



HAL
open science

Parametric mixed sequent calculus

Karim Nour, Olivier Laurent

► **To cite this version:**

| Karim Nour, Olivier Laurent. Parametric mixed sequent calculus. 2005. hal-00382689

HAL Id: hal-00382689

<https://hal.science/hal-00382689>

Submitted on 11 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PARAMETRIC MIXED SEQUENT CALCULUS

Olivier LAURENT

Université Denis Diderot
PPS - Case 7014
2, place Jussieu
75251 Paris Cedex 05 - France

Karim NOUR

Université de Savoie
LAMA - Equipe de logique
Campus scientifique
73376 Le Bourget du Lac - France

Abstract

In this paper, we present a propositional sequent calculus containing disjoint copies of classical and intuitionistic logics. We prove a cut-elimination theorem and we establish a relation between this system and linear logic.

1 Introduction

The systems which represent classical and intuitionistic logics have often been studied separately. For each of these logics we establish its proper properties. Although the systems which represent intuitionistic logic are sub-systems of those which represent classical logic, we usually do not find in the literature studies of the properties of these systems at the same time. We mainly find many external translations from classical to intuitionistic logic. The main novelty of the system which we look for is that intuitionistic and classical logics appear as fragments. For instance a proof of an intuitionistic formula may use classical lemmas without any restriction. This approach is radically different from the one that consists in changing the rule of the game when we want to change the logic. We want only one logic which, depending on its use, may appear classical or intuitionistic.

J.-Y. Girard presented in [2] a single sequent calculus (denoted LU) common to classical, intuitionistic and linear logics. Each formula is given with a polarity: positive, neutral and negative. For each connective the rules depend on the polarity of the formulas. On the other hand the system LU has a cut-elimination theorem and then the sub-formula property. Although the system LU is an answer to our question, we seek a simpler and more intuitive system only for classical and intuitionistic logics.

The usefulness of finding a system which mixes classical and intuitionistic logics is more and more recognized. For example, J.-L. Krivine and the second author introduced a second order mixed logic in order to type storage and control operators in $\lambda\mathcal{C}$ -calculus (see [3]). Indeed, they needed intuitionistic logic to characterize the operational behavior of the storage operators and classical logic for the control operators. The theoretical properties of this system are not difficult to prove because the only connectives are \rightarrow and \forall . Recently, C. Raffalli introduced in [5] a second order mixed logic which is slight extension of that of [3] in order to extract a program for some classical proofs. He applied his method to extract a program from a classical proof of Dickson's lemma.

In the paper [4], the second author and A. Nour presented a propositional logic with all connectives (denoted PML) containing three kinds of variables: minimal, intuitionistic and classical. The absurdity rules are restricted to the formulas containing the corresponding variables. They introduced for the system PML a Kripke semantics and they showed a completeness theorem. They deduced from this theorem a very significant result which is the following: "for a formula to be derivable in a logic,

it is necessary that the formula contains at least a variable which corresponds to this logic". They also presented a sequent calculus version of this system. The systems presented in [4] are not satisfactory because they do not have cut-elimination results.

We propose in this paper another approach to solve this problem. We fix a set of formulas \mathcal{P} which represents intuitively the set of formulas on which we can do classical reasoning. A sequent is a pair of multisets of formulas, denoted $\Gamma \vdash \Delta; \Pi$, where $\Delta \subseteq \mathcal{P}$ and Π contains at most one formula. We introduce a list of classical rules on the multiset Δ and intuitionistic rules on Π . Certain rules will require conditions on the membership of some formulas to the set \mathcal{P} . We prove a cut-elimination theorem and thus deduce the sub-formula property. We show how to code systems LK and LJ in our system which is coded in system LL.

The paper is organized as follows. In section 2 we present the rules of our system. We prove in section 3 the cut elimination properties of the system. The codings of systems LK and LJ in our system are given in section 4. We present in section 5 a coding of the system in system LL.

2 The $ML_{\mathcal{P}}$ sequent calculus

Definition 2.1

1. The set of formulas is defined by the following grammar:

$$F ::= 0 \mid \perp \mid X \mid F \wedge F \mid F \vee F \mid F \rightarrow F$$

where X ranges over a set of propositional variables \mathcal{V} .

