New Curry–Howard terms for full linear logic

David Albrecht, John N. Crossley, John S. Jeavons

*Department of Computer Science, Monash University, Australia
bDepartment of Computer Science, Monash University, Australia
cDepartment of Mathematics, Monash University, Caulfield East, P.O. Box 197, Clayton, Vic. 3168, Australia

Abstract

In this paper we (1) provide a natural deduction system for full first-order linear logic, (2) introduce Curry–Howard-style terms for this version of linear logic, (3) prove strong normalization for the system, and (4) prove that given a proof of $\forall x \exists y a(x, y)$ and any individual term $t$ we can compute a term $u$ such that $a(t, u)$ is provable.

Keywords: Linear logic; Natural deduction, Term calculus

1. Introduction

Increasing program complexity has meant formal methods are becoming an important technique in the software engineer’s toolkit. One aspect of these methods is the Curry–Howard correspondence (see [10]) between program specifications and propositions, programs and proofs, and computation and proof normalization. This correspondence is also one of the cornerstones of the logical foundations of functional programming (see [9]), and has been extensively studied for intuitionistic logic, where the programs correspond to typed-lambda terms.

In this paper we extend the Curry–Howard correspondence to full linear logic. Since its inception by Girard [7] linear logic has attracted a great deal of interest. This logic is based on the idea of resources and can be used to express a rich variety of program specifications, which include those written in classical logic, intuitionistic logic, and Girard’s system F.

There has already been some work done on developing an extension of the Curry–Howard correspondence to linear logic. Abramsky [1], developed a term system for full linear logic in which a “proof expression” is associated with a sequent-style proof in linear logic. The computational procedures for these proof expansions are inherently
parallel in nature and provide a clear dichotomy between lazy and eager evaluation procedures. Lincoln and Mitchell [11], Benton et al. [3], and Ronchi della Rocca and Roversi [12] have all developed a term calculus for the intuitionistic fragment of propositional linear logic. However, the intuitionistic fragment does not capture the essential nature of Girard's linear negation operator ($\neg$), where $(\alpha^+)\neg$ and $\alpha$ are equivalent formulae. In Albrecht et al. [2] we developed a term calculus for full linear logic; however, our proof of the Church–Rosser property contained an error.

The sources of the difficulty were the term reductions involving the contraction rule whereby multiple copies of a sub-term may proliferate during a reduction sequence. The effect of this is to complicate any arguments based on counting occurrences of certain types of sub-terms.

In this paper we overcome the problems associated with the contraction rule by introducing a new system of natural deduction for full linear logic in which contractions are handled implicitly.Troelstra [12] has introduced this technique for the intuitionistic fragment of linear logic. Our system has the usual style of introduction rules but has only a single elimination rule (the swap rule) which corresponds to the interchange of the conclusion and a hypothesis via linear negation (see the appendix). Our terms are functional in nature, the reductions satisfy the Church–Rosser property, as well as the strong normalization property (which is stronger than the corresponding property in Abramsky’s system) and allow the extraction of programs from the corresponding proofs.

Section 2 specifies the new natural deduction system and Section 3 introduces a term calculus for the system in the style of Gallier [6] and Benton et al. [3]. In Section 4 we define our reduction rules for the terms and in Section 5 we establish the strong normalization property of our system and also the Church–Rosser property. In Section 6 we establish a program extraction property by showing that if we can prove a formula of the form $\forall x \exists y \alpha(x, y)$, then given any individual term $t$ we can compute an individual term $u$ such that $\alpha(t, u)$ is also provable.

