Natural Deduction for Intuitionistic Non-Commutative Linear Logic

Jeff Polakow* and Frank Pfenning**

Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
e-mail: jpolakow@cs.cmu.edu, fp@cs.cmu.edu

Abstract. We present a system of natural deduction and associated term calculus for intuitionistic non-commutative linear logic (INCLL) as a conservative extension of intuitionistic linear logic. We prove subject reduction and the existence of canonical forms in the implicational fragment.

1 Introduction

Intuitionistic logic captures pure functional computation in a logical way, as can be seen from the Curry-Howard isomorphism between constructive proofs and functional programs. However, there are many structural properties of programs that are not captured within the intuitionistic framework, such as resource usage, computational complexity, and sequentiality.

Intuitionistic linear logic [Gir87,Abr93,Bar97] can be thought of as a refinement of intuitionistic logic in which resource consumption properties of functions can be expressed internally. Here, we refine it further to allow the expression of sequencing of computations. We achieve this by controlling the use of the structural rule of exchange to arrive at *intuitionistic non-commutative linear logic* (INCLL). Much research in non-commutative linear logic has been focused on simply removing the exchange rule from the underlying logic and only allowing exchange to be used in tandem with other structural rules on modal formulas. As an alternative we propose a system which distinguishes among unrestricted, linear, and ordered hypotheses.

Our presentation of INCLL is in the form of natural deduction with proof terms, thereby departing from previous formulations based on the sequent calculus [BG91,Abr90,Rue97]. This establishes the connection to functional computation by an extension of the Curry-Howard isomorphism. INCLL is a conservative extension of dual intuitionistic linear logic [Bar97] which means that we strictly increase its expressive power.

We have several motivating applications for this logic, although space does not permit their detailed analysis in this paper. One direct application is a

* Partially supported by the National Science Foundation under grant CCR-9804014.
** Partially supported by the National Science Foundation under grant CCR-9619584.

logical explanation for ordering properties of terms in continuation-passing style investigated by Danvy and the second author in [DP95]. The ordering inherent in non-commutative function arguments can be used to internalize stackability properties of program evaluation in a fragment of INCLL, which is large enough to capture the case of terms resulting from the standard CPS transformation.

Furthermore, our system integrates the Lambek calculus [Lam58] into a functional framework which also permits ordinary and linear functions in a consistent manner. With the coexistence of linear and ordered functions, we can logically describe more natural language phenomena than with either one by itself; for example, pied-piping and unbounded filler-gap dependencies [Par89,Hod94]. Related approaches to similar problems from computational linguistics are pursued, for example, by Kurtonina and Moortgat [KM96].

We show that our calculus permits canonical (that is, long $\beta\eta$-normal) forms, which means that it is a candidate for a foundation of a logical framework and logic programming language along the lines of Lolli [HM94] and linear LF [CP96]. In related work on a sequent calculus formulation of INCLL [PP99], we have developed an efficient proof search mechanism suitable for logic programming and applied it to algorithms for natural language parsing, sorting, and execution of abstract machines [PP98].

We begin in Section 2 by introducing the implicational fragment of INCLL which is characterized by four implications: intuitionistic ($\rightarrow$), linear ($\multimap$), left ordered ($\rightarrowtail$), and right ordered ($\twoheadrightarrow$). From a functional point of view, this corresponds to having four different types of functions—those which have no restrictions placed upon the use of their arguments; those which must use all their arguments once in any order; and those which which must use all of their arguments once in a specified order. We prove that this fragment satisfies subject reduction thereby validating the introduction and elimination rules. Strong normalization and the Church-Rosser property also hold, but are elided in this extended abstract.

In Section 3 we prove that every well-typed term has an equivalent canonical form, which is important for applications to logic programming and logical frameworks. The proof of this property employs logical relations and we develop the necessary machinery of substitutions. Then we introduce further logical connectives in Section 4 which include a modal operator for mobility (¡) and the usual connectives of linear logic. While subject reduction continues to hold, the existence of commutative conversions destroys the canonical form property.

2 The Implicational Fragment

We define intuitionistic non-commutative linear logic (INCLL) via a judgment

$$\Gamma; \Delta; \Omega \vdash M : A$$

where Γ is a context of unrestricted hypotheses (allowing exchange, weakening, and contraction), Δ is a context of linear hypotheses (allowing only exchange), Ω is a context of ordered hypotheses, M is a proof term, and A is a formula.

Associativity is assumed implicitly for all three contexts. In general, we use "formula" and "type" interchangeably, which is justified by the Curry-Howard isomorphism.

If we reflect the three kinds of hypotheses as connectives in the language of types, we obtain the familiar intuitionistic ($\rightarrow$) and linear ($\multimap$) implications, and two forms of ordered implication, depending on whether hypotheses are taken from the left ($\rightarrowtail$) or the right ($\twoheadrightarrow$) end of the ordered context. In the Lambek calculus [Lam58], the left ordered implication $A \rightarrowtail B$ is written as $A\backslash B$, while the right ordered implication $A \twoheadrightarrow B$ is written as B/A.

$$
\begin{array}{lllll}
\textit{Types} & A & ::= & P & \text{atomic types} \\
 & & \mid & A_1 \rightarrow A_2 & \text{intuitionistic implication} \\
 & & \mid & A_1 \multimap A_2 & \text{linear implication} \\
 & & \mid & A_1 \twoheadrightarrow A_2 & \text{ordered right implication} \\
 & & \mid & A_1 \rightarrowtail A_2 & \text{ordered left implication}
\end{array}
$$

Proof terms are drawn from a λ-calculus in the style of Church, that is, each valid term has a unique type, which seems essential for the logical framework applications we have in mind. We distinguish between intuitionistic (x), linear (y), and ordered (z) variables and write v if a variable might be declared in any of the three contexts.

$$
\begin{array}{lllll}
\textit{Terms} & M & ::= & x \mid y \mid z & \text{variables} \\
 & & \mid & \lambda x{:}A.\ M \mid M_1\, M_2 & \text{intuitionistic functions } (A \rightarrow B) \\
 & & \mid & \hat{\lambda} y{:}A.\ M \mid M_1 \,\hat{}\, M_2 & \text{linear functions } (A \multimap B) \\
 & & \mid & \overset{>}{\lambda} z{:}A.\ M \mid M_1 \,{}^{>}\, M_2 & \text{right ordered functions } (A \twoheadrightarrow B) \\
 & & \mid & \overset{<}{\lambda} z{:}A.\ M \mid M_1 \,{}^{<}\, M_2 & \text{left ordered functions } (A \rightarrowtail B)
\end{array}
$$

Contexts Γ, Δ, and Ω are simply lists of assumptions, $v{:}A$, where all variables v are distinct but of the same category (intuitionistic, linear, or ordered). We use "$\cdot$" to stand for the empty context, but we often omit it at the beginning of a context. We allow bound variables to be renamed tacitly.

