Using linear logic to reason about sequent systems *

Dale Miller[1] and Elaine Pimentel[2]

[1] Computer Science and Engineering Department, 220 Pond Lab,
Pennsylvania State University, University Park, PA 16802-6106 USA
dale@cse.psu.edu
[2] Departamento de Matemática,
Universidade Federal de Minas Gerais, Belo Horizonte, M.G. Brasil
elaine@mat.ufmg.br

Abstract. Linear logic can be used as a meta-logic for the specification of some sequent calculus proof systems. We explore in this paper properties of such linear logic specifications. We show that derivability of one proof system from another has a simple decision procedure that is implemented simply via bounded logic programming search. We also provide conditions to ensure that an encoded proof system has the cut-elimination property and show that this can be decided again by simple, bounded proof search algorithms.

1 Introduction

Various logical frameworks based on intuitionistic logic have been proposed [FM88,Pau89,HHP93] and used for specifying natural deduction proof systems. Given the intimate connection between natural deduction and λ-calculus, applications requiring object-level binding and substitutions have also been successfully implemented in these logical frameworks [Mil00].

In [Mil96], Miller proposed moving from intuitionistic logic to the more expressive setting of linear logic to capture the more general setting of sequent calculus proof system. This use of linear logic has been future explored in [Ric98,Pim01,MP]. In this paper we consider the structure of proofs in the Forum presentation of linear logic in order to show how various aspects of the meta-theory of linear logic can be used to conclude properties of the sequent calculus being specified. In particular, we describe a decision procedure for determining if one encoded proof system is derivable from another and we present conditions and their decision procedure that imply that an encoded proof system satisfies cut-elimination.

After providing an overview of Forum in Section 2 and the encoding into Forum of object-level sequents and inference rules in Section 3, we prove in

* Miller has been supported in part by NSF grants CCR-9912387, CCR-9803971, INT-9815645, and INT-9815731. Both authors wish to thank L'Institut de Mathématiques de Luminy, University Aix-Marseille 2 for the support to attend the Logic and Interaction Weeks in February 2002, during which much of this paper was written.

Section 4 that deduction between encodings of inference rules is captured by shallow Forum proofs. A decision procedure for determining if the encoding of one proof system is derivable from another proof system then follows by doing bounded depth proof search. Of course, we wish to know if an object-level proof system admits a cut-elimination theorem. Section 5 contains the basic background for this problem and Section 6 contains the main object-level cut-elimination theorem. Section 7 contains a specification and discussion of Girard's Logic of Unity. Finally, we conclude in Section 8.

2 The Forum presentation of linear logic

The Forum presentation of linear logic [Mil96] relies on the connectives $\bot$, $\parr$, ?, $\top$, &, $\multimap$, $\Rightarrow$, and $\forall$: this set of connectives is complete for linear logic, in the sense that all other linear logic connectives can be defined from these. Proof search using this collection of connectives can be restricted so that simple goal-directed proof search (using the technical device of multiple-conclusion uniform proofs [Mil93]) is complete. Thus, Forum makes it possible to claim that all of linear logic can be seen as an abstract logic programming language [MNPS91]. Forum has been used to specify a number of computation systems, ranging from object-oriented languages [DM95], imperative programming features [Mil96,Chi95], and a RISC processor [Chi95]. In this paper, we use Forum as a specification language for sequent calculus proof systems. For this purpose, we work often within a weaker fragment of Forum, called *Flat Forum*.

2.1 Flat Forum

A formula of Forum is a *flat goal* if it does not contain occurrences of $\multimap$ and $\Rightarrow$, and all occurrences of the modal ? have atomic scope. A formula of the form

$$\forall \bar{y}(G_1 \hookrightarrow \cdots \hookrightarrow G_m \hookrightarrow A_1 \parr \cdots \parr A_n), \quad (m, n \geq 0)$$

is called a *flat clause* if $G_1, \ldots, G_m$ are flat goals, $A_1, \ldots, A_n$ are atomic formulas, and occurrences of the symbol $\hookrightarrow$ are either occurrences of $\multimap$ or $\Rightarrow$. The formula $A_1 \parr \cdots \parr A_n$ is the *head* of such a clause, while for each $i = 1, \ldots, m$, the formula G_i is a *body* of this clause. If $n = 0$, then we write the head as simply $\bot$ and say that the head is *empty*. A flat clause is essentially a clause of the LinLog system [And92] except that heads of flat clauses may be empty.

A flat Forum formula is logically equivalent to a formula in *uncurried* form, namely, a formula of the form

$$\forall \bar{y}(B \multimap A_1 \parr \cdots \parr A_n)$$

where $n \geq 0$, $\bar{y}$ is the list of variables free in the head $A_1 \parr \cdots \parr A_n$, all free variables of B are also free in the head, and B may have occurrences of $\exists$, $\otimes$, 1, and !, but not in the scope of $\parr$, ?, &, $\forall$, $\multimap$, and $\Rightarrow$ (using the terminology

$$\frac{}{\Psi;\Delta \longrightarrow \mathcal{A}, \top, \Gamma; \Upsilon}\ \top R \qquad \frac{\Psi;\Delta \longrightarrow \mathcal{A}, B, \Gamma; \Upsilon \quad \Psi;\Delta \longrightarrow \mathcal{A}, C, \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, B \,\&\, C, \Gamma; \Upsilon}\ \& R$$

$$\frac{\Psi;\Delta \longrightarrow \mathcal{A}, \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, \bot, \Gamma; \Upsilon}\ \bot R \qquad \frac{\Psi;\Delta \longrightarrow \mathcal{A}, B, C, \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, B \parr C, \Gamma; \Upsilon}\ \parr R$$

$$\frac{\Psi; B, \Delta \longrightarrow \mathcal{A}, C, \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, B \multimap C, \Gamma; \Upsilon}\ \multimap R \qquad \frac{B, \Psi;\Delta \longrightarrow \mathcal{A}, C, \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, B \Rightarrow C, \Gamma; \Upsilon}\ \Rightarrow R$$

$$\frac{\Psi;\Delta \longrightarrow \mathcal{A}, B[y/x], \Gamma; \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, \forall_\tau x.B, \Gamma; \Upsilon}\ \forall R \qquad \frac{\Psi;\Delta \longrightarrow \mathcal{A}, \Gamma; B, \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}, ?\,B, \Gamma; \Upsilon}\ ?\,R$$

$$\frac{B, \Psi;\Delta \xrightarrow{B} \mathcal{A}; \Upsilon}{B, \Psi;\Delta \longrightarrow \mathcal{A}; \Upsilon}\ \mathit{decide}! \qquad \frac{\Psi;\Delta \longrightarrow \mathcal{A}, B; B, \Upsilon}{\Psi;\Delta \longrightarrow \mathcal{A}; B, \Upsilon}\ \mathit{decide}?$$

$$\frac{\Psi;\Delta \xrightarrow{B} \mathcal{A}; \Upsilon}{\Psi; B, \Delta \longrightarrow \mathcal{A}; \Upsilon}\ \mathit{decide} \qquad \frac{}{\Psi;\cdot \xrightarrow{A} A; \Upsilon}\ \mathit{initial} \qquad \frac{}{\Psi;\cdot \xrightarrow{A} \cdot; A, \Upsilon}\ \mathit{initial}?$$

$$\frac{}{\Psi;\cdot \xrightarrow{\bot} \cdot; \Upsilon}\ \bot L \qquad \frac{\Psi;\Delta \xrightarrow{B_i} \mathcal{A}; \Upsilon}{\Psi;\Delta \xrightarrow{B_1 \& B_2} \mathcal{A}; \Upsilon}\ \&\, L_i \qquad \frac{\Psi; B \longrightarrow \cdot; \Upsilon}{\Psi;\cdot \xrightarrow{?\,B} \cdot; \Upsilon}\ ?\,\mathrm{L}$$

$$\frac{\Psi;\Delta_1 \xrightarrow{B} \mathcal{A}_1; \Upsilon \quad \Psi;\Delta_2 \xrightarrow{C} \mathcal{A}_2; \Upsilon}{\Psi;\Delta_1,\Delta_2 \xrightarrow{B \parr C} \mathcal{A}_1, \mathcal{A}_2; \Upsilon}\ \parr L \qquad \frac{\Psi;\Delta \xrightarrow{B[t/x]} \mathcal{A}; \Upsilon}{\Psi;\Delta \xrightarrow{\forall_\tau x.B} \mathcal{A}; \Upsilon}\ \forall L$$

$$\frac{\Psi;\Delta_1 \longrightarrow \mathcal{A}_1, B; \Upsilon \quad \Psi;\Delta_2 \xrightarrow{C} \mathcal{A}_2; \Upsilon}{\Psi;\Delta_1,\Delta_2 \xrightarrow{B \multimap C} \mathcal{A}_1, \mathcal{A}_2; \Upsilon}\ \multimap L \qquad \frac{\Psi;\cdot \longrightarrow B; \Upsilon \quad \Psi;\Delta \xrightarrow{C} \mathcal{A}; \Upsilon}{\Psi;\Delta \xrightarrow{B \Rightarrow C} \mathcal{A}; \Upsilon}\ \Rightarrow L$$

Fig. 1. The Forum proof system. The rule ∀R has the proviso that y is not free in the lower sequent. In $\&L_i$, $i = 1$ or $i = 2$.

of [And92], no synchronous connective is in the scope of an asynchronous connective.) Although uncurried clauses are not Forum clauses, they can easily be rewritten to curried clauses using the following logical equivalences:

$$(B \otimes C) \multimap H \equiv B \multimap C \multimap H \qquad (\exists x.B\ x) \multimap H \equiv \forall x.(B(x) \multimap H)$$

$$(B \oplus C) \multimap H \equiv (B \multimap H)\,\&\,(C \multimap H) \qquad (!\,B) \multimap H \equiv B \Rightarrow H \qquad 1 \multimap H \equiv H.$$

(In the equivalence involving $\exists$, x is not free in H.)