2. Linear logic and the natural deduction system $\mathcal{N}$

2.1. Linear logic

In predicate calculus we usually have rules equivalent to the following two rules (given here in a Gentzen sequent calculus style). We write $\Gamma \vdash \alpha$ for "$\alpha$ is provable from $\Gamma$".

$$\frac{\Gamma, \alpha, \alpha \vdash \beta}{\Gamma, \alpha \vdash \beta} \quad \text{(Contraction)} \quad \frac{\Gamma \vdash \beta}{\Gamma, \alpha \vdash \beta} \quad \text{(Weakening)}$$

In linear logic instead of these general rules, weakening and contraction are only allowed when the premisses are of a special form, that is to say, formulae of the form $!\alpha$ with the new unary connective "!" (where we read "!" as "of course"). A formula of the form $!\alpha$ is to be thought of as a resource which can be stored, reused, or discarded,
i.e. !x can be duplicated, or discarded, without further ado. The rules in linear logic are equivalent to the following:

\[
\frac{\Gamma, !x, !x \vdash \beta}{\Gamma, !x \vdash \beta} \quad \text{(Contraction)} \quad \frac{\Gamma \vdash \beta}{\Gamma, !x \vdash \beta} \quad \text{(Weakening)}
\]

Another consequence of the absence of general weakening and contraction is that two, usually logically equivalent, versions of \( \wedge \) ("and") are no longer equivalent in linear logic.

\[
(\wedge, 1) \quad \frac{\Gamma \vdash \alpha \quad \Delta \vdash \beta}{\Gamma, \Delta \vdash (\alpha \wedge \beta)} \quad (\wedge, 2) \quad \frac{\Gamma \vdash \alpha \quad \Gamma \vdash \beta}{\Gamma \vdash (\alpha \wedge \beta)}
\]

This gives rise to two types of \( \wedge \) in linear logic which Girard defined as \textit{multiplicative and}, \( \otimes \) (times), which corresponds to \((\wedge, 1)\) and \textit{additive and}, \& (with), which corresponds to \((\wedge, 2)\). There are also two types of "or", \( \oplus \) (plus) and \( \lor \) (par), two types of truth, \( \top \) and \( \bot \), and two types of falsity, \( \bot \) and \( \emptyset \).

The language we use contains a countable number of literals with their duals \( p, p^\perp, q, q^\perp, r, r^\perp, \ldots \), the constants \( \top \) and \( \emptyset \), the binary connectives \( \otimes, \oplus, \& \) and \( \rightarrow \) (lollipop, or linear implication), the unary connectives \( ! \) and \( ? \) (why not), and the quantifiers \( \forall \) (for all) and \( \exists \) (there exists). We do not take \( \lor \) as a primitive symbol but instead define \( a \lor b \) by \( a \rightarrow b \). Formulae are generated in the usual manner from atomic formulae, using the connectives and quantifiers, and will be denoted by lower-case Greek letters. Upper-case Greek letters will denote sets of formulae (which may be empty).

We define \( \bot := !\top, \bot := ?\emptyset \) and follow Girard [7] in defining negation by De Morgan equations, thus,

\[
\begin{align*}
\top^\perp & := \emptyset & 0^\perp & := \top \\
(p)^\perp & := p^\perp & (p^\perp)^\perp & := p \\
(\alpha \otimes \beta)^\perp & := \alpha^\perp \circ \beta^\perp & (\alpha \circ \beta)^\perp & := \alpha \otimes \beta^\perp \\
(\alpha \oplus \beta)^\perp & := \alpha^\perp \oplus \beta^\perp & (\alpha \oplus \beta)^\perp & := \alpha^\perp \& \beta^\perp \\
(\forall \alpha)^\perp & := ?\alpha^\perp & (\exists \alpha)^\perp & := !\alpha^\perp \\
(\forall \forall \alpha)^\perp & := \exists \alpha \alpha \perp & (\exists \exists \alpha)^\perp & := \forall \forall \alpha \perp
\end{align*}
\]

It follows from these definitions, by induction on the complexity of the formula, that \((\alpha^\perp)^\perp = \alpha\), for every formula \( \alpha \).

2.2. Natural deduction

Gentzen (see [13]) originally developed the natural deduction system to reflect "the actual logical reasoning involved in mathematical proofs".

Natural deduction systems are logical systems which are closed under the substitution of proofs for other proofs. This property makes them very useful as a logical basis...
for functional languages, allowing programs to be "plugged" together, and types to be preserved during evaluation. (This does not occur for Abramsky's [1] parallel terms.)