In order to describe the inference rules, we need some auxiliary operations on contexts, *context concatenation* Ω, Ω' and *context merge* $\Delta \bowtie \Delta'$. Concatenation preserves the order of the assumptions, while the non-deterministic merge allows any interleaving of assumptions.

When viewing a natural deduction bottom-up, we think of context concatenation Ω_1, Ω_2 as *ordered context split* and context merge $\Delta_1 \bowtie \Delta_2$ as *context split.* Both of these are non-deterministic when read in this way, that is, there may be many ways to split a context $\Omega = \Omega_1, \Omega_2$ or $\Delta = \Delta_1 \bowtie \Delta_2$.

We now present the introduction and elimination rules for each implicational connective in turn. Other connectives are treated in Section 4. Generally, we use Γ, Δ and Ω to stand for contexts declaring intuitionistic, linear, and ordered variables, respectively.

Intuitionistic Functions $A \rightarrow B$. Since neither the linear nor the ordered context admit weakening, the rule for unrestricted variables requires them to be empty.

In the introduction rule, new variables are added at the right of Γ, but they could just as well be added on the left since the intuitionistic context admits exchange (see Lemma 1). In the elimination rule we cannot allow the derivation of the minor premise to depend on linear or ordered assumptions, since the use of A in the proof of $A \rightarrow B$ is unrestricted and subject reduction would fail. The intuitionistic context must be the same in both premises, which indicates that the rules are biased towards a bottom-up reading, where we distribute the hypotheses Γ to both premises, relying on the validity of contraction for the intuitionistic context.

$$\frac{}{(\Gamma_1, x{:}A, \Gamma_2); \cdot; \cdot \vdash x : A}\,\mathbf{ivar} \qquad \frac{(\Gamma, x{:}A); \Delta; \Omega \vdash M : B}{\Gamma; \Delta; \Omega \vdash \lambda x{:}A.\ M : A \rightarrow B} \rightarrow I$$

$$\frac{\Gamma; \Delta; \Omega \vdash M : A \rightarrow B \qquad \Gamma; \cdot; \cdot \vdash N : A}{\Gamma; \Delta; \Omega \vdash M\,N : B} \rightarrow E$$

Linear Functions $A \multimap B$. The rules for linear functions exhibit the new phenomenon that the linear contexts from the premises of the elimination rules are interleaved to form the linear context of the conclusion, which expresses the linearity condition concisely.

$$\frac{}{\Gamma; y{:}A; \cdot \vdash y : A}\,\mathbf{lvar} \qquad \frac{\Gamma; (\Delta, y{:}A); \Omega \vdash M : B}{\Gamma; \Delta; \Omega \vdash \hat{\lambda} y{:}A.\ M : A \multimap B} \multimap I$$

$$\frac{\Gamma; \Delta_1; \Omega \vdash M : A \multimap B \qquad \Gamma; \Delta_2; \cdot \vdash N : A}{\Gamma; (\Delta_1 \bowtie \Delta_2); \Omega \vdash M \,\hat{}\, N : B} \multimap E$$

Ordered Variables. Ordered variables must be the only ones in the hypothesis rule, which expresses that ordered variables must also be linear. In other words, order is seen as a further restriction on linearity, rather than as an independent property (which is also conceivable).

$$\frac{}{\Gamma; \cdot; z{:}A \vdash z : A}\,\mathbf{ovar}$$

Right Ordered Functions $A \twoheadrightarrow B$. In the introduction rule for right ordered functions the variable z must be new (by our general convention that variables in context are unique) and appear at the right end of the ordered context. In the matching elimination rule, the ordered contexts of the premises are concatenated in order to form the ordered context of the conclusion. The linear context is still interleaving, so as not to violate linearity.

$$\frac{\Gamma; \Delta; (\Omega, z{:}A) \vdash M : B}{\Gamma; \Delta; \Omega \vdash \overset{>}{\lambda} z{:}A.\ M : A \twoheadrightarrow B} \twoheadrightarrow I$$

$$\frac{\Gamma; \Delta_1; \Omega_1 \vdash M : A \twoheadrightarrow B \qquad \Gamma; \Delta_2; \Omega_2 \vdash N : A}{\Gamma; (\Delta_1 \bowtie \Delta_2); (\Omega_1, \Omega_2) \vdash M \overset{>}{\,}\, N : B} \twoheadrightarrow E$$

Left Ordered Functions $A \rightarrowtail B$. The rules for left ordered implication are symmetric to right ordered implication: the assumption $z{:}A$ appears at the left end of the ordered context in the introduction rule, and the contexts are concatenated in reverse order in the elimination rule. The fact that these rules are consistent is demonstrated by the subject reduction theorem 1.

$$\frac{\Gamma; \Delta; (z{:}A, \Omega) \vdash M : B}{\Gamma; \Delta; \Omega \vdash \overset{<}{\lambda} z{:}A.\ M : A \rightarrowtail B} \rightarrowtail I$$

$$\frac{\Gamma; \Delta_2; \Omega_2 \vdash M : A \rightarrowtail B \qquad \Gamma; \Delta_1; \Omega_1 \vdash N : A}{\Gamma; (\Delta_1 \bowtie \Delta_2); (\Omega_1, \Omega_2) \vdash M^{<} N : B} \rightarrowtail E$$

To give more intuition to our formulation, we now reconsider the rules as they would be used in the bottom-up construction of a proof.

In the three variable rules **ivar**, **lvar**, and **ovar**, the linear and ordered contexts must either be empty or contain only the subject variable, while the intuitionistic context is unrestricted. This forces linear and ordered assumptions to appear at least once in a term.