As in Church's Simple Theory of Types [Chu40], both terms and formulas are built using a simply typed λ-calculus. We assume the usual rules of α, β, and η-conversion and we identify terms and formulas up to α-conversion. A term is λ-normal if it contains no β and no η redexes. All terms are λ-convertible to a term in λ-normal form, and such a term is unique up to α-conversion. The substitution notation $B[t/x]$ denotes the λ-normal form of the β-redex $(\lambda x.B)t$. Following [Chu40], we shall also assume that formulas of Forum have type o.

There are two kinds of sequents in Forum, namely, $\Psi;\Delta \longrightarrow \Gamma;\Upsilon$ and $\Psi;\Delta \xrightarrow{B} \Gamma;\Upsilon$. The outermost contexts, labeled here as Ψ and Υ, are the left and right classical contexts: these contexts are sets of formulas. The innermost contexts, labeled here as Δ and Γ, are the left and right linear contexts: these

contexts are multisets of formulas. In the second sequent, the formula B over the sequent arrow is also formula and multiset Γ contains only atomic formulas. (Notice that the position of classical and linear contexts is different from that used in sequents within the LU proof system of Girard [Gir93].) The sequent system for Forum is given in Figure 1. The following is a consequence of the soundness and completeness result for Forum [Mil96].

Theorem 1. *Let Ψ be a set of flat clauses, Γ and Δ be multisets of flat goals, and Υ be a set of atomic formulas. Then the sequent $\Psi; \Delta \longrightarrow \Gamma; \Upsilon$ has a proof if and only if $!\Psi, \Delta \vdash \Gamma, ?\Upsilon$ has a proof in linear logic.*

The following lemma holds in general for Forum proofs, but it is particularly relevant here since the scope of ? will always be atomic.

Lemma 1. *If a sequent has a Forum proof, it has a proof in which there are no occurrences of decide? applied to an atomic formula.*

Proof Permute all occurrences of decide? involving an atomic formula up in a proof until they reach an instance of the initial rule, in which the combination of initial and decide? can be rewritten to an occurrence of initial?. ∎

Furthermore, a Forum proof of a sequent of the form $\Psi; \Delta \longrightarrow \Gamma; \Upsilon$, where Ψ, Δ, Γ and Υ are as in Theorem 1, is such that there are no occurrences of $-\circ$R, $\Rightarrow$R, and decide? inference rules and the left-hand linear context of all sequents in the proof is a subset of Δ.

3 Representing sequents and inference rules

This section summarizes material found in [MP,Pim01]: see also [Mil96,Ric98] for related material.

Since we now wish to represent one logic and proof system within another, we need to distinguish between the meta-logic, namely, linear logic as presented by Forum, and the various object-logics for which we wish to specify sequent proof systems. Formulas of the object-level will be identified with meta-level terms of type *bool*. Object-level logical connectives will be introduced as needed and as constructors of this type.

A two-sided sequent $\Delta \longrightarrow \Gamma$ is generally restricted so that Δ and Γ are either lists, multisets, or sets of formulas. Sets are used if all three structural rules (exchange, weakening, contraction) are implicit; multisets are used if exchange is implicit; and lists are used if no structural rule is implicit. Since our goal here is to encode object-level sequents into meta-level sequents as directly as possible, and since contexts in Forum are either multisets or sets, we will not be able to represent sequents that make use of lists. It is unlikely, for example, that non-commutative object-logics can be encoded into our linear logic meta theory along the lines we describe below.

3.1 Three schemes for encoding sequents

Consider the well-known, two-sided sequent proof systems for classical, intuitionistic, and linear logic. A convenient distinction between these logics can be described, in part, by where the structural rules of thinning and contraction can be applied. In classical logic, these structural rules are allowed on both sides of the sequent arrow; in intuitionistic logic, no structural rules are allowed on the right of the sequent arrow; and in linear logic, they are not allowed on either sides of the arrow. Thus a classical sequent is a pairing of two sets; a linear logic sequent is a pairing of two multisets; and an intuitionistic sequent is the pairing of a set (for the left-hand side) and a multiset (for the right-hand side). This discussion suggests the following representation of sequents in these three systems. Let $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ be two meta-level predicates, both of type $bool \to o$. These predicates are used to identify which object-level formulas appear on which side of the sequent arrow, and the ? modal is used to mark the formulas to which weakening and contraction can be applied.

We will identify three schemes for encoding sequents. The *linear scheme* encodes the (object-level) sequent $B_1, \ldots, B_n \longrightarrow C_1, \ldots, C_m$ $(n, m \geq 0)$ by the meta-level formula $\lfloor B_1 \rfloor \parr \cdots \parr \lfloor B_n \rfloor \parr \lceil C_1 \rceil \parr \cdots \parr \lceil C_m \rceil$ or by the Forum sequent

$$\cdot; \cdot \longrightarrow \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor, \lceil C_1 \rceil, \ldots, \lceil C_m \rceil; \cdot.$$

The *intuitionistic scheme* encodes $B_1, \ldots, B_n \longrightarrow C_1, \ldots, C_m$, where $n, m \geq 0$, with the meta-level formula $?\lfloor B_1 \rfloor \parr \cdots \parr ?\lfloor B_n \rfloor \parr \lceil C_1 \rceil \parr \cdots \parr \lceil C_m \rceil$ or by the Forum sequent

$$\cdot; \cdot \longrightarrow \lceil C_1 \rceil, \ldots, \lceil C_m \rceil; \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor.$$

Often intuitionistic sequents are additionally restricted to having one formula on the right. Finally, the *classical scheme* encodes the sequent $B_1, \ldots, B_n \longrightarrow C_1, \ldots, C_m$ $(n, m \geq 0)$ as the meta-level formula

$$?\lfloor B_1 \rfloor \parr \cdots \parr ?\lfloor B_n \rfloor \parr ?\lceil C_1 \rceil \parr \cdots \parr ?\lceil C_m \rceil$$

or by the Forum sequent

$$\cdot; \cdot \longrightarrow \cdot; \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor, \lceil C_1 \rceil, \ldots, \lceil C_m \rceil.$$

3.2 Encoding additive and multiplicative inference rules

We first illustrate how to encode object-level inference rules using the linear scheme.

Consider the specification of the logical inference rules for object-level conjunction, represented here as the infix constant $\wedge$ of type $bool \to bool \to bool$. Consider the additive inference rules for this connective.

$$\frac{\Delta, A \longrightarrow \Gamma}{\Delta, A \wedge B \longrightarrow \Gamma} \wedge L_1 \qquad \frac{\Delta, B \longrightarrow \Gamma}{\Delta, A \wedge B \longrightarrow \Gamma} \wedge L_2 \qquad \frac{\Delta \longrightarrow \Gamma, A \quad \Delta \longrightarrow \Gamma, B}{\Delta \longrightarrow \Gamma, A \wedge B} \wedge \mathrm{R}$$

These three inference rules can be specified in Forum using the clauses

$$(\wedge L_1) \quad \lfloor A \wedge B \rfloor \circ\!\!- \lfloor A \rfloor. \qquad (\wedge R) \quad \lceil A \wedge B \rceil \circ\!\!- \lceil A \rceil \,\&\, \lceil B \rceil.$$
$$(\wedge L_2) \quad \lfloor A \wedge B \rfloor \circ\!\!- \lfloor B \rfloor.$$

We shall assume that a formula displayed in this manner actually denote the formula you get when you add the modal ! to the universal closure of the formula displayed. We use the convention that capital letters will generally serve as variables.

Notice that the two clauses for left introduction can be written in the uncurried for as

$$\lfloor A \wedge B \rfloor \circ\!\!- \lfloor A \rfloor \oplus \lfloor B \rfloor.$$

Thus, these additive rules make use of two (dual) meta-level additive connectives: & and $\oplus$. Similarly, the following two clauses encode the multiplicative version of conjunction introduction rules:

$$(\wedge L) \quad \lfloor A \wedge B \rfloor \circ\!\!- \lfloor A \rfloor \parr \lfloor B \rfloor. \qquad (\wedge R) \quad \lceil A \wedge B \rceil \circ\!\!- \lceil A \rceil \circ\!\!- \lceil B \rceil.$$

The equivalent, uncurried form of the right introduction is

$$\lceil A \wedge B \rceil \circ\!\!- \lceil A \rceil \otimes \lceil B \rceil.$$

Thus, these multiplicative rules make use of two (dual) meta-level multiplicative connectives: $\otimes$ and $\parr$.

When using either the classical or intuitionistic (hybrid) encoding of an object-level sequent, inference rules can place occurrences of the ? modality where needed with the body of clauses. For example, the additive version of the $(\wedge R)$ rule for a classical sequent is encoded as

$$\lceil A \wedge B \rceil \circ\!\!- ?\lceil A \rceil \,\&\, ?\lceil B \rceil.$$

For additional examples, see Section 3.5.

