQ:

$$A \;::=\; X \;\mid\; \neg A \;\mid\; A \wedge A \;\mid\; A \vee A \;\mid\; \top \;\mid\; \bot$$

Sequents have the particular shape $\Pi \,;\, \Gamma \vdash \Delta$, where Γ and Δ are finite multisets of formulas and Π contains at most one formula.

The rules of LKT are given in Table 5.

$$\frac{}{A \,;\, \vdash A}\, ax \qquad \frac{\Pi \,;\, \Gamma \vdash \Delta, A \qquad A \,;\, \Gamma' \vdash \Delta'}{\Pi \,;\, \Gamma, \Gamma' \vdash \Delta, \Delta'}\, cut \qquad \frac{\,;\, \Gamma \vdash \Delta, A \qquad \Pi \,;\, A, \Gamma' \vdash \Delta'}{\Pi \,;\, \Gamma, \Gamma' \vdash \Delta, \Delta'}\, cut$$

$$\frac{A \,;\, \Gamma \vdash \Delta}{\,;\, A, \Gamma \vdash \Delta}\, der$$

$$\frac{\Pi \,;\, \Gamma \vdash \Delta}{\Pi \,;\, \Gamma, A \vdash \Delta}\, wkL \qquad \frac{\Pi \,;\, \Gamma \vdash \Delta}{\Pi \,;\, \Gamma \vdash \Delta, A}\, wkR \qquad \frac{\Pi \,;\, \Gamma, A, A \vdash \Delta}{\Pi \,;\, \Gamma, A \vdash \Delta}\, ctrL \qquad \frac{\Pi \,;\, \Gamma \vdash \Delta, A, A}{\Pi \,;\, \Gamma \vdash \Delta, A}\, ctrR$$

$$\frac{\Pi \,;\, \Gamma, A \vdash \Delta}{\Pi \,;\, \Gamma \vdash \neg A, \Delta}\, \neg R \qquad \frac{\,;\, \Gamma \vdash A, \Delta}{\neg A \,;\, \Gamma \vdash \Delta}\, \neg L$$

$$\frac{\Pi \,;\, \Gamma \vdash A, B, \Delta}{\Pi \,;\, \Gamma \vdash A \vee B, \Delta}\, \vee R \qquad \frac{A \,;\, \Gamma \vdash \Delta \qquad B \,;\, \Gamma' \vdash \Delta'}{A \vee B \,;\, \Gamma, \Gamma' \vdash \Delta, \Delta'}\, \vee L$$

$$\frac{\Pi \,;\, \Gamma \vdash A, \Delta \qquad \Pi \,;\, \Gamma \vdash B, \Delta}{\Pi \,;\, \Gamma \vdash A \wedge B, \Delta}\, \wedge R \qquad \frac{A \,;\, \Gamma \vdash \Delta}{A \wedge B \,;\, \Gamma \vdash \Delta}\, \wedge L \qquad \frac{B \,;\, \Gamma \vdash \Delta}{A \wedge B \,;\, \Gamma \vdash \Delta}\, \wedge L$$

$$\frac{\Pi \,;\, \Gamma \vdash \Delta}{\Pi \,;\, \Gamma \vdash \bot, \Delta}\, \bot R \qquad \frac{}{\bot \,;\, \vdash}\, \bot L \qquad \frac{}{\Pi \,;\, \Gamma \vdash \top, \Delta}\, \top R$$

Table 5: The sequent calculus LKT

The symmetry between sequents in LKQ and LKT gives a duality stressed in [DJS95].

Proposition 4 (Duality between LKQ and LKT)
Through the duality of Section 1.3, LKQ *and* LKT *are dual systems:*

$$\Gamma \vdash_{\mathsf{LKQ}} \Delta \,;\, \Pi \quad \Longleftrightarrow \quad \overline{\Pi} \,;\, \overline{\Delta} \vdash_{\mathsf{LKT}} \overline{\Gamma}$$

Proof: Studied for the connectives $\rightarrow$ and $\forall$ through linear logic in [DJS95]. Easy inductive checking left to the reader. □

From [DJS95], one knows that LKT has the expressive power of LK.

Proposition 5 (Classical provability of LKT)

- $\Pi \,;\, \Gamma \vdash_{\mathsf{LKT}} \Delta \Longrightarrow \Pi, \Gamma \vdash_{\mathsf{LK}} \Delta$
- $\Gamma \vdash_{\mathsf{LK}} \Delta \Longrightarrow \,;\, \Gamma \vdash_{\mathsf{LKT}} \Delta$

We end this section with a translation of the classical system LKT into LD_0.

Proposition 6 (From LKT to LD_0)
If $\Pi \,;\, \Gamma \vdash_{\mathsf{LKT}} \Delta$ *then* $\Pi \vdash_{\mathsf{LD}_0} \neg\Gamma, \Delta$.

PROOF: This can be obtained from Proposition 3 by duality. □

Putting together all the previous results, we have shown that the call-by-value/call-by-name duality of classical logic stressed in [Fil89, Sel01] is in exact correspondence with the left/right duality of intuitionistic/dual-intuitionistic logics (see Figure 1).

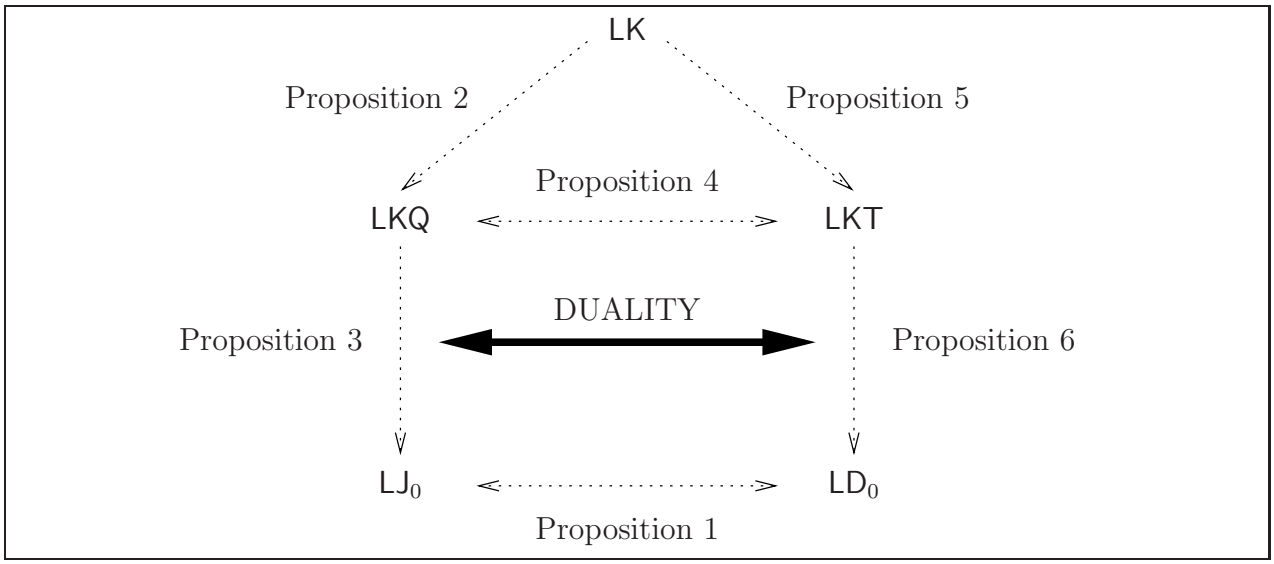


Figure 1: Dualities

The translation of call-by-name classical logic into LJ_0 described in [LRS93] can be decomposed in our setting into the translation from LKT to LD_0 followed by the duality embedding LD_0 into LJ_0.

2.3 Comparing provabilities

Concerning the various systems studied above, it was shown in [Cze77] that there is no loss of provability by moving from LK to LD_0. We can extend this result to a complete comparison of the provabilities of LK, LJ_0 and LD_0.

Proposition 7 (Provabilities)
The following statements are equivalents:

$$\begin{array}{ccccc}
\mathsf{LK} & & \mathsf{LJ}_0 & & \mathsf{LD}_0 \\
\vdash A & & & \iff & \vdash A \\
\iff \quad \neg A \vdash & \iff & \neg A \vdash & \iff & \neg A \vdash \\
\iff \quad \vdash \neg\neg A & \iff & \vdash \neg\neg A & \iff & \vdash \neg\neg A
\end{array}$$

PROOF: We prove enough implications:

- $\vdash_{\mathsf{LK}} A \iff \neg A \vdash_{\mathsf{LK}} \iff \vdash_{\mathsf{LK}} \neg\neg A$.
- $\vdash_{\mathsf{LK}} A \Longrightarrow \neg A \vdash_{\mathsf{LJ}_0}$ by translation through LKQ (Propositions 2 and 3).
- $\neg A \vdash_{\mathsf{LJ}_0} \Longrightarrow \vdash_{\mathsf{LJ}_0} \neg\neg A$.

- $\vdash_{\mathsf{LJ}_0} \neg\neg A \Longrightarrow \vdash_{\mathsf{LK}} \neg\neg A$ by inclusion.
- $\vdash_{\mathsf{LK}} A \Longrightarrow \vdash_{\mathsf{LD}_0} A$ by translation through LKT (Propositions 5 and 6).
- $\vdash_{\mathsf{LD}_0} A \Longrightarrow \neg A \vdash_{\mathsf{LD}_0} \Longrightarrow \vdash_{\mathsf{LD}_0} \neg\neg A$.
- $\vdash_{\mathsf{LD}_0} \neg\neg A \Longrightarrow \vdash_{\mathsf{LK}} \neg\neg A$ by inclusion. □

In particular, $\vdash_{\mathsf{LK}} A \iff \vdash_{\mathsf{LJ}_0} \neg\neg A$ is Glivenko's theorem [Gli29].

