A formal framework for specifying sequent calculus proof systems

Dale Miller[a], Elaine Pimentel[b]

[a] *INRIA Saclay and LIX/Ecole Polytechnique, 91128 Palaiseau, France*
[b] *Departamento de Matemáticas, Universidad del Valle, Cali, Colombia and Departamento de Matemática, UFMG, Belo Horizonte, Brazil*

Abstract

Intuitionistic logic and intuitionistic type systems are commonly used as frameworks for the specification of natural deduction proof systems. In this paper we show how to use classical linear logic as a logical framework to specify sequent calculus proof systems and to establish some simple consequences of encoded sequent calculus proof systems. In particular, derivability of an inference rule from a set of inference rules can be decided by bounded (linear) logic programming search on the encoded rules. We also present two simple and decidable conditions that guarantee that the cut rule and non-atomic initial rules can be eliminated.

Keywords: logical frameworks, linear logic, sequent calculus, proof systems, cut elimination

1. Introduction

Intuitionistic logic with quantification at (non-predicate) higher-order types [1, 2, 3] and dependently typed λ-calculi [4, 5, 6] can be used as meta-logics to specify proof systems for a range of object logics. A major advantage of using a "logical" or "type-theoretical" framework to specify a proof system is that they support levels of abstraction that facilitate writing declarative specifications of object-logic proof systems. For example, if the framework contains λ-binding, α, β, η-conversions, and higher-order quantification, then these can be used to encode all formula-level and proof-level bindings and the associated notions of equality and substitution. Similarly, these frameworks provide context management (via eigenvariables and hypothetical assumptions) that an object-level specification can exploit directly. Furthermore, implementations of such frameworks using unification and backtracking search can provide partial or complete implementations of natural deduction specifications [7, 8, 9]. Intuitionistic linear logic and a dependently typed λ-calculus with linear types have also been

Email addresses: `dale.miller@inria.fr` (Dale Miller), `elaine.pimentel@gmail.com` (Elaine Pimentel)

used as more expressive frameworks for the specification of a wider range of natural deduction proof systems [10, 11].

In this paper, we shall move from a meta-logical framework based on intuitionistic principles to one based on classical principles, particularly, those found within linear logic. The availability of classical principles in the framework allows us to capture directly the dualities that are present within the sequent calculi, such as De Morgan dualities, left/right duality, and the cut rule/initial rule duality. While intuitionistic linear logic can be used to encode sequent calculus [12], the associated dualities are not directly expressible in an intuitionistic framework. A linear logic meta-logic allows us to write simple linear logic formulas that capture the relationship between specification of left and right-introduction rules for an object-level logical connective. We show that if those formulas are, in fact, theorems of linear logic then the cut rule and non-atomic initial inference rules can be eliminated. Furthermore, we prove that if those linear logic formulas have proofs then they have short proofs: this makes it possible to show that this sufficient condition for cut-elimination and non-atomic initial-elimination is decidable.

This paper is structured as follows. Linear logic and a focused proof system for it are presented in Section 2. The general approach to specifying sequent calculus inference rules in linear logic is described in Section 3 and illustrated with examples in Section 4. Section 5 provides a decision procedure that determines if one set of inference rules entails another set. Sections 6 and 7 shows how linear logic can be used to provide simple and direct proofs of a number of properties of a specified sequent system, such as the elimination of cuts and non-atomic initial rules. Finally, we describe some related work in Section 8 and conclude in Section 9. This paper collects together most of the results from three conference papers [13, 14, 15] by the authors.

2. Linear logic

By the name LL we shall mean the logic that results from merging the logical connectives and proof rules of linear logic [16] with the term and quantificational structure of Church's Simple Theory of Types [17]. More precisely, simple types are either *primitive types*, of which o is a reserved primitive type denoting formulas, or *functional types* that are written using an infix arrow $\tau \to \tau'$. A type is a *predicate type* if it is of the form $\tau_1 \to \cdots \to \tau_n \to o$, where $n \geq 0$. The *order of a type* is a count of the number of occurrences of $\to$ that appear to the left of a $\to$: for example, primitive types are of order 0; $o \to o$ and $o \to o \to o$ are of order 1; and $(i \to o) \to o$ is of order 2. *Terms* are simply typed λ-terms and we identify two terms up to the usual α, β, and η-conversions. 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 $(\lambda x.B)t$. A *formula* is a term of type o.

The logical connectives for LL are those of linear logic: these can be divided into the following groups: the *multiplicative* version of conjunction, true, disjunction, and false, which are written as $\otimes$, 1, $⅋$, $\bot$, respectively; and the *additive* version of these connectives, which are written as &, $\top$, $\oplus$, 0, respectively; the *exponentials* ! and ? and the (typed) universal and existential quantifiers $\forall_\tau$ and $\exists_\tau$. In the quantifiers, the syntactic variable τ can range over all non-predicate types: $\forall_\tau$ and $\exists_\tau$ both have type $(\tau \to o) \to o$. The expressions $\forall_\tau \lambda x.B$ and $\exists_\tau \lambda x.B$ are abbreviated as the more usual $\forall_\tau x.B$ and $\exists_\tau x.B$. The subscript τ is often dropped when it is not important or it can be determined from context. Negation is a logical connective that has only atomic scope: if B is a general formula then $B^\perp$ denotes the result of moving negations inward until it has only atomic scope (such formulas are in *negation normal form*). For convenience, the expressions $A \multimap B$, $A \Rightarrow B$, and $A \equiv B$ are defined as, respectively, $A^\perp ⅋ B$, $(!\,A) \multimap B$, and $(A \multimap B) \mathbin{\&} (B \multimap A)$.

We write $\vdash_{LL} \Delta$ to denote entailment in the usual sense of linear logic [16] (we provide an explicit proof system for linear logic in Section 2.1).

2.1. A focused proof system for linear logic

In [18], Andreoli introduced a normal form of cut-free proofs in linear logic that plays an important role in our ability to use linear logic to specify inference rules. This normal form of proof is given by a *focused* proof system that is organized around providing two "phases" of proof construction, one for invertible inference rules and one for not-necessarily-invertible inference rules.

The connectives of linear logic can be divided into two classes. The *negative* connectives have invertible introduction rules: these connectives are $⅋$, $\bot$, &, $\top$, $\forall$, and ?. The *positive* connectives are the de Morgan duals of the negative connectives, namely, $\oplus$, 0, $\otimes$, 1, $\exists$, and !. A formula is *positive* if it is a negated atom or its top-level logical connective is positive. Similarly, a formula is *negative* if it is an atom or its top-level logical connective is negative.

The one-sided version of the focused proof system LLF is given in Figure 1 (the variable y in the $[\forall]$ rule is restricted so that it is not free in any formula of its conclusion). A *literal* is either an atomic formula or a negated atomic formula. In LLF, there are two kinds of sequents: $\Psi; \Delta \Uparrow L$ and $\Psi; \Delta \Downarrow F$, where Ψ is a set of formulas, Δ is a multiset of formulas, L is a list of formulas, and F is a formula. The inference rules with $\Uparrow$ in the premises and conclusion are the invertible rules. A sequence of these rules, reading them bottom-up, deals with the "don't-care non-determinism" of proof search: in this *negative* phase of proof construction, no backtracking on the selection of inference rules is necessary. The inference rules with $\Downarrow$ in the conclusion are the non-invertible rules. A sequence of these rules, reading them bottom-up, deals with the "don't-know non-determinism" of proof search: in this *positive* phase of proof construction, choices within inference rules can lead to failures for which one may need to backtrack. The negative phase ends (reading proofs bottom-up) when all formulas in L are "processed": that is, when L is the empty list. The positive phase begins by choosing (via one of the decide rules $[D_1]$ or $[D_2]$) a formula F on which to focus. Positive rules are applied to F until either a negated atom is encountered (and the proof must end using an initial rule $[I_1]$ or $[I_2]$) or a negative subformula is encountered (and the proof switches to the negative phase). This means that focused proofs

Negative rules

$$\frac{\Psi;\Delta\Uparrow L}{\Psi;\Delta\Uparrow\bot,L}\,[\bot]\qquad\frac{\Psi;\Delta\Uparrow F,G,L}{\Psi;\Delta\Uparrow F\parr G,L}\,[\parr]\qquad\frac{\Psi,F;\Delta\Uparrow L}{\Psi;\Delta\Uparrow ?F,L}\,[?]$$

$$\frac{}{\Psi;\Delta\Uparrow\top,L}\,[\top]\qquad\frac{\Psi;\Delta\Uparrow F,L\quad\Psi;\Delta\Uparrow G,L}{\Psi;\Delta\Uparrow F\,\&\,G,L}\,[\&]\qquad\frac{\Psi;\Delta\Uparrow F[y/x],L}{\Psi;\Delta\Uparrow\forall x.F,L}\,[\forall]$$

Positive rules

$$\frac{}{\Psi;\cdot\Downarrow 1}\,[1]\qquad\frac{\Psi;\Delta_1\Downarrow F\quad\Psi;\Delta_2\Downarrow G}{\Psi;\Delta_1,\Delta_2\Downarrow F\otimes G}\,[\otimes]\qquad\frac{\Psi;\cdot\Uparrow F}{\Psi;\cdot\Downarrow !\,F}\,[!]$$

$$\frac{\Psi;\Delta\Downarrow F_1}{\Psi;\Delta\Downarrow F_1\oplus F_2}\,[\oplus_l]\qquad\frac{\Psi;\Delta\Downarrow F_2}{\Psi;\Delta\Downarrow F_1\oplus F_2}\,[\oplus_r]\qquad\frac{\Psi;\Delta\Downarrow F[t/x]}{\Psi;\Delta\Downarrow\exists x.F}\,[\exists]$$

Identity, Decide, and Reaction rules

$$\frac{}{\Psi;A\Downarrow A^\bot}\,[I_1]\qquad\frac{}{\Psi,A;\cdot\Downarrow A^\bot}\,[I_2]\qquad\frac{\Psi;\Delta\Downarrow F}{\Psi;\Delta,F\Uparrow\cdot}\,[D_1]\qquad\frac{\Psi,F;\Delta\Downarrow F}{\Psi,F;\Delta\Uparrow\cdot}\,[D_2]$$

In $[I_1]$ and $[I_2]$, A is atomic; in $[D_1]$ and $[D_2]$, F is not an atom.

$$\frac{\Psi;\Delta,F\Uparrow L}{\Psi;\Delta\Uparrow F,L}\,[R\Uparrow]\qquad\text{provided that } F \text{ is positive or an atom}$$

$$\frac{\Psi;\Delta\Uparrow F}{\Psi;\Delta\Downarrow F}\,[R\Downarrow]\qquad\text{provided that } F \text{ is negative}$$

Figure 1: Focused proof search in linear logic LLF.

can be seen (bottom-up) as a sequence of alternations between negative and positive phases.

An earlier precursor to this style of focused proof construction was the work on *uniform proofs* that was used to provide a proof theory foundations for *logic programming*. In the terminology of uniform proofs [19, 20], goal-reduction corresponds to the negative phase and backchaining corresponds to the positive phase.

The following theorem is proved by Andreoli in [18].

Theorem 1. *If B is a linear logic formula, then $\vdash_{LL} B$ if and only if the sequent $;\Uparrow B$ is provable in LLF (Figure 1).*

Given that we are using a one-sided sequent calculus throughout this paper for linear logic, we modify the usual definition of *logical entailment* as follows.

Definition 2. Let $\mathcal{X}$ be a finite set of linear logic formulas. A formula C is *entailed* by $\mathcal{X}$ if the sequent $\mathcal{X};\cdot\Uparrow C$ has an LLF proof. Similarly, A *entails* B with respect to the set $\mathcal{X}$ if the sequent $\mathcal{X};\cdot\Uparrow A^\bot,B$ has an LLF proof.

Furthermore, we say that A and B are equivalent with respect to $\mathcal{X}$ if A *entails* B and B *entails* A both with respect to $\mathcal{X}$ or, equivalently, $A \equiv B$ is entailed by $\mathcal{X}$.

Focusing provides a rather immediate and natural operational semantics to proof search. Consider, for example, the sequent $\Psi; \Delta \Uparrow \cdot$ where the formula $A^\perp \otimes B^\perp \otimes (C \parr D \parr E)$ is a member of Ψ, where A, B, C, D, and E are atomic formulas, and where Ψ does not contain any atomic formulas. One way to proceed in attempting a proof of this sequent is to select (via $[D_2]$) this formula as focus: this yields the sequent

$$\Psi; \Delta \Downarrow A^\perp \otimes B^\perp \otimes (C \parr D \parr E).$$

By applying two occurrences of $[\otimes]$, this sequent reduces to proving the three sequents

$$\Psi; \Delta_1 \Downarrow A^\perp \qquad \Psi; \Delta_2 \Downarrow B^\perp \qquad \Psi; \Delta' \Downarrow C \parr D \parr E,$$

where Δ is the multiset union of Δ_1, Δ_2, and Δ'. Notice that the first two of these sequents are provable if and only if they are instances of the $[I_1]$, in which case, Δ_1 is $\{A\}$ and Δ_2 is $\{B\}$. Finally, the third sequent above is provable if and only if $\Psi; \Delta', C, D, E \Uparrow \cdot$ is provable. Thus, the selection of the formula $A^\perp \otimes B^\perp \otimes (C \parr D \parr E)$ for focus yields the "big-step" inference rule

$$\frac{\Psi; \Delta', C, D, E \Uparrow \cdot}{\Psi; A, B, \Delta' \Uparrow \cdot}$$

In other words, selecting this particular formula for focused proof construction is only possible if A and B are currently in the context and if the result of replacing them with C, D, and E is also provable. More generally, selecting the formula $A_1^\perp \otimes \cdots \otimes A_n^\perp \otimes (B_1 \parr \cdots \parr B_m)$ (where $n + m > 0$ and A's and B's are atomic) for focus yields the big-step rule

$$\frac{\Psi; \Delta, B_1, \ldots, B_m \Uparrow \cdot}{\Psi; \Delta, A_1, \ldots, A_n \Uparrow \cdot}$$

which can be interpreted as *multiset rewriting*: within a given multiset containing the multiset $\{A_1, \ldots, A_n\}$, replace that sub-multiset with $\{B_1, \ldots, B_m\}$.

For a second example, consider proving the sequent $\Psi; \Delta \Uparrow \cdot$ by focusing on the formula

$$A_3 \otimes {!}\, A_4 \otimes A_5 \otimes A_1^\perp \otimes A_2^\perp,$$

which is assumed to be a member of Ψ (once again, we will assume that Ψ does not contain any atomic formulas). Here, $A_1, \ldots, A_5$ are atomic formulas. This focusing phase is only successful if Δ is the multiset union of $\{A_1, A_2\}$ and two multisets Δ' and Δ'' and the sequents

$$\Psi; \Delta', A_3 \Uparrow \cdot \qquad \Psi; A_4 \Uparrow \cdot \qquad \Psi; \Delta'', A_5 \Uparrow$$

are all provable. Focusing on this particular formula provides the big-step inference rule

$$\frac{\Psi; \Delta', A_3 \Uparrow \cdot \quad \Psi; A_4 \Uparrow \cdot \quad \Psi; \Delta'', A_5 \Uparrow}{\Psi; A_1, A_2, \Delta', \Delta'' \Uparrow \cdot}$$

The use of the ! exponential allows one to control, to some extent, how the linear context Δ is divided during the search for a proof.