3.3 Encoding quantifier introduction rules

Using the quantification of higher-order types that is available in Forum, it is a simple matter to encode the inference rules for object-level quantifiers. For example, if we use the linear scheme for representing sequents, then the left and right introduction rules for object-level universal quantifier can be written as

$$(\forall L) \quad \lfloor \forall B \rfloor \circ\!\!- \lfloor Bx \rfloor. \qquad (\forall R) \quad \lceil \forall B \rceil \circ\!\!- \forall x \lceil Bx \rceil.$$

Here, the symbol $\forall$ is used for both meta-level and object-level quantification: at the object-level $\forall$ has the type $(i \to \mathit{bool}) \to \mathit{bool}$. Thus the variable B above has the type $i \to \mathit{bool}$. Consider the Forum sequent $\Psi; \cdot \longrightarrow \lceil \forall B \rceil, \Theta; \cdot$ where Ψ contains the above two clauses. Using *decide*! with the clause for $(\forall R)$ would cause the search for a proof of the above sequent to be reduced to the

search for a proof of the sequent $\Psi; \cdot \longrightarrow \lceil By \rceil, \Theta; \cdot$ where y is new. Here, the meta-level eigen-variable y also serves the role of an object-level eigen-variable. Dually, consider the Forum sequent $\Psi; \cdot \longrightarrow \lfloor \forall B \rfloor, \Theta; \cdot$. Using the *decide*! with the clause for $(\forall L)$ would cause proof search to reduce this sequent to the sequent $\Psi; \cdot \longrightarrow \lfloor Bt \rfloor, \Theta; \cdot$ where t is a term of type i. If we restrict appropriately the use of the type i, then terms of type i can be identified with object-level terms.

Notice that the clause for $(\forall L)$ is logically equivalent to the formula

$$\lfloor \forall B \rfloor \multimap \exists x \lfloor Bx \rfloor.$$

Thus, these quantifier rules make use of two (dual) meta-level quantifiers.

3.4 The initial and cut rules

Up to this point, all the Forum clauses used to specify an inference figure have been such that the head of the clause has been an atom. Clauses specifying the cut and initial rules will have rather different structure. In particular, the initial rule which asserts that the sequent $B \longrightarrow B$ is provable, can be represented simply by the following *initial clause*:

$$(\mathit{Initial}) \qquad \lfloor B \rfloor \parr \lceil B \rceil.$$

Notice that this clause has a head with two atoms and no body.

There appear to be several possible ways to encode the cut rule. As a proof rule, cut is given as

$$\frac{\Delta_1 \longrightarrow \Gamma_1, B \qquad \Delta_2, B \longrightarrow \Gamma_2}{\Delta_1, \Delta_2 \longrightarrow \Gamma_1, \Gamma_2}\ \mathit{Cut}$$

Depending on how structural rules are used in this encoding, this cut rule can be specified as one of the following clauses:

$$\begin{array}{llll} (\mathit{Cut}) & \forall B(\lceil B \rceil \multimap \lfloor B \rfloor \multimap \bot) & (\mathit{Cut}_1) & \forall B(?\lceil B \rceil \multimap \lfloor B \rfloor \multimap \bot) \\ (\mathit{Cut}_2) & \forall B(\lceil B \rceil \multimap ?\lfloor B \rfloor \multimap \bot) & (\mathit{Cut}_3) & \forall B(?\lceil B \rceil \multimap ?\lfloor B \rfloor \multimap \bot) \end{array}$$

Dual to the initial rule, these clauses have an empty head and two bodies. Other variations on the cut rule also seem possible: namely, one or both of the $\multimap$ can be replaced with $\Rightarrow$. The four displayed possibilities for the cut rule, however, entail these other variations, for example:

$$?\lceil B \rceil \multimap \lfloor B \rfloor \multimap \bot \quad \vdash \quad ?\lceil B \rceil \Rightarrow \lfloor B \rfloor \multimap \bot\ .$$

As a result, we shall not consider these variations any further here.

Notice that the *Initial* and *Cut* clauses together proves that $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are duals of each other: that is, $\forall B(\lfloor B \rfloor \multimap \lceil B \rceil)$ and $\forall B(\lceil B \rceil \multimap \lfloor B \rfloor)$ are proved from these two formulas.

3.5 Example Forum specifications

Consider a presentation of intuitionistic logic using the logical connective $\supset$, $\cap$, $\cup$, $\forall_i$, $\exists_i$, f_i, and t_i. The usual LJ proof system of Gentzen [Gen69] can be encoded as follows: rules for intuitionistic logic LJ and a cut rule (taken from [MP]).

$$
\begin{array}{llll}
(\supset L) & \lfloor A \supset B \rfloor \circ\!\!- \lceil A \rceil \circ\!\!- ?\lfloor B \rfloor. & (\supset R) & \lceil A \supset B \rceil \circ\!\!- ?\lfloor A \rfloor \parr \lceil B \rceil. \\
(\cap L_1) & \lfloor A \cap B \rfloor \circ\!\!- ?\lfloor A \rfloor. & (\cap R) & \lceil A \cap B \rceil \circ\!\!- \lceil A \rceil \,\&\, \lceil B \rceil. \\
(\cap L_2) & \lfloor A \cap B \rfloor \circ\!\!- ?\lfloor B \rfloor. & (\cup R_1) & \lceil A \cup B \rceil \circ\!\!- \lceil A \rceil. \\
(\cup L) & \lfloor A \cup B \rfloor \circ\!\!- ?\lfloor A \rfloor \,\&\, ?\lfloor B \rfloor. & (\cup R_2) & \lceil A \cup B \rceil \circ\!\!- \lceil B \rceil. \\
(\forall_i L) & \lfloor \forall_i B \rfloor \circ\!\!- ?\lfloor Bx \rfloor. & (\forall_i R) & \lceil \forall_i B \rceil \circ\!\!- \forall x \lceil Bx \rceil. \\
(\exists_i L) & \lfloor \exists_i B \rfloor \circ\!\!- \forall x\, ?\lfloor Bx \rfloor. & (\exists_i R) & \lceil \exists_i B \rceil \circ\!\!- \lceil Bx \rceil. \\
(f_i L) & \lfloor f_i \rfloor \circ\!\!- \top. & (t_i R) & \lceil t_i \rceil \circ\!\!- \top. \\
(Cut) & \bot \circ\!\!- ?\lfloor B \rfloor \circ\!\!- \lceil B \rceil. & (Initial) & \lfloor B \rfloor \parr \lceil B \rceil.
\end{array}
$$

Theorems that state that one's encoding of a proof system matches the original proof system are often called adequacy theorems. The following such theorem is easily proved by induction of the structure of proofs.

Adequacy Theorem The sequent $B_1, \ldots, B_n \longrightarrow B_0$ has an LJ-proof [Gen69] if and only if the sequent $LJ; \cdot \longrightarrow \lceil B_0 \rceil; \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor$ has a Forum proof. The sequent $B_1, \ldots, B_n \longrightarrow$ has an LJ-proof if and only if $LJ; \cdot \longrightarrow \cdot; \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor$ has a Forum proof ($n \geq 0$).

A number of other proof systems have been specified in Forum using this particular style of encoding. For example, Gentzen's LK and LJ [Gen69], linear logic, LKQ and LKT [DJS95], an optimization of LJ [LSS93,Dyc92], and Girard's LU [Gir93].

3.6 Advantages of such encodings

The encoding of an object-level proof system as Forum clauses has certain advantages over encoding them as inference figures. For example, the Forum specifications do not deal with context explicitly and instead they focus on the formulas that are directly involved in the inference rule. The distinction between making the inference rule additive or multiplicative is achieved in inference rule figures by explicitly presenting contexts and either splitting or copying them. The Forum clause representation achieves the same distinction using meta-level additive or multiplicative connectives. Object-level quantifiers can be handled directly using the meta-level quantification. Similarly, the structural rules of contraction and thinning can be captured together using the ? modal.

Since the encoding of proof systems is natural and direct, we might hope to be able to use the rich meta-theory of linear logic to help in drawing conclusions about object-level proof systems. An example of this kind of meta-level reason is given in [Mil96] where it is shown how a sequent calculus presentation of intuitionistic logic can be transformed into a natural deduction presentation by simple linear logic equivalences.

Since the encodings of object-level encodings result in logic programs (in the sense of Forum) and since there is significant knowledge and tools available to provide automatic and interactive tools to compute with those logic programs, encodings such as those described here can be important for the automation of various proof systems. In this paper, we explore automation of questions such as: Does one object-level sequent follow from the encoding of a proof system? Does one proof system's encoding entail another proof system's encoding? Can cut elimination be proved for the encoded logic? The last question has also been discussed by Avron and Lev [AL01] but their setting is limited the specification of propositional logics based on classical and additive maintenance of context.

There are, of course, some disadvantages to using linear logic as a meta-theory, the principle one being that it will not be possible to capture proof systems requiring non-commutativity. As we shall see, however, significant and interesting proof systems can be encoded into linear logic and for these systems, broad avenues of meta-level reasoning and automation should be available.