2. Let \mathcal{P} be a subset of formulas. A \mathcal{P} -sequent is a pair of multisets of formulas, denoted $\Gamma \vdash \Delta; \Pi$, where $\Delta \subseteq \mathcal{P}$ and Π contains at most one formula. The set Δ is called the body and Π (the space after “;”) is called the stoup. A \mathcal{P} -derivation may be constructed according to one of the rules below.

AXIOM/CUTS

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

$$\frac{\Gamma \vdash \Delta; A \quad \Gamma', A \vdash \Delta'; \Pi}{\Gamma, \Gamma' \vdash \Delta, \Delta'; \Pi} \text{ cut}_1$$

$$\frac{\Gamma \vdash \Delta, A; \Pi \quad \Gamma', A \vdash \Delta';}{\Gamma, \Gamma' \vdash \Delta, \Delta'; \Pi} \text{ cut}_2$$

STRUCTURE

$$\frac{\Gamma \vdash \Delta; A \quad A \in \mathcal{P}}{\Gamma \vdash \Delta, A;} \text{ der}$$

$$\frac{\Gamma, A, A \vdash \Delta; \Pi}{\Gamma, A \vdash \Delta; \Pi} \text{ c}_l$$

$$\frac{\Gamma \vdash \Delta, A, A; \Pi}{\Gamma \vdash \Delta, A; \Pi} \text{ c}_r$$

$$\frac{\Gamma \vdash \Delta; \Pi}{\Gamma, A \vdash \Delta; \Pi} \text{ w}_l$$

$$\frac{\Gamma \vdash \Delta; \Pi \quad A \in \mathcal{P}}{\Gamma \vdash \Delta, A; \Pi} \text{ w}_r$$

LOGIC

$$\begin{array}{c}
\frac{\Delta \subseteq \mathcal{P}}{\Gamma, 0 \vdash \Delta; \Pi} 0 \\
\\
\frac{}{\perp \vdash; \perp} \perp \\
\\
\frac{\Gamma, A, B \vdash \Delta; C \quad A \notin \mathcal{P} \text{ and } B \notin \mathcal{P}}{\Gamma, A \wedge B \vdash \Delta; C} \wedge_l^1 \qquad \frac{\Gamma \vdash \Delta; A \quad \Gamma' \vdash \Delta'; B}{\Gamma, \Gamma' \vdash \Delta, \Delta'; A \wedge B} \wedge_r^1 \\
\frac{\Gamma, A, B \vdash \Delta;}{\Gamma, A \wedge B \vdash \Delta;} \wedge_l^2 \qquad \frac{\Gamma \vdash \Delta, A; \quad \Gamma' \vdash \Delta', B;}{\Gamma, \Gamma' \vdash \Delta, \Delta'; A \wedge B} \wedge_r^2 \\
\frac{\Gamma \vdash \Delta; A \quad \Gamma' \vdash \Delta', B;}{\Gamma, \Gamma' \vdash \Delta, \Delta'; A \wedge B} \wedge_r^3 \\
\frac{\Gamma \vdash \Delta, A; \quad \Gamma' \vdash \Delta', B;}{\Gamma, \Gamma' \vdash \Delta, \Delta'; A \wedge B} \wedge_r^4 \\
\\
\frac{\Gamma, A \vdash \Delta; C \quad \Gamma, B \vdash \Delta; C \quad A \notin \mathcal{P} \text{ and } B \notin \mathcal{P}}{\Gamma, A \vee B \vdash \Delta; C} \vee_l^1 \qquad \frac{\Gamma \vdash \Delta; A}{\Gamma \vdash \Delta; A \vee B} \vee_r^1 \\
\frac{\Gamma \vdash \Delta; B}{\Gamma \vdash \Delta; A \vee B} \vee_r^2 \\
\frac{\Gamma, A \vdash \Delta; \quad \Gamma, B \vdash \Delta;}{\Gamma, A \vee B \vdash \Delta;} \vee_l^2 \qquad \frac{\Gamma \vdash \Delta, A;}{\Gamma \vdash \Delta; A \vee B} \vee_r^3 \\
\frac{\Gamma \vdash \Delta, B;}{\Gamma \vdash \Delta; A \vee B} \vee_r^4 \\
\\
\frac{\Gamma, B \vdash \Delta; C \quad \Gamma' \vdash \Delta'; A \quad B \notin \mathcal{P}}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta'; C} \rightarrow_l^1 \qquad \frac{\Gamma, A \vdash \Delta; B}{\Gamma \vdash \Delta; A \rightarrow B} \rightarrow_r^1 \\
\frac{\Gamma, B \vdash \Delta; \quad \Gamma' \vdash \Delta'; A}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta';} \rightarrow_l^2 \qquad \frac{\Gamma, A \vdash \Delta, B;}{\Gamma \vdash \Delta; A \rightarrow B} \rightarrow_r^2 \\
\frac{\Gamma, B \vdash \Delta; \quad \Gamma' \vdash \Delta', A; \Pi}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta'; \Pi} \rightarrow_l^3
\end{array}$$