2.2. Bipoles

The correspondence between focusing on a formula and an induced big-step inference rule is particularly interesting when the focused formula is a *bipole* [21].

Definition 3. A *monopole* formula is a linear logic formula that is built up from atoms and occurrences of the negative connectives, with the restriction that ? has atomic scope. A *bipole* is a formula built from monopoles and negated atoms using only positive connectives, with the additional restriction that ! can only be applied to a monopole. We shall also insist that a bipole is either a negated atom or has a top-level positive connective.

The last restriction on bipoles forces them to be different from monopoles: bipoles are always positive formulas. If M is a monopole then $M \otimes 1$ is a bipole that is logically equivalent to it. Using the linear logic distributive properties, monopoles are equivalent to formulas of the form

$$\forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} B_{i,j}],$$

where the $B_{i,j}$ are either atoms or the result of applying ? to an atomic formula. Similarly, bipoles can be rewritten as formulas of the form

$$\exists x_1 \ldots \exists x_p [\oplus_{i=1,\ldots,n} \otimes_{j=1,\ldots,m_i} C_{i,j}],$$

where $C_{i,j}$ are either negated atoms, monopole formulas, or the result of applying ! to a monopole formula. Notice that the units $\top$, 0, $\perp$, and 1 are 0-ary versions of $\&$, $\oplus$, $\parr$, and $\otimes$, respectively.

Given this normal representation of bipoles and according to the focusing discipline, it turns out that, once introduced, a bipole is completely decomposed into its atomic subformulas, a fact illustrated by the following bipole derivation.

$$\dfrac{\cdots \quad \dfrac{\dfrac{\cdots \quad \dfrac{\Psi'; \Gamma' \Uparrow \cdot}{\Psi; \Gamma' \Uparrow \parr_{j=1,\ldots,m_i} ?A_{i,j}}\,[\parr, ?] \quad \cdots}{\Psi; \Gamma' \Uparrow \forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?A_{i,j}]}\,[\forall, \&]}{\Psi; \Gamma' \Downarrow ! \forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?A_{i,j}]}\,[!] \quad \cdots}{\Psi; \Gamma \Downarrow \exists x_1 \ldots \exists x_t [\oplus_{i=1,\ldots,k} \otimes_{j=1,\ldots,q_i} C_{i,j}]}\,[\exists, \oplus, \otimes]$$

Here $A_{i,j}$ is atomic for all i, j. If the connective ! is not present, then the rule ! is replaced by the rule $R \Downarrow$. Notice that the derivation above contains a single positive and a single negative phase. This two phase decomposition will enable the adequate capturing of the application of an object-level inference rule by the meta-level logic.

3. Encoding sequent systems

We now consider the logic LL as a "meta-logic" and the formulas of a first-order logic as the "object-logic" and then illustrate how sets of bipoles in linear logic can be used to encode sequent calculi proof systems for the object-logic. Let *obj* be the type of object-level formulas and let $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ be two meta-level predicates, both of type $obj \to o$. Object-level sequents of the form $B_1, \ldots, B_n \longrightarrow C_1, \ldots, C_m$ (where $n, m \geq 0$) are encoded as the linear logic formula

$$\lfloor B_1 \rfloor \invamp \cdots \invamp \lfloor B_n \rfloor \invamp \lceil C_1 \rceil \invamp \cdots \invamp \lceil C_m \rceil$$

or, equivalently, as the multiset $\{\lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor, \lceil C_1 \rceil, \ldots, \lceil C_m \rceil\}$ within the LLF proof system. The $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ predicates identify which object-level formulas appear on which side of the sequent arrow. A useful mnemonic: the predicate that is used to mark left-hand-side formulas $\lfloor\cdot\rfloor$ is written with an L and its mirror image.

3.1. Encoding inference rules

Let $\mathcal{Q}$ be the set $\{\lfloor\cdot\rfloor, \lceil\cdot\rceil\}$: that is, $\mathcal{Q}$ contains the two predicates of LL that are central to our specification of object-level proof systems. The constants denoting object-level logical connectives have types of order 0, 1, or 2. Typical examples of object-level constants at various orders are: true and false are of order 0; conjunction, disjunction, and implication are of order 1; and universal and existential quantifiers are given using constants of order 2. We shall also assume that object-level quantification is first-order and over one domain, denoted at the meta-level by d. This is a simplifying assumption only: supporting multi-sorted object-level quantification is no challenge in this setting.

Definition 4. An *introduction clause* is a closed bipole formula of the form

$$\exists x_1 \ldots \exists x_n [(q(\diamond(x_1, \ldots, x_n)))^\perp \otimes F]$$

where $\diamond$ is an object-level connective of arity n ($n \geq 0$) and $q \in \mathcal{Q}$. Furthermore, F does not contain negated atoms and an atom occurring in F is either of the form $p(x_i)$ or $p(x_i(y))$ where $p \in \mathcal{Q}$ and $1 \leq i \leq n$. In the first case, x_i has type *obj* while in the second case x_i has type $d \to obj$ and y is a variable (of type d) quantified (universally or existentially) in F (in particular, y is not in $\{x_1, \ldots, x_n\}$).

In general, focusing on the introduction clause above replaces an atom $q(\diamond(t_1, \ldots, t_n))$ with the formula $F[t_1/x_1, \ldots, t_n/x_n]$. Since this formula is a bipole, it will be immediately decomposed into its atomic subformulas, hence capturing in one meta-level step of derivation the one object-level step of applying an inference rule.

Consider, for example, the specification of the logical inference rules for object-level conjunction, represented here as the infix constant $\wedge$ of type $obj \to$

$obj \to obj$. The additive version of the inference rules for this connective is the following

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

These three inference rules can be specified in linear logic using the clauses

$$(\wedge L) \quad \exists A, B(\lfloor A \wedge B \rfloor^{\perp} \otimes (\lfloor A \rfloor \oplus \lfloor B \rfloor)). \qquad (\wedge R) \quad \exists A, B(\lceil A \wedge B \rceil^{\perp} \otimes (\lceil A \rceil \,\&\, \lceil B \rceil)).$$

Thus, these additive rules make use of two (dual) meta-level additive connectives: & and $\oplus$. The multiplicative version of the introduction rules for $\wedge$ is usually specified as follows:

$$\frac{\Delta, A, B \longrightarrow \Gamma}{\Delta, A \wedge B \longrightarrow \Gamma} \wedge L \qquad \frac{\Delta_1 \longrightarrow \Gamma_1, A \qquad \Delta_2 \longrightarrow \Gamma_2, B}{\Delta_1, \Delta_2 \longrightarrow \Gamma_1, \Gamma_2, A \wedge B} \wedge \mathrm{R}.$$

These two inference rules can be specified by the following clauses

$$(\wedge L) \quad \exists A, B(\lfloor A \wedge B \rfloor^{\perp} \otimes (\lfloor A \rfloor \parr \lfloor B \rfloor)). \qquad (\wedge R) \quad \exists A, B(\lceil A \wedge B \rceil^{\perp} \otimes (\lceil A \rceil \otimes \lceil B \rceil)).$$

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

The introduction rules for implications are often written multiplicatively as

$$\frac{\Delta_1 \longrightarrow \Gamma_1, A \qquad \Delta_2 \longrightarrow \Gamma_2, B}{\Delta_1, \Delta_2, A \supset B \longrightarrow \Gamma_1, \Gamma_2} \supset \mathrm{L} \qquad \frac{\Delta, A \longrightarrow \Gamma, B}{\Delta \longrightarrow \Gamma, A \supset B} \supset R$$

These two inference rules can be specified as the following clauses

$$(\supset L) \quad \exists A, B(\lfloor A \supset B \rfloor^{\perp} \otimes (\lceil A \rceil \otimes \lfloor B \rfloor)). \qquad (\supset R) \quad \exists A, B(\lceil A \supset B \rceil^{\perp} \otimes (\lfloor A \rfloor \parr \lceil B \rceil)).$$

The quantification of higher-order types that is available in our meta-logic makes it a simple matter to encode inference rules for object-level quantifiers. In fact, the usual left and right introduction rules for the object-level universal quantifier can be written as

$$(\forall L) \quad \exists B(\lfloor \forall B \rfloor^{\perp} \otimes \exists x \lfloor Bx \rfloor). \qquad (\forall R) \quad \exists B(\lceil \forall B \rceil^{\perp} \otimes \forall x \lceil Bx \rceil).$$

Here, the symbol $\forall$ is used for both meta-level and object-level quantification: at the object-level $\forall$ has the type $(d \to obj) \to obj$. Thus the variable B above has the type $d \to obj$. These quantifier rules make use of two meta-level quantifiers.

3.2. Encoding weakening and contraction

There are at least three ways to treat the structural rules of weakening and contraction. One method encodes them directly as the bipoles

$$(Weak_L) \quad \exists B(\lfloor B \rfloor^{\perp} \otimes \perp). \qquad (Cont_L) \quad \exists B(\lfloor B \rfloor^{\perp} \otimes (\lfloor B \rfloor \parr \lfloor B \rfloor)).$$

$$(Weak_R) \quad \exists B(\lceil B \rceil^{\perp} \otimes \perp). \qquad (Cont_R) \quad \exists B(\lceil B \rceil^{\perp} \otimes (\lceil B \rceil \parr \lceil B \rceil)).$$

To illustrate how focusing on the $(Cont_L)$ formula yields a contraction rule for the encoded proof system, consider the following derivation (here, we assume that the formula $(Cont_L)$ is a member of the set Ψ):

$$
\dfrac{\dfrac{}{\Psi; \lfloor B \rfloor \Downarrow \lfloor B \rfloor^{\perp}}\ [I_1] \qquad \dfrac{\dfrac{\dfrac{\Psi; \Gamma, \lfloor B \rfloor, \lfloor B \rfloor \Uparrow \cdot}{\Psi; \Gamma \Uparrow \lfloor B \rfloor, \lfloor B \rfloor}\ [R\Uparrow] \times 2}{\Psi; \Gamma \Uparrow \lfloor B \rfloor \mathbin{\wp} \lfloor B \rfloor}\ [\wp]}{\Psi; \Gamma \Downarrow \lfloor B \rfloor \mathbin{\wp} \lfloor B \rfloor}\ [R\Uparrow]}{\dfrac{\Psi; \Gamma, \lfloor B \rfloor \Downarrow \lfloor B \rfloor^{\perp} \otimes (\lfloor B \rfloor \mathbin{\wp} \lfloor B \rfloor)}{\Psi; \Gamma, \lfloor B \rfloor \Uparrow \cdot}\ [D_2], [\exists]}\ [\otimes]
$$

Another treatment of the structural rules involves making direct use of linear logic exponentials. For example, consider an intuitionistic proof system with sequents having exactly one right-hand formula and where formulas on the left can be contracted and weakened. A natural encoding of a sequent in such a system is the LL formula

$$?\lfloor B_1 \rfloor \mathbin{\wp} \cdots \mathbin{\wp} ?\lfloor B_n \rfloor \mathbin{\wp} \lceil C \rceil$$

or the LLF sequent $\Psi, \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor; \lceil C \rceil \Uparrow$, where Ψ contains the LL encoding of inference rules as bipoles. Notice that, by using linear logic exponentials in this way, we can capture directly only two kinds of context maintenance: either both weakening and contraction are available or neither is available.

Focusing on a bipole representing an introduction rule typically reads-out an atomic formula from the context and then writes back some formulas. If that atomic formula is marked with a ? (or resides in the left-most context in an LLF sequent) then that formula is not deleted but it is maintained. Of the formulas written into the context, some or all of them might need to be marked by ? as well. One way to do this is to prefix formulas in the body of the introduction rule with ?. For example, the left-introduction for implication can be written as

$$\exists A, B(\lceil A \supset B \rceil^{\perp} \otimes (?\lfloor A \rfloor \mathbin{\wp} \lceil B \rceil)).$$

With such a clause, the newly inserted $\lfloor A \rfloor$ will be marked by ? and, as a result of focusing, it will find its way into the context where formulas are implicitly marked with ?.

Instead of doing this marking in the body of every introduction rule, we could adopt the following bipoles into the specification of our proof system.

$$(Neg) \quad \exists B(\lceil B \rceil^{\perp} \otimes ?\lceil B \rceil) \qquad\qquad (Pos) \quad \exists B(\lfloor B \rfloor^{\perp} \otimes ?\lfloor B \rfloor).$$

With these assumptions, the structural rules of *weakening* and *contraction* can be specified so that they both apply to all left-hand or all right-hand formulas. In the following, the clauses *Neg* and *Pos* will be called our *structural clauses*. Notice that a consequence of *Neg* is the equivalence of $\lceil B \rceil$ and $?\lceil B \rceil$ and a consequent of *Pos* is the equivalence of $\lfloor B \rfloor$ and $?\lfloor B \rfloor$.

Linear logic and the LLF proof system distinguishes between two kinds of formulas: linear formulas, on which no structural rule is applicable and unbounded formulas on which structural rules are applicable. This distinction corresponds to the difference between the two left-most contexts in LLF sequents. Thus, when encoding object-level sequents $\Delta \longrightarrow \Gamma$, the contexts Δ and Γ are either multisets or sets of formulas. As a result, non-commutative logics and their sequent calculus, which use lists as contexts, cannot be described in the framework we have described here. Similarly, the exponentials used in light linear logics [22] do not fit our framework. On the other hand, it is simple to generalize our framework here to include the so called *subexponentials* [23] in order to strengthen the expressiveness of LLF and to permit the encoding of more proof systems [24].

3.3. The initial and cut rules

The initial rule, which asserts that the sequent $B \longrightarrow B$ is provable, can be specified by the clause

$$(\mathit{Init}) \qquad \exists B(\lfloor B \rfloor^{\perp} \otimes \lceil B \rceil^{\perp}),$$

Operationally, focusing on this clause must lead to a complete LLF proof in just one phase, involving the one occurrence of both $[\exists]$ and $[\otimes]$ as well as two occurrences of initial rules (either $[I_1]$ or $[I_2]$).

The cut rule, which is usually written multiplicatively as the inference rule

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

can be specified by the clause

$$(\mathit{Cut}) \qquad \exists B(\lceil B \rceil \otimes \lfloor B \rfloor),$$

Operationally, focusing on this clause splits the multiset context of an LLF sequent into two parts: in one split a left-appearing copy of B is placed and into the other split a right-appearing copy of B is placed. The resulting two sequents remain to be proved.