4 Entailments between introduction rules

We now address the problem of how easy it is to prove that the encoding of some inference rules imply the encoding of some other inference rules. For this purpose, we need to make definitions that restrict flat Forum formulas further so that they encode object-level inference rules. We shall assume that we have fixed a set $\mathcal{Q}$ of unary meta-level predicates all of type $bool \to o$. Object-level logical constants will also be assumed to be fixed. These constants will have types of order 0, 1, or 2 and all will build terms of type $bool$. Examples of object-level constants at various orders are: order 0, true and false; order 1, conjunction and implication; and order 2, universal and existential quantifiers. We shall also assume that object-level quantification is first-order and over one domain, denoted at the meta-level by i.

Definition 1. *An* introduction clause *is a closed flat formula of the form*

$$\forall x_1 \ldots \forall x_n [q(\diamond(x_1, \ldots, x_n)) \hookleftarrow B_1 \hookleftarrow B_2 \hookleftarrow \ldots \hookleftarrow B_m],$$

where $n, m \geq 0$, $\diamond$ is an object-level connective of arity n ($n \geq 0$), and q is a meta-level predicate. Furthermore, an atom occurring in a body of this clause is either of the form $p(x_i)$ or $p(x_i(y))$ where p is a meta-level predicate and $1 \leq i \leq n$. In the first case, x_i has a type of order 0 while in the second case x_i has a type of order 1 and y is a variable quantified (universally or existentially) in a body of this clause (in particular, y is not in $\{x_1, \ldots, x_n\}$).

Notice that all the encodings of inference rules we have presented so far are examples of *introduction clauses*: the predicates $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are examples of meta-level predicates. Encodings of inference rules are also allowed to have other predicates defined for, say, side conditions, as long as they can be described using such clauses.

Definition 2. *A* premise atom *is an atomic formula of the form $q(t)$, where q is a meta-level predicate and t is a term of type bool with a variable as its head symbol. A* conclusion atom *is an atomic formula of the form $q(\diamond(x_1, \ldots, x_n))$, where q is a meta-level predicate, $\diamond$ is an object-level connective of arity n, and $x_1, \ldots, x_n$ is a list of variables.*

Definition 3. *Let Π be a Forum proof of the sequent $\Psi; \Delta \longrightarrow \Gamma; \Upsilon$, where Ψ is a set of flat clauses that are either initial, one of the cut clauses, an introduction clause, or a flat goal, Δ and Γ are multisets of flat goals, and Υ is a set of atoms. The depth of a Forum proof Π is defined as the maximum number of occurrences of the rules decide or decide! on a branch of Π.*

Notice that if Π is a proof of a sequent of the form $\Psi; \Delta \longrightarrow \Gamma; \Upsilon$, following the restrictions of the definition above, then all sequents in Π are such that the left classical context is equal to Ψ, the right classical context is always a set of atoms, and the two linear contexts are always multisets of flat goals.

It is also a simple matter to see that it is possible to search through all Forum proofs of such sequents which are bounded in depth. The Forum proof system is designed so that the only essential choices that need to be made are those involved with the decide, decide!, and decide? inference rules.

The following lemma can be used to build a decision procedure for certain kinds of inferences between introduction clauses.

Lemma 2. *Let Ψ be a set containing introduction clauses and possibly the initial clause. Let C be an introduction clause. If the sequent $\Psi; \cdot \longrightarrow C; \cdot$ has a proof, it has a proof of depth 3 or less.*

Proof We first argue that a sequent of the form

$$\Psi, \Delta_1; \Delta_2 \longrightarrow \mathcal{A}_1; \mathcal{A}_2, \qquad (*)$$

where Δ_1 is a set and Δ_2 is a multiset of flat goals over premise atoms and $\mathcal{A}_1$ is a multiset and $\mathcal{A}_2$ is a multiset of premise atoms, is provable if and only if it is provable with a proof of depth 2 or less. This sequent can only be proved using decide or decide!. Let B be the formula that is selected in one of these decide rules. If B is the initial clause, then this proof must have depth 1. Clearly, B is not an introduction clause from Ψ since eventually an instance of the head of that clause must be in either $\mathcal{A}_1$ or in $\mathcal{A}_2$, which is impossible since the instance of a conclusion atom cannot be a premise atom. Thus, B must be from either Δ_1 or Δ_2. A simple argument by structural induction of flat goals shows that in either of these cases, the depth of a proof is limited by 2. In particular, consider the case when B is $?\,A$, for some premise atom A. Then Δ_2 and $\mathcal{A}_1$ must be empty and the proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Psi, \Delta_1; \cdot \xrightarrow{A} \cdot; \mathcal{A}_2}\ \textit{initial}?}{\Psi, \Delta_1; A \longrightarrow \cdot; \mathcal{A}_2}\ \textit{decide}}{\Psi, \Delta_1; \cdot \xrightarrow{?\,A} \cdot; \mathcal{A}_2}\ ?\,L$$

Observe that the only rule that can be applied to the middle sequent is decide, since the left linear context is not empty. Also, note that since A is atomic, it must be in $\mathcal{A}_2$ and the last rule on the above proof has to be initial?. The depth in this case is exactly 2. The other cases of depending on the structure of B are similar and simpler.

Now consider a provable sequent of the form

$$\Psi, \Delta_1; \Delta_2 \longrightarrow A; \cdot, \qquad (**)$$

where Δ_1 is a set and Δ_2 is a multiset of flat goals over premise atoms and A is a conclusion atom. This sequent can be proved only using decide! with an introduction clause from Ψ since the atoms in Δ_1 and Δ_2 are premise atoms. The result of completing the backchaining leaves possibly several sequents that need to be proved, but all of these are of form $(*)$ above. Thus, sequents of the form $(**)$ can be proved in height 3 or less.

Finally, a sequent of the form $\Psi; \cdot \longrightarrow C; \cdot$ has a proof if and only if the right-rules for Forum reduce it to sequents of the form $(**)$ above. Thus, such a sequent is provable if and only if it has a proof of depth 3 or less. ∎

Lemma 2 shows that deciding whether or not one inference rule is derivable from other inference rules is rather simple: if such a derivation is possible, a very shallow proof witnesses that fact.

5 Canonical and coherent proof systems

In the inference systems we shall consider, the set of meta-level predicates $\mathcal{Q}$ is exactly the set $\{\lfloor\cdot\rfloor, \lceil\cdot\rceil\}$. In Section 7, we consider the LU proof system of Girard [Gir93] and there we will use additional meta-level predicates.

Definition 4. *Fix $\mathcal{Q}$ to be the set $\{\lfloor\cdot\rfloor, \lceil\cdot\rceil\}$. A* canonical proof system *is a set $\mathcal{P}$ of flat Forum clauses such that (i) the initial clause is a member of $\mathcal{P}$, (ii) exactly one cut clause is a member of $\mathcal{P}$, and (iii) all other clauses in $\mathcal{P}$ are introduction clauses with the additional restriction that, for every pair of atoms of the form $\lfloor T\rfloor$ and $\lceil S\rceil$ in a body, the head variable of T differs from head variable of S. A formula that statisfies condition (iii) is also called a* canonical clause.

Definition 5. *Consider a canonical proof system $\mathcal{P}$ and an object-level connective, say, $\diamond$ of arity $n \geq 0$. Consider all the (uncurried) formulas in $\mathcal{P}$ that specify a left-introduction rule for $\diamond$. These would be of the form*

$$\forall\bar{x}(\lfloor\diamond(x_1, \ldots, x_i)\rfloor \circ\!\!- L_1) \quad \cdots \quad \forall\bar{x}(\lfloor\diamond(x_1, \ldots, x_i)\rfloor \circ\!\!- L_p) \quad (p \geq 0)$$

Similarly, consider all the (uncurried) formulas in $\mathcal{P}$ that specify a right-introduction rule for $\diamond$. These would be of the form

$$\forall\bar{x}(\lceil\diamond(x_1, \ldots, x_i)\rceil \circ\!\!- R_1) \quad \cdots \quad \forall\bar{x}(\lceil\diamond(x_1, \ldots, x_i)\rceil \circ\!\!- R_q) \qquad (q \geq 0)$$

All of these $p+q$ displayed formulas can be replaced by the following two clauses

$$\forall\bar{x}(\lfloor\diamond(x_1,\ldots,x_i)\rfloor \circ\!\!- L_1 \oplus \cdots \oplus L_p) \text{ and } \forall\bar{x}(\lceil\diamond(x_1,\ldots,x_i)\rceil \circ\!\!- R_1 \oplus \cdots \oplus R_q)$$

(An empty $\oplus$ is written as the linear logic additive false 0.) We shall say that these last two formulas represent the introduction rules for $\diamond$ in their definition form. *While these formulas are not generally formulas of Forum, they are equivalent to the $p+q$ Forum formulas.*

Definition 6. *Consider a canonical proof system $\mathcal{P}$ and an object-level connective, say, $\diamond$ of arity $n \geq 0$. Let the formulas*

$$\forall\bar{x}(\lfloor\diamond(x_1,\ldots,x_n)\rfloor \circ\!\!- B_l) \quad \textit{and} \quad \forall\bar{x}(\lceil\diamond(x_1,\ldots,x_n)\rceil \circ\!\!- B_r)$$

be the definition form for the left and right introduction rules. Let C be the cut clause that appears in $\mathcal{P}$. The object-level connective $\diamond$ has dual left and right introduction rules *if $!\,C \vdash \forall\bar{x}(B_l \multimap B_r \multimap \bot)$ in linear logic.*

Definition 7. *A canonical system is called* coherent *if the left and right introduction rules for each object-level connective are duals.*