We write $\Gamma \vdash_{\mathcal{P}} \Delta; \Pi$ if the \mathcal{P} -sequent $\Gamma \vdash \Delta; \Pi$ is derivable in system $\text{ML}_{\mathcal{P}}$.

Remark 2.1 1. The conditions which we add on the set \mathcal{P} in some left logical rules are necessary to obtain a cut-elimination theorem. Indeed without these conditions the cuts on the principal formulas of the rules (\wedge_r^i $2 \leq i \leq 4$ and \wedge_l^1) or (\vee_r^i $3 \leq i \leq 4$ and \vee_l^1) or (\rightarrow_r^2 and \rightarrow_l^1) cannot be eliminate.

2. We can remove the rules \wedge_r^3 and \wedge_r^4 and replace the rule \wedge_l^1 by the following rule:

$$\frac{\Gamma, A, B \vdash \Delta; C \quad A \notin \mathcal{P} \text{ or } B \notin \mathcal{P}}{\Gamma, A \wedge B \vdash \Delta; C} \wedge_l^1$$

In this new system, the results of sections 3 and 4 remain true but not those of section 5.

3. We chose an additive “or” to facilitate the embedding of our system in LL. The results of sections 3 and 4 remain true if we add the two following left rules:

$$\frac{\Gamma, A \vdash \Delta; C \quad \Gamma, B \vdash \Delta; \quad A \notin \mathcal{P}}{\Gamma, A \vee B \vdash \Delta; C} \vee_l^3 \qquad \frac{\Gamma, A \vdash \Delta; \quad \Gamma, B \vdash \Delta; C \quad B \notin \mathcal{P}}{\Gamma, A \vee B \vdash \Delta; C} \vee_l^4$$

4. If we consider the connector \neg as primitive, we can add the following rules:

$$\frac{\Gamma \vdash \Delta; A}{\Gamma, \neg A \vdash \Delta; } \neg_l^1 \qquad \frac{\Gamma, A \vdash \Delta;}{\Gamma \vdash \Delta; \neg A} \neg_r$$

$$\frac{\Gamma \vdash \Delta, A; \Pi}{\Gamma, \neg A \vdash \Delta; \Pi} \neg_l^2$$

If $\perp \in \mathcal{P}$, we derive $\neg A \vdash_{\mathcal{P}}; A \rightarrow \perp$ and $A \rightarrow \perp \vdash_{\mathcal{P}}; \neg A$.

3 Properties of system $ML_{\mathcal{P}}$

Theorem 3.1 *The Hauptsatz holds for $ML_{\mathcal{P}}$.*

Proof The degree of a cut-rule in a \mathcal{P} -derivation is the pair of integers (l, k) where l is the length of the cut-formula and k is defined by:

- $k = 3$, if it is the rule cut_2 .
- $k = 2$, if it is the rule cut_1 and the cut-formula of the stoup of the left premise is the principal formula of a logical rule.
- $k = 1$, if it is the rule cut_1 and the cut-formula of the stoup of the left premise is the principal formula of a logical rule but not the cut-formula of the right premise.
- $k = 0$, if it is the rule cut_1 and the cut-formulas are the principal formulas of logical rules.