The *Init* and *Cut* clauses together prove that $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are duals of each other: that is, they entail the equivalence $\forall B(\lfloor B \rfloor^{\perp} \equiv \lceil B \rceil)$. The duality enforced by the cut and initial rules, made concise by the use of negation, is also stressed and similarly expressed in the Calculus of Structures [25].

3.4. Adequacy of encodings

The application of an inference rule in an object-level proof system is considered to be an atomic step. At the meta-level, however, such an inference rule might be encoded using an introduction clause containing several meta-logical connectives. It is important to qualify, in our setting, how well inference at the meta-level matches inference in the object-logic. Following [26], we will identify three levels of adequacy of encodings. The least restrictive level is based on *relative completeness*: that is, at this level, the set of provable sequents is the same – an object-level sequent has a proof if and only if the corresponding meta-level sequent has a proof. The second level of adequacy is based on *full completeness of proofs*: that is, the proofs of an object-level sequent are in one-to-one correspondence with the meta-level proofs of the encodings of the object sequent. Finally, the most restrictive level of adequacy is based on *full completeness of derivations*: that is, the object-level derivations (partial proofs, such as inference rules themselves) are in one-to-one correspondence Focusing is the critical device that allows us to state and prove such adequacy results.

Example 5. The additive version of the right-introduction for conjunction was given in Section 3.1 as

$$(\wedge R) \qquad \exists A, B(\lceil A \wedge B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil)).$$

To see the sense in which that clause encodes the object-level right-introduction of conjunction, consider the object-level sequent

$$D_1, \ldots, D_n \longrightarrow A \wedge B, C_1, \ldots, C_m$$

Let Ψ be some collection of linear logic formulas that includes the $(\wedge R)$ clause and let Δ be the multiset of formulas $\{\lfloor B_1\rfloor, \ldots, \lfloor B_n\rfloor, \lceil C_1\rceil, \ldots, \lceil C_m\rceil\}$. Consider the following focused derivation:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Psi; \lceil A \wedge B\rceil \Downarrow \lceil A \wedge B\rceil^{\perp}}\,[I_1] \qquad \dfrac{\dfrac{\dfrac{\Psi; \Delta, \lceil A\rceil \Uparrow \cdot}{\Psi; \Delta \Uparrow \lceil A\rceil}\,[R\Uparrow] \quad \dfrac{\Psi; \Delta, \lceil B\rceil \Uparrow \cdot}{\Psi; \Delta \Uparrow \lceil B\rceil}\,[R\Uparrow]}{\Psi; \Delta \Uparrow \lceil A\rceil \,\&\, \lceil B\rceil}\,[\&]}{\Psi; \Delta \Downarrow \lceil A\rceil \,\&\, \lceil B\rceil}\,[R\Downarrow]}{\Psi; \Delta, \lceil A \wedge B\rceil \Downarrow \lceil A \wedge B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil)}\,[\otimes]}{\Psi; \Delta, \lceil A \wedge B\rceil \Downarrow \exists A, B. \lceil A \wedge B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil)}\,[\exists] \times 2}{\Psi; \Delta, \lceil A \wedge B\rceil \Uparrow \cdot}\,[D_2]
$$

Thus, the process of selecting the $(\wedge R)$ clause and completing the positive and negative phases can be seen as replacing the need to prove the above object-level sequent with that of the two sequents $B_1, \ldots, B_n \longrightarrow A, C_1, \ldots, C_m$ and $B_1, \ldots, B_n \longrightarrow B, C_1, \ldots, C_m$.

3.5. Advantages of such encodings

The encoding of an object-level proof system into theories in linear logic has certain advantages over inference figures (in the style of, say, Gentzen). For example, the LL specifications do not deal with context explicitly and instead they concern only the formulas that are directly involved in the inference rule. The distinction between making the inference rule additive or multiplicative is achieved in inference figures by explicitly presenting contexts and either splitting or copying them. The representation through linear logic clauses achieves the same distinction using meta-level additive or multiplicative connectives. The meta-logic can also be used to handle object-level quantification directly: object-level instantiation of quantifiers is achieved by meta-level β-reductions and object-level eigenvariable conditions are handled by the corresponding condition built into the meta-logic. Similarly, the structural rules of contraction and weakening can be captured together using ?. Furthermore, encoding conventional inference figures as formulas has the advantage that the latter are easier to formally manipulate than the former.

Another advantage of this style of encoding is that a logic programming implementation of linear logic could provide a prototype implementation of any encoded object-logic. Some examples from Section 4 were implemented in the system TATU and are available at `http://www.logic.at/people/giselle/tatu/`.

There are some disadvantages to using linear logic as a meta-theory: in particular, proof systems containing non-commutative connectives or modal operators probably cannot be captured.

4. Some example specifications

We now present the linear logic encoding of a number of different kinds of sequent calculus proof systems as well as a natural deduction proof system.

4.1. Linear, classical, and intuitionistic logics

Figures 2, 3 and 4 present the linear logic specifications of **LL**, **LK**, and **LM**, respectively, for the well-known proof systems for linear, classical, and minimal logics (all clauses are implicitly existentially quantified). Object-level linear logic will be encoded reusing the same symbols that appear at the meta-level, namely, !, ?, $\otimes$, $\invamp$, $\bot$, 1, &, $\oplus$, 0, $\top$ $\multimap$, $\forall$, $\exists$ and negation $(\cdot)^\bot$ for atoms. Classical logic is encoded using $\wedge$, $\vee$, $\Rightarrow$, f_c, t_c, $\forall_c$, and $\exists_c$ for conjunction, disjunction, implication, false, true, and universal and existential quantification, respectively, while minimal logic is encoded with $\cap$, $\cup$, $\supset$, $\forall_i$, and $\exists_i$ for conjunction, disjunction, implication, and universal and existential quantification, respectively. As before, we use the type d to denote object-level individuals and obj to denote object-level formulas (our object-logics will all be first-order). All binary connectives have type $obj \to obj \to obj$ and will be written infix. Object-level constants representing quantification are all of the second order type $(d \to obj) \to obj$: we abbreviate expressions such as $\forall_i(\lambda x.B)$ as $\forall_i xB$.

The following adequacy theorems can be proved by structural induction over proof structures. Here, LK and LM proof systems are given by Gentzen [27][1] and LL is given by Girard [16].

[1]Note that we are not considering Gentzen's *interchange* rule since contexts are represented either by sets or multisets of formulas.

$$
\begin{array}{clcl}
(\multimap L) & \lfloor A \multimap B\rfloor^{\perp} \otimes (\lceil A\rceil \otimes \lfloor B\rfloor). & (\multimap R) & \lceil A \multimap B\rceil^{\perp} \otimes (\lfloor A\rfloor \parr \lceil B\rceil). \\
(\otimes L) & \lfloor A \otimes B\rfloor^{\perp} \otimes (\lfloor A\rfloor \parr \lfloor B\rfloor). & (\otimes R) & \lceil A \otimes B\rceil^{\perp} \otimes (\lceil A\rceil \otimes \lceil B\rceil). \\
(\& L_1) & \lfloor A \,\&\, B\rfloor^{\perp} \otimes \lfloor A\rfloor. & (\& R) & \lceil A \,\&\, B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil). \\
(\& L_2) & \lfloor A \,\&\, B\rfloor^{\perp} \otimes \lfloor B\rfloor. & (\oplus R_1) & \lceil A \oplus B\rceil^{\perp} \otimes \lceil A\rceil. \\
(\oplus L) & \lfloor A \oplus B\rfloor^{\perp} \otimes (\lfloor A\rfloor \,\&\, \lfloor B\rfloor). & (\oplus R_2) & \lceil A \oplus B\rceil^{\perp} \otimes \lceil B\rceil. \\
(\parr L) & \lfloor A \parr B\rfloor^{\perp} \otimes (\lfloor A\rfloor \otimes \lfloor B\rfloor). & (\parr R) & \lceil A \parr B\rceil^{\perp} \otimes (\lceil A\rceil \parr \lceil B\rceil). \\
(!\,L) & \lfloor !\,B\rfloor^{\perp} \otimes ?\lfloor B\rfloor. & (!\,R) & \lceil !\,B\rceil^{\perp} \otimes !\lceil B\rceil. \\
(?L) & \lfloor ?B\rfloor^{\perp} \otimes !\lfloor B\rfloor. & (?R) & \lceil ?B\rceil^{\perp} \otimes ?\lceil B\rceil. \\
(\forall_l L) & \lfloor \forall_l B\rfloor^{\perp} \otimes \exists x\lfloor Bx\rfloor. & (\forall_l R) & \lceil \forall_l B\rceil \otimes \forall x\lceil Bx\rceil. \\
(\exists_l L) & \lfloor \exists_l B\rfloor^{\perp} \otimes \forall x\lfloor Bx\rfloor. & (\exists_l R) & \lceil \exists_l B\rceil^{\perp} \otimes \exists x\lceil Bx\rceil. \\
(1L) & \lfloor 1\rfloor^{\perp} \otimes \perp. & (1R) & \lceil 1\rceil^{\perp} \otimes !\top. \\
(\perp L) & \lfloor \perp\rfloor^{\perp} \otimes !\top. & (\perp R) & \lceil \perp\rceil^{\perp} \otimes \perp. \\
(0L) & \lfloor 0\rfloor^{\perp} \otimes \top. & (\top R) & \lceil \top\rceil^{\perp} \otimes \top.
\end{array}
$$

Figure 2: Specification **LL** of object-level linear logic.

$$
\begin{array}{clcl}
(\Rightarrow L) & \lfloor A \Rightarrow B\rfloor^{\perp} \otimes (\lceil A\rceil \otimes \lfloor B\rfloor). & (\Rightarrow R) & \lceil A \Rightarrow B\rceil^{\perp} \otimes (\lfloor A\rfloor \parr \lceil B\rceil). \\
(\wedge L) & \lfloor A \wedge B\rfloor^{\perp} \otimes (\lfloor A\rfloor \oplus \lfloor B\rfloor). & (\wedge R) & \lceil A \wedge B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil). \\
(\vee L) & \lfloor A \vee B\rfloor^{\perp} \otimes (\lfloor A\rfloor \,\&\, \lfloor B\rfloor). & (\vee R) & \lceil A \vee B\rceil^{\perp} \otimes (\lceil A\rceil \oplus \lceil B\rceil). \\
(\forall_c L) & \lfloor \forall_c B\rfloor^{\perp} \otimes \exists x\lfloor Bx\rfloor. & (\forall_c R) & \lceil \forall_c B\rceil^{\perp} \otimes \forall x\lceil Bx\rceil. \\
(\exists_c L) & \lfloor \exists_c B\rfloor^{\perp} \otimes \forall x\lfloor Bx\rfloor. & (\exists_c R) & \lceil \exists_c B\rceil^{\perp} \otimes \exists x\lceil Bx\rceil. \\
(f_c L) & \lfloor f_c\rfloor^{\perp} \otimes \top. & (t_c R) & \lceil t_c\rceil^{\perp} \otimes \top.
\end{array}
$$

Figure 3: Specification **LK** classical logic.

$$
\begin{array}{clcl}
(\supset L) & \lfloor A \supset B\rfloor^{\perp} \otimes (\lceil A\rceil \otimes \lfloor B\rfloor). & (\supset R) & \lceil A \supset B\rceil^{\perp} \otimes (\lfloor A\rfloor \parr \lceil B\rceil). \\
(\cap L) & \lfloor A \cap B\rfloor^{\perp} \otimes (\lfloor A\rfloor \oplus \lfloor B\rfloor). & (\cap R) & \lceil A \cap B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil). \\
(\cup L) & \lfloor A \cup B\rfloor^{\perp} \otimes (\lfloor A\rfloor \,\&\, \lfloor B\rfloor). & (\cup R) & \lceil A \cup B\rceil^{\perp} \otimes (\lceil A\rceil \oplus \lceil B\rceil). \\
(\forall_i L) & \lfloor \forall_i B\rfloor^{\perp} \otimes \exists x\lfloor Bx\rfloor. & (\forall_i R) & \lceil \forall_i B\rceil^{\perp} \otimes \forall x\lceil Bx\rceil. \\
(\exists_i L) & \lfloor \exists_i B\rfloor^{\perp} \otimes \forall x\lfloor Bx\rfloor. & (\exists_i R) & \lceil \exists_i B\rceil^{\perp} \otimes \exists x\lceil Bx\rceil.
\end{array}
$$

Figure 4: Specification **LM** minimal logic

$$
\begin{array}{ll}
APos = \exists A(\lfloor A\rfloor^{\perp} \otimes ?\lfloor A\rfloor \otimes atomic(A)). & Pos = \exists B(\lfloor B\rfloor^{\perp} \otimes ?\lfloor B\rfloor). \\
ANeg = \exists A(\lceil A\rceil^{\perp} \otimes ?\lceil A\rceil \otimes atomic(A)). & Neg = \exists B(\lceil B\rceil^{\perp} \otimes ?\lceil B\rceil). \\
AInit = \exists A(\lfloor A\rfloor^{\perp} \otimes \lceil A\rceil^{\perp} \otimes atomic(A)). & Init = \exists B(\lfloor B\rfloor^{\perp} \otimes \lceil B\rceil^{\perp}). \\
ACut = \exists A(\lfloor A\rfloor \otimes \lceil A\rceil \otimes atomic(A)). & Cut = \exists B(\lfloor B\rfloor \otimes \lceil B\rceil)
\end{array}
$$

Figure 5: Some named formulas.

Proposition 6. *The sequent* $\longrightarrow B$ *has a linear logic proof if and only if the sequent*

$$\boldsymbol{LL}, \mathit{Cut}, \mathit{Init}; \lceil B \rceil \Downarrow$$

is provable in linear logic; has an LK proof if and only if

$$\boldsymbol{LK}, \mathit{Cut}, \mathit{Init}, \mathit{Pos}, \mathit{Neg}; \lceil B \rceil \Downarrow$$

is provable in linear logic; and has an LM proof if and only if

$$\boldsymbol{LM}, \mathit{Cut}, \mathit{Init}, \mathit{Pos}; \lceil B \rceil \Downarrow$$

is provable in linear logic.

While this proposition states that these proof system encodings are adequate at the level of *relative completeness* (see Section 3.4), the direct proofs of these statements easily show that the actual adequacy result is at the level of *full completeness of proofs*. As the next example illustrates, adequacy at the level of *full completeness of derivations* is not achieved by these specifications: going for this further precision in specification rules is illustrated in Section 4.2 and in [26]. The additional structure provided by full completeness of derivations is not needed for most of the goals of this paper.