Example 1. Consider the specification of LJ in Section 3.5. The introduction rules for $\cap$, for example, in definition form are the two formulas

$$\forall A\forall B(\lfloor A \cap B\rfloor \circ\!\!- ?\lfloor A\rfloor \oplus ?\lfloor B\rfloor) \quad \text{and} \quad \forall A\forall B(\lceil A \cap B\rceil \circ\!\!- \lceil A\rceil \,\&\, \lceil B\rceil).$$

The definition form for the introduction rules for the other logical connectives can be computed easily. In the end, to determine that the LJ specification is coherent, the following must be proved:

$$\begin{array}{ll}
(\supset) & !\,Cut_2 \vdash \forall A\forall B[(?\lfloor A\rfloor \oplus ?\lceil B\rceil) \multimap (\lceil A\rceil \,\&\, \lceil B\rceil) \multimap \bot] \\
(\cap) & !\,Cut_2 \vdash \forall A\forall B[(\lceil A\rceil \otimes ?\lfloor B\rfloor) \multimap (?\lfloor A\rfloor \parr \lceil B\rceil) \multimap \bot] \\
(\cup) & !\,Cut_2 \vdash \forall A\forall B[(?\lfloor A\rfloor \,\&\, ?\lfloor B\rfloor) \multimap (\lceil A\rceil \oplus \lceil B\rceil) \multimap \bot] \\
(\forall_i) & !\,Cut_2 \vdash \forall B[\exists x(?\lfloor Bx\rfloor) \multimap \forall x\lceil Bx\rceil \multimap \bot] \\
(\exists_i) & !\,Cut_2 \vdash \forall B[\forall x(?\lfloor Bx\rfloor) \multimap \exists x\lceil Bx\rceil \multimap \bot] \\
(t_i) & !\,Cut_2 \vdash 0 \multimap \top \multimap \bot \\
(f_i) & !\,Cut_2 \vdash \top \multimap 0 \multimap \bot
\end{array}$$

All of these sequents have simple Forum proofs.

Definition 8. *A Forum proof is said to encode* an object-level cut-free proof *if no occurrence of the decide*! *inference rule is used on a cut clause.*

Definition 9. *The* degree $d(B)$ *of an object-level formula B is the number of occurrences of object-level logical connectives in B. Thus, $d(A) = 0$ if and only if A is atomic. Logical constants of arity 0 have degree 1. If a cut-clause is used in a decide*! *rule in a Forum proof, then the* degree of that occurrence of decide! *is the degree of the object-level formula used to instantiate that occurrence of the cut clause. The degree of a Forum proof Π, written as $d(\Pi)$, is the multiset of the degree of all occurrences of decide*! *in the proof. Thus, a Forum proof Π encodes a cut-free object-level proof if and only if $d(\Pi)$ is the empty multiset.*

We now show that, for coherent systems, it is possible to exchange a Forum proof by another one with a smaller degree: here we use the multiset well-ordering [DM79] induced by the ordering on non-negative integers.

Lemma 3. *Let $\mathcal{P}$ be a coherent system, Ψ be a set, Δ be a multiset of flat goals which contain no occurrences of object-level logical constants, and Γ be a multiset and Υ be a set of atomic formulas. If there is a Forum proof Π of the sequent $\mathcal{P}, \Psi; \Delta \longrightarrow \Gamma; \Upsilon$ such that $d(\Pi)$ contains a positive integer, then there is a proof of the same sequent with smaller multiset order.*

Proof We shall first assume that the cut clause in $\mathcal{P}$ is the clause without the ? modal. Let Π be a Forum proof for $\mathcal{P}, \Psi; \Delta \longrightarrow \Gamma; \Upsilon$ such that $d(\Pi)$ contains a positive integer. There is thus a subproof of Π of the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1}{\mathcal{P}, \Psi; \Delta_1 \longrightarrow \lfloor D \rfloor, \Gamma_1; \Upsilon'} \quad \dfrac{\Pi_2}{\mathcal{P}, \Psi; \Delta_2 \longrightarrow \lceil D \rceil, \Gamma_2; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_1, \Delta_2 \xrightarrow{\lceil D \rceil \multimap \lfloor D \rfloor \multimap \bot} \Gamma_1, \Gamma_2; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_1, \Delta_2 \xrightarrow{\forall B (\lceil B \rceil \multimap \lfloor B \rfloor \multimap \bot)} \Gamma_1, \Gamma_2; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_1, \Delta_2 \longrightarrow \Gamma_1, \Gamma_2; \Upsilon'}$$

where $D = \diamond(D_1, \ldots, D_n)$ for some object-level connective $\diamond$, $n \geq 0$, and $\Upsilon \subseteq \Upsilon'$.

We may assume that the occurrences of $\lfloor D \rfloor$ in Π_1 and of $\lceil D \rceil$ in Π_2 are *principal formulas* in their respective proofs: that is, these proofs end with a decide or decide! rule and these atoms are the ones rewritten by the backchaining step. If this is not the case, the decide or decide! rule together with the cut clause can be permuted upward in the Forum proof.

We can distinguish three cases. In one case, Π_1 ends in a decide of an initial clause: thus Γ_1 is $\lceil D \rceil$, Δ_1 is empty, the displayed proof fragment above can be replaced by Π_2, and the degree of the resulting proof decreases. In another case, Π_2 ends in a decide of an initial clause, Γ_2 is $\lfloor D \rfloor$, Δ_2 is empty, the displayed proof fragment above can be replaced by Π_1, and the degree of the resulting proof decreases. The only other case is that Π_1 and Π_2 end in a decide or decide! rule selecting some formula from $\mathcal{P}$, Ψ, and Δ_1 or Δ_2, respectively. Since D must contain an object level logical constant and since the formulas in Ψ, Δ_1, and Δ_2 do not contain such constants, the only possible selections are of formulas from $\mathcal{P}$. Thus, there are clauses

$$\forall \bar{x}. \lfloor \diamond(x_1, \ldots, x_i) \rfloor \circ\!\!- B_l \quad \text{and} \quad \forall \bar{x}. \lceil \diamond(x_1, \ldots, x_i) \rceil \circ\!\!- B_r$$

in $\mathcal{P}$ such that Π_1 and Π_2 are the following two proofs:

$$\dfrac{\dfrac{\Pi_1'}{\mathcal{P}, \Psi; \Delta_1 \longrightarrow \theta B_l, \Gamma_1; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_1 \longrightarrow \lfloor D \rfloor, \Gamma_1; \Upsilon'} \qquad \dfrac{\dfrac{\Pi_2'}{\mathcal{P}, \Psi; \Delta_2 \longrightarrow \theta B_r, \Gamma_2; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_2 \longrightarrow \lceil D \rceil, \Gamma_2; \Upsilon'}$$

where θ is the appropriate substitution for the variables in $\bar{x}$. (Here we have assumed that the clauses in $\mathcal{P}$ are written in the logically equivalent uncurried

form). Finally, since $\mathcal{P}$ is a coherent system, we know that $Cut; \theta B_l, \theta B_r \longrightarrow \bot; \cdot$ is provable.

Using the soundness and completeness theorem for Forum (Theorem 1) the three sequents

$$!\,\mathcal{P}, !\,\Psi, \Delta_1 \vdash \theta B_l, \Gamma_1, ?\,\Upsilon' \qquad !\,\mathcal{P}, !\,\Psi, \Delta_2 \vdash \theta B_r, \Gamma_2, ?\,\Upsilon' \qquad !\,Cut, \theta B_l, \theta B_r \vdash \bot$$

are provable in linear logic. Using cut twice, we can then conclude that

$$!\,\mathcal{P}, !\,\Psi, \Delta_1, \Delta_2 \vdash \Gamma_1, \Gamma_2\, ?\,\Upsilon'$$

is provable in linear logic (remember that $Cut \in \mathcal{P}$) and by cut-elimination in linear logic and Theorem 1, we have that

$$\mathcal{P}, \Psi; \Delta_1, \Delta_2 \longrightarrow \Gamma_1, \Gamma_2; \Upsilon'$$

has a Forum proof. The process of translating to and from linear logic and using cut-elimination in linear logic will not change the degree of the Forum proof except to replace the one selected occurrence with possibly several smaller uses of decide! with Cut in the proof of $!\,Cut, \theta B_l, \theta B_r \vdash \bot$. As a result, the degree of the overall proof has reduced.

If the cut clause in $\mathcal{P}$ has the modal ?, the result follows with a slightly different proof. ∎

As an immediate corollary of this lemma, if the sequent $\mathcal{P}, \Psi; \Delta \longrightarrow \Gamma; \Upsilon$ (assuming the restrictions of this lemma) has a proof with a degree containing a positive integer, that sequent has a proof with a degree that contains at most zeros. That is, this lemma shows how to reduce object-level cuts to only object-level atomic cuts. The following result shows that, in fact, these cuts can be removed.

Lemma 4. *Let C be a canonical clause that encodes an introduction rule and let $\mathcal{P}$ be a coherent system. If $!\,\mathcal{P} \vdash C$ in linear logic then there is an object-level cut-free Forum proof of $\mathcal{P}; \cdot \longrightarrow C; \cdot$.*

Proof Given that $!\,\mathcal{P} \vdash C$ in linear logic, there is a Forum proof of the sequent $\mathcal{P}; \cdot \longrightarrow C; \cdot$. Applying right introduction rules to this sequent forces the sequent $\mathcal{P}, \Psi; \Delta \longrightarrow A; \cdot$, where Ψ is a set and Δ is a multiset of premise atoms and A is a conclusion atom, to have a Forum proof Π. Given that all clauses in Ψ are flat clauses, a simple induction show that every sequent occurring in Π is of the form $\mathcal{P}, \Psi; \Delta' \stackrel{\{B\}}{\longrightarrow} \Gamma; \Upsilon$ where $\Delta' \subseteq \Delta$, Γ is a multiset of flat goals, Υ is a set of atomic formulas and $\{B\}$ indicates that a flat clause labeling the arrow might be present.

