Using linear logic to reason about sequent systems

Dale Miller, Penn State University
(Sept 2002: INRIA and École Polytechnique)

Elaine Pimentel, Departamento de Matemática,
Universidade Federal de Minas Gerais, Belo Horizonte Brazil

Outline

1. Metalogical settings for specifying proof systems.
2. Some generalizations of multiset rewriting in linear logic
3. Representing sequents and inference rules
4. Entailment between encodings of proof systems
5. Establishing object-level cut elimination

Intuitionistic-based frameworks and natural deduction

Higher-order logics and dependently typed λ-calculi based on intuitionistic logic have been proposed for encoding natural deduction systems.

- higher-order hereditary Harrop formulas: Isabelle, λProlog, etc.
- LF: Twelf, etc.

$$\dfrac{\begin{array}{c}(A)\\ \vdots \\ B\end{array} \quad C}{D} \qquad (\mathit{prove}\ A \supset \mathit{prove}\ B) \wedge \mathit{prove}\ C \supset \mathit{prove}\ D.$$

Advantages of using meta-logics and frameworks

Bound variables — in formulas and in proofs (eigenvariable) — are treated uniformly and declaratively by the meta-level (higher-order abstract syntax is generally supported).

Meta-level β-normalization can directly provide object-level substitution.

When reasoning about specifications, "substitution lemmas" often come for free.

Proof search in intuitionistic logic is well studied and has several robust implementations.

Which framework for specifying sequent calculus?

Clearly, sequents can be encoded into existing frameworks by representing them as pairs of lists of formulas, etc.

But sequent calculus has numerous *dualities*:

left	right
positive	negative
initial	cut
synchronous	asynchronous

A framework should account for such dualities directly (problematic in intuitionistic logic).

Structural rules play a significant role in defining logical connectives in sequent calculi.

Sequent calculus seems to be more general than natural deduction.

Linear logic makes a good candidate: it has a involutive negation, allows contraction and weakening to be controlled, and refines intuitionistic logic.

The Forum presentation of linear logic

Three abstract logic programming languages:

hereditary Harrop	$\top, \&, \supset, \forall$
Lolli	$\top, \&, \supset, \forall, \quad \multimap$
Forum	$\top, \&, \supset, \forall, \quad \multimap, \quad \bot, \parr, ?$

The Forum set of connectives is complete for linear logic, in the sense that all other linear logic connectives can be defined from these.

Proof search using this collection of connectives can be restricted so that simple goal-directed proof search (using the technical device of multiple-conclusion uniform proofs) is complete.

Flat Forum

For the purpose of specifying sequent calculus, we need only a subset of Forum.

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

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

is called a *flat clause* if $G_1, \ldots, G_m$ are flat goals, $A_1, \ldots, A_n$ are atomic formulas, and occurrences of the symbol $\hookrightarrow$ are either occurrences of $\multimap$ or $\supset$.

The formula $A_1 \parr \cdots \parr A_n$ is the *head* of such a clause, while for each $i = 1, \ldots, m$, the formula G_i is a *body* of this clause. If $n = 0$, then we write the head as simply $\perp$ and say that the head is *empty*.

Negation $B^\perp$ is equivalent to $B \multimap \perp$.

We sometimes use *uncurried* clauses via the logical equivalences:

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

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

† Provided x is not free in H.

Flat Forum sequents

Sequents in full Forum (linear logic) are of the form $\Psi; \Delta \longrightarrow \Gamma; \Upsilon$ where Ψ and Υ are sets of formulas that can be used unbounded number of times and Δ and Γ are multisets of formulas that are bounded in their use.

The logic program is generally identified with Ψ.

When using flat Forum, the zone Ψ does not change in proof search and Δ will be empty. Thus, we do not write the left-hand side of Forum sequents.