In the $\multimap E$, $\twoheadrightarrow E$, and $\rightarrowtail E$ rules, the linear context is split into two disjoint parts (when reading from the bottom up), which means that each assumption can be used at most once. In the $\rightarrow E$ rules, all linear assumptions propagate to the left premise. These observations together show that each linear variable is used at most once. Since it is also used at least once by the observation made about the variable rules, linear assumptions occur exactly once.

In the $\twoheadrightarrow E$ rule, the ordered context is split in an order-preserving way, with the leftmost assumptions Ω_1 going to the left premise and the rightmost assumptions Ω_2 going to the right premise. The converse applies to the $\rightarrowtail E$ rule. In the $\multimap E$ and $\rightarrow E$ rules the whole ordered context Ω goes to the left premise. These observations, together with the observation on the variable rules, show that ordered assumptions occur exactly once and in the order they were made.

As we will see, the emptiness restrictions on the linear and ordered contexts in the $\multimap E$ and $\rightarrow E$ rules are necessary to guarantee subject reduction. The reduction rules are simply β-reduction for all three kinds of functions. We will later also consider η-expansion.

Reduction Rules.

$$(\lambda x{:}A.\ M)\, N \Longrightarrow [N/x]M \qquad (\hat{\lambda} y{:}A.\ M)\,\hat{}\, N \Longrightarrow [N/y]M$$
$$(\overset{>}{\lambda} z{:}A.\ M)^{>} N \Longrightarrow [N/z]M \qquad (\overset{<}{\lambda} z{:}A.\ M)^{<} N \Longrightarrow [N/z]M$$

In order to prove subject reduction we proceed to establish the expected structural properties for contexts and substitution lemmas.

Lemma 1 (Structural Properties).

1. *If $(\Gamma_1, x{:}A, x'{:}A', \Gamma_2); \Delta; \Omega \vdash M : B$ then $(\Gamma_1, x'{:}A', x{:}A, \Gamma_2); \Delta; \Omega \vdash M : B$.*

2. *If* $(\Gamma_1, \Gamma_2); \Delta; \Omega \vdash M : B$ *then* $(\Gamma_1, x{:}A, \Gamma_2); \Delta; \Omega \vdash M : B$.
3. *If* $(\Gamma_1, x{:}A, x'{:}A, \Gamma_2); \Delta; \Omega \vdash M : B$ *then* $(\Gamma_1, x{:}A, \Gamma_2); \Delta; \Omega \vdash [x/x']M : B$.
4. *If* $\Gamma; (\Delta_1, y{:}A, y'{:}A', \Delta_2); \Omega \vdash M : B$ *then* $\Gamma; (\Delta_1, y'{:}A', y{:}A, \Delta_2); \Omega \vdash M : B$.

Proof: By induction on the structure of the given derivations. □

Lemma 2 (Substitution Properties).

1. *If* $(\Gamma_1, x{:}A, \Gamma_2); \Delta; \Omega \vdash M : B$ *and* $\Gamma_1; \cdot; \cdot \vdash N : A$ *then* $(\Gamma_1, \Gamma_2); \Delta; \Omega \vdash [N/x]M : B$.
2. *If* $\Gamma; (\Delta_1, y{:}A, \Delta_2); \Omega \vdash M : B$ *and* $\Gamma; \Delta'; \cdot \vdash N : A$ *then* $\Gamma; (\Delta_1, \Delta', \Delta_2); \Omega \vdash [N/y]M : B$.
3. *If* $\Gamma; \Delta; (\Omega_1, z{:}A, \Omega_2) \vdash M : B$ *and* $\Gamma; \Delta'; \Omega' \vdash N : A$ *then* $\Gamma; (\Delta \bowtie \Delta'); (\Omega_1, \Omega', \Omega_2) \vdash [N/z]M : B$.

Proof: By induction over the structure of the given typing derivation for M in each case, using Lemma 1. □

Subject reduction now follows immediately.

Theorem 1 (Subject Reduction).
If $M \Longrightarrow M'$ *and* $\Gamma; \Delta; \Omega \vdash M : A$ *then* $\Gamma; \Delta; \Omega \vdash M' : A$.

Proof: For each reduction, we apply inversion to the given typing derivation and then use the substitution lemma 2 to obtain the typing derivation for the conclusion. □

Subject reduction demonstrates that an introduction rule immediately followed by an elimination rule for the same connective can be reduced. This is a form of a *local soundness* theorem expressing that the elimination rules are not too strong. The corresponding global soundness property states that every derivation can be normalized entirely. This is easy to establish via a standard forgetful interpretation into the simply-typed λ-calculus. The normal form is also unique, which is a direct consequence of confluence. We will not formally state these theorems here, since they are besides the main interest of this paper. The proof of confluence is also completely standard (either developing a theory of residuals or using the Tait/Martin-Löf method of parallel reduction).

Local soundness (expressed as subject reduction) guarantees that, for each connective, the elimination rules are not too strong. To check that they are not too weak, we need to show that there is a way to apply elimination rules so that the original judgment can be recovered by introduction rules. This property of local completeness is expressed on proof terms as *subject expansion*, where "expansion" refers to η-expansion.

Theorem 2 (Subject Expansion).

1. *If* $\Gamma; \Delta; \Omega \vdash M : A \rightarrow B$ *then* $\Gamma; \Delta; \Omega \vdash \lambda x{:}A.\ M\, x : A \rightarrow B$.
2. *If* $\Gamma; \Delta; \Omega \vdash M : A \multimap B$ *then* $\Gamma; \Delta; \Omega \vdash \hat{\lambda} y{:}A.\ M \,\hat{}\, y : A \multimap B$.

3. *If* $\Gamma; \Delta; \Omega \vdash M : A \twoheadrightarrow B$ *then* $\Gamma; \Delta; \Omega \vdash \overset{>}{\lambda} z{:}A.\ M^{>}\ z : A \twoheadrightarrow B$.
4. *If* $\Gamma; \Delta; \Omega \vdash M : A \rightarrowtail B$ *then* $\Gamma; \Delta; \Omega \vdash \overset{<}{\lambda} z{:}A.\ M^{<}\ z : A \rightarrowtail B$.

Proof: By a direct derivation in each case, using weakening (lemma 1(2)) in part 1. □

A corresponding global property is the existence of long normal forms. This is the subject of the next section.