3 Extensions and restrictions

We discuss some of the possible variations on the systems of the previous section.

3.1 Implication and difference

The *implication connective* which is often at the heart of logical systems is omitted in the above presentations. We can find in [DJS95] the appropriate rules for LKQ and LKT. They come from considerations related with linear logic. In order to preserve duality, we also add the *subtraction connective* [Cro01].

We extend LKQ:

$$\frac{\Gamma, A \vdash B, \Delta\ ;}{\Gamma \vdash \Delta\ ;\ A \to B} \to R \qquad \frac{\Gamma \vdash \Delta\ ;\ A \qquad \Gamma', B \vdash \Delta'\ ;}{\Gamma, \Gamma', A \to B \vdash \Delta, \Delta'\ ;} \to L$$

$$\frac{\Gamma \vdash \Delta\ ;\ A \qquad \Gamma', B \vdash \Delta'\ ;}{\Gamma, \Gamma' \vdash \Delta, \Delta'\ ;\ A - B} -R \qquad \frac{\Gamma, A \vdash B, \Delta\ ;\ \Pi}{\Gamma, A - B \vdash \Delta\ ;\ \Pi} -L$$

These rules are precisely those which can be derived from the definitions $A \to B \equiv \neg(A \wedge \neg B)$ and $A - B \equiv A \wedge \neg B$.

We also extend LKT:

$$\frac{\Pi\ ;\ \Gamma, A \vdash B, \Delta}{\Pi\ ;\ \Gamma \vdash A \to B, \Delta} \to R \qquad \frac{;\ \Gamma \vdash A, \Delta \qquad B\ ;\ \Gamma' \vdash \Delta'}{A \to B\ ;\ \Gamma, \Gamma' \vdash \Delta, \Delta'} \to L$$

$$\frac{;\ \Gamma \vdash A, \Delta \qquad B\ ;\ \Gamma' \vdash \Delta'}{;\ \Gamma, \Gamma' \vdash A - B, \Delta, \Delta'} -R \qquad \frac{;\ \Gamma, A \vdash B, \Delta}{A - B\ ;\ \Gamma \vdash \Delta} -L$$

These rules are precisely those which can be derived from the definitions $A \to B \equiv \neg A \vee B$ and $A - B \equiv \neg(\neg A \vee B)$.

In order to preserve Proposition 3, LJ_0 is extended with the following new rules:

$$\frac{\Gamma, A, \neg B \vdash}{\Gamma \vdash A \to B} \to R \qquad \frac{\Gamma \vdash A \qquad \Delta, B \vdash}{\Gamma, \Delta, A \to B \vdash} \to L$$

$$\frac{\Gamma \vdash A \qquad \Delta, B \vdash}{\Gamma, \Delta \vdash A - B} -R \qquad \frac{\Gamma, A, \neg B \vdash \Pi}{\Gamma, A - B \vdash \Pi} -L$$

Of course, they also correspond to $A \to B \equiv \neg(A \wedge \neg B)$ and $A - B \equiv A \wedge \neg B$. These rules do not define the usual intuitionistic connectives. In particular $(\to L)$ is a restriction of the usual rule and $(\to R)$ is more general than the usual one. As a consequence, the provable sequents of LJ_0 using

$\to$ are not the usual intuitionistic ones: $\vdash \neg\neg A \to A$ is provable in LJ_0 and $A, A \to B \vdash B$ is *not* provable in general in LJ_0.

In the same spirit, we extend LD_0:

$$\frac{\Pi \vdash \neg A, B, \Gamma}{\Pi \vdash A \to B, \Gamma} \to R \qquad \frac{\vdash A, \Gamma \qquad B \vdash \Delta}{A \to B \vdash \Gamma, \Delta} \to L$$

$$\frac{\vdash A, \Gamma \qquad B \vdash \Delta}{\vdash A - B, \Gamma, \Delta} -R \qquad \frac{\vdash \neg A, B, \Gamma}{A - B \vdash \Gamma} -L$$

This corresponds to $A \to B \equiv \neg A \vee B$ and $A - B \equiv \neg(\neg A \vee B)$.

The duality of Section 1.3 is then extended with:

$$\overline{A \to B} \;=\; \overline{B} - \overline{A} \qquad\qquad \overline{A - B} \;=\; \overline{B} \to \overline{A}$$

All these definitions allow us to preserve Propositions 1, 2, 3, 4, 5, 6 and 7.

Example 2 (Peirce's law)
The classical sequent $(A \to B) \to A \vdash A$ corresponds to Peirce's law. We consider the refinement $(A \to B) \to A \,;\, \vdash A$ in LKT. The dual sequent in LKQ is $\overline{A} \vdash \,;\, \overline{A} - (\overline{B} - \overline{A})$. Using $\overline{A} - (\overline{B} - \overline{A}) \equiv \overline{A} \wedge \neg(\overline{B} \wedge \neg\overline{A}) \equiv \overline{A} \wedge (\overline{B} \to \overline{A})$ in LKQ, it is equivalent to $\overline{A} \vdash \,;\, \overline{A} \wedge (\overline{B} \to \overline{A})$. Associated proofs are:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \,;\, \vdash A}\,ax}{;\, A \vdash A}\,der}{;\, A \vdash B, A}\,wkR}{;\, \vdash A \to B, A} \to R \qquad \dfrac{}{A \,;\, \vdash A}\,ax}{(A \to B) \to A \,;\, \vdash A, A} \to L}{(A \to B) \to A \,;\, \vdash A}\,ctrR$$

$$\dfrac{\dfrac{\dfrac{}{\overline{A} \vdash \,;\, \overline{A}}\,ax \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\overline{A} \vdash \,;\, \overline{A}}\,ax}{\overline{A} \vdash \overline{A} \,;}\,der}{\overline{A}, \overline{B} \vdash \overline{A} \,;}\,wkL}{\overline{A} \vdash \,;\, \overline{B} \to \overline{A}} \to R}{\overline{A}, \overline{A} \vdash \,;\, \overline{A} \wedge (\overline{B} \to \overline{A})} \wedge R}{\overline{A} \vdash \,;\, \overline{A} \wedge (\overline{B} \to \overline{A})}\,ctrL$$

The corresponding translations in LD_0 and LJ_0 are:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax}{\vdash \neg A, A}\,\neg R}{\vdash \neg A, B, A}\,wkR}{\vdash A \to B, A} \to R \qquad \dfrac{}{A \vdash A}\,ax}{(A \to B) \to A \vdash A, A} \to L}{(A \to B) \to A \vdash A}\,ctrR$$

$$\dfrac{\dfrac{\dfrac{}{\overline{A} \vdash \overline{A}}\,ax \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\overline{A} \vdash \overline{A}}\,ax}{\overline{A}, \neg\overline{A} \vdash}\,\neg L}{\overline{A}, \overline{B}, \neg\overline{A} \vdash}\,wkL}{\overline{A} \vdash \overline{B} \to \overline{A}} \to R}{\overline{A}, \overline{A} \vdash \overline{A} \wedge (\overline{B} \to \overline{A})} \wedge R}{\overline{A} \vdash \overline{A} \wedge (\overline{B} \to \overline{A})}\,ctrL$$

3.2 Multiplicative fragments

Following the terminology of linear logic, the connectives $\wedge$ and $\top$ of LJ_0 as well as the connectives $\vee$ and $\bot$ of LD_0 are called *multiplicative*. The connectives $\vee$ and $\bot$ of LJ_0 as well as the connectives $\wedge$ and $\top$ of LD_0 are called *additive*. The multiplicative fragments MLJ_0 and MLD_0 of LJ_0 and LD_0 are obtained by removing the additive connectives.

It is possible to justify that this restriction is not too strong in an important number of cases. Indeed, the formulas $\neg(A \vee B)$ and $\neg A \wedge \neg B$ are equivalent in LJ_0:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax}{A \vdash A \vee B}\vee R}{A, \neg(A \vee B) \vdash}\neg L}{\neg(A \vee B) \vdash \neg A}\neg R \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash B}\,ax}{A \vdash A \vee B}\vee R}{B, \neg(A \vee B) \vdash}\neg L}{\neg(A \vee B) \vdash \neg B}\neg R}{\dfrac{\neg(A \vee B), \neg(A \vee B) \vdash \neg A \wedge \neg B}{\neg(A \vee B) \vdash \neg A \wedge \neg B}\,ctrL}\wedge R$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax}{A, \neg A \vdash}\neg L}{A, \neg A, \neg B \vdash}\,wkL \qquad \dfrac{\dfrac{\dfrac{}{B \vdash B}\,ax}{B, \neg B \vdash}\neg L}{B, \neg A, \neg B \vdash}\,wkL}{A \vee B, \neg A, \neg B \vdash}\vee L}{\dfrac{\neg A, \neg B \vdash \neg(A \vee B)}{\neg A \wedge \neg B \vdash \neg(A \vee B)}\wedge L}\neg R$$

This is not only an equivalence but even a syntactical isomorphism [Lau05]. In the same spirit, $\neg\bot$ and $\top$ are equivalent. This means that formulas of LJ_0 such that all the additive connectives are under the scope of at least one negation can be studied in MLJ_0 without any loss of precision.

Dually, $\neg(A \wedge B)$ and $\neg A \vee \neg B$ and $\neg\top$ and $\bot$ are equivalent in LD_0.

3.3 Reversal

Another way of constraining our systems is to restrict structural rules. It will be used in Sections 4.6 and 5.1.