The sequent

$$\longrightarrow B_1, \ldots, B_n; C_1, \ldots, C_m$$

is related to the linear logic formula

$$B_1 \parr \ldots \parr B_n \parr ?\, C_1 \parr \ldots \parr ?\, C_m.$$

Backchaining and multiset rewriting

Multiset rewriting can be captured naturally in proof search. Assume, for example, that the clause

$$a \parr b \multimapinv c \parr d \parr e.$$

is a member of the logic program specification (in Ψ). Consider the following proof fragment.

$$\dfrac{\dfrac{\dfrac{\dfrac{\longrightarrow c, d, e, \Gamma; \Upsilon}{\longrightarrow c, d \parr e, \Gamma; \Upsilon}}{\longrightarrow c \parr d \parr e, \Gamma; \Upsilon} \qquad \dfrac{\dfrac{}{\xrightarrow{a} a; \Upsilon} \quad \dfrac{}{\xrightarrow{b} b; \Upsilon}}{\xrightarrow{a \parr b} a, b; \Upsilon}}{\xrightarrow{c \parr d \parr e \multimap a \parr b} a, b, \Gamma; \Upsilon}}{\longrightarrow a, b, \Gamma; \Upsilon}\ \mathit{decide}\,!$$

We can interpret this proof fragment as a reduction of the multiset a, b, Γ to the multiset c, d, e, Γ by backchaining on the clause displayed above.

Members of Υ are considered permanent members of the multiset, as well.

Splitting and copying of contexts

Backchaining on the clause $G_1 \multimap G_2 \supset G_3 \multimap B_1 \parr B_2$ (which is logically equivalent to $(G_1 \otimes ! G_2 \otimes G_3) \multimap B_1 \parr B_2$) to prove the sequent

$$\longrightarrow B_1, B_2, \mathcal{A}; \Upsilon$$

yields an attempt to prove the three sequents

$$\longrightarrow G_1, \mathcal{A}_1; \Upsilon \qquad \longrightarrow G_2; \Upsilon \qquad \longrightarrow G_3, \mathcal{A}_2; \Upsilon$$

where $\mathcal{A}$ is *split* into $\mathcal{A}_1$ and $\mathcal{A}_2$.

Backchaining on the clause $G_1 \mathbin{\&} G_2 \multimap B$ to prove the sequent

$$\longrightarrow B, \mathcal{A}; \Upsilon$$

yields an attempt to prove the two sequents

$$\longrightarrow G_1, \mathcal{A}; \Upsilon \qquad \longrightarrow G_2, \mathcal{A}; \Upsilon.$$

Here, context is *copied*.

Encoding sequents

Let $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ be two meta-level predicates, both of type $\mathit{bool} \rightarrow o$, used to identify which object-level formulas appear on the left and right of the sequent arrow. The ? modal is used to mark the formulas to which weakening and contraction can be applied.

Consider encoding the object-level sequent $B_1, \ldots, B_n \longrightarrow C_1, \ldots, C_m$ $(n, m \geq 0)$ as a meta-level formula or a meta-level sequent. Examples of encodings might be the following.

Linear scheme: $\lfloor B_1\rfloor \parr \cdots \parr \lfloor B_n\rfloor \parr \lceil C_1\rceil \parr \cdots \parr \lceil C_m\rceil$ or

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

Classical scheme: $?\lfloor B_1\rfloor \parr \cdots \parr ?\lfloor B_n\rfloor \parr ?\lceil C_1\rceil \parr \cdots \parr ?\lceil C_m\rceil$ or

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

Intuitionistic scheme: $?\lfloor B_1\rfloor \parr \cdots \parr ?\lfloor B_n\rfloor \parr \lceil C_1\rceil \parr \cdots \parr \lceil C_m\rceil$ or

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

Encoding additive and multiplicative inference rules

Consider the **additive introduction rules** for conjunction.

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

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

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

Consider the **multiplicative inference rules** for this connective.

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

These two rules can be encoded using the following Forum clauses.

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

Notice that the clause for right introduction could be written equivalently in linear logic as

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

Encoding quantifier introduction rules

Using quantification at higher-order types, it is a simple matter to encode the inference rules for object-level quantifiers.