3 Canonical Forms

The existence of *canonical* (or long $\beta\eta$-normal) forms is critical in logical framework applications of our calculus, since it is the canonical forms which are in bijective correspondence with the objects to be represented. This property is inherited both from the logical framework LF [HHP93] and its linear refinement LLF [CP96]. For the intuitionistic case, both syntactic and semantic proofs exist (see, for example, [Gha97]). Here we pursue a proof by logical relations, whose development also sheds light on the nature of substitutions in our calculus.

We first formalize the property that a term can be converted to canonical form via a deductive system which can easily be related to the usual notion of long $\beta\eta$-normal form. This deductive system can also be read as an algorithm for converting a term to canonical form.

We then prove that any well-typed term can indeed be converted to canonical form. Our proof will be an argument by Kripke logical relations (also called Tait's method) consisting of two parts: (1) If M is a well-typed term of type A then M is in the logical relation represented by A, and (2) if M is in the logical relation represented by A then there is some canonical term N convertible to M. Our reduction strategy is based on weak head reduction defined below.

$$\frac{}{(\lambda x{:}A.\ M)\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ [N/x]M}\,\beta_{\rightarrow} \qquad \frac{M \ \overset{\mathbf{whr}}{\longrightarrow}\ M'}{M\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ M'\,N}\,\mathbf{whr}_{\rightarrow}$$

$$\frac{}{(\hat{\lambda} y{:}A.\ M)\,\hat{}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ [N/y]M}\,\beta_{\multimap} \qquad \frac{M \ \overset{\mathbf{whr}}{\longrightarrow}\ M'}{M\,\hat{}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ M'\,\hat{}\,N}\,\mathbf{whr}_{\multimap}$$

$$\frac{}{(\overset{<}{\lambda} z{:}A.\ M)^{<}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ [N/z]M}\,\beta_{\rightarrowtail} \qquad \frac{M \ \overset{\mathbf{whr}}{\longrightarrow}\ M'}{M^{<}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ M'^{<}\,N}\,\mathbf{whr}_{\rightarrowtail}$$

$$\frac{}{(\overset{>}{\lambda} z{:}A.\ M)^{>}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ [N/z]M}\,\beta_{\twoheadrightarrow} \qquad \frac{M \ \overset{\mathbf{whr}}{\longrightarrow}\ M'}{M^{>}\,N \ \overset{\mathbf{whr}}{\longrightarrow}\ M'^{>}\,N}\,\mathbf{whr}_{\twoheadrightarrow}$$

Intuitively, canonical terms are atomic terms of atomic type or λ-abstractions of canonical terms. Atomic terms are variables or applications of atomic terms to canonical terms. This is formalized in the judgments $\Gamma;\Delta;\Omega \vdash M \Uparrow M' : A$, which denotes that M has canonical form M' at type A, and $\Gamma;\Delta;\Omega \vdash M \downarrow M' : A$, which denotes that M has atomic form M' at type A.

Atomic Types.

$$\frac{\Gamma;\Delta;\Omega \vdash M \downarrow M' : P}{\Gamma;\Delta;\Omega \vdash M \Uparrow M' : P}\,\textbf{coercion}$$

$$\frac{M \stackrel{\textbf{whr}}{\longrightarrow} M' \qquad \Gamma;\Delta;\Omega \vdash M' \Uparrow M'' : P}{\Gamma;\Delta;\Omega \vdash M \Uparrow M'' : P}\,\textbf{reduction}$$

Intuitionistic Functions.

$$\frac{}{(\Gamma_1, x{:}A, \Gamma_2);\cdot;\cdot \vdash x \downarrow x : A}\,\textbf{ivar} \qquad \frac{(\Gamma, x{:}A);\Delta;\Omega \vdash M\,x \Uparrow M' : B}{\Gamma;\Delta;\Omega \vdash M \Uparrow \lambda x{:}A.\ M' : A \rightarrow B}\,{\rightarrow}I$$

$$\frac{\Gamma;\Delta;\Omega \vdash M \downarrow M' : A \rightarrow B \qquad \Gamma;\cdot;\cdot \vdash N \Uparrow N' : A}{\Gamma;\Delta;\Omega \vdash M\,N \downarrow M'\,N' : B}\,{\rightarrow}E$$

Linear Functions.

$$\frac{}{\Gamma;y{:}A;\cdot \vdash y \downarrow y : A}\,\textbf{lvar} \qquad \frac{\Gamma;(\Delta, y{:}A);\Omega \vdash M\,\hat{}\,y \Uparrow M' : B}{\Gamma;\Delta;\Omega \vdash M \Uparrow \hat{\lambda} y{:}A.\ M' : A \multimap B}\,{\multimap}I$$

$$\frac{\Gamma;\Delta;\Omega \vdash M \downarrow M' : A \multimap B \qquad \Gamma;\Delta_A;\cdot \vdash N \Uparrow N' : A}{\Gamma;(\Delta \bowtie \Delta_A);\Omega \vdash M\,\hat{}\,N \downarrow M'\,\hat{}\,N' : B}\,{\multimap}E$$

Ordered Functions.

$$\frac{}{\Gamma;\cdot;z{:}A \vdash z \downarrow z : A}\,\textbf{ovar}$$

$$\frac{\Gamma;\Delta;(\Omega, z{:}A) \vdash M\,{}^{>}\,z \Uparrow M' : B}{\Gamma;\Delta;\Omega \vdash M \Uparrow \overset{>}{\lambda} z{:}A.\ M' : A \twoheadrightarrow B}\,{\twoheadrightarrow}I$$

$$\frac{\Gamma;\Delta;\Omega \vdash M \downarrow M' : A \twoheadrightarrow B \qquad \Gamma;\Delta_A;\Omega_A \vdash N \Uparrow N' : A}{\Gamma;(\Delta \bowtie \Delta_A);(\Omega, \Omega_A) \vdash M\,{}^{>}\,N \downarrow M'\,{}^{>}\,N' : B}\,{\twoheadrightarrow}E$$

$$\frac{\Gamma;\Delta;(z{:}A, \Omega) \vdash M\,{}^{<}\,z \Uparrow M' : B}{\Gamma;\Delta;\Omega \vdash M \Uparrow \overset{<}{\lambda} z{:}A.\ M' : A \rightarrowtail B}\,{\rightarrowtail}I$$

$$\frac{\Gamma;\Delta;\Omega \vdash M \downarrow M' : A \rightarrowtail B \qquad \Gamma;\Delta_A;\Omega_A \vdash N \Uparrow N' : A}{\Gamma;(\Delta \bowtie \Delta_A);(\Omega_A, \Omega) \vdash M\,{}^{<}\,N \downarrow M'\,{}^{<}\,N' : B}\,{\rightarrowtail}E$$

We remark that the expected structural properties of the intuitionistic and linear contexts also hold for this system. Furthermore, if $\Gamma;\Delta;\Omega \vdash M \Uparrow M' : A$ then $\Gamma;\Delta;\Omega \vdash M' : A$ and M' is in long $\beta\eta$-normal form. These properties follow by immediate structural inductions.