Using Lemma 3, we can conclude that Π contains only atomic object level cuts. We now argue that these cuts can be removed. If Π is not object-level cut-free, there is a subproof of Π of the form

$$\dfrac{\dfrac{\mathcal{P}, \Psi; \Delta_1 \stackrel{\Pi_1}{\longrightarrow} \lfloor D \rfloor, \Gamma_1; \Upsilon' \quad \mathcal{P}, \Psi; \Delta_2 \stackrel{\Pi_2}{\longrightarrow} \lceil D \rceil, \Gamma_2; \Upsilon'}{\mathcal{P}, \Psi; \Delta_1, \Delta_2 \stackrel{\forall B(\lceil B \rceil \multimap \lfloor B \rfloor \multimap \bot)}{\longrightarrow} \Gamma_1, \Gamma_2; \Upsilon'}}{\mathcal{P}, \Psi; \Delta_1, \Delta_2 \longrightarrow \Gamma_1, \Gamma_2; \Upsilon'}\ \textit{decide}!$$

where Π_1 and Π_2 are object-level cut-free Forum proofs, $\Upsilon \subseteq \Upsilon'$, and D is an object-level atomic formula.

We can distinguish three possibilities for the last inference rules of Π_1 and Π_2. If Π_1 ends a decide! with the initial clause then Δ_1 is empty and Γ_1 is the multiset set containing just $\lceil D \rceil$. In that case, the entire displayed proof can be replaced by Π_2, which removes one decide! on a cut-clause. Similarly, if Π_2 ends a decide! with the initial clause, then the displayed proof can be replaced by Π_1. Finally, the last inference rules of Π_1 and Π_2 could be (meta-level) left-introduction rules: in Π_1 a formula could be selected from Ψ or Δ_1 for backchaining and in Π_2 a formula could be selected from Ψ or Δ_2 for backchaining. In these cases, the formula $\lfloor D \rfloor$ would be a subformula of a formula in Ψ or Δ_1 and $\lceil D \rceil$ would be a subformula of a formula in Ψ or Δ_2. Then both $\lfloor D \rfloor$ and $\lceil D \rceil$ occur in bodies of the clause C, something that is explicitly ruled out by the definition of canonical proof systems (Definition 4).

For the case where another cut rule is present in $\mathcal{P}$, the analysis is the same except that either one or both of $\lfloor D \rfloor$ and $\lceil D \rceil$ could be in the right classical context and references to the Forum rule initial might need to be initial?. ∎

This result is important since it provides a way of controlling the use of cut clauses during proof search. In general, it is desirable to control meta-level proof search when clauses with empty head are available. Using the decide inference rule with such clauses can produce redundant steps in a proof, in a possibly endless process. The same behavior can be observed at the object-level with the use of a cut rule.

Next, we describe a decision procedure for determining if a proof system is derivable from another.

Theorem 2. *Let $\mathcal{P}$ be a coherent proof system. Let Ψ be a set of canonical clauses together with a cut clause and the initial clause and let P be the formula $!C_1 \& \ldots \& !C_m$ where $\Psi = \{C_1, \ldots, C_m\}$. If there is a proof in Forum of $\mathcal{P}; \cdot \longrightarrow P; \cdot$ then it has a proof of depth less than or equal to 3.*

Proof Clearly, $\mathcal{P}; \cdot \longrightarrow P; \cdot$ is provable with depth less than or equal to 3 if and only if for all $C \in \Psi$, the Forum sequent $\mathcal{P}; \cdot \longrightarrow C; \cdot$ is provable with depth less than or equal to 3. Thus, we only need to prove this depth restriction for $\mathcal{P}; \cdot \longrightarrow C; \cdot$ for $C \in \Psi$.

The case where C is the initial clause is trivial since $C \in \mathcal{P}$. In the case that C is a cut rule, the proof of $\mathcal{P}; \cdot \longrightarrow C; \cdot$ must look like

$$\dfrac{\begin{array}{c}\Pi \\ \mathcal{P}; \lfloor B \rfloor, \lceil B \rceil \longrightarrow \cdot; \cdot\end{array}}{\mathcal{P}; \cdot \longrightarrow \forall B. \lfloor B \rfloor \multimap \lceil B \rceil \multimap \bot; \cdot}$$

where B is an eigen-variable of the proof. Using Lemma 3, we may assume that the rest of this proof contains only object-level atomic cuts. Furthermore, the last inference rule of Π must be decide! using a cut rule. Thus, Π must be of

the form

$$\dfrac{\dfrac{\dfrac{}{\mathcal{P};\cdot \xrightarrow{\lfloor B\rfloor} \lfloor D\rfloor;\cdot}\ \textit{initial}}{\mathcal{P};\lfloor B\rfloor \longrightarrow \lfloor D\rfloor;\cdot}\ \textit{decide} \quad \dfrac{\dfrac{}{\mathcal{P};\cdot \xrightarrow{\lceil B\rceil} \lceil D\rceil;\cdot}\ \textit{initial}}{\mathcal{P};\lceil B\rceil \longrightarrow \lceil D\rceil;\cdot}\ \textit{decide}}{\mathcal{P};\lfloor B\rfloor,\lceil B\rceil \xrightarrow{\forall B(\lceil B\rceil \multimap \lfloor B\rfloor \multimap \bot)} \cdot;\cdot}$$

where D is some object-level atomic formula, which must be B. In this case, the depth is 2. If another cut rule with occurrences of ? is present in $\mathcal{P}$, the depth is 3.

The remaining case is where C encodes an introduction rule. Then it follows from Lemma 3 that the cut rule of $\mathcal{P}$ is not used. Lemmas 2 then provides us with our conclusion: if the cut rule is removed from $\mathcal{P}$, we are left with only the initial rule and introduction clauses. ∎

Notice that if the encoded proof system $\mathcal{P}$ entails the encoded system Ψ (as describe in the above theorem), then a simple consequence of cut-elimination at the meta-level is that whenever $\mathcal{P}$ proves a object-level sequent, the proof system Ψ also proves that same sequent.

6 Cut-elimination for coherent systems

The cut-elimination theorem for a particular logic can often be divided into two parts. The first part shows that a cut involving a non-atomic formula can be replaced by possibly multiple cuts involving subformulas of the original cut formula. This part of the proof works because left and right introduction rules for each logical connective are duals (formalized here in Definition 6). The second part of the proof argues how cuts with atomic formulas can be removed. Cut-elimination for coherent object-level proof systems is proved similarly: Lemma 3 shows that non-atomic cuts can be removed and the following theorem proves that object-level atomic cuts can also be removed.

Theorem 3. *Let $\mathcal{P}$ be a coherent system and B be an object-level formula. If $!\mathcal{P} \vdash B$ is provable (that is, if there is an object-level proof of B using the proof system encoded as $\mathcal{P}$), then there is an object-level cut-free proof of the Forum sequent $\mathcal{P};\cdot \longrightarrow \lceil B\rceil;\cdot$.*

Proof Since $\lceil B\rceil$ is an atomic meta-level formula and $\mathcal{P}$ contains only flat formulas, the left linear context is empty for all sequents in the proof Π of $\mathcal{P};\cdot \longrightarrow \lceil B\rceil;\cdot$. Also, any formula that occurs in the right classical context of any sequent in Π is atomic.

Assume that Π is not free of object-level cuts. That is, there is a subproof of Π of the form

$$\dfrac{\dfrac{\mathcal{P};\cdot \longrightarrow \lfloor D\rfloor, \Gamma_1;\Upsilon \quad \mathcal{P};\cdot \longrightarrow \lceil D\rceil, \Gamma_2;\Upsilon}{\mathcal{P};\cdot \xrightarrow{\forall B(\lfloor B\rfloor \multimap \lceil B\rceil \multimap \bot)} \Gamma_1,\Gamma_2;\Upsilon}}{\mathcal{P};\cdot \longrightarrow \Gamma_1,\Gamma_2;\Upsilon}\ \textit{decide}!$$

such that Π_1 and Π_2 are object-level cut-free, Γ_1 and Γ_2 are multisets of atomic formulas, and D is an object-level formula. As before, we may also assume that $\lfloor D \rfloor$ is a principal formula in Π_1. By Lemma 3, we may assume that D is actually an object-level atomic formula. Thus the last rule of Π_1 must be decide! over the *Initial* rule, since Π_1 is cut-free and the heads of all the other clauses in $\mathcal{P}$ (except *Initial*) have an object-level connective. Hence, Γ_1 is $\lceil D \rceil$ and the subproof displayed above can be replaced by Π_2, which is cut-free by hypothesis. Thus, we have eliminated one instance of the use of the object-level cut rule. This procedure shows that it is possible to eliminate all the top-most cuts. Continuing in this manner, we can finally arrive at a object-level cut-free proof of $\mathcal{P}; \cdot \longrightarrow \lceil B \rceil; \cdot$. ∎