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

Here, the symbol $\forall$ is used for both meta-level and object-level quantification: at the object-level $\forall$ has the type $(i \rightarrow \mathit{bool}) \rightarrow \mathit{bool}$. Thus the variable B above has the type $i \rightarrow \mathit{bool}$.

Meta-level treatment of substitution and eigenvariables directly implements the appropriate restrictions at the object-level.

Notice that the clause for $(\forall L)$ in uncurried form is

$$\lfloor \forall B \rfloor \circ\!- \exists x \lfloor Bx \rfloor.$$

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

The cut and initial rules

Until now, all clauses are introduction rules have heads that are (meta-level) atomic formulas. Encodings of **cut** and **initial** rules are different.

The initial rule

$$\frac{}{B \longrightarrow B}$$

is encoded using the clause

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

This clause has two atoms in its head and none in its body. The cut rule

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

can be specified simply as the clause

$$(Cut) \qquad \bot \multimapinv \lfloor B \rfloor \multimapinv \lceil B \rceil.$$

This clause has an empty head and two atoms in its body. Other cut-rules are possible:

$$\bot \multimapinv ?\lfloor B \rfloor \multimapinv \lceil B \rceil \qquad \bot \multimapinv \lfloor B \rfloor \multimapinv ?\lceil B \rceil \qquad \bot \multimapinv ?\lfloor B \rfloor \multimapinv ?\lceil B \rceil$$

Cut and initial provide dual information

The initial formula $\lfloor B \rfloor \parr \lceil B \rceil$ is logically equivalent to $\lfloor B \rfloor^{\perp} \multimap \lceil B \rceil$.

The cut formula $\perp \multimapinv \lfloor B \rfloor \multimapinv \lceil B \rceil$ is logically equivalent to $\lceil B \rceil \multimap \lfloor B \rfloor^{\perp}$.

Taken together, we have (the not surprising fact) that left and right are duals of each other:

$$\lceil B \rceil \equiv \lfloor B \rfloor^{\perp}.$$

Of course, if the cut and initial rules involve some modals (as in intuitionistic or classical encodings of sequents), then this equivalence will also involve modals.

Advantages of such encodings

• The Forum specifications do not deal with context explicitly (side formulas): they only mention the formulas that are directly involved in the inference rule.

• The distinction between additive and multiplicative inference rules is achieved using the appropriate linear logic connective.

• Object-level quantifiers and substitution is handled directly by the meta-logic.

• The structural rules of contraction and thinning can be captured together using the ? modal.

• Since the encodings yield abstract logic programming, procedures for proof search and unification in linear logic can be used to help fashion implementations of object-logics.

• Since the encoding of proof systems is natural and direct, we hope to be able to use the rich meta-theory of linear logic to help draw conclusions about object-level proof systems.

Disadvantages of such encodings

• Since the meta-level is commutative, **non-commutative proof systems** cannot be encoded directly. One might turn this into a test: can a proposed non-commutative logic be used at the meta-level to capture non-commutative object-logics?

• Logics that require hypersequents for their characterized seem unlikely candidates for this framework.

• This kind of work generally only captures “conventional” proof systems, not the avant guard ones done at, say, Tableaux.