The following unary Kripke logical relation is the crux of our argument. It is defined by induction on the type A. Note how the structural properties of intuitionistic, linear, and ordered contexts are captured in this definition.

$\Gamma;\Delta;\Omega \vdash M \in [\![P]\!]$ iff $\Gamma;\Omega;\Delta \vdash M \Uparrow N : P$ for some N.

$\Gamma;\Delta;\Omega \vdash M \in [\![A_1 \rightarrow A_2]\!]$ iff for all Γ_N and N,
if $\Gamma,\Gamma_N;\cdot;\cdot \vdash N \in [\![A_1]\!]$ then $\Gamma,\Gamma_N;\Omega;\Delta \vdash M\, N \in [\![A_2]\!]$.

$\Gamma;\Delta;\Omega \vdash M \in [\![A_1 \multimap A_2]\!]$ iff for all Δ_N and N,
if $\Gamma;\Delta_N;\cdot \vdash N \in [\![A_1]\!]$ then $\Gamma;\Delta \bowtie \Delta_N;\Omega \vdash M\,\hat{}\,N \in [\![A_2]\!]$.

$\Gamma;\Delta;\Omega \vdash M \in [\![A_1 \twoheadrightarrow A_2]\!]$ iff for all Δ_N, Ω_N and N,
if $\Gamma;\Delta_N;\Omega_N \vdash N \in [\![A_1]\!]$ then $\Gamma;\Delta \bowtie \Delta_N;\Omega,\Omega_N \vdash M^{>} N \in [\![A_2]\!]$.

$\Gamma;\Delta;\Omega \vdash M \in [\![A_1 \rightarrowtail A_2]\!]$ iff for all Δ_N, Ω_N and N,
if $\Gamma;\Delta_N;\Omega_N \vdash N \in [\![A_1]\!]$ then $\Gamma;\Delta \bowtie \Delta_N;\Omega_N,\Omega \vdash M^{<} N \in [\![A_2]\!]$.

We can now formally state and prove the second part of our proof— that well-typed terms in the logical relation at all types have canonical forms. We can prove this only simultaneously with the reverse statement for terms with an atomic form.

Lemma 3 (Logical Relations and Canonical Forms).

1. *If $\Gamma;\Delta;\Omega \vdash M \in [\![A]\!]$ then $\Gamma;\Delta;\Omega \vdash M \Uparrow N : A$ for some N.*
2. *If $\Gamma;\Delta;\Omega \vdash M \downarrow N : A$ then $\Gamma;\Delta;\Omega \vdash M \in [\![A]\!]$.*

Proof: By induction on A using structural properties of contexts. □

Lemma 4 (Closure Under Head Expansion).
If $M \stackrel{\mathbf{whr}}{\longrightarrow} M'$ and $\Gamma;\Delta;\Omega \vdash M' \in [\![A]\!]$ then $\Gamma;\Delta;\Omega \vdash M \in [\![A]\!]$.

Proof: By induction on A making use of lemma 3. □

In order to show $\Gamma;\Delta;\Omega \vdash M : A$ implies $\Gamma;\Delta;\Omega \vdash M \in [\![A]\!]$, we need to explicitly manipulate substitutions. We shall define a substitution to be a triple, $(\gamma;\delta;\omega)$, where each component is a list of term/variable pairs.

$$(\gamma;\delta;\omega) = (\cdot;\cdot;\cdot) \mid (\gamma, M/x;\delta;\omega) \mid (\gamma;\delta, M/y;\omega) \mid (\gamma;\delta;\omega, M/z)$$

We assume no variable is defined more than once in $(\gamma;\delta;\omega)$ and we write $(\gamma;\delta;\omega)(v) = M$ if M/v occurs in $(\gamma;\delta;\omega)$. We define well-typed substitutions with the judgment $\Gamma'; \Delta'; \Omega' \vdash (\gamma; \delta; \omega) : \Gamma; \Delta; \Omega$ which means that $\gamma; \delta; \omega$ supply appropriate terms for the variables declared in $\Gamma; \Delta; \Omega$, respectively.

$$\frac{}{\Gamma'; \Delta'; \Omega' \vdash (\cdot; \cdot; \cdot) : \cdot; \cdot; \cdot}$$

$$\frac{\Gamma'; \Delta'; \Omega' \vdash (\gamma; \delta; \omega) : \Gamma; \Delta; \Omega \qquad \Gamma'; \cdot; \cdot \vdash M : A}{\Gamma'; \Delta'; \Omega' \vdash (\gamma, M/x; \delta; \omega) : \Gamma, x{:}A; \Delta; \Omega}$$

$$\frac{\Gamma'; \Delta'_1; \Omega' \vdash (\gamma; \delta; \omega) : \Gamma; \Delta; \Omega \qquad \Gamma'; \Delta'_2; \cdot \vdash M : A}{\Gamma'; \Delta'_1 \bowtie \Delta'_2; \Omega' \vdash (\gamma; \delta, M/y; \omega) : \Gamma; \Delta, y{:}A; \Omega}$$

$$\frac{\Gamma'; \Delta'_1; \Omega'_1 \vdash (\gamma; \delta; \omega) : \Gamma; \Delta; \Omega \qquad \Gamma'; \Delta'_2; \Omega'_2 \vdash M : A}{\Gamma'; \Delta'_1 \bowtie \Delta'_2; \Omega'_1, \Omega'_2 \vdash (\gamma; \delta; \omega, M/z) : \Gamma; \Delta; \Omega, z{:}A}$$

Note the restrictions which prohibit, for example, that the substitution term for a linear variable depends on an ordered variable. Such a dependence would falsify Theorem 5.