The order we consider on degrees is the lexicographic order.

The degree of a \mathcal{P} -derivation is the finite list of increasing degrees of its cuts. We consider also the lexicographic order on these degrees.

Let \mathcal{D} be a \mathcal{P} -derivation. We will explain how to reduce a cut in \mathcal{D} to obtain a \mathcal{P} -derivation of smaller degree.

We consider a cut of degree (l, k) where its premises are derivable without the cut-rules.

- If $k = 0$, we replace this cut by other cuts of degrees (l', k') where $l' < l$.
- If $k = 1$, we move up the left premise in the \mathcal{P} -derivation of the right premise at the places where the cut-formula was the principal formula of a logical rule. We thus replace this cut by other cuts of degrees $(l, 0)$.
- If $k = 2$, we move up the right premise in the \mathcal{P} -derivation of the left premise at the places where the cut-formula was the principal formula of a logical rule. We thus replace this cut by other cuts of degrees $(l, 1)$ or $(l, 0)$.
- If $k = 3$, we move up the right premise in the \mathcal{P} -derivation of the left premise at the places where the cut-formula was introduced using the rules der , w_r or 0 . We thus replace this rule cut_2 by other rules cut_1 of degrees (l, k') where $0 \leq k' \leq 2$.

We notice that in each case the degree of the obtained \mathcal{P} -derivation decreases strictly. □

Corollary 3.1 *The $ML_{\mathcal{P}}$ has the sub-formula property.*

Proof By theorem 3.1. □

Corollary 3.2 1. If $\vdash_{\mathcal{P}}; A \vee B$, then $\vdash_{\mathcal{P}}; A$ or $\vdash_{\mathcal{P}}; B$ or $\vdash_{\mathcal{P}}; A$ or $\vdash_{\mathcal{P}}; B$;

2. If $A, B \notin \mathcal{P}$ and $\vdash_{\mathcal{P}}; A \vee B$, then $\vdash_{\mathcal{P}}; A$ or $\vdash_{\mathcal{P}}; B$.

Proof We consider a normal derivation of $\vdash_{\mathcal{P}}; A \vee B$ and we look at the last used rule. \square

4 Codings of LK and LJ in $ML_{\mathcal{P}}$

We consider systems LK and LJ constructed respectively over the sets of variables $\mathcal{V} \cup \{\perp\}$ and $\mathcal{V} \cup \{0\}$. We suppose that \perp (resp. 0) is the symbol for the absurdity of LK (resp. LJ). We will give some conditions to code separately systems LK and LJ in $ML_{\mathcal{P}}$.

Definition 4.1 A set of formulas \mathcal{S} is said to be stable iff for every $c \in \{\wedge, \vee, \rightarrow\}$, if $AcB \in \mathcal{S}$, then $A, B \in \mathcal{S}$

Theorem 4.1 Let \mathcal{K} be a stable set such that $\mathcal{K} \subseteq \mathcal{P}$ and $0 \notin \mathcal{K}$. If $\Gamma, \Delta \subseteq \mathcal{K}$, then $\Gamma \vdash_{\mathcal{P}} \Delta$; iff $\Gamma \vdash_{LK} \Delta$.

Proof \Rightarrow : If we replace “;” by “,” the rules of $ML_{\mathcal{P}}$ are rules of LK.

\Leftarrow : We consider a derivation \mathcal{D} of $\Gamma \vdash_{LK} \Delta$. We check that we can move up the rules used in \mathcal{D} without putting formulas in the stoups. The rule *cut* and the left rules of LK correspond to the rules *cut*₂, *c*_l, *w*_l, \perp , \wedge_l^2, \vee_l^2 and \rightarrow_l^3 (without stoups). The axiom and the right rules of LK correspond to *ax*, *c*_r, *w*_r, $\wedge_r^2, \vee_r^3, \vee_r^4$ and \rightarrow_r^2 using the rule *der*. \square

Remark 4.1 1. The set $\mathcal{K} = \mathcal{P}$ of all formulas on $\mathcal{V} \cup \{\perp\}$ satisfies the hypothesis of theorem 4.1.