Specification of the *LK* sequent calculus

$(\supset L)$	$\lfloor A \supset B \rfloor \multimapinv ?\lceil A \rceil \multimapinv ?\lfloor B \rfloor.$	$(\supset R)$	$\lceil A \supset B \rceil \multimapinv ?\lfloor A \rfloor \parr ?\lceil B \rceil.$
$(\wedge L_1)$	$\lfloor A \wedge B \rfloor \multimapinv ?\lfloor A \rfloor.$	$(\wedge R)$	$\lceil A \wedge B \rceil \multimapinv ?\lceil A \rceil \mathbin{\&} ?\lceil B \rceil.$
$(\wedge L_2)$	$\lfloor A \wedge B \rfloor \multimapinv ?\lfloor B \rfloor.$	$(\vee R_1)$	$\lceil A \vee B \rceil \multimapinv ?\lceil A \rceil.$
$(\vee L)$	$\lfloor A \vee B \rfloor \multimapinv ?\lfloor A \rfloor \mathbin{\&} ?\lfloor B \rfloor.$	$(\vee R_2)$	$\lceil A \vee B \rceil \multimapinv ?\lceil B \rceil.$
$(\forall L)$	$\lfloor \forall B \rfloor \multimapinv ?\lfloor Bx \rfloor.$	$(\forall R)$	$\lceil \forall B \rceil \multimapinv \forall x\, ?\lceil Bx \rceil.$
$(\exists L)$	$\lfloor \exists B \rfloor \multimapinv \forall x\, ?\lfloor Bx \rfloor.$	$(\exists R)$	$\lceil \exists B \rceil \multimapinv ?\lceil Bx \rceil.$
(fL)	$\lfloor f \rfloor \multimapinv \top.$	(tR)	$\lceil t \rceil \multimapinv \top.$
(Cut)	$\bot \multimapinv ?\lfloor B \rfloor \multimapinv ?\lceil B \rceil.$	$(Initial)$	$\lfloor B \rfloor \parr \lceil B \rceil.$

Specification of the LJ sequent calculus

$(\supset L)$	$\lfloor A \supset B \rfloor \multimapinv \lceil A \rceil \multimapinv ?\lfloor B \rfloor.$	$(\supset R)$	$\lceil A \supset B \rceil \multimapinv ?\lfloor A \rfloor \parr \lceil B \rceil.$
$(\wedge L_1)$	$\lfloor A \wedge B \rfloor \multimapinv ?\lfloor A \rfloor.$	$(\wedge R)$	$\lceil A \wedge B \rceil \multimapinv \lceil A \rceil \mathbin{\&} \lceil B \rceil.$
$(\wedge L_2)$	$\lfloor A \wedge B \rfloor \multimapinv ?\lfloor B \rfloor.$	$(\vee R_1)$	$\lceil A \vee B \rceil \multimapinv \lceil A \rceil.$
$(\vee L)$	$\lfloor A \vee B \rfloor \multimapinv ?\lfloor A \rfloor \mathbin{\&} ?\lfloor B \rfloor.$	$(\vee R_2)$	$\lceil A \vee B \rceil \multimapinv \lceil B \rceil.$
$(\forall L)$	$\lfloor \forall B \rfloor \multimapinv ?\lfloor Bx \rfloor.$	$(\forall R)$	$\lceil \forall B \rceil \multimapinv \forall x \lceil Bx \rceil.$
$(\exists L)$	$\lfloor \exists B \rfloor \multimapinv \forall x\, ?\lfloor Bx \rfloor.$	$(\exists R)$	$\lceil \exists B \rceil \multimapinv \lceil Bx \rceil.$
(fL)	$\lfloor f \rfloor \multimapinv \top.$	(tR)	$\lceil t \rceil \multimapinv \top.$
(Cut)	$\bot \multimapinv ?\lfloor B \rfloor \multimapinv \lceil B \rceil.$	$(Initial)$	$\lfloor B \rfloor \parr \lceil B \rceil.$

Introducing polarities

$$\begin{aligned} (Pos) \quad & \lfloor B \rfloor \multimapinv ?\lfloor B \rfloor. \\ (Neg) \quad & \lceil B \rceil \multimapinv ?\lceil B \rceil. \end{aligned}$$

The converses of the Pos and Neg implications are, of course, linear logic theorems.

If one studies the LU (Logic of Unity) logic of Girard, these two clauses are applied not to all formula but only to certain formulas. Controlling polarity makes it possible for classical, intuitionistic, and linear logic to co-exist in one logic. The paper in the proceedings contains a new proof of cut-elimination of LU.

Modular presentations of classical and intuitionistic logics

The essential difference between the theories LJ and LK is the different set of occurrences of the ? modal. Let LK_0 and LJ_0 be the result of removing the cut and initial rules as well as deleting the ? modal from the LK and LJ.