Example 7. [LM] The encoding of ($\supset L$) in Figure 4 is not at the level of full completeness for LM derivations. For example, let Ψ be the classical context $\{\lfloor D \rfloor, \lfloor A \supset B \rfloor\}$, for object-level formulas A, B, and D and consider proving the sequent $\mathbf{LM}, \mathit{Cut}, \mathit{Init}, \mathit{Pos}, \Psi; \lceil C \rceil \Uparrow$, which encodes the intuitionistic sequent $D, A \supset B \longrightarrow C$ in LM. Focusing on ($\supset L$) and then reducing the resulting positive and negative phases leads to two different sets of sequents: these two different reductions encode the two object-level inference rules

$$\frac{D \longrightarrow A, C \qquad D, B \longrightarrow}{D, A \supset B \longrightarrow C} \qquad \text{and} \qquad \frac{D \longrightarrow A \qquad D, B \longrightarrow C}{D, A \supset B \longrightarrow C}.$$

Within LM, the first inference rule is not allowed since its left-most premise contains two right-hand-side formulas. Thus, of the two possible reductions that arise from focusing on the encoding of ($\supset L$), one is not allowed in LM. While full adequacy for derivations fails for this encoding of LM, full adequacy for proofs does hold for **LM**: notice that a meta-level sequent containing two right-hand-side formulas (such as $\lceil A \rceil, \lceil C \rceil$) is never entailed by **LM**.

Example 8. [LJ] In order to specify the well known proof system LJ [27] for intuitionistic logic (obtaining the system **LJ**), we add to **LM** the clauses

$$(f_i L) \quad \lfloor f_i \rfloor^\perp \otimes \top. \qquad (t_i R) \quad \lceil t_i \rceil^\perp \otimes \top.$$

It is then possible to prove an adequacy result for LJ as stated in Proposition 6, but only at the level of "relative completeness"[2]. In fact, the sequent $\Phi; \cdot \Downarrow$

[2] Note that the proof of such a proposition is not by simple structural induction.

$\lceil(t_i \supset f_i) \supset A\rceil$, where $\Phi = \{\mathbf{LJ}, \mathit{Cut}, \mathit{Init}, \mathit{Pos}\}$, is provable for any object level formula A, but the proof

$$\dfrac{\dfrac{\dfrac{}{\Phi; \lceil A\rceil, \lceil t_i\rceil \Uparrow \cdot} \quad \dfrac{}{\Phi; \lfloor f_i\rfloor \Uparrow \cdot}}{\Phi; \lfloor(t_i \supset f_i)\rfloor, \lceil A\rceil \Uparrow \cdot}}{\Phi; \lceil(t_i \supset f_i) \supset A\rceil \Uparrow \cdot}$$

does not have any correspondent in LJ.

In order to achieve adequacy at the level of "full completeness of derivations", we substitute **LJ**'s implication left and cut clauses by

$$(\supset L) \quad \lfloor A \supset B\rfloor^{\perp} \otimes (!\lceil A\rceil \otimes \lfloor B\rfloor). \qquad (\mathit{Cut}) \quad \lfloor B\rfloor \otimes !\lceil B\rceil.$$

respectively. The bang preceding the right formulas forces *any* other formula appearing in the right-hand-side of the conclusion to move to the correct premise. This placement of right formulas is illustrated by the following derivation, where $\Theta = \{\mathbf{LJ}, \mathit{Cut}, \mathit{Init}, \mathit{Pos}, \Psi\}$ and Ψ is as in Example 7.

$$\dfrac{\dfrac{\dfrac{\Theta; \lceil A\rceil \Uparrow \cdot}{\Theta; \cdot \Downarrow !\lceil A\rceil} \quad \dfrac{\dfrac{\Theta, \lfloor B\rfloor; \lceil C\rceil \Uparrow \cdot}{\Theta; \lceil C\rceil \Uparrow \lfloor B\rfloor}}{\Theta; \lceil C\rceil \Downarrow \lfloor B\rfloor}}{\Theta; \lceil C\rceil \Downarrow !\lceil A\rceil \otimes \lfloor B\rfloor}}{\Theta; \lceil C\rceil \Uparrow \cdot}$$

In the rest of the paper, we will call **LJ** the system **LM** plus the clauses for the units, since the results stated here only depend on the least level of adequacy.

Since the introduction clauses for **LJ** and **LK** are identical except for a systematic renaming of logical constants, the distinction between the intuitionistic (**LJ**) and classical (**LK**) proof systems is that the former assumes *Pos* while the latter assumes both *Pos* and *Neg*. The following equivalences are provable using some of the named formulas in Figure 5.

1. *Cut* and *Init* prove the equivalence $\lceil B\rceil \equiv \lfloor B\rfloor^{\perp}$.
2. *Cut*, *Init*, *Pos* and *Neg* prove the following equivalences: $\lceil B\rceil \equiv \lfloor B\rfloor^{\perp}$, $?\lfloor B\rfloor \equiv \lfloor B\rfloor$, $?\lceil B\rceil \equiv \lceil B\rceil$, $?\lceil B\rceil \equiv (?\lfloor B\rfloor)^{\perp}$, $?\lceil B\rceil \equiv !\lceil B\rceil$, $?\lfloor B\rfloor \equiv !\lfloor B\rfloor$, $!\lfloor B\rfloor \equiv \lfloor B\rfloor$, and $?\lceil B\rceil \equiv \lceil B\rceil$.
3. *Cut*, *Init* and *Pos* prove the equivalences $\lceil B\rceil \equiv (?\lfloor B\rfloor)^{\perp}$, $\lceil B\rceil \equiv !\lceil B\rceil$, $\lfloor B\rfloor \equiv ?\lfloor B\rfloor$, and $\lceil B\rceil \equiv \lfloor B\rfloor^{\perp}$.

Thus, the cut and initial rules imply that $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ are duals of each other. In the cases of **LJ** and **LK**, however, that duality also forces the collapse of some of *exponential modalities*. In particular, linear logic has 7 distinct modalities derived from the exponentials [28], namely, $!$, $?$, $?\,!$, $!\,?$, $!\,?\,!$, $?\,!\,?$, and the empty sequence of exponentials. However, it is straightforward to show that, in the **LK** theory, all those modals are equivalent when applied to either a $\lfloor\cdot\rfloor$-atom or a $\lceil\cdot\rceil$-atom. It is also easy to see that, in the **LJ** theory, these modals collapse to four when applied to either the $\lceil\cdot\rceil$-atoms or the $\lfloor\cdot\rfloor$-atoms.

$$
\begin{array}{llll}
(\Rightarrow L) & \lfloor A \Rightarrow B \rfloor^{\perp} \otimes (\lceil A \rceil \;\&\; \lfloor B \rfloor). & (\Rightarrow R) & \lceil A \Rightarrow B \rceil^{\perp} \otimes (\lfloor A \rfloor \parr \lceil B \rceil). \\
(\wedge L) & \lfloor A \wedge B \rfloor^{\perp} \otimes (\lfloor A \rfloor \parr \lfloor B \rfloor). & (\wedge R) & \lceil A \wedge B \rceil^{\perp} \otimes (\lceil A \rceil \;\&\; \lceil B \rceil). \\
(\vee R) & \lceil A \vee B \rceil^{\perp} \otimes (\lceil A \rceil \parr \lceil B \rceil). & (\vee L) & \lfloor A \vee B \rfloor^{\perp} \otimes (\lfloor A \rfloor \;\&\; \lfloor B \rfloor). \\
(f_c L) & \lfloor f_c \rfloor^{\perp} \otimes \top. & &
\end{array}
$$

Figure 6: Specification of **G3c**.

Example 9. The **G3c** proof system of [29] provides inference rules for propositional classical logic that are all invertible. A linear logic encoding of that proof system is given in Figure 6. The invertibility of the encoded rules follows immediately from the fact that the only logical connectives used in the introduction rules—&, $\parr$, $\top$—are negative and, hence, their introduction rules in linear logic are invertible.

4.2. Three focused proof systems: ILU, LKQ, LKT

We now present the specification of three proof systems that are "focused", meaning that after certain introduction rules are applied, the next introduction rule (in a bottom-up reading) can be constrained significantly. For the sake of presenting these examples, we shall consider the fragments of intuitionistic and classical logics that involve just implication and universal quantification.

Figure 7 contains a variant of the ILU proof system given in [28] in which quantification is limited to first-order and the structural rules of contraction and weakening are not explicitly given but are built into the other inference rules. Figure 8 contains the linear logic theory **ILU** that encodes ILU. The sequents in this proof system have the form $\Pi; \Gamma \longrightarrow A$, where Γ and Π denote multisets, and Π contains at most one formula. Such sequents are encoded at the meta-level as the sequent **ILU**, $\lfloor \Gamma \rfloor; \lfloor \Pi \rfloor, \lceil A \rceil \Uparrow \cdot$, where $\lfloor \Gamma \rfloor$ and $\lceil \Gamma \rceil$ denote the multiset of formulas resulting from applying the corresponding predicate to all formulas in Γ. Proofs in ILU are focused in the sense that the left rules $(\supset L)$ and $(\forall_i L)$ can only be applied to formulas in the left linear context Π. This restriction, which is enforced using exponentials in the encoding, constrains proof search significantly.

Observe that this specification does not have a "Pos" rule as in **LJ**. Instead, the exponential ? is applied to some left-atoms (as in the clauses $\supset R$ and *Mid-cut*) whenever they should have a classical behavior. All the other left-atoms behave linearly and these formulas maintain the "stoup" (the member of Π when it is non-empty). A notable feature of this specification is that the exponential ! is used in the body of the two left-rules and in the *Mid-cut* in order to ensure that only the stoup is selected and not some other left-atom.

ILU has adequacy at the level of *full completeness of proofs* but not at the level of *full completeness of derivations* since the use of the clause *Head-cut* could place more than one formula in the left linear context. Cut-free proofs have, however, adequacy on the level of full completeness of derivations. Note

$$\frac{B; B, \Gamma \longrightarrow A}{\cdot; B, \Gamma \longrightarrow A}\ D \qquad \frac{}{A; \Gamma \longrightarrow A}\ \mathit{initial} \qquad \frac{A[t/x]; \Gamma \longrightarrow B}{\forall_i x A; \Gamma \longrightarrow B}\ \forall_i L$$

$$\frac{\cdot; \Gamma \longrightarrow A \quad B; \Gamma \longrightarrow C}{A \supset B; \Gamma \longrightarrow C}\ \supset L \qquad \frac{\Pi; \Gamma, A \longrightarrow B}{\Pi; \Gamma \longrightarrow A \supset B}\ \supset R \qquad \frac{\Pi; \Gamma \longrightarrow A[y/x]}{\Pi; \Gamma \longrightarrow \forall_i x A}\ \forall_i R$$

$$\frac{\Pi; \Gamma \longrightarrow A \quad A; \Gamma \longrightarrow B}{\Pi; \Gamma \longrightarrow B}\ \mathit{head\text{-}cut} \qquad \frac{\cdot; \Gamma \longrightarrow A \quad \Pi; A, \Gamma \longrightarrow B}{\Pi; \Gamma \longrightarrow B}\ \mathit{mid\text{-}cut}$$

Figure 7: The sequent calculus ILU

$(\supset L)$	$\lfloor A \supset B \rfloor^{\perp} \otimes !\lceil A \rceil \otimes \lfloor B \rfloor.$	$(\supset R)$	$\lceil A \supset B \rceil^{\perp} \otimes (?\lfloor A \rfloor \parr \lceil B \rceil).$
$(\forall_i L)$	$\lfloor \forall_i B \rfloor^{\perp} \otimes !\lfloor Bx \rfloor.$	$(\forall_i R)$	$\lceil \forall_i B \rceil^{\perp} \otimes \forall x \lceil Bx \rceil.$
(*Head-cut*)	$\lceil A \rceil \otimes \lfloor A \rfloor.$	(*Init*)	$\lfloor A \rfloor^{\perp} \otimes \lceil A \rceil^{\perp}.$
(*Mid-cut*)	$!\lceil A \rceil \otimes ?\lfloor A \rfloor.$		

Figure 8: Specification **ILU**

that the *Head-cut* and *Init* rules of **ILU** prove the equivalence $\lceil B \rceil \equiv \lfloor B \rfloor^{\perp}$: no additional equivalences between linear logic exponential modals can be proved.

Two sequent calculi, LKQ and LKT, which provide focused proof systems for classical logic, are also presented in [28]. Sequents of the calculus LKQ, written as $\Gamma \longrightarrow \Delta; \Pi$, are encoded as linear logic formulas $\lceil \Pi \rceil \parr ?\lfloor \Gamma \rfloor \parr ?\lceil \Delta \rceil$ where Π represents a multiset containing at most one formula. The specification for LKQ is presented in Figure 9. Sequents of the LKT proof system, written as $\Pi; \Gamma \longrightarrow \Delta$, are encoded as $\lfloor \Pi \rfloor \parr ?\lfloor \Gamma \rfloor \parr ?\lceil \Delta \rceil$, where again Π is a multiset containing at most one formula (see Figure 10 for the specification of LKT). Observe that LKT is a classical equivalent of ILU; that is, the intuitionistic calculus is obtained from LKT by the usual restriction of having exactly one formula on the right side of the sequent.

The linear logic encodings of the LKQ and LKT proof systems are designed to be adequate at the demanding level of full completeness for deviations. Recent work [24] has demonstrated that it is possible to extend the expressive strength of linear logic (to include the so called *subexponentials*) so that more proof systems, such as ILU, can also be captured at this more demanding level of adequacy.