2. For every formula A on $\mathcal{V} \cup \{\perp\}$, let \mathcal{K}_A be the finite set of the sub-formulas of A . If $\mathcal{K}_A \subseteq \mathcal{P}$, then $\vdash_{\mathcal{P}}; A$; iff $\vdash_{LK} A$.

Theorem 4.2 Let \mathcal{I} be a stable set such that $\mathcal{I} \cap \mathcal{P} = \emptyset$ and $\perp \notin \mathcal{I}$. If $\Gamma, A \subseteq \mathcal{I}$, then $\Gamma \vdash_{\mathcal{P}}; A$ iff $\Gamma \vdash_{LJ} A$.

Proof \Rightarrow : We consider a normal \mathcal{P} -derivation \mathcal{D} of $\Gamma \vdash_{\mathcal{P}}; A$. The fact that $\Gamma, A \subseteq \mathcal{I}$ allows to move up the rules using in \mathcal{D} without putting formulas in the bodies. Then the only rules used in \mathcal{D} are intuitionistic rules.

\Leftarrow : The rules of LJ correspond to the rules *ax*, *cut*₁, *c*_l, *w*_l, 0, $\wedge_l^1, \wedge_r^1, \vee_l^1, \vee_r^1, \vee_r^2, \rightarrow_l^1$ and \rightarrow_r^1 (without bodies). \square

Remark 4.2 1. Let $\mathcal{P} = \{\perp\}$. The set \mathcal{I} of all formulas on $\mathcal{V} \cup \{0\}$ satisfies the hypothesis of theorem 4.2.

2. Let A be a formula on $\mathcal{V} \cup \{0\}$. We have $\vdash_{\{\perp\}}; A$ iff $\vdash_{LJ} A$.

To code, at the same time, the two systems, it is necessary to realize all the conditions of theorems 4.1 and 4.2. We give an example of such a system.

Definition 4.2 1. We suppose that we have two disjoint sets of propositional variables: $\mathcal{V}_i = \{X_i, Y_i, Z_i, \dots\}$ the set of intuitionistic variables and $\mathcal{V}_c = \{X_c, Y_c, Z_c, \dots\}$ the set of classical variables. Let $\mathcal{V} = \mathcal{V}_i \cup \mathcal{V}_c$.

2. If A is a formula, we denote by $var(A)$ the set of variables and constants of A .

3. Let $\tilde{\mathcal{F}}$ be the set of all formulas,
 $\tilde{\mathcal{K}} = \{F \in \tilde{\mathcal{F}} / \text{var}(F) \subseteq \mathcal{V}_c \cup \{\perp\}\}$ the set of classical formulas,
 $\tilde{\mathcal{I}} = \{F \in \tilde{\mathcal{F}} / \text{var}(F) \subseteq \mathcal{V}_i \cup \{0\}\}$ the set of intuitionistic formulas and
 $\tilde{\mathcal{P}} = \tilde{\mathcal{F}} - \tilde{\mathcal{I}}$.

Corollary 4.1 1. If $\Gamma, \Delta \subseteq \tilde{\mathcal{K}}$, then $\Gamma \vdash_{\tilde{\mathcal{P}}} \Delta$; iff $\Gamma \vdash_{\text{LK}} \Delta$.

2. If $\Gamma, A \subseteq \tilde{\mathcal{I}}$, then $\Gamma \vdash_{\tilde{\mathcal{P}}} A$ iff $\Gamma \vdash_{\text{LJ}} A$.