Define the two new theories

$$LJ' = LJ_0 \cup \{Cut, Initial, Pos_2\} \text{ and}$$

$$LK' = LK_0 \cup \{Cut, Initial, Pos_2, Neg_2\}.$$

While LJ' is a strengthening of LJ, they can both prove the same object-level, intuitionistic sequents. Similarly for LK' and LK.

Collapsing of modal prefixes

The Cut and Initial rules of *LK* prove the equivalences

$$\forall B.\, ?\lceil B\rceil \equiv (?\lfloor B\rfloor)^{\perp} \qquad \forall B.\, ?\lceil B\rceil \equiv\, !\lceil B\rceil \qquad \forall B.\, ?\lfloor B\rfloor \equiv\, !\lfloor B\rfloor.$$

The Cut and Initial rules of *LJ* prove the equivalences

$$\forall B.\lceil B\rceil \equiv (?\lfloor B\rfloor)^{\perp} \qquad \forall B.\lceil B\rceil \equiv\, !\lceil B\rceil.$$

In the cases of *LJ* and *LK*, that duality forces the collapse of some of modals.

As is well known, linear logic has 7 distinct modalities:

$$\text{empty}, \quad !, \quad ?, \quad ?!, \quad !?, \quad !?!, \quad ?!?\,.$$

In the *LK* theory, however, all those modals collapse into just two when applied to a $\lfloor\cdot\rfloor$-atom or a $\lceil\cdot\rceil$-atom. In the presence of *LJ*, they collapse to four when applied to $\lceil\cdot\rceil$-atoms.

Such collapsing limits the distinction available for controlling the use of formulas during proof search.

The calculus *LKQ* and *LKT* calculi

$$\begin{array}{llll}
(\supset L) & \lfloor A \supset B \rfloor \multimapinv \lceil A \rceil \Leftarrow ?\lfloor B \rfloor. & (\supset R) & \lceil A \supset B \rceil \Leftarrow ?\lfloor A \rfloor \parr ?\lceil B \rceil. \\
(\forall L) & \lfloor \forall B \rfloor \Leftarrow ?\lfloor Bx \rfloor. & (\forall R) & \lceil \forall B \rceil \Leftarrow \forall x\, ?\lceil Bx \rceil. \\
(Cut) & \bot \multimapinv \lceil A \rceil \multimapinv ?\lfloor A \rfloor. & (Initial) & \lfloor A \rfloor \parr \lceil A \rceil. \\
 & \bot \multimapinv ?\lceil A \rceil \Leftarrow ?\lfloor A \rfloor. & &
\end{array}$$

$$\begin{array}{llll}
(\supset L) & \lfloor A \supset B \rfloor \Leftarrow ?\lceil A \rceil \multimapinv \lfloor B \rfloor. & (\supset R) & \lceil A \supset B \rceil \multimapinv ?\lfloor A \rfloor \parr ?\lceil B \rceil. \\
(\forall L) & \lfloor \forall B \rfloor \multimapinv \lfloor Bx \rfloor. & (\forall R) & \lceil \forall B \rceil \multimapinv \forall x\, ?\lceil Bx \rceil. \\
(Cut) & \bot \multimapinv ?\lceil A \rceil \multimapinv \lfloor A \rfloor. & (Initial) & \lfloor A \rfloor \parr \lceil A \rceil. \\
 & \bot \Leftarrow ?\lceil A \rceil \multimapinv ?\lfloor A \rfloor. & &
\end{array}$$

See: Danos, Joinet, and Schellinx, *LKQ and LKT: sequent calculi for second order logic based upon dual linear decompositions of classical implication*, Workshop on Linear Logic 1993.