4.3. A more ad hoc *example*

In a final example, we present the encoding a proof system that deviates from the previous ones in several ways. An optimized version of (the implicational fragment of) LM, called IIL*, is presented in [30] (see also [31]). The proof system for IIL* is given in Figure 11: in those inference rules, the syntactic variable p is used to range over atomic formulas. Notice that IIL* does not contain contraction or cut rules, and weakening is only allowed at the leaves of a proof; that is, when the *Init* rule is applied (to atomic formulas). A key property of IIL* is that the *principal* formula is not duplicated in the premises

$$
\frac{}{A, \Gamma \longrightarrow \Delta; A}\ \mathit{initial}
\qquad
\frac{\Gamma \longrightarrow \Delta; A}{\Gamma \longrightarrow \Delta, A;}\ D
\qquad
\frac{\Gamma \longrightarrow \Delta; A \quad B, \Gamma \longrightarrow \Delta; \cdot}{\Gamma, A \Rightarrow B \longrightarrow \Delta; \cdot}\ {\Rightarrow}L
$$

$$
\frac{\Gamma, A[x/t] \longrightarrow \Delta; \cdot}{\Gamma, \forall_c x A \longrightarrow \Gamma; \cdot}\ \forall_c L
\qquad
\frac{\Gamma \longrightarrow \Delta, A[x/y]; \cdot}{\Gamma \longrightarrow \Delta; \forall_c x A}\ \forall_c R
\qquad
\frac{\Gamma, A \longrightarrow \Delta, B; \cdot}{\Gamma \longrightarrow \Delta; A \Rightarrow B}\ {\Rightarrow}R
$$

$$
\frac{\Gamma \longrightarrow \Delta; A \quad A, \Gamma \longrightarrow \Delta; \Pi}{\Gamma \longrightarrow \Delta; \Pi}\ \mathit{tail\text{-}cut}
\qquad
\frac{\Gamma \longrightarrow \Delta, A; \Pi \quad A, \Gamma \longrightarrow \Delta; \cdot}{\Gamma \longrightarrow \Delta; \Pi}\ \mathit{mid\text{-}cut}
$$

$$
\begin{array}{llll}
({\Rightarrow}L) & \lfloor A \Rightarrow B \rfloor^{\perp} \otimes !(\lceil A \rceil \otimes ?\lfloor B \rfloor). & ({\Rightarrow}R) & \lceil A \Rightarrow B \rceil^{\perp} \otimes !(?\lfloor A \rfloor \invamp ?\lceil B \rceil). \\
(\forall_c L) & \lfloor \forall_c B \rfloor^{\perp} \otimes !?\lfloor Bx \rfloor. & (\forall_c R) & \lceil \forall_c B \rceil^{\perp} \otimes ! \forall x ?\lceil Bx \rceil. \\
(\mathit{Cut}) & !\lceil A \rceil \otimes ?\lfloor A \rfloor. & (\mathit{Init}) & \lfloor A \rfloor^{\perp} \otimes \lceil A \rceil^{\perp}. \\
 & ?\lceil A \rceil \otimes !?\lfloor A \rfloor. & &
\end{array}
$$

Figure 9: The sequent calculus for LKQ and its encoding as **LKQ**

$$
\frac{}{A; \Gamma \longrightarrow \Delta, A}\ \mathit{initial}
\qquad
\frac{A; \Gamma \longrightarrow \Delta}{\cdot; A, \Gamma \longrightarrow \Delta}\ D
\qquad
\frac{\cdot; \Gamma \longrightarrow \Delta, A \quad B; \Gamma \longrightarrow \Delta}{A \Rightarrow B; \Gamma \longrightarrow \Delta}\ {\Rightarrow}L
$$

$$
\frac{A[x/t]; \Gamma \longrightarrow \Delta}{\forall_c x A; \Gamma \longrightarrow \Delta}\ \forall_c L
\qquad
\frac{\Pi; \Gamma \longrightarrow \Delta, A[x/y]}{\Pi; \Gamma \longrightarrow \Delta, \forall_c x A}\ \forall_c R
\qquad
\frac{\Pi; \Gamma, A \longrightarrow \Delta, B}{\Pi; \Gamma \longrightarrow \Delta, A \Rightarrow B}\ {\Rightarrow}R
$$

$$
\frac{\Pi; \Gamma \longrightarrow \Delta, A \quad A; \Gamma \longrightarrow \Delta}{\Pi; \Gamma \longrightarrow \Delta}\ \mathit{head\text{-}cut}
\qquad
\frac{\cdot; \Gamma \longrightarrow \Delta, A \quad \Pi; A, \Gamma \longrightarrow \Delta}{\Pi; \Gamma \longrightarrow \Delta}\ \mathit{mid\text{-}cut}
$$

$$
\begin{array}{llll}
({\Rightarrow}L) & \lfloor A \Rightarrow B \rfloor^{\perp} \otimes !?\lceil A \rceil \otimes \lfloor B \rfloor. & ({\Rightarrow}R) & \lceil A \Rightarrow B \rceil^{\perp} \otimes ?\lfloor A \rfloor \invamp ?\lceil B \rceil. \\
(\forall_c L) & \lfloor \forall_c B \rfloor^{\perp} \otimes !\lfloor Bx \rfloor. & (\forall_c R) & \lceil \forall_c B \rceil^{\perp} \otimes \forall x ?\lceil Bx \rceil. \\
(\mathit{Cut}) & ?\lceil A \rceil \otimes !\lfloor A \rfloor. & (\mathit{Init}) & \lfloor A \rfloor^{\perp} \otimes \lceil A \rceil^{\perp}. \\
 & !?\lceil A \rceil \otimes ?\lfloor A \rfloor. & &
\end{array}
$$

Figure 10: The sequent calculus for LKT and its encoding as **LKT**

of any of the rules. This suggests the encoding $\lfloor \Gamma \rfloor \invamp \lceil D \rceil$ for the IIL* sequent $\Gamma \longrightarrow D$. The linear logic encoding of IIL* is also given in Figure 11: there the predicate $\mathit{atomic}(\cdot)$ (of type $obj \to o$) is assumed to be defined to hold for all these atomic formulas. The encodings of the *Init* and $\supset L$ rules are different from what we have seen so far: the *Init* rule uses the additive truth $\top$ to allow weakening (via "erasing"), and the $\supset L_1$ and $\supset L_2$ rules use the additive conjunction & to copy the left context and uses a "two headed clause" in order to avoid copying the right context and to place it in the correct sequent of the premise.

The resulting logic specification is not composed of introduction clauses as formally defined in Definition 4. In fact, it is impossible to specify IIL* in our setting using only "one-headed" clauses since it is necessary to have two different linear contexts for encoding the system: one for the formulas on the left and one for the ones on the right side of the sequent. Linear logic allows only for one linear and one classical context, a shortcoming that can be overcome by using subexponentials [23, 32].

$$\frac{}{\Gamma, p \longrightarrow p}\ \text{Init} \qquad \frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B} \supset R \qquad \frac{\Gamma \longrightarrow p \qquad \Gamma, B \longrightarrow C}{\Gamma, p \supset B \longrightarrow C} \supset L_1$$

$$\frac{\Gamma, B \supset C \longrightarrow A \supset B \qquad \Gamma, C \longrightarrow D}{\Gamma, (A \supset B) \supset C \longrightarrow D} \supset L_2$$

$$\begin{array}{ll}
(\mathit{Init}) & \lfloor A \rfloor^{\perp} \otimes \lceil A \rceil^{\perp} \otimes \mathit{atomic}(A) \otimes \top. \\
(\supset R) & \lceil A \supset B \rceil^{\perp} \otimes (\lfloor A \rfloor \mathbin{⅋} \lceil B \rceil). \\
(\supset L_1) & \lfloor A \supset B \rfloor^{\perp} \otimes \lceil C \rceil^{\perp} \otimes \mathit{atomic}(A) \otimes (\lceil A \rceil \mathbin{\&} (\lfloor B \rfloor \mathbin{⅋} \lceil C \rceil)). \\
(\supset L_2) & \lfloor (A \supset B) \supset C \rfloor^{\perp} \otimes \lceil D \rceil^{\perp} \otimes ((\lfloor B \supset C \rfloor \mathbin{⅋} \lceil A \supset B \rceil) \mathbin{\&} (\lfloor C \rfloor \mathbin{⅋} \lceil D \rceil)).
\end{array}$$

Figure 11: The sequent calculus proof system IIL* and its encoding **IIL*** in linear logic.

4.4. Natural deduction

To illustrate an application of using meta-level reasoning to draw conclusions about proof systems for an object-logic, we show how a specification for natural deduction can be derived from a specification of sequent calculus for intuitionistic logic. For simplicity, we consider the fragment of minimal logic involving only $\supset$, $\cap$, and $\forall_i$ (disjunction and existential quantification are similarly addressed in [20]).

Given the equivalences arising from the cut, initial, and structural rules in **LM** listed in Section 4, the specification for $(\supset L)$ is logically equivalent to the following formulas.

$$\begin{aligned}
(\lfloor B \rfloor \otimes \lceil A \rceil) \otimes \lfloor A \supset B \rfloor^{\perp} &\equiv (\lceil B \rceil^{\perp} \otimes \lceil A \rceil) \otimes \lceil A \supset B \rceil \\
&\equiv (\lceil A \supset B \rceil \otimes \lceil A \rceil) \otimes \lceil B \rceil^{\perp}
\end{aligned}$$

The later can be recognized as a specification of the $\supset$ elimination rule. Similarly, the specification for $(\supset R)$ is equivalent to the following formulas, which encode the introduction rule for implication.

$$\begin{aligned}
(\lfloor A \rfloor \mathbin{⅋} \lceil B \rceil) \otimes \lceil A \supset B \rceil^{\perp} &\equiv (\lceil A \rceil^{\perp} \mathbin{⅋} \lceil B \rceil) \otimes \lceil A \supset B \rceil^{\perp} \\
&\equiv ((!\, \lceil A \rceil)^{\perp} \mathbin{⅋} \lceil B \rceil) \otimes \lceil A \supset B \rceil^{\perp} \\
&\equiv (!\, \lceil A \rceil \multimap \lceil B \rceil) \otimes \lceil A \supset B \rceil^{\perp}
\end{aligned}$$

Continuing in such a manner, we can systematically replace all occurrences of $\lfloor \cdot \rfloor$ with occurrences of $\lceil \cdot \rceil$, resulting in the specification in Figure 12. The clauses in this figure, named **NM**, can easily be seen as specifying the introduction and elimination rules for this particular fragment of minimal logic. The usual specification of natural deduction rules for minimal logic [2, 5] has intuitionistic implications replacing the top-level linear implications in Figure 12, but as observed in [10], the choice of which implication to use for these top-level occurrences does not change the set of provable atomic formulas.

As a result of this natural connection between clauses in **LM** and **NM**, the following propositions have direct proofs (see [20] for details).

$$
\begin{array}{llll}
(\supset I) & \lceil A \supset B \rceil^{\perp} \otimes (!\lceil A \rceil \multimap \lceil B \rceil). & (\supset E) & \lceil B \rceil^{\perp} \otimes \lceil A \rceil \otimes \lceil A \supset B \rceil. \\
(\forall_i I) & \lceil \forall_i B \rceil^{\perp} \otimes \forall x \lceil Bx \rceil. & (\forall_i E) & \lceil Bx \rceil^{\perp} \otimes \lceil \forall_i B \rceil. \\
(\cap I) & \lceil A \cap B \rceil^{\perp} \otimes (\lceil A \rceil \mathbin{\&} \lceil B \rceil). & (\cap E_1) & \lceil A \rceil^{\perp} \otimes \lceil A \cap B \rceil. \\
 & & (\cap E_2) & \lceil B \rceil^{\perp} \otimes \lceil A \cap B \rceil.
\end{array}
$$

Figure 12: Specification of the **NM** natural deduction calculus.

Proposition 10. *Let* $\boldsymbol{LM}$ *denote the set of formulas containing Cut, Init, Pos and introduction rules for* $\supset$, $\cap$, *and* $\forall_i$. *Let* $\boldsymbol{NM}$ *be the set of formulas displayed in Figure 12. Then* $!\,\boldsymbol{LM} \equiv\, ![(\&\boldsymbol{NM}) \mathbin{\&} \mathit{Init} \mathbin{\&} \mathit{Cut}]$.

Proposition 11. *If* B *is an object-level formula, then* $\boldsymbol{NM}; \lceil B \rceil \Downarrow$ *if and only if* $\boldsymbol{LM}; \lceil B \rceil \Downarrow$.

A consequence of the last proposition and the adequacy of the encodings of **LM** and **NM** is the mutual relative completeness of natural deduction and sequent calculus: B has a sequent calculus proof if and only if it has a natural deduction proof.

More about using linear logic to specify natural deduction-style proof systems can be found in [26].

4.5. Relating Meta-level Formulas and Object-level Rules

So far, we have shown how to use linear logic formulas of the form

$$\exists x_1 \ldots \exists x_n [(q(\diamond(x_1, \ldots, x_n)))^{\perp} \otimes F]$$

to formally define introduction rules. We have argued that when F is a bipole (with some other restrictions, see Definition 4), then focusing on such a formula yields an inference rule of the encoded logic. For example, if $n = 3$ and the body of the introduction rule is $\lceil x_1 \rceil \otimes (\lfloor x_2 \rfloor \mathbin{\&} \lceil x_3 \rceil)$, then that linear logic formula determines the big-step rule

$$\frac{\Psi; \Gamma_1 \Downarrow \lceil x_1 \rceil \qquad \Psi; \Gamma_2 \Uparrow \lfloor x_2 \rfloor \qquad \Psi; \Gamma_2 \Uparrow \lceil x_3 \rceil}{\Psi; \Gamma_1, \Gamma_2 \Downarrow \lceil x_1 \rceil \otimes (\lfloor x_2 \rfloor \mathbin{\&} \lceil x_3 \rceil)}\ .$$

It is a simple matter to show that every introduction clause corresponds to a specification of a sequent calculus introduction rule. The following example illustrates that if a clause has a body that is not a bipole, then the meta-level can take steps that are not available at the object-level and, as a result, the meta-level encoding of inference rules might not be adequate.

Example 12. Consider the following non-bipolar formulas.

$$\lceil \diamond(A, B, C) \rceil^{\perp} \otimes (\lceil A \rceil \mathbin{\&} (\lceil B \rceil \otimes \lceil C \rceil)) \qquad \lfloor \diamond(A, B, C) \rfloor^{\perp} \otimes (\lfloor A \rfloor \oplus (\lfloor B \rfloor \parr \lfloor C \rfloor))$$

The following inference rules are the natural candidates encoded by these formulas.

$$\frac{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, A \qquad \Gamma_1 \vdash \Delta_1, B \qquad \Gamma_2 \vdash \Delta_2, C}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \diamond(A, B, C)}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \diamond(A, B, C) \vdash \Delta} \qquad\qquad \frac{\Gamma, B, C \vdash \Delta}{\Gamma, \diamond(A, B, C) \vdash \Delta}$$

If Ψ is the set of formulas that contains *Init* and the two non-bipolar formulas above, then the LLF sequent

$$\Psi; \Uparrow \lceil \diamond(A, B, C) \rceil, \lfloor \diamond(A, B, C) \rfloor$$

has essentially two LLF proofs: one focusing on just the *Init* rule and another in which both non-bipolar formulas are focused on prior to focusing on *Init*. On the other hand, there is only one proof of the object-level sequent $\diamond(A, B, C) \vdash \diamond(A, B, C)$, namely, the proof that is just the initial rule. In other words, the initial rule cannot be restricted to just atomic formulas. Thus, the linear logic encoding of these inference rules is not adequate at the level of proofs.

In Sections 6 and 7 we identify additional properties (*cut* and *initial coherence*) that "sensible" introduction clauses should satisfy beyond the requirement to be bipoles.

5. Entailments between cut-free proof system theories

A *proof system theory* is a finite set of linear logic formulas all of which are either *Init*, *Cut*, *Neg*, or *Pos* of Figure 5 or an introduction clause as given by Definition 4 in Section 3.1. Such a proof system theory is *cut-free* if it does not contain the formula *Cut*. In this section, we consider cut-free proof system theories exclusively.

Let $\{C_1, \ldots, C_n\}$ and $\{D_1, \ldots, D_m\}$ be two (cut-free) proof system theories. We wish to determine whether or not the LLF sequent

$$C_1, \ldots, C_n; \cdot \Uparrow \&_{i=1}^{m} D_i^{\perp} \qquad (*)$$

is provable. Notice that if this sequent is provable then any formula entailed by $\{D_1, \ldots, D_m\}$ is also entailed by $\{C_1, \ldots, C_n\}$. Notice also that $(*)$ is provable in LLF iff $C_1, \ldots, C_n; \cdot \Uparrow D_i^{\perp}$ is provable for all $i = 1, \ldots, m$.