Proof We use theorems 4.1 and 4.2. □

5 Coding of $\text{ML}_{\mathcal{P}}$ in LL

Definition 5.1 1. We define the following two translations b and t from $\text{ML}_{\mathcal{P}}$ to LL:

- if $A \notin \mathcal{P}$, then $b(A) = t(A)$
- if $A \in \mathcal{P}$, then $b(A) = ?t(A)$

and

- $t(0) = t(\perp) = 0$
- $t(X) = !X$, for every $X \in \mathcal{V}$
- $t(A \wedge B) = !b(A) \otimes !b(B)$
- $t(A \vee B) = !b(A) \oplus !b(B)$
- $t(A \rightarrow B) = !(t(A) \multimap b(B))$

2. If $\Gamma = A_1, \dots, A_n$, then $t(\Gamma) = t(A_1), \dots, t(A_n)$.

Lemma 5.1 1. If $\Gamma, t(A), t(A) \vdash_{\text{LL}} \Delta$, then $\Gamma, t(A) \vdash_{\text{LL}} \Delta$.

2. If $\Gamma \vdash_{\text{LL}} \Delta$, then $\Gamma, t(A) \vdash_{\text{LL}} \Delta$.

3. If $t(\Gamma), A \vdash_{\text{LL}} ?t(\Delta)$, then $t(\Gamma), ?A \vdash_{\text{LL}} ?t(\Delta)$.

4. If $t(\Gamma) \vdash_{\text{LL}} ?t(\Delta), A$, then $t(\Gamma) \vdash_{\text{LL}} ?t(\Delta), !A$.

Proof See [1]. □

Theorem 5.1 If $\Gamma \vdash_{\mathcal{P}} \Delta; \Pi$, then $t(\Gamma) \vdash_{\text{LL}} ?t(\Delta), t(\Pi)$

Proof By induction on a \mathcal{P} -derivation of $\Gamma \vdash_{\mathcal{P}} \Delta; \Pi$. We look at the last rule used.

- For the rules $ax, cut_1, der, c_r, w_r, 0, \perp, \wedge_l^1, \vee_l^1$ and \rightarrow_l^1 , the proof is easy.
- For the rules c_l and w_l , we use 1. and 2. of lemma 5.1.
- For the rules $cut_2, \wedge_l^2, \vee_l^2$ and \rightarrow_l^2 , we use 3. of lemma 5.1.
- For the rules \wedge_r^i ($1 \leq i \leq 4$), \vee_r^i ($1 \leq i \leq 4$) and \rightarrow_r^i ($1 \leq i \leq 2$), we use 4. of lemma 5.1.

- For the rule \rightarrow_l^3 , we have, by induction hypothesis, $t(\Gamma), b(B) \vdash_{\text{LL}} ?t(\Delta)$ and $t(\Gamma') \vdash_{\text{LL}} p?t(\Delta'), b(A), t(\Pi)$. We deduce that

$$\frac{\frac{\frac{\frac{t(\Gamma), b(B) \vdash_{\text{LL}} ?t(\Delta)}{t(\Gamma), t(A) \multimap b(B), t(A) \vdash ?t(\Delta)}}{t(\Gamma), !(t(A) \multimap b(B)), t(A) \vdash ?t(\Delta)}}{t(\Gamma), !(t(A) \multimap b(B)), b(A) \vdash ?t(\Delta)}}{\frac{t(\Gamma') \vdash_{\text{LL}} ?t(\Delta'), b(A), t(\Pi)}{t(\Gamma), t(\Gamma'), t(A \rightarrow B) \vdash ?t(\Delta), ?t(\Delta'), t(\Pi)}}}{t(\Gamma), t(\Gamma'), t(A \rightarrow B) \vdash ?t(\Delta), ?t(\Delta'), t(\Pi)} \quad \square$$

References

- [1] V. Danos, J.-B. Joinet and H. Schellinx, *A new deconstructive logic: Linear Logic*, Journal of Symbolic Logic, vol. 62, num. 3, pp. 755-807, 1997.
- [2] J.-Y. Girard, *On the unity of logic*, Annals of pure and applied logic, vol. 59, pp. 201-217, 1993.
- [3] K. Nour, *Mixed logic and storage operators*, Archive for Mathematical Logic, vol. 39, pp. 261-280, 2000.
- [4] K. Nour and A. Nour; *Propositional mixed logic: its syntax and semantics*, Journal of Applied Non-Classical Logics, vol. 13, pp. 377-390, 2003.
- [5] C. Raffali, *Getting results from programs extracted from classical proofs*, Theoretical Computer Science, vol. 323, pp. 49-70, 2004.