A non-standard inference system: *IIL**

$$
\begin{array}{ll}
(\mathit{Initial}) & \lfloor A \rfloor \parr \lceil A \rceil \multimapinv \top \multimapinv \mathit{atomic}(A). \\
(\supset R) & \lceil B \supset C \rceil \multimapinv \lceil B \rceil \parr \lfloor C \rfloor. \\
(\supset 1L) & \lfloor A \supset B \rfloor \parr \lceil D \rceil \multimapinv \lceil A \rceil \mathbin{\&} (\lfloor B \rfloor \parr \lceil D \rceil) \multimapinv \mathit{atomic}(A). \\
(\supset 2L) & \lfloor (A \supset B) \supset C \rfloor \parr \lceil D \rceil \multimapinv (\lfloor B \supset C \rfloor \parr \lceil A \supset B \rceil) \mathbin{\&} (\lfloor C \rfloor \parr \lceil D \rceil).
\end{array}
$$

Due to Dyckhoff; Lincoln, Scedrov, and Shankar [APAL93]; and several others.

Deriving NJ from LJ

From LJ we have:

$$\forall B.\lceil B\rceil^{\perp} \equiv ?\lfloor B\rfloor \qquad \forall B.\lceil B\rceil \equiv !\lceil B\rceil \qquad \forall B.\lfloor B\rfloor \equiv ?\lfloor B\rfloor$$

Thus, all occurrences of $\lfloor B\rfloor$ and $?\lfloor B\rfloor$ can be replaced by $\lceil B\rceil^{\perp}$. The introduction rules for implication

$$\begin{array}{ll} (\supset L) & \lfloor A \supset B\rfloor \multimapinv \lceil A\rceil \multimapinv ?\lfloor B\rfloor. \\ (\supset R) & \lceil A \supset B\rceil \multimapinv ?\lfloor A\rfloor \parr \lceil B\rceil. \end{array}$$

are thus transformed to

$$\begin{array}{c} \lceil A \supset B\rceil^{\perp} \multimapinv \lceil A\rceil \multimapinv \lceil B\rceil^{\perp}. \\ \lceil A \supset B\rceil \multimapinv \lceil A\rceil^{\perp} \parr \lceil B\rceil. \end{array}$$

which are equivalent to

$$\begin{array}{lr} (\supset E) & \lceil B\rceil \multimapinv \lceil A\rceil \multimapinv \lceil A \supset B\rceil \\ (\supset I) & \lceil A \supset B\rceil \multimapinv \lceil A\rceil \supset \lceil B\rceil. \end{array}$$

These are the usual $\supset$ elimination rule of natural deduction.

Disjunction and existential in LJ

The story for the disjunction and existential is (predictably) more complicated.

$$
\begin{array}{ll}
(\vee R) & \lceil A \vee B \rceil \multimapinv \lceil A \rceil \oplus \lceil B \rceil. \\
(\vee L) & \lfloor A \vee B \rfloor \multimapinv ?\lfloor A \rfloor \mathbin{\&} ?\lfloor B \rfloor. \\
(\exists R) & \lceil \exists B \rceil \multimapinv \lceil Bx \rceil. \\
(\exists L) & \lfloor \exists B \rfloor \multimapinv \forall x.\, ?\lfloor Bx \rfloor).
\end{array}
$$

$$
\begin{array}{ll}
(\vee I)' & \lceil A \vee B \rceil \multimapinv \lceil A \rceil \oplus \lceil B \rceil. \\
(\vee E)' & \bot \multimapinv \lceil A \vee B \rceil \multimapinv (\lceil A \rceil \supset \bot) \mathbin{\&} (\lceil B \rceil \supset \bot). \\
(\exists I)' & \lceil \exists B \rceil \multimapinv \lceil Bx \rceil. \\
(\exists E)' & \bot \multimapinv \lceil \exists B \rceil \multimapinv \forall x. \lceil Bx \rceil \supset \bot.
\end{array}
$$

$$
\begin{array}{ll}
(\vee I) & \lceil A \vee B \rceil \multimapinv \lceil A \rceil \oplus \lceil B \rceil. \\
(\vee E) & \lceil E \rceil \multimapinv \lceil A \vee B \rceil \multimapinv (\lceil A \rceil \supset \lceil E \rceil) \multimapinv (\lceil B \rceil \supset \lceil E \rceil). \\
(\exists I) & \lceil \exists B \rceil \multimapinv \lceil Bx \rceil. \\
(\exists E) & \lceil E \rceil \multimapinv \lceil \exists B \rceil \multimapinv \forall x. \lceil Bx \rceil \supset \lceil E \rceil.
\end{array}
$$

Some results about encoded proofs systems

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

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

where $n, m \geq 0$, $\diamond$ is an object-level connective of arity n ($n \geq 0$), and atoms occurring in a body of this clause are either of the form $p(x_i)$ or $p(x_i(y))$. Here, p and q are either $\lfloor\cdot\rfloor$ or $\lceil\cdot\rceil$.