We consider the following restrictions of the contraction, weakening and right negation rules in LJ_0:

$$\dfrac{\Gamma \vdash \Pi}{\Gamma, X \vdash \Pi}\,wkL \qquad \dfrac{\Gamma \vdash \Pi}{\Gamma, \neg A \vdash \Pi}\,wkL \qquad \dfrac{\Gamma, X, X \vdash \Pi}{\Gamma, X \vdash \Pi}\,ctrL \qquad \dfrac{\Gamma, \neg A, \neg A \vdash \Pi}{\Gamma, \neg A \vdash \Pi}\,ctrL$$

$$\dfrac{\neg\Gamma, \Xi, A \vdash}{\neg\Gamma, \Xi \vdash \neg A}\neg R$$

with Ξ containing only propositional variables. The reason for taking negation into account in this restriction comes once again from linear logic (and is important with respect to cut elimination) and all this is related with the ρ-constraint of [QTdF96, LQTdF05]. This constraint is a logical way of imposing that proofs use the reversibility of the left logical rules (except $\neg L$) of LJ_0: these logical rules are applied as late as possible (from a top down point of view) in the constrained system.

The restricted system is called LJ_0^ρ, and there is no loss of provability through this restriction.

Proposition 8 (Reversal in LJ_0)
$\Gamma \vdash \Pi$ *is provable in* LJ_0 *if and only if it is provable in* LJ_0^ρ.

Proof: We prove by induction on A that any occurrence of a contraction or weakening rule on A with main connective $\wedge$, $\vee$, $\top$ or $\bot$ can be replaced by occurrences of this rule applied to strict sub-formulas of A:

$$\dfrac{\Gamma, A \wedge B, A \wedge B \vdash \Pi}{\Gamma, A \wedge B \vdash \Pi}\,ctrL \qquad \mapsto$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax \quad \dfrac{}{B \vdash B}\,ax}{A, B \vdash A \wedge B}\wedge R \qquad \dfrac{\dfrac{}{A \vdash A}\,ax \quad \dfrac{}{B \vdash B}\,ax}{A, B \vdash A \wedge B}\wedge R}{\Gamma, A \wedge B, A, B \vdash \Pi}\,cut}{\Gamma, A, B, A, B \vdash \Pi}\,cut}{\Gamma, A, B, B \vdash \Pi}\,ctrL}{\Gamma, A, B \vdash \Pi}\,ctrL}{\Gamma, A \wedge B \vdash \Pi}\wedge L$$

$$\frac{\Gamma, A \vee B, A \vee B \vdash \Pi}{\Gamma, A \vee B \vdash \Pi}\, ctrL \qquad \mapsto$$

$$\dfrac{\dfrac{\dfrac{\Gamma, A \vee B, A \vee B \vdash \Pi \qquad \dfrac{\dfrac{}{A \vdash A}\, ax}{A \vdash A \vee B}\, \vee R}{\Gamma, A, A \vee B \vdash \Pi}\, cut \qquad \dfrac{\dfrac{}{A \vdash A}\, ax}{A \vdash A \vee B}\, \vee R}{\Gamma, A, A \vdash \Pi}\, cut}{\Gamma, A \vdash \Pi}\, ctrL \qquad \dfrac{\dfrac{\dfrac{\Gamma, A \vee B, A \vee B \vdash \Pi \qquad \dfrac{\dfrac{}{B \vdash B}\, ax}{B \vdash A \vee B}\, \vee R}{\Gamma, B, A \vee B \vdash \Pi}\, cut \qquad \dfrac{\dfrac{}{B \vdash B}\, ax}{B \vdash A \vee B}\, \vee R}{\Gamma, B, B \vdash \Pi}\, cut}{\Gamma, B \vdash \Pi}\, ctrL}{\Gamma, A \vee B \vdash \Pi}\, \vee L$$

$$\frac{\Gamma, \top, \top \vdash \Pi}{\Gamma, \top \vdash \Pi}\, ctrL \qquad \mapsto \qquad \dfrac{\dfrac{\Gamma, \top, \top \vdash \Pi \qquad \dfrac{}{\vdash \top}\, \top R}{\Gamma, \top \vdash \Pi}\, cut \qquad \dfrac{}{\vdash \top}\, \top R}{\Gamma, \top \vdash \Pi}\, cut}{\Gamma \vdash \Pi}\, \top L$$

$$\frac{\Gamma, \bot, \bot \vdash \Pi}{\Gamma, \bot \vdash \Pi}\, ctrL \qquad \mapsto \qquad \frac{}{\Gamma, \bot \vdash \Pi}\, \bot L$$

$$\frac{\Gamma \vdash \Pi}{\Gamma, A \wedge B \vdash \Pi}\, wkL \qquad \mapsto \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma, A \vdash \Pi}\, wkL}{\Gamma, A, B \vdash \Pi}\, wkL}{\Gamma, A \wedge B \vdash \Pi}\, \wedge L$$

$$\frac{\Gamma \vdash \Pi}{\Gamma, A \vee B \vdash \Pi}\, wkL \qquad \mapsto \qquad \dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma, A \vdash \Pi}\, wkL \qquad \dfrac{\Gamma \vdash \Pi}{\Gamma, B \vdash \Pi}\, wkL}{\Gamma, A \vee B \vdash \Pi}\, \vee L$$

$$\frac{\Gamma \vdash \Pi}{\Gamma, \top \vdash \Pi}\, wkL \qquad \mapsto \qquad \frac{\Gamma \vdash \Pi}{\Gamma, \top \vdash \Pi}\, \top L$$

$$\frac{\Gamma \vdash \Pi}{\Gamma, \bot \vdash \Pi}\, wkL \qquad \mapsto \qquad \frac{}{\Gamma, \bot \vdash \Pi}\, \bot L$$

The right negation rule is treated in a similar way. □

In a symmetric way, we can restrict contraction, weakening and left negation rules in LD_0 to define LD_0^ρ:

$$\frac{\Pi \vdash \Gamma}{\Pi \vdash \Gamma, X}\, wkR \qquad \frac{\Pi \vdash \Gamma}{\Pi \vdash \Gamma, \neg A}\, wkR \qquad \frac{\Pi \vdash \Gamma, X, X}{\Pi \vdash \Gamma, X}\, ctrR \qquad \frac{\Pi \vdash \Gamma, \neg A, \neg A}{\Pi \vdash \Gamma, \neg A}\, ctrR$$

$$\frac{\vdash \neg\Gamma, \Xi, A}{\neg A \vdash \neg\Gamma, \Xi}\, \neg L$$

with Ξ containing only propositional variables.

Again, there is no loss of provability.

Proposition 9 (Reversal in LD_0)
$\Pi \vdash \Gamma$ *is provable in* LD_0 *if and only if it is provable in* LD_0^ρ.

Proof: By duality, from Proposition 8. □

We define the systems MLJ_0^ρ and MLD_0^ρ as the multiplicative restrictions of LJ_0^ρ and LD_0^ρ, or equivalently as the reversed versions of MLJ_0 and MLD_0.

4 Nets

We define a graphical syntax incarnating the duality between LJ_0 and LD_0 without making a specific choice between them. The notion of *id-nets* we introduce is strongly inspired from polarized proof-nets [Lau03, LTdF04].

We choose a presentation of id-nets in the spirit of Lafont's interaction nets [Laf90]. However id-nets are not true interaction nets (at least cells with more than one principal port are required).

It is well known, in the theory of proof-nets for linear logic, that dealing with the multiplicative connectives only is much easier than the introduction of the additive connectives which induce more global behaviours. This is why we concentrate on the multiplicative fragments of LJ_0 and LD_0. Moreover we have explained in Section 3.2 how the restriction to multiplicative connectives is often harmless. A possible treatment of the additive connectives will be shortly addressed in Section 5.1.

4.1 Definitions

Since we want our nets to be fair with both LJ_0 and LD_0, we use a specific grammar for formulas appearing on edges of nets. We also prefer the terminology *type* rather than formula to make clear the distinction.

Types are built with the connectives $\circ$, $\mathtt{I}$ and $\neg$:

$$A \quad ::= \quad X \quad | \quad \neg A \quad | \quad A \circ A \quad | \quad \mathtt{I}$$

A *partial directed graph* is a directed graph in which some edges can be without source or without target (or without source nor target). A *root* in a partial directed graph is an edge without source. A *leaf* is an edge without target.

In order to build our *id-nets* which are partial directed graphs with edges labelled with types, we will rely on a given set of *kinds of nodes* with a given label which specifies: the number of incoming edges, the number of outgoing edges, and the constraints relating the types labelling these edges.

There are seven *basic* kinds of nodes:

- $\circ$-up: one incoming edge and two outgoing edges, the type of the incoming edge is $A \circ B$ if A is the type of the first outgoing edge and B is the type of the second one.
- $\mathtt{I}$-up: one incoming edge of type $\mathtt{I}$ and no outgoing edge.
- $\circ$-dn: two incoming edges and one outgoing edge, the type of the outgoing edge is $A \circ B$ if A is the type of the first incoming edge and B is the type of the second one.
- $\mathtt{I}$-dn: no incoming edge and one outgoing edge of type $\mathtt{I}$.
- $\mathtt{C}$-dn: two incoming edges and one outgoing edge, the three edges have the same type.
- $\mathtt{W}$-dn: no incoming edge and one outgoing edge, with no particular constraint on the type of the edge.
- $\neg$-dn: no incoming edge and two outgoing edges, the type of the first outgoing edge is A if the type of the second one is $\neg A$.

The graphical representations of these basic kinds of nodes are given in Figure 2.

Definition 1 (Id-nets)
Id-nets and kinds of nodes are defined in a mutually recursive way.