We now show that determining if $C_1, \ldots, C_n; \cdot \Uparrow D_i^{\perp}$ is a decidable problem. Proposition 13 proves the decidability of this entailment when D_i is either *Init*, *Neg*, or *Pos*, and Proposition 16 proves the decidability of this entailment when D_i is an introduction rule.

Proposition 13. *If Δ is a (cut-free) proof system theory and D is either Init, Neg, or Pos, then $\Delta; \cdot \Uparrow D^{\perp}$ has an LLF proof if and only if $D \in \Delta$.*

Proof The "if" direction is immediate. To show the converse, we illustrate the case when D is initial, since the cases when D is *Neg* or *Pos* are similar and simpler. The sequent $\Delta; \cdot \Uparrow \forall B(\lfloor B \rfloor \invamp \lceil B \rceil)$ is provable if and only if $\Delta; \lfloor x \rfloor, \lceil x \rceil \Uparrow \cdot$ is provable (for an eigenvariable x). We now prove the following claim: if (*i*) Γ_1 is a multiset subset of the multiset $\{\lfloor x \rfloor, \lceil x \rceil\}$ and (*ii*) Γ_2 is a subset of the set $\{\lfloor x \rfloor, \lceil x \rceil\}$ and (*iii*) $\Gamma_1 \cup \Gamma_2 = \{\lfloor x \rfloor, \lceil x \rceil\}$ then the sequent $\Delta, \Gamma_2; \Gamma_1 \Uparrow \cdot$ is provable in LLF if and only if *Init* is a member of Δ. The "if" part of this claim is trivial to show. For the converse, we proceed by induction by showing that for all natural numbers n, if Γ_1 and Γ_2 satisfy the three conditions above and $\Delta, \Gamma_2; \Gamma_1 \Uparrow \cdot$ has a proof in LLF of height n or less, then *Init* is a member of Δ. Let Ξ be an LLF proof of $\Delta, \Gamma_2; \Gamma_1 \Uparrow \cdot$ of height $n+1$ and consider the last inference rule of Ξ: that rule is a decide rule ($[D_1]$ or $[D_2]$). If the decided formula is the *Init* formula, then the proof can be completed. It is not possible for the decide rule to select an introduction rule since it would be impossible to complete the resulting $\Downarrow$ proof phase. The restrictions on the decide rules (Figure 1) do not allow selecting either $\lfloor x \rfloor$ or $\lceil x \rceil$. Thus, the only remaining cases to consider select either *Pos* or *Neg*: in both of these cases, Ξ has a proper subproof for which the inductive assumption can be applied. ∎

In other words, a structural or initial clause is derivable from a proof system if and only if it is present in that proof system. The following two definitions are used to prove our next theorem.

Definition 14. Let Π be a proof in LLF. The *depth* of Π is the maximum number of decide rules (either $[D_1]$ or $[D_2]$) along any path in Π from the root.

Definition 15. 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 *obj* 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 distinct variables.

Theorem 16. *Let Δ be a (cut-free) proof system theory and let D be an introduction clause. It is decidable whether or not the sequent $\Delta; \cdot \Uparrow D^\perp$ is provable.*

Proof Let $D = \exists \overline{X}.(q(\diamond(X_1, \ldots, X_n))^\perp \otimes F)$. Since F is a bipole without negated atoms, $D^\perp$ is equivalent to a formula of the form

$$\forall \overline{X} \forall \overline{Y}. [\&_{i=1,\ldots,n}[q(\diamond(X_1, \ldots, X_n)) \invamp (\invamp_{j=1,\ldots,m_i} F_{i,j})]]$$

where $F_{i,j}$ are either negated monopoles or the application of ? to a negated monopole. Given the definition of monopoles (Definition 3), such negated monopoles are constructed from negated atoms, ! applied to negated atoms, and the positive connectives. A *component of* $D^\perp$ is a formula C such that either C or $?C$ is a substitution instance of $F_{i,j}$ (for some i, j). The structure of LLF proofs implies that $\Delta; \cdot \Uparrow D^\perp$ has a proof if and only if a collection of sequents is provable: those sequents are all of the form

$$\Delta'; q(\diamond(x_1, \ldots, x_n)), \Psi \Uparrow \cdot, \qquad (**)$$

where Δ' is Δ plus possibly some components of $D^{\perp}$ and Ψ is a multiset of components of $D^{\perp}$. Thus, decidability of proving $\Delta; \cdot \Uparrow D^{\perp}$ reduces to the decidability of proving sequents of the form $(**)$.

On proving such a sequent, it is easy to see that it is not possible to start with the rule $[D_1]$. In fact, by the restriction on $[D_1]$, the focus cannot be $q(\diamond(x_1, \ldots, x_n))$ so it must be a component of $D^{\perp}$. Since these formulas are composed of only positive connectives or negated atoms, that positive phase must terminate with either $[I_1]$ or $[I_2]$ or $[!]$. None of these cases are possible, however: the initial rules are not possible since atoms are members of neither Δ' nor Ψ and no component can contain the negation of $q(\diamond(x_1, \ldots, x_n))$ since $\diamond$ does not occur in the body of an introduction rule. Similarly, the last rule cannot be $[!]$ since the linear context to the left of $\Downarrow$ is not empty (it contains $q(\diamond(x_1, \ldots, x_n))$). Thus, a proof of a sequent of the form $(**)$ must end with a $[D_2]$ rule and that decide rule can select either a structural rule *Neg* or *Pos* or an introduction clause (selecting *Init* is not possible). We consider these two cases below.

Case 1: The sequent $(**)$ is proved as a consequence of a $[D_2]$ on an introduction rule, which has the structure

$$\exists \overline{X}.[q(\diamond(X_1, \ldots, X_n))^{\perp} \otimes C(X_1, \ldots, X_n)].$$

Here $C(X_1, \ldots, X_n)$ is a bipole (restricted to not contain negated atoms). When the positive and then negative phases finish (reading the proof bottom-up) the resulting frontier of sequents will be of the form $\Delta''; \Psi' \Uparrow \cdot$, where Δ'' and Ψ' extend Δ' and Ψ, respectively, by the possible addition of premise atoms. Proofs of such sequents are now rather simple: they can involve deciding on structural clauses or a component of $D^{\perp}$. The application of structural rules can be limited to the number of atomic formulas in Ψ' and selecting a component for focus must immediately yield a proof of depth 1 or 2 (since a component might contain an expression of the form $!\,A^{\perp}$, for atomic A, the focus can be lost before the $!\,A$ is again selected for focus). Thus, the depth of such a proof can be limited to $v + 3$, where v is the number of atomic formulas in Ψ'.

Case 2: In the case that a structural clause is selected for focus at the root of Π, the result of that focus is the sequent $\Delta', q(\diamond(x_1, \ldots, x_n)); \Psi \Uparrow \cdot$. In this case, the atom $q(\diamond(x_1, \ldots, x_n))$ persists in all the premises and can enable focusing on an introduction rule repeated. The following three observations hold in this situation.

i. Selecting an introduction rule for focus transforms the sequent

$$\Delta'', q(\diamond(x_1, \ldots, x_n)); \Psi' \Uparrow \cdot \quad \text{to} \quad \Delta''', q(\diamond(x_1, \ldots, x_n)); \Psi'' \Uparrow \cdot$$

where Δ''' and Ψ'' extend Δ'' and Ψ', respectively, with premise atoms.

ii. Selecting *Pos* or *Neg* for focus transforms the sequent

$$\Delta'', q(\diamond(x_1, \ldots, x_n)); \Psi', A \Uparrow \cdot \quad \text{to} \quad \Delta'', q(\diamond(x_1, \ldots, x_n)), A; \Psi' \Uparrow \cdot.$$

iii. If $\Delta''', q(\diamond(x_1, \ldots, x_n)); \Psi'' \Uparrow \cdot$ is proved by selecting a component C from either the classical or linear context, then the number of atoms in Ψ'' is bounded by the number of atoms in C.

Thus, by (*i*), selecting an introduction rule can make the side formula contexts increase; by (*ii*), selecting a structural rule moves an atom from the bounded to unbounded context; and by (*iii*) selecting a component ends the proof only if the classical context is not too big. Also, notice that while multiplicity of occurrences matters in the multiset Ψ'', it does not matter in the set Δ'''. Thus, the search for a loop-free proof of $\Delta', q(\diamond(x_1, \ldots, x_n)); \Psi \Uparrow \cdot$ must always terminate. ∎

It is easy to see that the sequent[3] $\mathbf{LK} \vdash \mathbf{LJ}$ can be proved with an LLF proof of depth 2. Here we are assuming also that the same symbols are used for the same object-logical connectives in these two theories.

Theorem 16 is concerned with the *derivability* of inference rules: this is different from determining whether or not an inference is *admissible*. Determining admissibility generally requires inductive arguments and, hence, is harder to determine than derivability. In Section 6, we address the admissibility of cut in object-level proof systems.

6. Cut-elimination for cut-coherent systems

We now present a necessary condition for characterizing systems having the cut-elimination property.

Definition 17. A *canonical clause* is an introduction clause restricted so 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 the head variable of S. A *canonical proof system theory* is a set $\mathcal{X}$ of bipoles such that (*i*) the *Init* and *Cut* clauses are members of $\mathcal{X}$, (*ii*) structural clauses (*Pos* and *Neg*) may be members of $\mathcal{X}$, and (*iii*) all other clauses in $\mathcal{X}$ are canonical (introduction) clauses.

Definition 18. Let $\mathcal{X}$ be a canonical proof system theory and $\diamond$ an object-level connective of arity $n \geq 0$. Furthermore, let the formulas

$$\exists \bar{x}(\lfloor \diamond(\bar{x}) \rfloor^{\perp} \otimes F_l) \quad \text{and} \quad \exists \bar{x}(\lceil \diamond(\bar{x}) \rceil^{\perp} \otimes F_r)$$

be the left and right introduction rules for $\diamond$: here, the free variables of F_l and F_r are in the list of variables $\bar{x}$. The object-level connective $\diamond$ has *cut-coherent introduction rules* if the sequent $Cut; \cdot \Uparrow \forall \bar{x}(F_l^{\perp} \invamp F_r^{\perp})$ is provable in LLF. A canonical proof system theory is called *cut-coherent* if all object-level connectives have cut-coherent introduction rules.

[3]Meaning that the set of clauses specifying the **LK** rules entails the conjunction of the clauses specifying **LJ** rules in LLF. Observe that this result implies that every formula provable in LJ is provable in LK.

Example 19. The cut-coherence of the **LJ** specification is established by proving all the following linear logic sequents.

$$
\begin{array}{ll}
(\supset) & Cut; \cdot \Uparrow \forall A \forall B[(\lceil A \rceil^\perp \parr \lfloor B \rfloor^\perp) \parr (\lfloor A \rfloor^\perp \otimes \lceil B \rceil^\perp)] \\
(\cap) & Cut; \cdot \Uparrow \forall A \forall B[(\lfloor A \rfloor^\perp \,\&\, \lfloor B \rfloor^\perp) \parr (\lceil A \rceil^\perp \oplus \lceil B \rceil^\perp)] \\
(\cup) & Cut; \cdot \Uparrow \forall A \forall B[(\lfloor A \rfloor^\perp \oplus \lfloor B \rfloor^\perp) \parr (\lceil A \rceil^\perp \,\&\, \lceil B \rceil^\perp)] \\
(\forall_i) & Cut; \cdot \Uparrow \forall B[\forall x \lfloor Bx \rfloor^\perp \parr \exists x \lceil Bx \rceil^\perp] \\
(\exists_i) & Cut; \cdot \Uparrow \forall B[\exists x \lfloor Bx \rfloor^\perp \parr \forall x \lceil Bx \rceil^\perp] \\
(f_i) & Cut; \cdot \Uparrow 0^\perp \parr \top^\perp \\
(t_i) & Cut; \cdot \Uparrow \top^\perp \parr 0^\perp
\end{array}
$$

All these sequents have simple linear proofs. In general, deciding whether or not canonical systems are cut-coherent involves a simple algorithm (see Theorem 22).

The two following theorems imply that if the specification $\mathcal{X}$ is a cut-coherent proof system theory then the cut inference rule can be eliminated from the object-level proof system that $\mathcal{X}$ encodes.

Theorem 20. *Let the disjoint union $\mathcal{X} \cup \{Cut\}$ be a cut-coherent proof system and let Γ and Δ be a multiset and set, respectively, of atomic formulas. If $\mathcal{X}, Cut, \Gamma; \Delta \Uparrow \cdot$ is provable, then $\mathcal{X}, ACut, \Gamma; \Delta \Uparrow \cdot$ is provable.*

Proof The proof of this theorem follows the usual line of removing cuts on general formulas for cuts on atomic formulas for first-order logic. In particular, we can permute phases in LLF just as we might permute inference rules in the encoded proof system.

Let Ξ be a cut-free LLF proof of the sequent $\mathcal{X}, Cut, \Gamma; \Delta \Uparrow \cdot$. If $[D_2]$ is used to focus on the *Cut* formula in this proof, then the premise of that decide rule is the conclusion of an $[\exists]$ infer rule. Let B be the substitution term used to instantiate the existential quantifier. We say that this occurrence of the $[D_2]$ inference rule is an *object-level cut* which has *cut formula* B. We also define a measure on formulas and proofs as follows: $|B|$ is the natural number denoting the number of occurrences of object-level logical connectives in B and $|\Xi|$ is the multiset of natural numbers $|B|$ for every occurrence of an object-level cut in Ξ with cut formula B. Multisets of natural numbers are well ordered using the usual lifting to multisets of the less-than ordering on the natural numbers [33].