In a sequent, $\Gamma \vdash x$, $\Gamma$ will be called the declared premises of the sequent, and $x$ will be called the conclusion of the sequent. We shall not distinguish the order of the formulae in the premises $\Gamma$. In this way we do not need an exchange rule. Nevertheless, the system can be easily modified to use ordered premises and an exchange rule.

For each connective we have an introduction rule. Our new rule (S) is the only elimination rule and corresponds to the following sequent rule:

$$
\frac{\Gamma, x \vdash \beta}{\Gamma, \beta^\bot \vdash x^\bot}
$$

Viewing the premisses of a sequent as the input of a program and the conclusion as the output, the swap rule allows one to interchange an input with the output. It is this feature of interchanging input and output that provides the connection between the programs (terms) developed in this paper and the parallel programs (terms) discussed in Abramsky [1].

In Girard's system (see the appendix) there is an inherent symmetry (shown by the involutory nature of $(\ )^\bot$) exhibited by the axiom $\vdash x, x^\bot$ and the fact that $x^\bot^\bot = x$. In ordinary intuitionistic logic we lose such symmetry, since we only have that $x \vdash \beta$ gives $\neg\beta \vdash \neg x$. On the other hand, in ordinary classical logic we have $x \vdash \beta$ if, and only if, $\neg\beta \vdash \neg x$. Our swap rule (S) restores this symmetry and, as we show in the appendix, we get a system directly equivalent to Girard's.

Troelstra [14] has presented a system of natural deduction for intuitionistic linear logic in which there is no explicit rule for contraction. The contractions are managed implicitly via the use of labelled assumption formulae. An open assumption $a_x$ which occurs $k$ times ($k > 1$) in a deduction tree denotes the implicit application of a generalized contraction application, so that such occurrences count as a single instance of an hypothesis. We employ this notation in our system for natural deduction specified by the rules below. It is important to note that the use of multiple occurrences of a label is restricted by the conditions specified following the rules, so that a rule can be applied only if the deductions for the premisses satisfy the rule, and so also does the resulting deduction obtained by the rule application.

We use Troelstra's rule for $(!)$ and we use a rule for $(\&)$ mentioned by Bierman [4] but not included in his system. The rule $\top$ is used by Bierman and the weakening rule (W) can be found in Benton et al. [3].

### 2.3. The natural deduction system $\mathcal{N}$

Deductions are constructed using labelled assumptions and the following rules, subject to the restrictions specified below. The simplest deduction is just a labelled formula.
2.4. Restrictions

In the rule (\forall) the usual restriction applies that the individual variable \(x\) does not occur free in any of the open assumptions for the deduction. In the rule (\exists), \(a(x/t)\) denotes the result of substituting \(t\) for all free occurrences of \(x\) in \(a\), subject to the usual conditions for avoiding variable capture. The rule (\&) assumes that we have deductions of \(\alpha\) and \(\beta\) from the complete set of open assumptions \(\gamma_x, \ldots, \eta_Y\) and we discharge
the assumptions in both deductions by the introduction of deductions of \( \gamma, \ldots, \eta \) as shown. The rule \((!) \) assumes that we have a deduction of \( \beta \) from a complete set of open assumptions \( !\gamma, \ldots, !\eta \) and deductions \( \pi_\gamma, \ldots, \pi_\eta \) of \( !\gamma, \ldots, !\eta \), respectively. The deduction \( \pi_\gamma \) is substituted for \( !\gamma \), etc. It should be noted that in rules \((\land), (\neg \to), (\neg) \) where a labelled assumption is discharged, then all occurrences of this labelled formula are considered discharged. A deduction tree with set of open labelled assumptions \( \{!\gamma, \ldots, !\eta\} \) and conclusion \( \beta \) is a proof of the sequent \( \gamma, \ldots, \eta \vdash \beta \) in our system. Note that a given formula may occur more than once in the multiset of assumptions \( \gamma, \ldots, \eta \).