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

Coherent proof systems

Definition: Write the all the left and right introduction rules for the connective $\diamond$ in the uncurried form as

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

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

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

To show, for example, that LJ is coherent, the following must be proved.

$$\begin{array}{ll}
(\supset) & !\,Cut_2 \vdash \forall A \forall B[(?\lfloor A\rfloor \oplus ?\lceil B\rceil) \multimap (\lceil A\rceil \mathbin{\&} \lceil B\rceil) \multimap \bot] \\
(\wedge) & !\,Cut_2 \vdash \forall A \forall B[(\lceil A\rceil \otimes ?\lfloor B\rfloor) \multimap (?\lfloor A\rfloor \parr \lceil B\rceil) \multimap \bot] \\
(\vee) & !\,Cut_2 \vdash \forall A \forall B[(?\lfloor A\rfloor \mathbin{\&} ?\lfloor B\rfloor) \multimap (\lceil A\rceil \oplus \lceil B\rceil) \multimap \bot] \\
(\forall) & !\,Cut_2 \vdash \forall B[\exists x(?\lfloor Bx\rfloor) \multimap \forall x\lceil Bx\rceil \multimap \bot] \\
(\exists) & !\,Cut_2 \vdash \forall B[\forall x(?\lfloor Bx\rfloor) \multimap \exists x\lceil Bx\rceil \multimap \bot] \\
(t) & !\,Cut_2 \vdash 0 \multimap \top \multimap \bot \\
(f) & !\,Cut_2 \vdash \top \multimap 0 \multimap \bot
\end{array}$$

All are provable in Forum easily. Here, Cut_2 is $\forall B(\lceil B\rceil \multimap ?\lfloor B\rfloor \multimap \bot)$.

Derivability of one proof system from another

Theorem: If $\mathcal{P}$ is a coherent proof system and $\{C_1, \ldots, C_n\}$ is a set of canonical clauses (possibly including the initial clause) then $\mathcal{P} \vdash \,!\,C_1 \,\&\, \ldots \,\&\, !\,C_n$ if and only if forall $i = 1, \ldots, n$, there is a Forum proof of height 3 or less of $\mathcal{P} \vdash C_i$.

NB: *derivability* is generally much simpler to establish than *admissibility*. The later generally requires induction.

We show now an admissibility result: cut is admissible in a system without cut.

Object-level cut-elimination holds for coherent systems

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

Theorem: Determining whether or not a canonical proof system is coherent is decidable. In particular, determining duality of a right and left introduction rule connective can be done by bounding proof search to a depth of $v + 2$ where v is the maximum number of meta-level atomic subformulas in the bodies of the introduction clauses. (Usually $v = 2$.)

Related work:

- Arnon Avron and Iddo Lev, *Canonical Propositional Gentzen-Type Systems*, IJCAR 2001.
- Frank Pfenning, *Structural Cut Elimination*, LICS95.

Future Work

Stating and proving that *atomically closed* sequents are complete.

Of the various proposals for non-commutative variants of classical linear logic: it would be interesting to see if these can be used to capture non-commutative object-level logics in a manner done here.

To deal with admissibility of inference rules, induction over formulas and proofs are needed. Considering an extension to linear logic with notions of *definitions* and *induction* similar to the $FO\lambda^{\Delta\mathbb{N}}$ extension to intuitionistic logic [McDowell & Miller, TCS 2000].

What is a good choice of proof terms to witness sequent calculus proofs in this setting. They should allow for induction (needed to prove that atomic cuts can be eliminated) and support natural notions of proof normalization and substitution.