When computing the result of applying a substitution to a term, we would like to maintain the invariant that the substitution matches the contexts in which the term is well-formed. This means we have to split the substitution at applications. Thus, we define the application of a substitution to a term as follows:

$$\begin{aligned}
&[(\gamma; \delta; \omega)]v = (\gamma; \delta; \omega)(v) \\
&[(\gamma; \delta; \omega)](\lambda x{:}A.\ M) = \lambda x{:}A.\ [(\gamma, x/x; \delta; \omega)]M \\
&[(\gamma; \delta; \omega)](M\ N) = ([(\gamma; \delta; \omega)]M)([\gamma; \cdot; \cdot]N) \\
&[(\gamma; \delta; \omega)](\hat{\lambda} y{:}A.\ M) = \hat{\lambda} y{:}A.\ [(\gamma; \delta, y/y; \omega)]M \\
&[(\gamma; \delta_1 \bowtie \delta_2; \omega)](M\,\hat{}\,N) = ([(\gamma; \delta_1; \omega)]M)\,\hat{}\,([\gamma; \delta_2; \cdot]N) \\
&[(\gamma; \delta; \omega)](\overset{>}{\lambda} z{:}A.\ M) = \overset{>}{\lambda} z{:}A.\ [(\gamma; \delta; \omega, z/z)]M \\
&[(\gamma; \delta_1 \bowtie \delta_2; \omega_1, \omega_2)](M^{>} N) = ([(\gamma; \delta_1; \omega_1)]M)^{>}([\gamma; \delta_2; \omega_2]N) \\
&[(\gamma; \delta; \omega)](\overset{<}{\lambda} z{:}A.\ M) = \overset{<}{\lambda} z{:}A.\ [(\gamma; \delta; z/z, \omega)]M \\
&[(\gamma; \delta_1 \bowtie \delta_2; \omega_2, \omega_1)](M^{<} N) = ([(\gamma; \delta_1; \omega_1)]M)^{<}([\gamma; \delta_2; \omega_2]N)
\end{aligned}$$

At first glance the substitution splitting may seem non-deterministic. However, the proper split can be easily determined from the typing derivation of the term we substitute into. Since typing derivations are unique, there is no ambiguity. We rely on this in the proof of the fundamental theorem of logical relations (Lemma 7).

Lemma 5 (Typing and Substitutions). *If $\Gamma; \Delta; \Omega \vdash M : A$ and $\Gamma'; \Delta'; \Omega' \vdash (\gamma; \delta; \omega) : \Gamma; \Delta; \Omega$ then $\Gamma'; \Delta'; \Omega' \vdash [(\gamma; \delta; \omega)]M : A$.*

Proof: By induction on the structure of the derivation of $\Gamma; \Delta; \Omega \vdash M : A$. □

Substitutions compose in the obvious way, although we do not investigate properties of substitutions further here. We write $id_{\Gamma;\Delta;\Omega}$ for the identity substitution on the variables declared in Γ, Δ, and Ω. We define logical relations on substitutions by induction on the structure of contexts.

$$\Gamma'; \cdot; \cdot \vdash \cdot \in [\![\cdot; \cdot; \cdot]\!]$$

$$\begin{array}{l}\Gamma'; \Delta'; \Omega' \vdash (\gamma, M/x; \delta; \omega) \in [\![\Gamma, x{:}A; \Delta; \Omega]\!] \quad \text{iff} \\ \quad \Gamma'; \Delta'; \Omega' \vdash (\gamma; \delta; \omega) \in [\![\Gamma; \Delta; \Omega]\!] \quad \text{and} \quad \Gamma'; \cdot; \cdot \vdash M \in [\![A]\!]\end{array}$$

$$\begin{array}{l}\Gamma'; \Delta'_1 \bowtie \Delta'_2; \Omega' \vdash (\gamma; \delta, M/y; \delta; \omega) \in [\![\Gamma; \Delta, y{:}A; \Omega]\!] \quad \text{iff} \\ \quad \Gamma'; \Delta'_1; \Omega' \vdash (\gamma; \delta; \omega) \in [\![\Gamma; \Delta; \Omega]\!] \quad \text{and} \quad \Gamma'; \Delta'_2; \cdot \vdash M \in [\![A]\!]\end{array}$$

$$\begin{array}{l}\Gamma'; \Delta'_1 \bowtie \Delta'_2; \Omega'_1 \Omega'_2 \vdash (\gamma; \delta; \omega, M/z) \in [\![\Gamma; \Delta; \Omega, z{:}A]\!] \quad \text{iff} \\ \quad \Gamma'; \Delta'_1; \Omega'_1 \vdash (\gamma; \delta; \omega) \in [\![\Gamma; \Delta; \Omega]\!] \quad \text{and} \quad \Gamma'; \Delta'_2; \Omega'_2 \vdash M \in [\![A]\!]\end{array}$$

Lemma 6 (Identity). $\Gamma; \Delta; \Omega \vdash id_{\Gamma;\Delta;\Omega} \in [\![\Gamma; \Delta; \Omega]\!]$

Proof: Immediate by definition and lemma 3. □

Lemma 7 (Typing and Logical Relations). *If $\Gamma; \Delta; \Omega \vdash M : A$ then for any $\Gamma'; \Delta'; \Omega' \vdash (\gamma; \delta; \omega) \in [\![\Gamma; \Delta; \Omega]\!]$ we have $\Gamma'; \Delta'; \Omega' \vdash [(\gamma; \delta; \omega)]M \in [\![A]\!]$.*

Proof: By induction on the structure of the given derivation using lemma 4. □

Theorem 3 (Canonical Forms).
If $\Gamma; \Delta; \Omega \vdash M : A$ then for some N, $\Gamma; \Delta; \Omega \vdash M \Uparrow N : A$.

Proof: Immediate from lemmas 7, 3, and 6. □

4 Other Logical Connectives

Before considering the other standard connectives from linear logic, we note further structural properties.

Theorem 4 (Demotion).