Let Π be a derivation of $\mathcal{X}, Cut, \Gamma; \Delta \Uparrow \cdot$. If all object-level cut formulas occurring in Π are atomic, then we can change the proof Π to be a proof of $\mathcal{X}, ACut, \Gamma; \Delta \Uparrow \cdot$. Thus, assume that Π has a non-atomic cut-formula and consider the highest occurrence of a $[D_2]$ rule that selects such a non-atomic cut-formula. By mimicking the usual arguments for permuting inference rules, we can assume that this cut is a *principle cut* in which the cut-formula is the non-atomic formula $\diamond(B_1,\ldots,B_n)$: that is, Π contains a subproof of the form

$$
\cfrac{\cfrac{\cfrac{\begin{array}{c}\Pi_1\\ \mathcal{X}, Cut, \Psi; \Delta_1 \Downarrow F_l[\bar{B}/\bar{x}]\end{array}}{\mathcal{X}, Cut, \Psi; \Delta_1, \lfloor\diamond(\bar{B})\rfloor \Uparrow \cdot}\ [D_2,\exists,\otimes,I_1]}{\mathcal{X}, Cut, \Psi; \Delta_1 \Downarrow \lfloor\diamond(\bar{B})\rfloor}\ [R\Downarrow, R\Uparrow] \quad \cfrac{\cfrac{\begin{array}{c}\Pi_2\\ \mathcal{X}, Cut, \Psi; \Delta_2 \Downarrow F_r[\bar{B}/\bar{x}]\end{array}}{\mathcal{X}, Cut, \Psi; \Delta_2, \lceil\diamond(\bar{B})\rceil \Uparrow \cdot}\ [D_2,\exists,\otimes,I_1]}{\mathcal{X}, Cut, \Psi; \Delta_2 \Downarrow \lceil\diamond(\bar{B})\rceil}\ [R\Downarrow, R\Uparrow]}{\cfrac{\mathcal{X}, Cut, \Psi; \Delta_1, \Delta_2 \Downarrow \lfloor\diamond(\bar{B})\rfloor \otimes \lceil\diamond(\bar{B})\rceil}{\mathcal{X}, Cut, \Psi; \Delta_1, \Delta_2 \Uparrow \cdot}\ [D_2,\exists]}\ [\otimes]
$$

where Ψ (respectively Δ_1 and Δ_2) is a set (are multisets) of atomic formulas, and Π_1, Π_2 do not contain object-level non-atomic cuts (the expression $\bar{B}$ denotes the list $B_1,\ldots,B_n$ and the expression $[\bar{B}/\bar{x}]$ denotes the simultaneous substitution of B_i for x_i for $i = 1,\ldots,n$). Here there are three occurrences of the $[D_2]$ inference rule: one occurrence encodes the object-level cut and the other two encode the left and right introduction rules for the $\diamond$ connective.

Since $\mathcal{X}$ is a cut-coherent proof system theory the sequent

$$Cut; \cdot \Uparrow \forall\bar{x}(F_l^{\perp} \parr F_r^{\perp})$$

is provable. By instantiating eigenvariables in that proof, we have a proof for $Cut; \cdot \Uparrow F_l[\bar{B}/\bar{x}]^{\perp}, F_r[\bar{B}/\bar{x}]^{\perp}$. Thus, the following three sequents all have cut-free linear logic proofs (using the soundness direction of Theorem 1)

$$\vdash ?\mathcal{X}, ?Cut, ?\Psi, \Delta_1, F_l[\bar{B}/\bar{x}] \qquad \vdash ?\mathcal{X}, ?Cut, ?\Psi, \Delta_2, F_r[\bar{B}/\bar{x}]$$
$$\vdash ?Cut, F_l[\bar{B}/\bar{x}]^{\perp}, F_r[\bar{B}/\bar{x}]^{\perp}$$

By using two instances of linear logic cut, we can conclude that

$$\vdash ?\mathcal{X}, ?Cut, ?\Psi, \Delta_1, \Delta_2$$

has a proof with cut. Applying the cut-elimination process for linear logic will yield a cut-free linear logic proof of the same sequent: the elimination process might instantiate eigenvariables of the proof with arbitrary terms but since the only eigenvariables in these proofs are of type d, the sizes of object-level cut formulas in the resulting cut-free proof does not increase. Using the completeness direction of Theorem 1 (completeness is proved by permuting inference rules), we know that

$$\mathcal{X}, Cut, \Psi; \Delta_1, \Delta_2 \Uparrow \cdot$$

has a proof of smaller measure since we have removed a cut on the formula $\diamond(B_1,\ldots,B_n)$ but replaced with possibly many cuts on smaller formulas.

We can repeatedly perform this rewriting of object-level cuts into linear logic cuts and smaller object-level cuts: the multiset ordering on proofs will force this rewriting process to terminate. ∎

Theorem 21. *Let the disjoint union* $\mathcal{X} \cup \{Cut\}$ *be a cut-coherent proof system and let* $\Gamma_o \longrightarrow \Delta_o$ *be an object-level sequent. If* $\mathcal{X}, ACut; \cdot \Uparrow \lfloor\Gamma_o\rfloor, \lceil\Delta_o\rceil$ *is provable, then* $\mathcal{X}; \cdot \Uparrow \lfloor\Gamma_o\rfloor, \lceil\Delta_o\rceil$ *is provable.*

Proof The usual proof that permutes an atomic cut up in a proof can be applied here. Any occurrence of an instance of $[D_2]$ on the *ACut* formula can be moved up in a proof until it can either be dropped entirely or until one of the premises is proved by an instance of $[D_2]$ on the *Init*: in that case, the proof of the other premise is used as the proof for the conclusion of the cut inference. ∎

We show now that it is decidable to check whether or not a proof system encoding is cut-coherent.

Theorem 22. *Determining whether or not a canonical proof system is cut-coherent is decidable. In particular, determining if the cut clause proves the duality of the introduction rules for a given connective can be achieved by proof search in LLF bounded by the depth $v + 3$ where v is the maximum number of premise atoms in the bodies of the introduction clauses.*

Proof Let P be a cut-coherent proof system theory and let the formulas

$$\exists\bar{x}(\lfloor\diamond(x_1,\ldots,x_i)\rfloor^\perp \otimes F_l) \quad \text{and} \quad \exists\bar{x}(\lceil\diamond(x_1,\ldots,x_i)\rceil^\perp \otimes F_r)$$

be the introduction rules for the object level connective $\diamond$. By cut-coherence, $Cut; \cdot \Uparrow \forall\bar{x}(F_l^\perp \invamp F_r^\perp)$ has a proof Π in LLF. The structure of LLF proofs implies that such a sequent is provable if and only if sequents of the form $Cut, \Gamma; \Delta \Uparrow \cdot$ are provable, where Γ is a set and Δ is a multiset of *components* of $F_l^\perp$ and $F_r^\perp$ (see Theorem 16). In particular, Γ is a set and Δ is a multiset of negated monopoles. Hence deciding the provability of $Cut; \cdot \Uparrow \forall\bar{x}(F_l^\perp \invamp F_r^\perp)$ reduces to determining the decidability of the provability of such sequents.

In any LLF proof of the sequent $Cut, \Gamma; \Delta \Uparrow \cdot$, the action of selecting the *Cut* clause as the focus formula will simply add atoms to the linear context, hence the number of such actions can be limited to the number v of atomic formulas in $\Delta \cup \Gamma$. Furthermore, selecting a component for focus must immediately yield a proof of deep 1 or 2. Thus, the depth of such a proof can be limited to $v + 3$. ∎

As an example, it is possible to prove cut-coherence for **LJ** by bounding proof search at depth 4 during proof search for the formulas in Example 19.

7. Coherent systems

The notion of cut-coherence implies that non-atomic cuts can be replaced by simpler ones: hence, all cuts can removed or reduced to just atomic cuts. A separate argument allows us to also remove atomic cuts. We now consider the dual problem of replacing non-atomic initial rules with atomic initial rules.

Example 23. Consider the sequent system

$$\frac{}{\Gamma, A \vdash \Delta, A}\ Initial \qquad \frac{\Gamma_1 \vdash \Delta_1, A \quad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ Cut$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, \diamond(A,B,C)}\ (\diamond R_1) \qquad \frac{\Gamma \vdash \Delta, B \quad \Gamma \vdash \Delta, C}{\Gamma \vdash \Delta, \diamond(A,B,C)}\ (\diamond R_2)$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, \diamond(A,B,C) \vdash \Delta}\ (\diamond L_1) \qquad \frac{\Gamma, A \vdash \Delta \quad \Gamma, C \vdash \Delta}{\Gamma, \diamond(A,B,C) \vdash \Delta}\ (\diamond L_2)$$

These rules for the connective $\diamond(A, B, C)$ are specified by the clauses

$$\lceil \diamond(A, B, C) \rceil^{\perp} \otimes (\lceil A \rceil \oplus (\lceil B \rceil \mathbin{\&} \lceil C \rceil))$$

$$\lfloor \diamond(A, B, C) \rfloor^{\perp} \otimes (\lfloor A \rfloor \mathbin{\&} \lfloor B \rfloor) \oplus (\lfloor A \rfloor \mathbin{\&} \lfloor C \rfloor)$$

It is easy to see that

$$\mathit{Cut}; \cdot \Uparrow \forall X, Y, Z.((\lceil X \rceil^{\perp} \mathbin{\&} (\lceil Y \rceil^{\perp} \oplus \lceil Z \rceil^{\perp})) \mathbin{⅋} ((\lfloor X \rfloor^{\perp} \oplus \lfloor Y \rfloor^{\perp}) \mathbin{\&} (\lfloor X \rfloor^{\perp} \oplus \lfloor Z \rfloor^{\perp})))$$

is provable and hence the system above is cut-coherent. The initial rule, however, cannot be reduced to just atomic initial rules since the sequent $\diamond(A, B, C) \vdash \diamond(A, B, C)$ would not be provable anymore. As we will see below, this failure is reflected by the fact that the sequent

$$\mathit{Init}; \cdot \Uparrow \forall X, Y, Z.((\lceil X \rceil \oplus (\lceil Y \rceil \mathbin{\&} \lceil Z \rceil)) \mathbin{⅋} ((\lfloor X \rfloor \mathbin{\&} \lfloor Y \rfloor) \oplus (\lfloor X \rfloor \mathbin{\&} \lfloor Z \rfloor)))$$

is not provable.

We now introduce a condition on the meta-level specification of the left and right introductions rules for a connective that guarantees that non-atomic initial rules can be eliminated.

Definition 24. Let $\mathcal{X}$ be a canonical proof system and $\diamond$ an object-level connective of arity $n \geq 0$. Furthermore, let the formulas

$$\exists \bar{x}(\lfloor \diamond(x_1, \ldots, x_n) \rfloor^{\perp} \otimes F_l) \quad \text{and} \quad \exists \bar{x}(\lceil \diamond(x_1, \ldots, x_n) \rceil^{\perp} \otimes F_r)$$

be the left and right introduction clauses for $\diamond$. The object-level connective $\diamond$ has *initial-coherent introduction rules* if

$$\mathit{Init}; \Uparrow \forall \bar{x}(F_l \mathbin{⅋} F_r)$$

is provable in LLF. A canonical system is called *initial-coherent* if all object-level connectives have initial-coherent introduction rules.

It is a simple matter to show that a bounded proof search in LLF yields a decision procedure for determining whether or not an object-level connective has initial-coherent introduction rules: the only occurrences of contractions are with the formula *Init* (whose selection as a focus must immediately end proof search) and with possibly (premise) atoms, which can never be selected for focus. It is also easy to see that initial-coherence does not imply cut-coherence. For example, a canonical proof theory containing the left and right introduction clauses for the object level connective $\diamond$:

$$\lfloor \diamond(A, B) \rfloor^{\perp} \otimes \lfloor A \rfloor \quad \text{and} \quad \lceil \diamond(A, B) \rceil^{\perp} \otimes (\lceil A \rceil \oplus \lceil B \rceil).$$

is initial-coherent but not cut-coherent. In general, we take both of these coherence properties together.

Definition 25. A *cut-coherent* theory that is also *initial-coherent* is called a *coherent* theory.

Recall that the notation $A \equiv B$ is an abbreviation of $(A \multimap B) \mathbin{\&} (B \multimap A)$. If F_l and F_r are the bodies of the left and right introduction rules for the same connective from a coherent proof theory, then $Cut, Init; \Uparrow \forall \bar{x}(F_l^{\perp} \equiv F_r)$ is provable (see Proposition 26). Thus in coherent systems, the body of the left and the right introduction clauses are essentially de Morgan duals of each other. Another way to see this is to first let ng be the following function on linear logic formulas.

1. $ng(\lfloor B \rfloor) = \lceil B \rceil$, $ng(\lceil B \rceil) = \lfloor B \rfloor$;
2. $ng(0) = \top$, $ng(\top) = 0$, $ng(\perp) = 1$, $ng(1) = \perp$;
3. $ng(C \star D) = ng(C) \bar{\star}\, ng(D)$, where $\star$ is a binary linear logic connective and $\bar{\star}$ its dual;
4. $ng(\bullet C) = \bar{\bullet}\, ng(C)$, where $\bullet$ is an unary linear logic connective (*i.e.*, the exponentials and quantifiers) and $\bar{\bullet}$ its dual.

Thus the $ng(\cdot)$ function computes the de Morgan dual of its argument as well as switching between $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$. We will say that the left and right bodies of an introduction clause (F_l and F_r respectively) are *duals* if $\cdot; \cdot \Uparrow \forall \bar{x}(F_l \equiv ng(F_r))$ is provable.

The next proposition shows that coherence implies this duality for left and right inference rules.

Proposition 26. *Let $\mathcal{X}$ be a coherent proof theory and let $\diamond$ be an object-level connective of arity $n \geq 0$ with left and right introduction rules*

$$\exists \bar{x}(\lfloor \diamond(x_1, \ldots, x_n) \rfloor^{\perp} \otimes F_l) \quad \textit{and} \quad \exists \bar{x}(\lceil \diamond(x_1, \ldots, x_n) \rceil^{\perp} \otimes F_r).$$

Then F_r and F_l are duals.

Proof From the definition of cut-coherent, F_l entails $F_r^{\perp}$ in a theory containing *Cut*. Similarly, from the definition of initial-coherence, $F_r^{\perp}$ entails F_l in a theory containing *Init*. Thus, the equivalence $F_r^{\perp} \equiv F_l$ is provable in a theory containing *Cut* and *Init*. As noted in Section 4, such a theory also entails that $\lfloor \cdot \rfloor^{\perp} \equiv \lceil \cdot \rceil$. Thus it follows that F_r and F_l are duals. ∎

Finally, the next theorem states that, in coherent systems, the initial rule can be restricted to its atomic version. For this theorem, we need to axiomatize the meta-level predicate $atomic(\cdot)$. Given that our object-logic is first-order, this axiomatization can be achieved by collecting into the theory Δ all formulas of the form $\exists \bar{x}.(atomic(p(x_1, \ldots, x_n)))^{\perp}$ for every predicate of the object logic (here, n is the arity of the predicate p).