To determine that a proof system satisfies cut-elimination, we can check if its encoding as flat clauses is coherent, since this guarantees cut-elimination for the encoded proof system (Theorem 3). Thus, the complete automation of proof of cut-elimination for coherent systems is given by the following result:

Theorem 4. *Determining whether or not a canonical proof system is coherent is decidable. In particular, determining if a cut clause proves the duality of the definitions of introduction rules for a given connective can be done by bounding proof search to a depth of $v + 2$ where v is the maximum number of meta-level atomic subformulas in the bodies of the introduction clauses.*

Proof Assume that we have a given coherent proof system $\mathcal{P}$. For every object-level logical constant $\diamond$, let the introduction rules be given in definition format as

$$\forall \bar{x}(\lfloor \diamond(x_1, \ldots, x_i) \rfloor \circ\!\!- B_l) \quad \text{and} \quad \forall \bar{x}(\lceil \diamond(x_1, \ldots, x_i) \rceil \circ\!\!- B_r)$$

By coherence, $Cut \vdash \forall \bar{x}(B_l \multimap B_r \multimap \bot)$ in linear logic. Let C be the curried form of $\forall \bar{x}(B_l \multimap B_r \multimap \bot)$ and consider the Forum sequent $Cut; \cdot \longrightarrow C; \cdot$. Various right rules will decompose this sequent to a set of sequents of the form $Cut, \Psi; \Delta \longrightarrow \cdot; \cdot$ where Ψ is a set and Δ is a multiset of flat goals that do not contain any occurrences of object-level logical constants. Thus, we need to describe a decision procedure for the Forum provability of such sequents.

Let Π be a proof for the sequent $Cut, \Psi; \Delta \longrightarrow \cdot; \cdot$ above mentioned. Unless $\Delta = \bot$ or $\bot \in \Psi$ and $\Delta = \emptyset$ (in which cases the proof is trivial), Π is not cut-free. We claim that every cut formula is a subformula of some formula in Ψ or Δ. In fact, no formula is added to the left-hand side of sequents in Π and the formulas on the right-hand side of sequents are introduced by decide! on the cut rule. Then Π has the form:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Cut, \Psi; \Delta_1' \longrightarrow \lceil B \rceil, \Gamma_1; \Upsilon}\,\Pi_1}}{\vdots}}{Cut, \Psi; \Delta_1 \longrightarrow \lceil B \rceil; \cdot} \cdots \quad \dfrac{\dfrac{\dfrac{\dfrac{}{Cut, \Psi; \Delta_2' \longrightarrow \lfloor B \rfloor, \Gamma_2; \Upsilon}\,\Pi_2}}{\vdots}}{Cut, \Psi; \Delta_2 \longrightarrow \lfloor B \rfloor; \cdot} \cdots}{Cut, \Psi; \Delta \xrightarrow{\forall B(\lfloor B \rfloor \multimap \lceil B \rceil \multimap \bot)} \bot; \cdot}\, \forall L, \multimap L}{Cut, \Psi; \Delta \longrightarrow \bot; \cdot}\, \textit{decide!}$$

where Π_1 and Π_2 are cut-free, $\Delta'_i \subseteq \Delta_i$ and Γ_1 and Γ_2 are multisets of atoms, introduced after various applications of decide! on the cut rule. Let $\Gamma'_1 = \Gamma_1 \cup \lceil B \rceil$. Since Π_1 is cut-free, its last rule has to be decide! selecting one formula from Ψ (and in this case Δ'_1 is empty) or decide selecting one formula from Δ'_1 (Δ'_1 is a singleton). That is, the last inference rule of Π_1 is of the form:

$$\frac{Cut, \Psi; \cdot \xrightarrow{D} \Gamma'_1; \Upsilon}{Cut, \Psi; \Delta'_1 \longrightarrow \Gamma'_1; \Upsilon}$$

Hence every formula in Γ'_1 is a subformula of D, a formula in Ψ or in Δ'_1. Moreover, all the formulas in Γ'_1 must follow from the formula D and hence the number of formulas in Γ'_1 cannot exceed the number of the atomic subformulas of D.

Note that the total number of formulas in Γ'_1 is exactly the number of cuts applied in this path of the proof Π. Hence, for every branch of the proof, the number of decide! on cut rules is less than or equal to the maximum number of atomic subformulas of formulas in Ψ and Δ and this is less or equal to v. Since the number of decide or decide! in a branch of Π over a formula in Ψ or Δ is at most 2 (? has atomic scope), the maximum depth is $v + 2$.

A similar result holds if the cut rule in the proof system is one with the ? modal. ∎

For example, it is possible to prove coherence for LJ by bounding proof search at depth 4 during the check for duality.

7 LU

In [Gir93], Girard introduced the sequent system LU (logic of unity) in which classical, intuitionistic, and linear logics appear as fragments. In this logic, all three of these logics keep their own characteristics but they can also communicate via formulas containing connectives mixing these logics. The key to allowing these logics to share one proof system lies in using *polarities*. In terms of the encoding we have presented here, polarities allow the meta-level atom $\lfloor B \rfloor$ be replaced by $?\lfloor B \rfloor$ if B is positive and the meta-level atom $\lceil B \rceil$ be replaced by $?\lceil B \rceil$ if B is negative. This possibility of replacement is in contrast to the examples of classical and intuitionistic sequent proof systems presented earlier where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ atoms are either all preceded by the ? modal or all are not so prefixed. The neutral polarity is also available and corresponds to the case where this replacement with a ? modal is not allowed. Many of the LU inference rules for classical and intuitionistic connectives are specified in Figure 2. The definition of the predicates $pos(\cdot)$, $neg(\cdot)$, and $neu(\cdot)$ can be directly obtained from the various polarity tables given in [Gir93].

As noted before, LU is not canonical since the side conditions in its rules require meta-level predicates other than simply $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$. For a future work, we intend to generalize the notion of canonical clauses in order to handle more general systems like LU. Still, it is possible to introduce the notions of coherence

Identity and structure

$\lfloor B \rfloor \parr \lceil B \rceil.$

$\bot \circ\!\!- \lfloor B \rfloor \circ\!\!- \lceil B \rceil.$

$\lceil N \rceil \circ\!\!- ?\lceil N \rceil \Leftarrow neg(N).$

$\lfloor P \rfloor \circ\!\!- ?\lfloor P \rfloor \Leftarrow pos(P).$

Conjunction

$\lceil u \wedge v \rceil \Leftarrow \lceil u \rceil \Leftarrow \lceil v \rceil \quad \Leftarrow pos(u) \oplus pos(v).$

$\lceil u \wedge v \rceil \circ\!\!- \lceil u \rceil \,\&\, \lceil v \rceil \quad \Leftarrow notpos(u) \,\&\, notpos(v).$

$\lfloor u \wedge v \rfloor \circ\!\!- ?\lfloor u \rfloor \parr ?\lfloor v \rfloor \quad \Leftarrow pos(u) \oplus pos(v).$

$\lfloor u \wedge v \rfloor \circ\!\!- \lfloor u \rfloor \oplus \lfloor v \rfloor \quad \Leftarrow notpos(u) \,\&\, notpos(v).$

Intuitionistic implication

$\lceil u \supset v \rceil \circ\!\!- ?\lfloor u \rfloor \parr \lceil v \rceil.$

$\lfloor u \supset v \rfloor \Leftarrow \lceil u \rceil \circ\!\!- \lfloor v \rfloor.$

Quantifiers

$\lceil \forall_c u \rceil \circ\!\!- \forall x\, ?\lceil ux \rceil.$

$\lfloor \forall_c u \rfloor \Leftarrow \lfloor ux \rfloor.$

$\lceil \exists_c u \rceil \Leftarrow \lceil ux \rceil.$

$\lfloor \exists_c u \rfloor \circ\!\!- \forall x\, ?\lfloor ux \rfloor.$

Disjunction

$\lceil u \vee v \rceil \circ\!\!- !\lceil u \rceil \oplus !\lceil v \rceil \quad \Leftarrow notneg(u) \,\&\, notneg(v).$

$\lceil u \vee v \rceil \circ\!\!- ?\lceil u \rceil \parr ?\lceil v \rceil \quad \Leftarrow (pos(u) \,\&\, neg(v)) \oplus (neg(u) \,\&\, notneu(v)).$

$\lceil u \vee v \rceil \circ\!\!- \lceil u \rceil \parr ?!\lceil v \rceil \quad \Leftarrow neg(u) \,\&\, neu(v).$

$\lceil u \vee v \rceil \circ\!\!- ?!\lceil u \rceil \parr \lceil v \rceil \quad \Leftarrow neu(u) \,\&\, neg(v).$

$\lfloor u \vee v \rfloor \circ\!\!- ?\lfloor u \rfloor \,\&\, ?\lfloor v \rfloor \quad \Leftarrow notneg(u) \,\&\, notneg(v).$

$\lfloor u \vee v \rfloor \Leftarrow \lfloor u \rfloor \Leftarrow \lfloor v \rfloor \quad \Leftarrow (pos(u) \,\&\, neg(v)) \oplus (neg(u) \,\&\, notneu(v)).$

$\lfloor u \vee v \rfloor \circ\!\!- \lfloor u \rfloor \Leftarrow ?\lfloor v \rfloor \quad \Leftarrow neg(u) \,\&\, neu(v).$

$\lfloor u \vee v \rfloor \Leftarrow ?\lfloor u \rfloor \circ\!\!- \lfloor v \rfloor \quad \Leftarrow neu(u) \,\&\, neg(v).$

Classical implication

$\lceil u \Rightarrow v \rceil \circ\!\!- ?\lfloor u \rfloor \parr ?\lceil v \rceil \quad \Leftarrow (neg(u) \,\&\, neg(v)) \oplus (pos(u) \,\&\, notneu(v)).$

$\lceil u \Rightarrow v \rceil \circ\!\!- \lceil v \rceil \oplus \lfloor u \rfloor \quad \Leftarrow neg(u) \,\&\, pos(v).$

$\lfloor u \Rightarrow v \rfloor \circ\!\!- \lceil u \rceil \,\&\, \lfloor v \rfloor \quad \Leftarrow neg(u) \,\&\, pos(v).$

$\lfloor u \Rightarrow v \rfloor \Leftarrow \lceil u \rceil \Leftarrow \lfloor v \rfloor \quad \Leftarrow (neg(u) \,\&\, neg(v)) \oplus (pos(u) \,\&\, notneu(v)).$

Fig. 2. *LU* rules

and duality in LU using extra clauses: these clauses play the role of the *Cut* rule on determining the dual predicates for polarity.