1. *If $\Gamma; (\Delta_1, y{:}A, \Delta_2); \Omega \vdash M : B$ then $(\Gamma, x{:}A); (\Delta_1, \Delta_2); \Omega \vdash [x/y]M : B$.*
2. *If $\Gamma; \Delta; (\Omega_1, z{:}A, \Omega_2) \vdash M : B$ then $\Gamma; (\Delta, y{:}A); (\Omega_1, \Omega_2) \vdash [y/z]M : B$.*

Proof: In both cases by induction on the structure of the given derivation. □

When considering the typing rules for the new connectives, we shall take care that the preceding property continues to hold. The subject reduction and strong normalization theorems also continue to hold, with straightforward extensions of the proofs mentioned in Section 2.

Some of the new connectives, namely an ordered conjunction ($\bullet$), multiplicative unit (1), disjunction ($\oplus$), falsehood (0), mobility (¡) and exponential (!) introduce commutative conversions into the proof term calculus. Unique canonical forms no longer exist, even though each connective remains locally sound and complete. This means that these connectives must be ruled out or restricted in logic programming or logical frameworks applications of INCLL. Fortunately, this does not seem to be a serious drawback in practice [PP98].

Ordered Conjunction $A \bullet B$.

$$\frac{\Gamma; \Delta_1; \Omega_1 \vdash M{:}A \qquad \Gamma; \Delta_2; \Omega_2 \vdash N{:}B}{\Gamma; (\Delta_1 \bowtie \Delta_2); (\Omega_1, \Omega_2) \vdash M \bullet N : A \bullet B} \bullet I$$

$$\frac{\Gamma; \Delta_2; \Omega_2 \vdash M : A \bullet B \qquad \Gamma; \Delta_1; (\Omega_1, z{:}A, z'{:}B, \Omega_3) \vdash N : C}{\Gamma; (\Delta_1 \bowtie \Delta_2); (\Omega_1, \Omega_2, \Omega_3) \vdash \mathbf{let}\, z \bullet z' = M \,\mathbf{in}\, N : C} \bullet E$$

We have the following reduction rule:

$$\mathbf{let}\, z \bullet z' = M \bullet M' \,\mathbf{in}\, N \Longrightarrow [M/z, M'/z']N$$

Multiplicative Unit 1. This is the right and left unit element for the ordered conjunction connective. We have $1 \twoheadrightarrow C$ iff C iff $1 \rightarrowtail C$, and $A \bullet 1$ iff A iff $1 \bullet A$. The introduction rule shows why there is only one multiplicative unit.

$$\frac{}{\Gamma; \cdot; \cdot \vdash \star : 1} 1I$$

$$\frac{\Gamma; \Delta_2; \Omega_2 \vdash M : 1 \qquad \Gamma; \Delta_1; (\Omega_1, \Omega_3) \vdash N : C}{\Gamma; (\Delta_1 \bowtie \Delta_2); (\Omega_1, \Omega_2, \Omega_3) \vdash \mathbf{let}\, \star = M \,\mathbf{in}\, N : C} 1E$$

We have the following reduction rule:

$$\mathbf{let}\, \star = \star \,\mathbf{in}\, N \Longrightarrow N$$

Additive Conjunction $A \,\&\, B$. This is additive on both the linear and ordered contexts, in order to preserve Theorem 4.

$$\frac{\Gamma; \Delta; \Omega \vdash M : A \qquad \Gamma; \Delta; \Omega \vdash N : B}{\Gamma; \Delta; \Omega \vdash \langle M, N \rangle : A \,\&\, B} \&I$$

$$\frac{\Gamma; \Delta; \Omega \vdash M : A \,\&\, B}{\Gamma; \Delta; \Omega \vdash \mathbf{fst}\, M : A} \&E_1 \qquad \frac{\Gamma; \Delta; \Omega \vdash M : A \,\&\, B}{\Gamma; \Delta; \Omega \vdash \mathbf{snd}\, M : B} \&E_2$$

We have the following reduction rules:

$$\begin{aligned} \mathbf{fst}\, \langle M, N \rangle &\Longrightarrow M \\ \mathbf{snd}\, \langle M, N \rangle &\Longrightarrow N \end{aligned}$$

Additive Unit $\top$. Because it is additive, the left and right units for & coincide.

$$\frac{}{\Gamma; \Delta; \Omega \vdash \langle\,\rangle : \top} \top I$$

Since there is no elimination rule, there are no reductions for the additive unit.

Disjunction $\oplus$. The disjunction is additive and therefore does not split into left and right versions.

$$\frac{\Gamma;\Delta;\Omega \vdash M : A}{\Gamma;\Delta;\Omega \vdash \mathbf{inl}^B\, M : A \oplus B}\oplus I_1 \qquad \frac{\Gamma;\Delta;\Omega \vdash M : B}{\Gamma;\Delta;\Omega \vdash \mathbf{inr}^A\, M : A \oplus B}\oplus I_2$$

$$\frac{\begin{array}{l}\Gamma;\Delta_2;\Omega_2 \vdash M : A \oplus B\\ \Gamma;\Delta_1;(\Omega_1, z{:}A, \Omega_3) \vdash N : C\\ \Gamma;\Delta_1;(\Omega_1, z'{:}B, \Omega_3) \vdash N' : C\end{array}}{\Gamma;(\Delta_1 \bowtie \Delta_2);(\Omega_1,\Omega_2,\Omega_3) \vdash \mathbf{case}\, M\ \mathbf{of}\ \mathbf{inl}\, z \Rightarrow N \mid \mathbf{inr}\, z' \Rightarrow N' : C}\oplus E$$

We have the following reduction rules:

$$\begin{array}{c}\mathbf{case}\, \mathbf{inl}^B\, M\ \mathbf{of}\ \mathbf{inl}\, z \Rightarrow N \mid \mathbf{inr}\, z' \Rightarrow N' \Longrightarrow [M/z]N\\ \mathbf{case}\, \mathbf{inr}^A\, M'\ \mathbf{of}\ \mathbf{inl}\, z \Rightarrow N \mid \mathbf{inr}\, z' \Rightarrow N' \Longrightarrow [M'/z']N'\end{array}$$

Additive Falsehood 0. This is the unit for disjunction.

$$\frac{\Gamma;\Delta_2;\Omega_2 \vdash M : 0}{\Gamma;(\Delta_1 \bowtie \Delta_2);(\Omega_1,\Omega_2,\Omega_3) \vdash \mathbf{abort}^C\, M : C}0E$$

Since there is no introduction rule for 0, there are no new reductions.