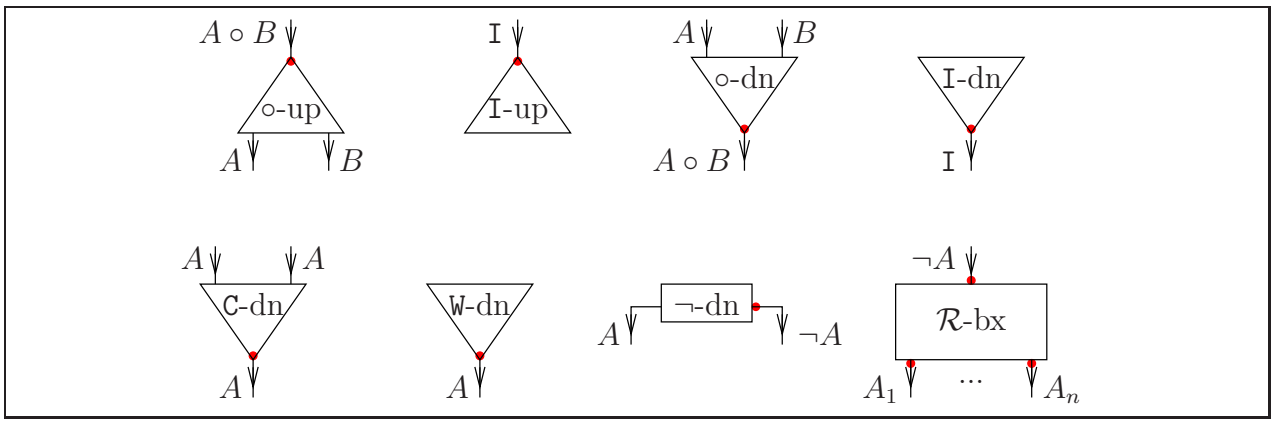


Figure 2: Kinds of nodes (with active ports)

A partial directed acyclic graph, built with basic kinds of nodes and with edges labelled with types respecting the constraints associated with the basic kinds of nodes, and with exactly one root or one node of kind ¬-dn, is an id-net with *depth* 0.

If $\mathcal{R}$ is an id-net with leaves typed with A_1, ..., A_n, A and no root, $\mathcal{R}$-bx is a kind of node (called a *box kind*) with one incoming edge with type $\neg A$ and n outgoing edges with types A_1, ..., A_n (see Figure 2).

A partial directed acyclic graph $\mathcal{R}$, built with kinds of nodes (either basic or box) and with edges labelled with types respecting the constraints associated with the kinds of nodes, and with exactly one root or one node of kind ¬-dn, is an id-net with depth $d + 1$ (where d is the maximal depth of the id-nets $\mathcal{R}_b$ such that $\mathcal{R}$ contains an $\mathcal{R}_b$-bx node).

Proof-nets are often defined in two steps: first a notion of proof-structure and then a notion of correctness criterion selecting valid proof-structures [Gir87]. Since we have no use of proof-structures here, we have done everything in one step. The underlying notion of correctness criterion [Lau03] can be found in the acyclicity requirement and in the constraint on the number of roots and ¬-dn nodes.

We use the notation *-dn for any of $\circ$-dn, I-dn, C-dn or W-dn. We use the notation *-up for any of $\circ$-up, I-up or $\mathcal{R}$-bx.

We define the notion of *active* source or target of edges by looking at the kind of the associated node:

- The target of the incoming edge of a *-up node is active.
- The source of the outgoing edge of a *-dn node is active.
- The source of the second outgoing edge of a ¬-dn node is active.
- The sources of the outgoing edges of an $\mathcal{R}$-bx node are active.
- The other sources or targets are passive.

An edge is a *cut edge* if both its source and its target are active (in particular both the source and the target must exist). An edge is an *axiom edge* if both its source and its target are passive (they may not exist).

An example of id-net is given in Figure 3.

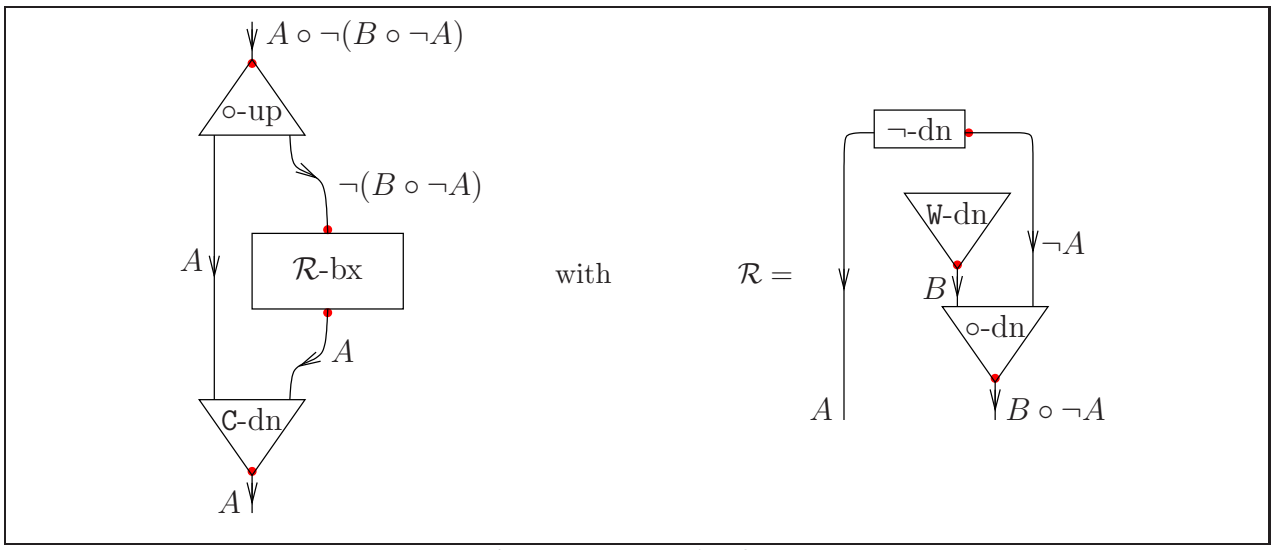


Figure 3: Peirce's id-net

4.2 Translations

We define translations of MLJ_0 and MLD_0 into id-nets allowing us to show how id-nets can represent in one setting both sequent calculi.

4.2.1 MLJ_0

A proof in MLJ_0 of the sequent $\Gamma \vdash \Pi$ is translated as an id-net with root corresponding to Π (no root if Π is empty) and leaves corresponding to Γ. This is done by induction on the proof:

- An *ax*-rule introducing $A \vdash A$ is translated as a single (axiom) edge (without source nor target) typed with A.
- If $\mathcal{R}_1$ is the translation of a proof π_1 of $\Gamma \vdash A$ and $\mathcal{R}_2$ is the translation of a proof π_2 of $A, \Delta \vdash \Pi$, the proof obtained from π_1 and π_2 by adding a *cut*-rule is translated as the id-net obtained by identifying the root (typed A) in $\mathcal{R}_1$ with the leaf typed A in $\mathcal{R}_2$.
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma \vdash \Pi$, the proof obtained from π by adding a weakening rule is translated by adding a W-dn node to $\mathcal{R}$.
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma, A, A \vdash \Pi$, the proof obtained from π by adding a contraction rule is translated by adding a C-dn node to $\mathcal{R}$ with incoming edges the two leaves of $\mathcal{R}$ corresponding to A and A.
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma, A \vdash$, the proof obtained from π by adding a right negation rule is translated as the id-net reduced to an $\mathcal{R}$-bx node.
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma \vdash A$, the proof obtained from π by adding a left negation rule is translated by adding a ¬-dn node to $\mathcal{R}$ with first outgoing edge the root of $\mathcal{R}$ and second outgoing edge a new leaf of the whole id-net.
- If $\mathcal{R}_1$ is the translation of a proof π_1 of $\Gamma \vdash A$ and $\mathcal{R}_2$ is the translation of a proof π_2 of $\Delta \vdash B$, the proof obtained from π_1 and π_2 by adding a right conjunction rule is translated as the id-net obtained by adding a $\circ$-up node with outgoing edges the roots of $\mathcal{R}_1$ and $\mathcal{R}_2$.

- If $\mathcal{R}$ is the translation of a proof π of $\Gamma, A, B \vdash \Pi$, the proof obtained from π by adding a left conjunction rule is translated by adding a $\circ$-dn node to $\mathcal{R}$ with incoming edges the two leaves of $\mathcal{R}$ corresponding to A and B.
- A right $\top$ rule is translated as an id-net reduced to a `I`-up node.
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma \vdash \Pi$, the proof obtained from π by adding a left $\top$ rule is translated by adding a `I`-dn node to $\mathcal{R}$.

4.2.2 MLD_0

A proof in MLD_0 of the sequent $\Pi \vdash \Gamma$ is translated as an id-net with root corresponding to Π (no root if Π is empty) and leaves corresponding to Γ. This is done by induction on the proof in a very similar way as for MLJ_0. We only give the key ingredients:

- An *ax*-rule introducing $A \vdash A$ is translated as a single (axiom) edge (without source nor target) typed with A.
- If $\mathcal{R}_1$ is the translation of a proof π_1 of $\Pi \vdash \Gamma, A$ and $\mathcal{R}_2$ is the translation of a proof π_2 of $A \vdash \Delta$, the proof obtained from π_1 and π_2 by adding a *cut*-rule is translated as the id-net obtained by identifying the leaf typed A in $\mathcal{R}_1$ with the root (typed A) in $\mathcal{R}_2$.
- The weakening rule is translated by using a `W`-dn node.
- The contraction rule is translated by using a `C`-dn node.
- The right negation rule is translated by using a $\neg$-dn node.
- The left negation rule is translated by using an $\mathcal{R}$-bx node.
- The right disjunction rule is translated by using a $\circ$-dn node.
- The left disjunction rule is translated by using a $\circ$-up node.
- The right $\bot$ rule is translated by using a `I`-dn node.
- The left $\bot$ rule is translated by using a `I`-up node.