Since in LU formulas have only one polarity, it is reasonable to consider the clauses:

$$\begin{aligned} \mathit{notpos}(u) &\multimap (\mathit{neu}(u) \oplus \mathit{neg}(u)). \\ \mathit{notneg}(u) &\multimap (\mathit{neu}(u) \oplus \mathit{pos}(u)). \\ \mathit{notneu}(u) &\multimap (\mathit{neg}(u) \oplus \mathit{pos}(u)). \\ \mathit{pos}(u) &\multimap \mathit{neg}(u) \multimap 0. \\ \mathit{pos}(u) &\multimap \mathit{neu}(u) \multimap 0. \\ \mathit{neg}(u) &\multimap \mathit{neu}(u) \multimap 0. \end{aligned}$$

The first three clauses define the predicates $\mathit{notpos}(\cdot), \mathit{notneg}(\cdot)$ and $\mathit{notneu}(\cdot)$ while the last three indicate that $\mathit{pos}(\cdot), \mathit{neg}(\cdot)$ and $\mathit{neu}(\cdot)$ are dual predicates. Let $\mathcal{L}$ be the set of clauses above. It is straightforward to prove the following for LU: for every connective $\diamond$ of LU, if the left and right introduction clauses for $\diamond$ in their definition form are:

$$\forall \bar{x}(\lfloor \diamond(x_1, \ldots, x_i) \rfloor \circ\!\!- B_l) \quad \text{and} \quad \forall \bar{x}(\lceil \diamond(x_1, \ldots, x_i) \rceil \circ\!\!- B_r)$$

then

$$!\,\mathcal{L}, !\,\mathit{Cut}, !\,\mathit{Pos}, !\,\mathit{Neg} \vdash \forall \bar{x}(B_l \multimap B_r \multimap \bot) \qquad (***)$$

in linear logic. Here, *Neg* is the third and *Pos* the fourth clause in Figure 2. This suggests that such an entailment might be used as a natural generalization of coherence to this setting.

Example 2. Consider the definition form for the left and right introduction rules for the conjunction:

$$\lceil u \wedge v \rceil \circ\!\!- !(\lceil u \rceil \& \lceil v \rceil \& (\mathit{pos}(u) \oplus \mathit{pos}(v))) \oplus ((\lceil u \rceil \& \lceil v \rceil) \otimes !(\mathit{notpos}(u) \& \mathit{notpos}(v))).$$

$$\lfloor u \wedge v \rfloor \circ\!\!- (?\lfloor u \rfloor \parr ?\lfloor v \rfloor \otimes !(\mathit{pos}(u) \oplus \mathit{pos}(v))) \oplus \lfloor u \rfloor \oplus \lfloor v \rfloor \otimes !(\mathit{notpos}(u) \& \mathit{notpos}(v)).$$

Due to the $\oplus$ operator that occurs on B_r and B_l, the proof of the sequent $(***)$ will have four sub-proofs, two of them inferring the same polarity for each object-level formula involved and the other two with incompatible polarities for at least one formula. If the polarities are the same, the proof follows mostly as the usual duality check described in Section 5 (some extra steps may be necessary due to the compact way in which the set of rules of LU was written). On the other hand, if dual polarities appear the proof follows easily and the only rules applied are the last rules listed in $\mathcal{L}$.

The intuitive way we motivated a notion of coherence for LU suggests that it may be possible to extend the definitions and results obtained in earlier sections to more elaborate proofs systems containing certain kinds of side conditions.

8 Conclusion and future work

We have argued here that the use of linear logic as a meta-logic for the specification of sequent calculi allows us to use some of the meta-theory of linear logic to draw conclusions about the object-level proof systems. For example, the notion of duality within coherent proof systems is basically the notion of de Morgan duals in linear logic. The proof of Lemma 3, used to prove object-level cut-elimination, makes a critical use of meta-level cut-elimination. We also showed that for coherent proof systems, the question of whether or not one proof system's encoding entails another proof system's encoding is decidable.

An implementation of Forum can also provide a vehicle for the implementation of a number of object-level proof systems. To experiment with the decision procedures described in this paper, the authors used a simple and direct implementation of the Forum proof system within λProlog [NM88]: this implementation could then be used to do proof search restricted to bounded depth.

There are certainly numerous directions for future work related to what has been presented here. For example, most sequent calculi remain complete when restricting to atomically closed initial sequents. Checking the completeness of such a restriction should certainly be handled using techniques such as those for proving that coherent proofs systems satisfy cut-elimination. Also, there have been various proposals for non-commutative variants of classical linear logic [AR99,GS01,Ret97]: it would be interesting to see if these can be used to capture non-commutative object-level logics in a manner done here.

Finally, while we addressed the question of whether or not an inference rule is derivable from other inference rules, the more interesting and useful question is whether or not an inference rule is *admissible* in another proof system. For this, induction is generally required. It seems natural to consider adding to linear logic forms of induction along the lines found in [MM00,Pim01].

References

[AL01] Arnon Avron and Iddo Lev. Canonical propositional gentzen-type systems. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR 2001*, volume 2083 of *LNAI*, pages 529–544. Springer-Verlag, 2001.

[And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[AR99] V. Michele Abrusci and Paul Ruet. Non-commutative logic I: The multiplicative fragment. *Annals of Pure and Applied Logic*, 101(1):29–64, 1999.

[Chi95] Jawahar Chirimar. *Proof Theoretic Approach to Specification Languages*. PhD thesis, University of Pennsylvania, February 1995.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[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 Girard, Lafont, and Regnier, editors, *Workshop on Linear Logic*, pages 211–224. London Mathematical Society Lecture Notes 222, Cambridge University Press, 1995.

[DM79] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.

[DM95] Giorgio Delzanno and Maurizio Martelli. Objects in Forum. In *Proceedings of the International Logic Programming Symposium*, 1995.

[Dyc92] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *Journal of Symbolic Logic*, 57(3):795–807, September 1992.

[FM88] Amy Felty and Dale Miller. Specifying theorem provers in a higher-order logic programming language. In *Ninth International Conference on Automated Deduction*, pages 61–80, Argonne, IL, May 1988. Springer-Verlag.

[Gen69] Gerhard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Co., Amsterdam, 1969.

[Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.

[GS01] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In L. Fribourg, editor, *CSL 2001*, volume 2142 of *LNCS*, pages 54–68, 2001.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, 1993.

[LSS93] Patrick Lincoln, Andre Scedrov, and Natarajan Shankar. Linearizing intuitionistic implication. In *Annals of Pure and Applied Logic*, pages 151–177, 1993.

[Mil93] Dale Miller. The π-calculus as a theory in linear logic: Preliminary results. In E. Lamma and P. Mello, editors, *Proceedings of the 1992 Workshop on Extensions to Logic Programming*, number 660 in LNCS, pages 242–265. Springer-Verlag, 1993.

[Mil96] Dale Miller. Forum: A multiple-conclusion specification language. *Theoretical Computer Science*, 165(1):201–232, September 1996.

[Mil00] Dale Miller. Abstract syntax for variable binders: An overview. In John Lloyd and et. al., editors, *Computational Logic - CL 2000*, number 1861 in LNAI, pages 239–253. Springer, 2000.

[MM00] Raymond McDowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232:91–119, 2000.

[MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[MP] Dale Miller and Elaine Pimentel. Linear logic as a framework for specifying sequent calculus. To appear in the Proceedings of Logic Colloquium 1999.

[NM88] Gopalan Nadathur and Dale Miller. An Overview of λProlog. In *Fifth International Logic Programming Conference*, pages 810–827, Seattle, August 1988. MIT Press.

[Pau89] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–397, September 1989.

[Pim01] Elaine Gouvêa Pimentel. *Lógica linear e a especificação de sistemas computacionais*. PhD thesis, Universidade Federal de Minas Gerais, Belo Horizonte, M.G., Brasil, December 2001. (written in English).

[Ret97] Christian Retoré. Pomset logic: a non-commutative extension of classical linear logic. In *Proceedings of TLCA*, volume 1210, pages 300–318, 1997.

[Ric98] Giorgia Ricci. *On the expressive powers of a Logic Programming presentation of Linear Logic (FORUM)*. PhD thesis, Department of Mathematics, Siena University, December 1998.