Theorem 27. *Given an object level formula B, let $\mathrm{Init}(B)$ denote the formula $\lfloor B \rfloor^{\perp} \otimes \lceil B \rceil^{\perp}$, let AInit be the formula presented in Figure 5, and let Δ be the theory that axiomatizes the meta-level predicate $\mathrm{atomic}(\cdot)$. If $\mathcal{X}$ is a coherent proof theory, then the sequent $\mathcal{X}, AInit, \Delta; \cdot \Uparrow (\mathrm{Init}(B))^{\perp}$ is provable.*

Proof The proof is by induction on the structure of B. If B is atomic, then the result follows trivially. Suppose $B = \diamond(B_1, \ldots, B_s)$ for some object level connective $\diamond$ of arity $s \geq 0$ with coherent introduction clauses

$$\exists\bar{x}.[\lfloor\diamond(\bar{x})\rfloor^{\perp} \otimes F_l] \text{ and } \exists\bar{x}.[\lceil\diamond(\bar{x})\rceil^{\perp} \otimes F_r]$$

Observe that, since F_r and F_l are bipoles and duals, one is purely positive and the other purely negative, and the exponential ! has atomic scope. Hence the negative formula is equivalent to a formula of the form

$$\forall\bar{X}[\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} F_{i,j}]$$

where the $F_{i,j}$ are either atomic, or the exponential ? applied to an atomic formula, while the positive formula is equivalent to

$$\exists\bar{X}[\oplus_{i=1,\ldots,n} \otimes_{j=1,\ldots,m_i} ng(F_{i,j})].$$

Thus, for proving the sequent $\mathcal{X}, AInit, \Delta; \cdot \Uparrow F_r[\bar{B}/\bar{x}], F_l[\bar{B}/\bar{x}]$ it is sufficient to prove $\mathcal{X}, AInit, \Delta; \cdot \Uparrow \widetilde{F}_{i,j}[\bar{B}/\bar{x}], ng(\widetilde{F}_{i,j}[\bar{B}/\bar{x}])$, for all i, j, where $\bar{B}$ denotes the list $B_1, \ldots, B_s$ and $\widetilde{F}_{i,j}$ represents an instance of $F_{i,j}$. Hence, the result follows by the inductive hypothesis. ∎

8. Related work

In [34, 20], the first author illustrated how linear logic could be used to capture sequent calculus proof systems. The dissertation of the second author [35] and the series of conference papers [14, 13, 15] expanded on this theme. The current paper collects together most of the results of those previous publications.

Multiset rewriting can be modeled on either the right-hand side of the sequent arrow, as it is done in this paper, or on the left-hand side, in which case, such modeling can be captured within *intuitionistic* linear logic. In [12], Pfenning made such a transition and specified a number of sequent calculus proof systems within a fragment of intuitionistic logic. He was then able to provide new proofs of cut elimination for those logics based on those specifications. Those proofs are also obtainable using the automated analysis of the Elf [6] implementation of dependent typed λ-calculus. Later, Cervesato and Pfenning [11] developed the linear logical framework LLF and extended their earlier work to the specification of sequent calculus proof systems for linear logic. In each of these cases, the induction proofs were conducted not in the specification language but in informal and formal external languages. Since intuitionistic linear logic contains neither the linear logic unit $\perp$ nor the linear logic negation $(\cdot)^{\perp}$, their specification language cannot state directly the various dualities (see Section 4) that play a central role in our treatment of sequent calculus. None-the-less, similar specifications can be written using the *focused* intuitionistic proof system LJF [36]: in particular, A. S. Henriksen [37] showed that LJF can be used to encode all of the proof systems that were encoded using LLF in [26].

McDowell and Miller [38] have proposed a *two-level logic* approach to reasoning about proof systems. One level of logic described a sequent calculus proof system (using a restricted intuitionistic logic) and a second level of logic used inductive principles to prove that cut-elimination holds of the sequent calculus specifications. The theorem prover Abella [39] provides an implementation of this two-level logic framework. Our work here could be used to allow for the flexible specification of a range of sequent calculus systems in that theorem prover.

In the setting of classical, propositional logics that use only additive maintenance of context, Avron has provided some necessary and sufficient conditions for sequent calculus specification to satisfy the cut elimination theorem [40]. That work introduced the notion of *coherence* and it is that condition that we extend in Section 6.

In [41], Ciabattoni et al. introduced a systematic procedure to relate large classes of linear logic formulas with equivalent structural inference rules in sequent and hypersequent calculi. In that work, the classes $\mathcal{N}_i$ and $\mathcal{P}_i$ are defined so that $\mathcal{N}_i \subset \mathcal{N}_{i+1}$, $\mathcal{N}_i \subset \mathcal{P}_{i+1}$, $\mathcal{P}_i \subset \mathcal{P}_{i+1}$, $\mathcal{P}_i \subset \mathcal{N}_{i+1}$ and $\mathcal{P}_i$ is built using *positive* connectives, while $\mathcal{N}_i$ is built using *negative* ones. Although the definition is over intuitionistic linear logic without exponentials, it is straightforward to extend these classes to the whole linear logic using monopoles and bipoles. In fact, monopoles are in $\mathcal{N}_1$ while bipoles are in $\mathcal{P}_2$, both with the restriction that ? must have atomic scope.

In [41] there are two main results concerning this classification: 1) every axiom in $\mathcal{N}_2$ is equivalent to a finite set of structural rules; 2) every axiom in $\mathcal{P}_3$ is equivalent to a finite set of hyperstructural rules. It turns out that, in our approach, we can completely characterize a sub-set of formulas in $\mathcal{P}_2$.

Necessary and sufficient conditions for *reductive* cut elimination for single conclusion sequent systems is given by Ciabattoni and Terui in [42]. Their setting can capture a range of propositional logics, including LJ, intuitionistic linear logic extended with knotted structural rules, and the Full Lambek Calculi. As in Avron's work, Ciabattoni and Terui present neither a decision procedure for determining if a proof system falls into their framework nor any automation of the proof of cut elimination. In [41], Ciabattoni, Galatos, and Terui present a systematic procedure to transform large classes of axioms into equivalent structural rules in sequent and hypersequent calculi: a general proof of cut elimination for hypersequents is also given. In that paper, a hierarchy of formulas in intuitionistic linear logic without exponentials is given.

As we discussed in Section 3.4, the adequacy of the encoding of a proof system can be defined into three levels. In this paper we are interested in providing a framework that guarantees the admissibility of the cut and the (non-atomic) initial rules. Since these theorems generally refer to *provability*, we have concerned ourselves with only the most shallow level of adequacy, namely, the level of *relative completeness*. It is, of course, desirable to see if deeper levels of adequacy can be captured via linear logic specification. Nigam and Miller in [26] showed that it is possible to capture a wider range of proof systems (sequent, natural deduction, tableaux, etc) by a generalization of the style of specifications presented in this paper. And in [32] Nigam, Pimentel and Reis proposed using focused linear logic with subexponentials as the meta-logic for specifying proof systems, together with a generalization of the notion of coherence presented in this paper. But it is worthy noticing that, while subexpontentials allow for the specification of systems at the level of *full completeness of derivations*, finding general conditions for verifying properties becomes a tricky task. In fact, the finer is the specification, the harder is the verification.

9. 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 20, 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.

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 [43, 44, 45]: it would be interesting to see if these can be used to capture non-commutative object-level logics in a similar manner as done here.

Finally, while we addressed the question of whether or not an inference rule is derivable from other inference rules, it would be interesting and useful to study the question of whether or not an inference rule is *admissible* in another proof system.

Acknowledgments. This work has been supported in part by the ACI grants Geocal and Rossignol, the INRIA "Equipes Associées" Slimmer, and the Brazilian agencies CNPq and FAPEMIG. We thank Kaustuv Chaudhuri and the anonymous reviewers for their comments on an earlier draft of this paper.

References

[1] D. Miller, G. Nadathur, A logic programming approach to manipulating formulas and programs, in: S. Haridi (Ed.), IEEE Symposium on Logic Programming, San Francisco, 1987, pp. 379–388.
URL `http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/slp87.pdf`

[2] A. Felty, D. Miller, Specifying theorem provers in a higher-order logic programming language, in: Ninth International Conference on Automated Deduction, no. 310 in LNCS, Springer, Argonne, IL, 1988, pp. 61–80.

[3] L. C. Paulson, The foundation of a generic theorem prover, Journal of Automated Reasoning 5 (1989) 363–397.

[4] R. Harper, F. Honsell, G. Plotkin, A framework for defining logics, in: 2nd Symp. on Logic in Computer Science, Ithaca, NY, 1987, pp. 194–204.

[5] R. Harper, F. Honsell, G. Plotkin, A framework for defining logics, Journal of the ACM 40 (1) (1993) 143–184.

[6] F. Pfenning, Elf: A language for logic definition and verified metaprogramming, in: 4th Symp. on Logic in Computer Science, Monterey, CA, 1989, pp. 313–321.

[7] G. Nadathur, D. Miller, An Overview of λProlog, in: Fifth International Logic Programming Conference, MIT Press, Seattle, 1988, pp. 810–827. URL http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/iclp88.pdf

[8] F. Pfenning, C. Schürmann, System description: Twelf — A meta-logical framework for deductive systems, in: H. Ganzinger (Ed.), 16th Conf. on Automated Deduction (CADE), no. 1632 in LNAI, Springer, Trento, 1999, pp. 202–206.

[9] T. Nipkow, L. C. Paulson, M. Wenzel, Isabelle/HOL: A Proof Assistant for Higher-Order Logic, Springer, 2002, lNCS Tutorial 2283.

[10] J. Hodas, D. Miller, Logic programming in a fragment of intuitionistic linear logic, Information and Computation 110 (2) (1994) 327–365.

[11] I. Cervesato, F. Pfenning, A Linear Logical Framework, Information & Computation 179 (1) (2002) 19–75.

[12] F. Pfenning, Structural cut elimination, in: 10th Symp. on Logic in Computer Science, IEEE Computer Society Press, San Diego, California, 1995, pp. 156–166.

[13] D. Miller, E. Pimentel, Linear logic as a framework for specifying sequent calculus, in: J. van Eijck, V. van Oostrom, A. Visser (Eds.), Logic Colloquium '99: Proceedings of the Annual European Summer Meeting of the Association for Symbolic Logic, Lecture Notes in Logic, A K Peters Ltd, 2004, pp. 111–135.

[14] D. Miller, E. Pimentel, Using linear logic to reason about sequent systems, in: U. Egly, C. G. Fermüller (Eds.), International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, Vol. 2381 of LNCS, Springer, 2002, pp. 2–23.

[15] E. Pimentel, D. Miller, On the specification of sequent systems, in: LPAR 2005: 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, no. 3835 in LNAI, 2005, pp. 352–366.

[16] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1987) 1–102.

[17] A. Church, A formulation of the simple theory of types, J. of Symbolic Logic 5 (1940) 56–68.

[18] J.-M. Andreoli, Logic programming with focusing proofs in linear logic, J. of Logic and Computation 2 (3) (1992) 297–347.

[19] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, Uniform proofs as a foundation for logic programming, Annals of Pure and Applied Logic 51 (1991) 125–157.

[20] D. Miller, Forum: A multiple-conclusion specification logic, Theoretical Computer Science 165 (1) (1996) 201–232.

[21] J.-M. Andreoli, Focussing and proof construction, Annals of Pure and Applied Logic 107 (1) (2001) 131–163.

[22] J.-Y. Girard, Light linear logic, Information and Computation 143 (2) (1998) 175–204.

[23] V. Nigam, D. Miller, Algorithmic specifications in linear logic with subexponentials, in: ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP), 2009, pp. 129–140.
URL `http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/ppdp09.pdf`

[24] V. Nigam, E. Pimentel, G. Reis, Specifying proof systems in linear logic with subexponentials, in: Proceedings of the Fifth Logical and Semantic Frameworks, with Applications Workshop (LSFA 2010), Vol. 269 of ENTCS, 2011, pp. 109–123.

[25] A. Guglielmi, A system of interaction and structure, ACM Trans. on Computational Logic 8 (1) (2007) 1–64.

[26] V. Nigam, D. Miller, A framework for proof systems, J. of Automated Reasoning 45 (2) (2010) 157–188.
URL `http://springerlink.com/content/m12014474287n423/`

[27] G. Gentzen, Investigations into logical deduction, in: M. E. Szabo (Ed.), The Collected Papers of Gerhard Gentzen, North-Holland, Amsterdam, 1969, pp. 68–131, translation of articles that appeared in 1934-35.

[28] V. Danos, J.-B. Joinet, H. Schellinx, LKT and LKQ: sequent calculi for second order logic based upon dual linear decompositions of classical implication, in: J.-Y. Girard, Y. Lafont, L. Regnier (Eds.), Advances in Linear Logic, no. 222 in London Mathematical Society Lecture Note Series, Cambridge University Press, 1995, pp. 211–224.

[29] A. S. Troelstra, H. Schwichtenberg, Basic Proof Theory, Cambridge University Press, 1996.

[30] P. Lincoln, A. Scedrov, N. Shankar, Linearizing intuitionistic implication, in: Annals of Pure and Applied Logic, 1993, pp. 151–177.

[31] R. Dyckhoff, Contraction-free sequent calculi for intuitionistic logic, J. of Symbolic Logic 57 (3) (1992) 795–807.

[32] V. Nigam, E. Pimentel, G. Reis, An extended framework for specifying and reasoning about proof systems, submitted to Logical Journal of the IGPL (2012).

[33] N. Dershowitz, Z. Manna, Proving termination with multiset orderings, Communications of the ACM 22 (8) (1979) 465–476.

[34] D. Miller, A multiple-conclusion meta-logic, in: S. Abramsky (Ed.), 9th Symp. on Logic in Computer Science, IEEE Computer Society Press, Paris, 1994, pp. 272–281.

[35] E. G. Pimentel, Lógica linear e a especificação de sistemas computacionais, Ph.D. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, M.G., Brasil, written in English (Dec. 2001).

[36] C. Liang, D. Miller, Focusing and polarization in linear, intuitionistic, and classical logics, Theoretical Computer Science 410 (46) (2009) 4747–4768. URL `http://dx.doi.org/10.1016/j.tcs.2009.07.041`

[37] A. S. Henriksen, Adversarial models for cooperative interactions, Ph.D. thesis, University of Copenhagen (Dec. 2011).

[38] R. McDowell, D. Miller, Reasoning with higher-order abstract syntax in a logical framework, ACM Trans. on Computational Logic 3 (1) (2002) 80–136.

[39] A. Gacek, D. Miller, G. Nadathur, A two-level logic approach to reasoning about computations, accepted to the J. of Automated Reasoning (Aug. 2010).
URL `http://arxiv.org/abs/0911.2993`

[40] A. Avron, I. Lev, Canonical propositional Gentzen-type systems, in: R. Goré, A. Leitsch, T. Nipkow (Eds.), IJCAR 2001, Vol. 2083 of LNAI, Springer, 2001, pp. 529–544.

[41] A. Ciabattoni, N. Galatos, K. Terui, From axioms to analytic rules in nonclassical logics, in: 23th Symp. on Logic in Computer Science, IEEE Computer Society Press, 2008, pp. 229–240.

[42] A. Ciabattoni, K. Terui, Towards a semantic characterization of cut-elimination, Studia Logica 82 (1) (2006) 95–119.

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

[44] A. Guglielmi, L. Straßburger, Non-commutativity and MELL in the calculus of structures, in: L. Fribourg (Ed.), CSL 2001, Vol. 2142 of LNCS, 2001, pp. 54–68.

[45] C. Retoré, Pomset logic: a non-commutative extension of classical linear logic, in: Proceedings of TLCA, Vol. 1210, 1997, pp. 300–318.