4.2.3 Duality

A crucial intrinsic property of the two translations above is that two dual proofs in MLJ_0 and MLD_0 are translated as the same id-net.

Example 3 (Peirce's id-net)
For the following two dual proofs of in LJ_0 and LD_0 (see Example 2 for comments about these proofs):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax}{A, \neg A \vdash}\,\neg L}{A, B, \neg A \vdash}\,wkL}{A, B \wedge \neg A \vdash}\,\wedge L \qquad \dfrac{\dfrac{}{A \vdash A}\,ax \qquad \dfrac{A \vdash \neg(B \wedge \neg A)}{}\,\neg R}{A, A \vdash A \wedge \neg(B \wedge \neg A)}\,\wedge R}{A \vdash A \wedge \neg(B \wedge \neg A)}\,ctrL$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,ax}{\vdash \neg A, A}\,\neg R}{\vdash B, \neg A, A}\,wkR}{\vdash B \vee \neg A, A}\,\vee R \qquad \dfrac{\dfrac{}{A \vdash A}\,ax \qquad \dfrac{\neg(B \vee \neg A) \vdash A}{}\,\neg L}{A \vee \neg(B \vee \neg A) \vdash A, A}\,\vee L}{A \vee \neg(B \vee \neg A) \vdash A}\,ctrR$$

the common associated id-net is given in Figure 3.

4.3 Sequentializations

Starting from a given id-net, it is possible to recover both an MLJ_0 proof and an MLD_0 proof. Moreover they are dual to each other.

4.3.1 MLJ_0

We first introduce a specific kind of id-net: *up trees*. They are defined inductively: an edge is an up tree, a `I`-up node with its incoming edge is an up tree, an $\mathcal{R}$-bx node with its incoming and outgoing edges is an up tree, and one gets an up tree by branching the roots of two up trees on the outgoing edges of a $\circ$-up node. An up tree is always an id-net.

In an id-net, an edge is always the root of an up tree (the up tree reduced to the edge). We call *the* up tree of an edge, the maximal up tree (with respect to inclusion) having this edge as root.

Lemma 1
*A directed path, starting from an edge whose target is a *-dn node and going to a leaf of the id-net, either crosses a cut or is such that the source of its last edge is a *-dn node.*

We now extract a proof of $\Gamma \vdash \Pi$ in MLJ_0 from an id-net $\mathcal{R}$ with root Π and leaves Γ. This is done by induction on the size of $\mathcal{R}$. The *size* of an id-net is its number of nodes at all depths (*i.e.* the number of nodes of the graph — at depth 0 — plus the sizes of the id-nets of the box nodes).

- If there is a leaf of $\mathcal{R}$ with source a $\circ$-dn, `I`-dn, `C`-dn or `W`-dn, we remove this node, we still have an id-net.

  On the MLJ_0 side, it corresponds to a left conjunction rule, a left $\top$ rule, a contraction rule or a weakening rule.

- Otherwise, if there is a cut edge, we consider a maximal one in the id-net (*i.e.* such that there is no path in the directed graph going from this cut edge to another one). We split this edge into a new root and a new leaf. We show that we obtain two components (which are both id-nets). We look at the leaves of the up tree of the cut edge. They cannot have a *-dn node as target otherwise, by Lemma 1, we would contradict the absence of leaf with source a *-dn node or the maximality of the cut edge. This means that the nodes and edges reachable from the target of the cut edge exactly constitute an up tree not connected to the rest of the id-net.

  By induction hypothesis, we have proofs in MLJ_0 corresponding to the two id-nets. We just have to add a cut rule between them.

- Otherwise, if there is a $\neg$-dn node, its second outgoing edge is a leaf: its target cannot be a *-up node otherwise we have a cut edge, and it cannot be a *-dn node by Lemma 1 as before. We remove the $\neg$-dn node and we still have an id-net.

  On the MLJ_0 side, it corresponds to a left negation rule.

- Otherwise, the id-net is an up tree: the id-net has a root (since it does not contain $\neg$-dn nodes) and we consider the up tree of the root. The leaves of this up tree are leaves of the id-net otherwise we could apply Lemma 1 again. The id-net is reduced to this up tree otherwise we have a node not reachable from the root thus reachable from a `I`-dn or `W`-dn node. This is impossible without cut edges nor *-dn nodes as sources of leaves, by Lemma 1. We now look at the up tree. If the up tree is reduced to an edge, it is the translation of an *ax*-rule. If the up

tree is reduced to a I-up node, it is the translation of a right $\top$ rule. If the up tree is reduced to an $\mathcal{R}$-bx node, by induction hypothesis, we can build a proof of MLJ_0 corresponding to $\mathcal{R}$. The whole id-net is the translation of the proof obtained by adding a right negation rule to the proof associated with $\mathcal{R}$. If the root of the up tree has target a $\circ$-up node, we remove it, we obtain two up trees. By induction hypothesis, these two up trees correspond to proofs in MLJ_0, to which we just have to add a right conjunction rule.

4.3.2 MLD_0

We can also extract a proof of $\Pi \vdash \Gamma$ in MLD_0 from an id-net $\mathcal{R}$ with root Π and leaves Γ. The procedure follows the same decomposition of the id-net as for MLJ_0:

- If there is a leaf with source a $\circ$-dn, I-dn, C-dn or W-dn, we get a right disjunction rule, a right $\bot$ rule, a contraction rule or a weakening rule in MLD_0.
- If there is a cut edge, we get a cut rule in MLD_0.
- If there is a $\neg$-dn node, we get a right negation rule in MLD_0.
- If the id-net is reduced to an edge, we get an *ax*-rule in MLD_0.
- If the id-net is reduced to a I-up node, we get a left $\bot$ rule in MLD_0.
- If the id-net is reduced to an $\mathcal{R}$-bx node, we get a left negation rule in MLD_0.
- If the id-net is an up tree with root having a $\circ$-up node as target, we get a left disjunction rule in MLD_0.

By applying sequentializations towards MLJ_0 and towards MLD_0 from the same id-net, and by choosing nodes in the same order, one gets dual proofs.

4.4 Cut elimination

A crucial property of sequent calculi is cut elimination. A major advantage of the proof-net approach to proof systems is to provide simplified cut elimination procedures where commutative steps almost disappear.

Given any cut edge in an id-net, we define a way to rewrite it by a cut elimination step. These steps are described in Figure 4.

By using the proofs of [Lau03], one can show that cut elimination is *confluent* and *strongly normalizing* in id-nets.

4.5 Axiom expansion

Another nice transformation of id-nets is *axiom expansion* which turns an id-net into one with all axiom edges typed with propositional variables.

The corresponding rewriting steps are presented in Figure 5.

It is very easy to see, by induction on the types, that axiom expansion applied to axiom edges always terminates and never introduces cut edges since, by definition of an axiom edge, its source and target are not active.

On the sequent calculus side, this corresponds to being able to restrict *ax*-rules to the introduction of $X \vdash X$.