Linear Exponential $!A$.

$$\frac{\Gamma;\cdot;\cdot \vdash M : A}{\Gamma;\cdot;\cdot \vdash\ !M : !A}!I$$

$$\frac{\Gamma;\Delta_2;\Omega_2 \vdash M : !A \qquad (\Gamma, x{:}A);\Delta_1;(\Omega_1,\Omega_3) \vdash N : C}{\Gamma;(\Delta_1 \bowtie \Delta_2);(\Omega_1,\Omega_2,\Omega_3) \vdash \mathbf{let}\ !x = M\ \mathbf{in}\ N : C}!E$$

We have the following reduction rule:

$$\mathbf{let}\ !x = !M\ \mathbf{in}\ N \Longrightarrow [M/x]N$$

Mobility Modal $¡A$. We may also consider a modality not present in linear logic which allows an ordered hypothesis to be used out of order. In analogy with !, we wish to have $¡A \rightarrowtail B \equiv ¡A \twoheadrightarrow B \equiv A \multimap B$.

$$\frac{\Gamma;\Delta;\cdot \vdash M : A}{\Gamma;\Delta;\cdot \vdash ¡M : ¡A}¡I$$

$$\frac{\Gamma;\Delta_2;\Omega_2 \vdash M : ¡A \qquad \Gamma;(\Delta_1, y{:}A);(\Omega_1,\Omega_3) \vdash N : C}{\Gamma;(\Delta_1 \bowtie \Delta_2);(\Omega_1,\Omega_2,\Omega_3) \vdash \mathbf{let}\ ¡y = M\ \mathbf{in}\ N : C}¡E$$

We have the following reduction rule:

$$\mathbf{let}\ ¡y = ¡M\ \mathbf{in}\ N \Longrightarrow [M/y]N$$

5 Conclusion and Future Work

We have presented a natural deduction version of intuitionistic non-commutative linear logic which conservatively extends intuitionistic linear logic. We have shown that the proof term calculus satisfies subject reduction and strong normalization, and that canonical forms exist for the implicational fragment. In [PP99] we present a sequent calculus for INCLL, prove cut-elimination and show that it closely corresponds to the natural deduction system presented here.

Applications lie in the areas of logical frameworks, functional programming, logic programming, and natural language processing. These applications are sketched in the introduction and are the subject of current research. At present, for example, we have shown that the ordering properties of functional programs which result from CPS conversion discovered by Danvy [Dan94] can be captured completely internally in the INCLL term calculus. We have also shown that uniform derivations are sound and complete with respect to our calculus, which means that the implicational fragment of INCLL can be considered an *abstract logic programming language* [MNPS91]. A prototype implementation using advanced resource management strategies analogous to Lolli [Hod94] has been used for the concise expression of various algorithms for sorting, natural language parsing, and the execution of abstract machines. The systems and examples may be found in [PP98].

We have also given an operational semantics to an extension of the functional core presented here and are investigating the connection between stackability of intermediate values and ordered function arguments.

References

[Abr90] V. Michele Abrusci. Non-commutative intuitionistic linear propositional logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 36:297–318, 1990.

[Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[Bar97] Andrew Barber. *Linear Type Theories, Semantics and Action Calculi*. PhD thesis, Department of Computer Science, University of Edinburgh, 1997.

[BG91] C. Brown and D. Gurr. Relations and non-commutative linear logic. Technical Report DAIMI PB-372, Computer Science Department, Aarhus University, November 1991.

[CP96] Iliano Cervesato and Frank Pfenning. A linear logical framework. In E. Clarke, editor, *Proceedings of the Eleventh Annual Symposium on Logic in Computer Science — LICS'96*, pages 264–275, New Brunswick, New Jersey, 27–30 July 1996. IEEE Computer Society Press. This work also appeared as Preprint 1834 of the Department of Mathematics of Technical University of Darmstadt, Germany.

[Dan94] Olivier Danvy. Back to direct style. *Science of Computer Programming*, 22(3):183–195, 1994.

[DP95] Olivier Danvy and Frank Pfenning. The occurrence of continuation parameters in CPS terms. Technical Report CMU-CS-95-121, Department of Computer Science, Carnegie Mellon University, February 1995.

[Gha97] Neil Ghani. Eta-expansion in dependent type theory — the Calculus of Constructions. In Philippe de Groote and J. Roger Hindley, editors, *Proceedings of the 3rd International Conference on Typed Lambda Calculi and Applications (TLCA'97)*, pages 164–180, Nancy, France, April 1997. Springer-Verlag LNCS 1210.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

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

[HM94] Joshua S. Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994. Extended abstract in the Proceedings of the Sixth Annual Symposium on Logic in Computer Science, Amsterdam, July 15–18, 1991.

[Hod94] Joshua S. Hodas. *Logic Programming in Intuitionistic Linear Logic: Theory, Design and Implementation*. PhD thesis, University of Pennsylvania, Department of Computer and Information Science, 1994.

[KM96] Natasha Kurtonina and Michael Moortgat. Structural control. In P. Blackburn and M. de Rijke, editors, *Specifying Syntactic Structures*. CSLI Publications, 1996.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:363–386, 1958.

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

[Par89] Remo Pareschi. *Type-Driven Natural Language Analysis*. PhD thesis, University of Edinburgh, Edinburgh, Scotland, July 1989. Available as technical report MS-CIS-89-45, Department of Computer and Information Sciences, University of Pennsylvania.

[PP98] Jeff Polakow and Frank Pfenning. Ordered linear logic programming. Technical Report CMU-CS-98-183, Department of Computer Science, Carnegie Mellon University, December 1998.

[PP99] Jeff Polakow and Frank Pfenning. Relating natural deduction and sequent calculus for intuitionistic non-commutative linear logic. In Andre Scedrov and Achim Jung, editors, *Proceedings of the 15th Conference on Mathematical Foundations of Programming Semantics*, New Orleans, Louisiana, April 1999. To appear.

[Rue97] Paul Ruet. *Non-Commutative Logic and Concurrent Constraint Programming*. PhD thesis, Université Denis Diderot, Paris 7, 1997.