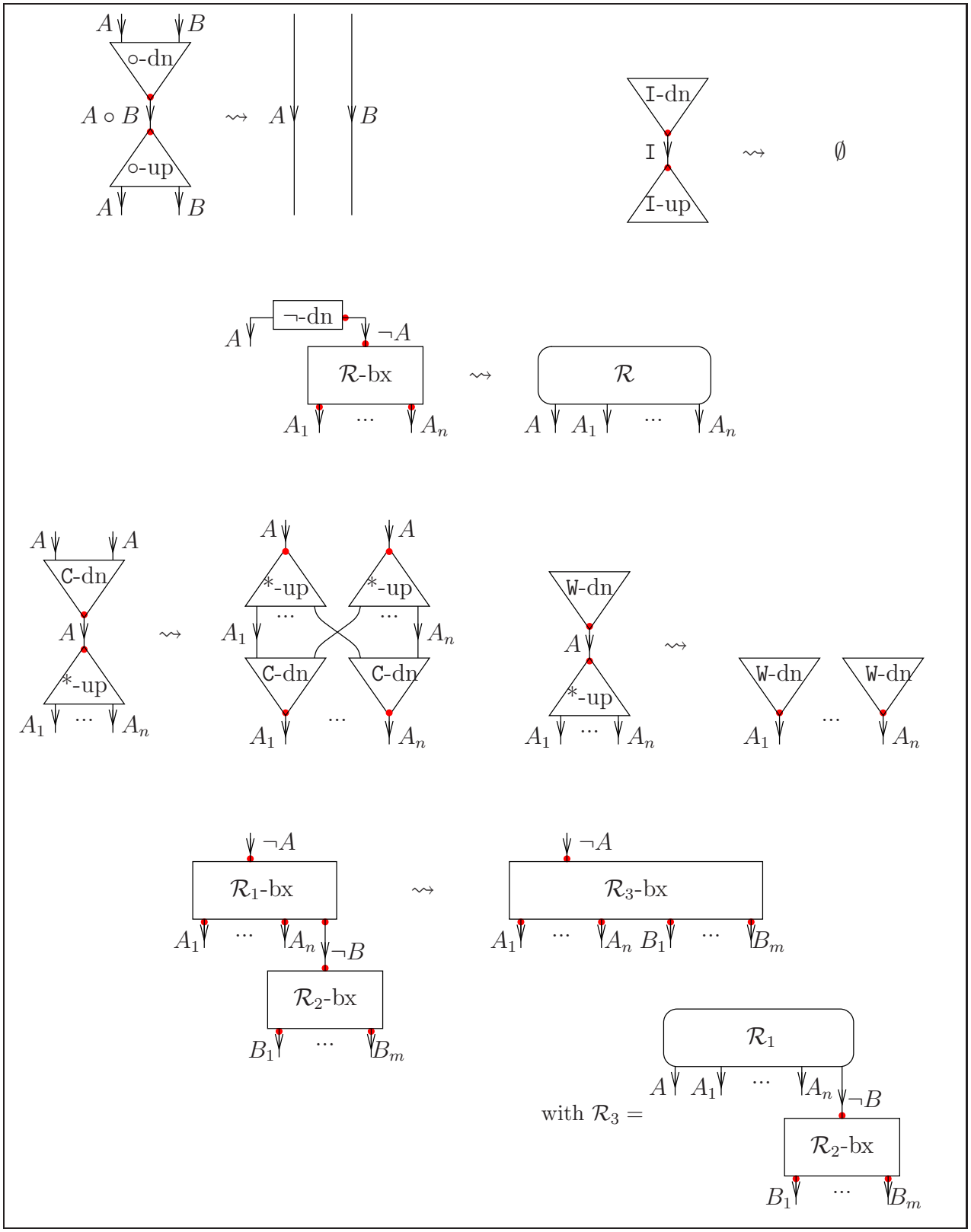


Figure 4: Cut elimination steps

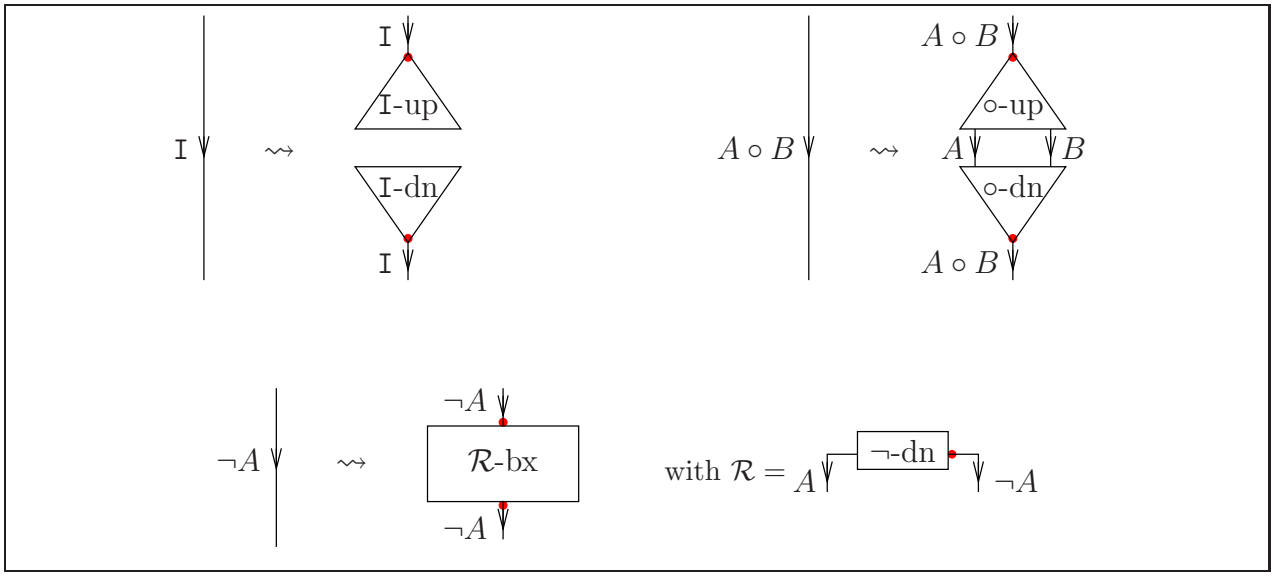


Figure 5: Axiom expansion steps

4.6 Embedding in essential nets

In the literature on proof-nets, one finds two main correctness criteria based on directed graphs. The first one has been introduced by Lamarche [Lam94] for intuitionistic linear logic. The second one is for polarized linear logic [Lau99] and is the one we are relying on for id-nets.

We prove here how these two criteria can be related[1] through a translation of id-nets (using the polarized directed criterion) into essential nets [Lam94] (using the intuitionistic directed criterion).

We use a slight modification of essential nets using explicit boxes. This has no impact on the heart of the correctness criteria: they mainly deal with the multiplicative structure.

The formulas used for typing essential structures are formulas of unit-free intuitionistic multiplicative exponential linear logic. They are given by two dual classes: *output formulas* and *input formulas*.

$$\begin{array}{rcccccccc} O & ::= & X & | & O \otimes O & | & I \parr O & | & !O \\ I & ::= & X^\perp & | & I \parr I & | & O \otimes I & | & ?I \end{array}$$

Essential structures are partial directed graphs built from the nodes of Figure 6. An essential structure is required to have no root and exactly one of its leaves which is of output type. Moreover each $?w$-node n is the target of an additional untyped edge called a *jump* coming from a node which is not reachable by a directed path starting from n. Finally, as for id-nets, with each !-node with outgoing edges typed $?\Gamma$ and $!O$ is associated an essential structure with leaves typed $?\Gamma$ and O.

Based on the output/input polarities of formulas, we define a new orientation on the edges of essential structures by reversing the orientation of edges typed with output formulas. The new orientation is called the *polarized orientation*.

Trails of an essential structure $\mathcal{S}$ are paths starting from the output leaf of $\mathcal{S}$ in the partial directed graph obtained from $\mathcal{S}$ by: deleting edges with target an output $\parr$-node (*i.e.* a $\parr$-node with outgoing edge of output type) — the source of such a deleted edge is called a *sink* [MO00];

[1] Thanks to F. Lamarche who asked me if such a relation exists in 1998 and to L. Strassburger who asked it again ten years later in 2008 at LMRC-08. I am now able to answer them positively.

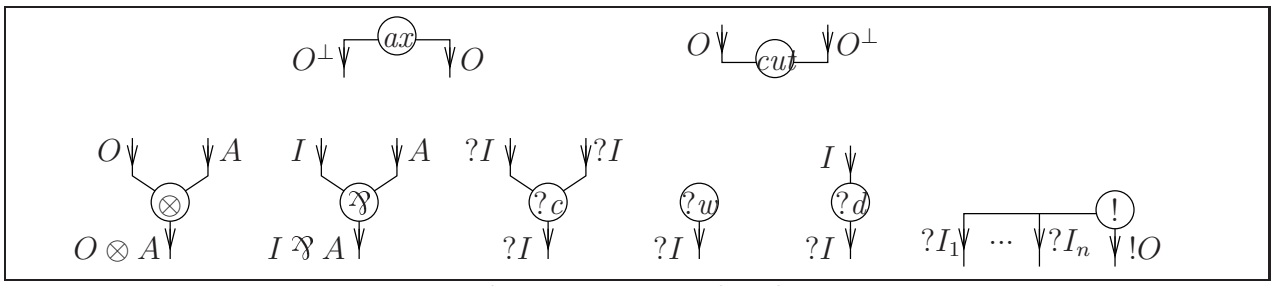


Figure 6: Essential nodes

by considering jumps as normal edges (directed towards the $?w$-node); and by using the polarized orientation for the other (typed) edges.

We now give the correctness criterion of essential nets [Lam94, MO00] which selects "valid" essential structures.

Definition 2 (Essential net)
An essential structure is an *essential net* if:

- The set of its trails is finite.
- Each node belongs to a trail.
- Each trail reaching a sink goes (previously) through the target of its outgoing edge.
- The previous three conditions are recursively satisfied in the essential structures associated with the !-nodes.

In order to define a translation of id-nets into essential nets, we consider id-nets corresponding to the multiplicative reversed systems MLJ_0^ρ and MLD_0^ρ without unit ($\top$ or $\bot$).

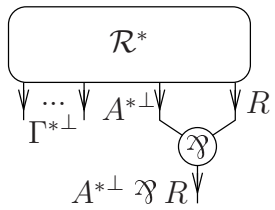
We choose a propositional variable R of intuitionistic linear logic and we translate types by output formulas:

$$X^* = !X \qquad (A \circ B)^* = A^* \otimes B^* \qquad (\neg A)^* = !({A^*}^\perp \parr R)$$

An id-net with root of type A and leaves of types Γ is translated as an essential structure with leaves typed ${\Gamma^*}^\perp, A^*$. An id-net without root and with leaves of types Γ is translated as an essential structure with leaves typed ${\Gamma^*}^\perp, R$.

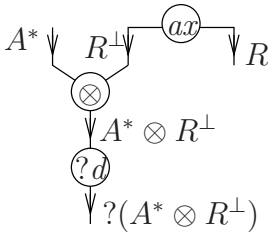
Starting from an id-net, we work by induction on its depth (see Definition 1):

- We translate each $\circ$-up node as a $\otimes$-node, and we reverse the orientation of its incoming and outgoing edges.
- We translate each $\circ$-dn node as a $\parr$-node, each `C`-dn node as a $?c$-node and each `W`-dn node as a $?w$-node, without changing the orientation of edges.
- Given an $\mathcal{R}$-bx node, by induction hypothesis, $\mathcal{R}$ is translated as an essential structure $\mathcal{R}^*$ with leaves typed ${\Gamma^*}^\perp, {A^*}^\perp, R$ if $\mathcal{R}$ has leaves typed Γ, A. The essential structure $\mathcal{S}$ is obtained from $\mathcal{R}^*$ by adding a $\parr$-node to it:



We translate the $\mathcal{R}$-bx node as a !-node with outgoing edges typed $\Gamma^{*\perp}, !(A^{*\perp} \parr R)$, and $\mathcal{S}$ is the essential structure associated with it. This entails that we reverse the orientation of the incoming edge of the $\mathcal{R}$-bx node.

- We translate each $\neg$-dn node as the following graph:



thus we reverse the first outgoing edge, which becomes an incoming edge of the $\otimes$-node.

- We translate each cut edge as a *cut*-node with two incoming edges with sources the source and the target of the original edge.
- We translate each axiom edge as an *ax*-node with two outgoing edges with targets the source and the target of the original edge.

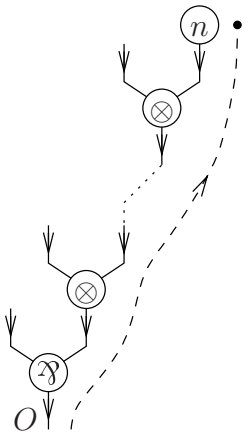
Since some edges are reversed and some are not, it may seem contradictory. We have to check that the orientation of edges in the translation is properly defined. Edges with an orientation which could be both reversed and preserved are:

- edges which have a $\circ$-up node or a $\neg$-dn node (first outgoing edge of the node) as source and a *-dn node as target, and these are axiom edges, which are in fact translated as a pair of edges having the same *ax*-node as source (thus no contradiction);
- edges which have a *-dn node or a $\neg$-dn node (second outgoing edge of the node) as source and a *-up node as target, and these are cut edges, which are in fact translated as a pair of edges having the same *cut*-node as target (thus no contradiction).

The obtained partial graph has no root, the only possibilities would be a leaf of the id-net which has a $\circ$-up node as source or which is the first outgoing edge of a $\neg$-dn node, or a root of the id-net which has a *-dn node as target, but all these edges are axiom edges and thus do not generate roots in the obtained partial graph.

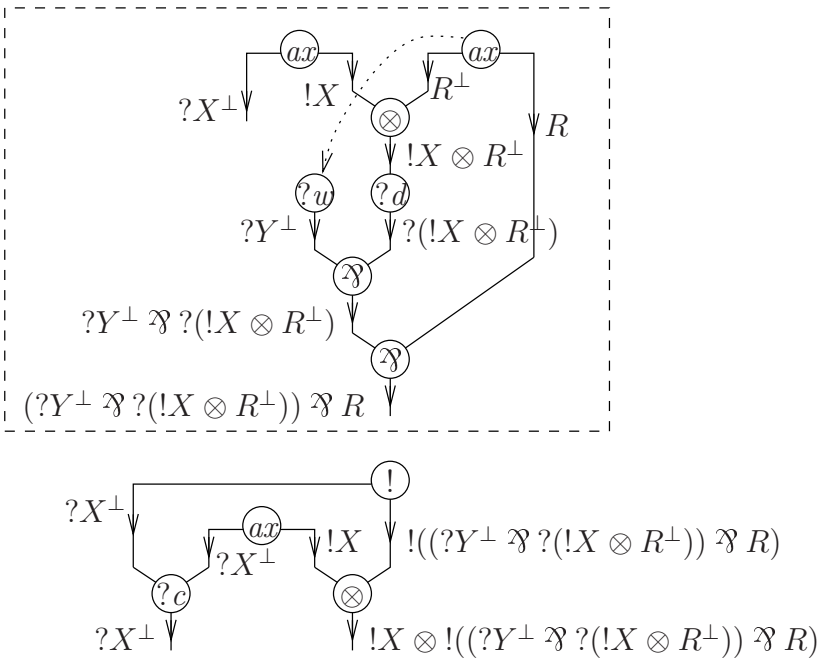
Concerning the leaves of the obtained partial graph: they correspond to the roots and leaves of the id-net plus one leaf of type R for each $\neg$-dn node of the id-net. In particular, there is exactly one leaf of output type.

Finally, let n be the target of the last edge of the maximal path obtained by starting from the output leaf and by always going up through the second incoming edge of the reached node (it stops when reaching a node with less than two incoming edges):



The node n is an *ax*-node or a !-node. Jumps are put from n to all the $?w$-nodes of the essential structure.

Example 4 (Peirce's essential net)
The essential net corresponding to the id-net of Figure 3 is:



where the essential net associated with the !-node is represented above it in the dashed box.

Proposition 10 (Correctness)
The translation of an id-net is an essential net.

Proof: The key remark is that a trail of the essential structure $\mathcal{R}^*$ associated with the id-net $\mathcal{R}$ is exactly a path in the directed partial graph $\mathcal{G}$ obtained from $\mathcal{R}$ by adding edges corresponding to jumps. All these jump edges are starting from the same node n, where n is obtained from the root (or from the first output edge of the ¬-dn node) by always going to the second

outgoing edge of the reached $\circ$-up node and by stopping when reaching a node which is not a $\circ$-up node (or a leaf, and then n is the source of this leaf). Indeed the polarized orientation on $\mathcal{R}^*$ corresponds to the orientation of edges in $\mathcal{R}$.

We work by induction on the depth of the id-net.

The set of trails is infinite if and only if at least one of them is cyclic. Jumps have no impact on cycles in $\mathcal{G}$. We immediately obtain the acyclicity of trails of $\mathcal{R}^*$ from the acyclicity of $\mathcal{R}$.

Nodes of the id-net without incoming edge are $\neg$-dn nodes and `W`-dn nodes. If $\mathcal{R}$ has a root it is translated as the output leaf of $\mathcal{R}^*$. If $\mathcal{R}$ has a $\neg$-dn node, all the nodes coming from the translation of this node belong to trails. Next, the node n which is source of the jumps belongs to a trail. From the fact that, in $\mathcal{G}$, all the nodes without incoming edge (and the unique node with a root as source if it exists) are reachable by a path corresponding to a trail, we easily conclude that all the nodes are.

In $\mathcal{R}^*$, there is no sink!

Finally, we consider an essential structure $\mathcal{S}$ associated with a !-node. Let $\mathcal{S}_0$ be the essential structure obtained from $\mathcal{S}$ by removing the output $⅋$-node above the output leaf, $\mathcal{S}_0$ is the translation of an id-net thus an essential net, by induction hypothesis. The trails of $\mathcal{S}$ are obtained from the trails of $\mathcal{S}_0$ by prefixing all of them with the additional $⅋$-node. Since the only sink in $\mathcal{S}$ is the source of the input incoming edge of the $⅋$-node, it is easy to check that $\mathcal{S}$ is an essential net. □

It is easy to see that replacing each !-node with its associated essential net followed by appropriate unary !-nodes and ¡-nodes gives back the original syntax of essential nets in [Lam94].

To conclude, from the point of view developed in this section, Proposition 10 shows how the correctness criterion of polarized proof-nets can be considered as a particular case of the correctness criterion of essential nets.

5 Additional directions

We end this paper with the short presentation of how it would be possible to extend the work on id-nets to the additive connectives and of how to go beyond propositional logic.

5.1 Nets with additive connectives

Following the work in [LTdF04], we define *slices* as id-nets built with four new basic kinds of nodes:

- $\Box_1$-up: one incoming edge and one outgoing edge, the type of the incoming edge is $A \Box B$ if A is the type of the outgoing edge.
- $\Box_2$-up: one incoming edge and one outgoing edge, the type of the incoming edge is $A \Box B$ if B is the type of the outgoing edge.
- $\Box_1$-dn: one incoming edge and one outgoing edge, the type of the outgoing edge is $A \Box B$ if A is the type of the incoming edge.
- $\Box_2$-dn: one incoming edge and one outgoing edge, the type of the outgoing edge is $A \Box B$ if B is the type of the incoming edge.

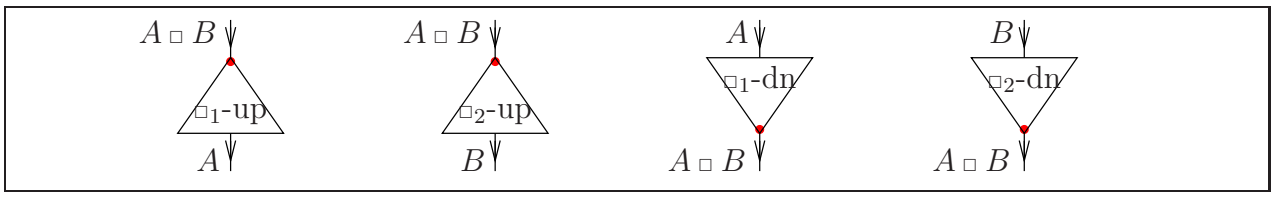


Figure 7: Additive kinds of nodes (with active ports)

The graphical representations of these new basic kinds of nodes are given in Figure 7.

Types are extended with two new connectives:

$$A \quad ::= \quad \ldots \quad | \quad A \,\square\, A \quad | \quad \mathtt{J}$$

An *additive id-net* is a set of slices with root and leaves with the same types satisfying two specific properties:

- enough slices: mainly at least one slice for each boolean valuation with domain the set of occurrences of $\square$ in the types of the leaves (but also taking into account the presence of $\mathtt{J}$, dependencies between occurrences and scopes of $\neg$), see [LTdF04, Lau02];
- not too many slices: no more than one slice for each valuation.

The disjunction of LJ_0 is translated as $\square$ and $\bot$ as $\mathtt{J}$. The additive rules of LJ_0 are translated in the following way:

- If $\mathcal{R}$ is the translation of a proof π of $\Gamma \vdash A$, the proof obtained from π by adding the first right disjunction rule is translated by adding a $\square_1$-up node to each element of $\mathcal{R}$ with outgoing edge the root (corresponding to A).
- If $\mathcal{R}$ is the translation of a proof π of $\Gamma \vdash B$, the proof obtained from π by adding the second right disjunction rule is translated by adding a $\square_2$-up node to each element of $\mathcal{R}$ with outgoing edge the root (corresponding to B).
- If $\mathcal{R}_1$ is the translation of a proof π_1 of $\Gamma, A \vdash \Pi$ and $\mathcal{R}_2$ is the translation of a proof π_2 of $\Gamma, B \vdash \Pi$, the proof obtained from π_1 and π_2 by adding a left disjunction rule is translated as the additive id-net obtained by adding a $\square_1$-dn node to each element of $\mathcal{R}_1$ with incoming edge the leaf of type A and by adding the a $\square_2$-dn node to each element of $\mathcal{R}_2$ with incoming edge the leaf of type B, and by finally taking the union of the two obtained sets of slices.
- A left $\bot$ rule is translated as the empty set of slices.

We work in a dual way with LD_0 ($\wedge$ translated as $\square$ and $\top$ as $\mathtt{J}$):

- The first left conjunction rule is translated by using $\square_1$-up nodes.
- The second left conjunction rule is translated by using $\square_2$-up nodes.
- The right conjunction rule is translated by using $\square_1$-dn and $\square_2$-dn nodes, and by taking the union of the two sets of slices.
- A right $\top$ rule is translated as the empty set of slices.

This approach allows us to define id-nets for the cut-free sub-systems of the reversed restrictions LJ_0^ρ and LD_0^ρ (see Section 3.3) with expanded axioms. The appropriate way of re-introducing cuts is to start from two cut-free id-nets, to add a cut between them and to apply the required cut elimination steps until reaching a cut-free id-net. It is not guaranteed that intermediate steps are valid nets. However this describes a big-step cut elimination procedure for id-nets with additive connectives (see [LTdF04] for more details).

5.2 Quantifiers

We have only worked with propositional systems, but there is no particular problem in adding (first or second order) quantifiers. We just have to do it properly, as for implication and difference (see Section 3.1).

We extend the rules of LJ_0 with:

$$\frac{\Gamma, \neg A \vdash}{\Gamma \vdash \forall\alpha.A}\,\forall R \quad \alpha \notin \Gamma \qquad \frac{\Gamma, A[^{\Phi}/_{\alpha}] \vdash}{\Gamma, \forall\alpha.A \vdash}\,\forall L \qquad \frac{\Gamma \vdash A[^{\Phi}/_{\alpha}]}{\Gamma \vdash \exists\alpha.A}\,\exists R \qquad \frac{\Gamma, A \vdash \Pi}{\Gamma, \exists\alpha.A \vdash \Pi}\,\exists L \quad \alpha \notin \Gamma, \Pi$$

While the rules for $\exists$ are the natural ones, the rules for $\forall$ correspond to the encoding $\forall\alpha.A \equiv \neg\exists\alpha\neg A$.

We extend the rules of LD_0 with:

$$\frac{\Pi \vdash \Gamma, A}{\Pi \vdash \Gamma, \forall\alpha.A}\,\forall R \quad \alpha \notin \Gamma, \Pi \qquad \frac{A[^{\Phi}/_{\alpha}] \vdash \Gamma}{\forall\alpha.A \vdash \Gamma}\,\forall L \qquad \frac{\vdash \Gamma, A[^{\Phi}/_{\alpha}]}{\vdash \Gamma, \exists\alpha.A}\,\exists R \qquad \frac{\vdash \neg A, \Gamma}{\exists\alpha.A \vdash \Gamma}\,\exists L \quad \alpha \notin \Gamma$$

The rules for $\exists$ correspond to $\exists\alpha.A \equiv \neg\forall\alpha\neg A$.

The rules for the universal quantifier in LKQ are given in [DJS95]. We add the rules for the existential quantifier.

$$\frac{\Gamma \vdash \Delta, A\ ;}{\Gamma \vdash \Delta\ ; \forall\alpha.A}\,\forall R \quad \alpha \notin \Gamma, \Delta \qquad \frac{\Gamma, A[^{\Phi}/_{\alpha}] \vdash \Delta\ ;}{\Gamma, \forall\alpha.A \vdash \Delta\ ;}\,\forall L \qquad \frac{\Gamma \vdash \Delta\ ; A[^{\Phi}/_{\alpha}]}{\Gamma \vdash \Delta\ ; \exists\alpha.A}\,\exists R \qquad \frac{\Gamma, A \vdash \Delta\ ; \Pi}{\Gamma, \exists\alpha.A \vdash \Delta\ ; \Pi}\,\exists L \quad \alpha \notin \Gamma, \Delta, \Pi$$

The same for LKT:

$$\frac{\Pi\ ; \Gamma \vdash \Delta, A}{\Pi\ ; \Gamma \vdash \Delta, \forall\alpha.A}\,\forall R \quad \alpha \notin \Gamma, \Delta, \Pi \qquad \frac{A[^{\Phi}/_{\alpha}]\ ; \Gamma \vdash \Delta}{\forall\alpha.A\ ; \Gamma \vdash \Delta}\,\forall L \qquad \frac{;\ \Gamma \vdash \Delta, A[^{\Phi}/_{\alpha}]}{;\ \Gamma \vdash \Delta, \exists\alpha.A}\,\exists R \qquad \frac{;\ \Gamma, A \vdash \Delta}{\exists\alpha.A\ ; \Gamma \vdash \Delta}\,\exists L \quad \alpha \notin \Gamma, \Delta$$

It is also possible to extend id-nets with quantifiers by following the method of [Gir91b, Lau02].

Acknowledgements

I would like to thank Pierre-Louis Curien for his helpful comments on this work.

References

[CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *Proceedings of the International Conference on Functional Programming*, volume 35(9) of *ACM SIGPLAN Notices*, pages 233–243. Association for Computing Machinery, ACM Press, September 2000.

[Cro01] Tristan Crolard. Subtractive logic. *Theoretical Computer Science*, 254(1–2):151–185, March 2001.

[Cze77] Johannes Czermak. A remark on Gentzen's calculus of sequents. *Notre Dame Journal of Formal Logic*, 18(3):471–474, July 1977.

[DJS95] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. LKQ and LKT: Sequent calculi for second order logic based upon dual linear decompositions of classical implication. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 211–224. Cambridge University Press, 1995.

[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.

[Fil89] Andrzej Filinski. Declarative continuations and categorical duality. Master's thesis, computer science department, University of Copenhagen, August 1989. DIKU Report 89/11.

[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.

[Gir99] Jean-Yves Girard, editor. *Typed Lambda Calculi and Applications '99*, volume 1581 of *Lecture Notes in Computer Science*. Springer, April 1999.

[Gli29] Valerii Glivenko. Sur quelques points de la logique de M. Brouwer. Bulletins de la classe des sciences, Académie Royale de Belgique, 1929.

[Gri90] Timothy Griffin. A formulae-as-types notion of control. In *Proceedings of the 1990 Principles of Programming Languages Conference* [IEE90], pages 47–58.

[HS02] Martin Hofmann and Thomas Streicher. Completeness of continuation models for lambda-mu-calculus. *Information and Computation*, 179(2):332–355, December 2002.

[IEE90] IEEE. *Proceedings of the 1990 Principles of Programming Languages Conference*. IEEE Computer Society Press, 1990.

[Laf90] Yves Lafont. Interaction nets. In *Proceedings of the 1990 Principles of Programming Languages Conference* [IEE90], pages 95–108.

[Lam94] François Lamarche. Proof nets for intuitionistic linear logic I: Essential nets. Preliminary report, April 1994.

[Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In Girard [Gir99], pages 213–227.

[Lau02] Olivier Laurent. *Étude de la polarisation en logique*. Thèse de doctorat, Université Aix-Marseille II, March 2002.

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

[Lau05] Olivier Laurent. Classical isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5):969–1004, October 2005.

[Lev99] Paul Blain Levy. Call-by-push-value: A subsuming paradigm. In Girard [Gir99], pages 228–242.

[LQTdF05] Olivier Laurent, Myriam Quatrini, and Lorenzo Tortora de Falco. Polarized and focalized linear and classical proofs. *Annals of Pure and Applied Logic*, 134(2–3):217–264, July 2005.

[LRS93] Yves Lafont, Bernhard Reus, and Thomas Streicher. Continuation semantics or expressing implication by negation. Technical Report 93-21, Ludwig-Maximilians-Universität, München, 1993.

[LTdF04] Olivier Laurent and Lorenzo Tortora de Falco. Slicing polarized additive normalization. In Thomas Ehrhard, Jean-Yves Girard, Paul Ruet, and Philip Scott, editors, *Linear Logic in Computer Science*, volume 316 of *London Mathematical Society Lecture Note Series*, pages 247–282. Cambridge University Press, November 2004.

[MO00] Andrzej Murawski and Luke Ong. Dominator trees and fast verification of proof nets. In *Proceedings of the fifteenth annual symposium on Logic In Computer Science*, pages 181–191, Santa Barbara, June 2000. IEEE, IEEE Computer Society Press.

[Oga98] Ichiro Ogata. Cut elimination for classical proofs as continuation passing style computation. In Jieh Hsiang and Atsushi Ohori, editors, *Advances in Computing Science ASIAN*, volume 1538 of *Lecture Notes in Computer Science*, pages 61–78. Springer, 1998.

[Par92] Michel Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proceedings of International Conference on Logic Programming and Automated Reasoning*, 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. *Comptes Rendus de l'Académie des Sciences de Paris*, 323:113–116, 1996.

[Sel01] Peter Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11(2):207–260, April 2001.

[SR98] Thomas Streicher and Bernhard Reus. Classical logic, continuation semantics and abstract machines. *Journal of Functional Programming*, 8(6):543–572, November 1998.