Troelstra restricts the use of multiple label occurrences as follows. If a label \( X \) for an open assumption \( \alpha \) in a deduction \( \pi \) occurs \( k \) times \( (k > 1) \), then \( \pi \) must be of the form

\[
\begin{array}{c}
\vdash \pi_1 \\
\vdash \pi_2 \\
\vdots \\
\vdash \pi_k \\
[!\beta, !\beta, \ldots, !\beta]
\end{array}
\]

where each \( \pi_i \) \( (i = 1, \ldots, k) \) is an isomorphic copy of a deduction \( \hat{\pi} \) of \( !\beta \), such that the label \( X \) occurs exactly once in each \( \pi_i \). Troelstra calls the set of occurrences of \( !\beta \) a substitution location, indicating that the substitution of a deduction at a location site must be carried out simultaneously at each occurrence. Then the deduction trees constructed via the rules above are all subject to the restriction on the use of multiple occurrences of a label; that is, to apply some rule \((R)\)

\[
\begin{array}{c}
\vdash \pi_1 \\
\vdash \pi_n \\
\vdots \\
\beta 
\end{array}
\]

we require that each deduction \( \pi_i \) satisfy the restriction of multiple label occurrences, and also that the resulting figure, after application of \((R)\) and any associated discharging of hypotheses, still satisfies the restrictions. Thus in the rule \((\&), \) if the discharge of hypotheses was not carried out then the resulting figure might no longer satisfy the restriction on label occurrences.

Our system with the label restrictions still gives a system of natural deduction which is equivalent to Girard's [6] sequent calculus in the following sense. Girard's system appears in the appendix to this paper.

**Lemma.** If in Girard's calculus the sequent \( \vdash \alpha_1, \ldots, \alpha_n \) is provable then we have \( \alpha_1^\perp, \ldots, \alpha_{i-1}^\perp, \alpha_{i+1}^\perp, \ldots, \alpha_n^\perp \vdash \alpha_i \) for each \( i = 1, \ldots, n \). Conversely, if in our system \( \beta_1, \ldots, \beta_n \vdash \alpha \), then the sequent \( \vdash \beta_1^\perp, \ldots, \beta_n^\perp, \alpha \) is provable in Girard's system.

**Proof.** Induction on the length of the proofs/deductions – see the appendix.
3. The Curry–Howard terms

This term system is based on the typed lambda calculus defined by Gallier [6] for a natural deduction system for intuitionistic first-order logic and the term system of Benton et al. [3] for intuitionistic linear logic. The principle of the system is to assign a Curry–Howard term to each stage of the deduction which encodes the construction of the deduction so far. This is done by representing a proof of a sequent \( \alpha_1, \ldots, \alpha_n \vdash \beta \) by a judgement \( \Gamma \vdash \beta \). Here \( \Gamma \) is called the context and is a finite set of the form \( X_1 : \alpha_1, \ldots, X_n : \alpha_n \), where the \( X_i \) are distinct term variables of type \( \alpha_i \) (though the \( \alpha_i \) need not be distinct), \( F \) is the Curry–Howard term, and \( \beta \) is its type. The terms are constructed as follows. For a deduction consisting of just \( \alpha_X \) we have the judgement \( X : \alpha \vdash X : \alpha \). Corresponding to each one of the logical rules we have a rule as listed below. The restriction on the use of term variables \( X : \alpha \) is precisely analogous to the restrictions on the use of labelled assumptions \( \alpha_X \) in our system \( \mathcal{N} \).

**Structural Rules:**

\[
\begin{align*}
\text{(S)} & \quad \frac{\Gamma \vdash \beta \quad \Delta \vdash \gamma}{\Gamma \cup \Delta \vdash \text{swap } X \text{ and } G \text{ using } F} : \beta, \gamma \\
\text{(W)} & \quad \frac{\Gamma \vdash F : \beta}{\Gamma \cup \Delta \vdash (\text{discard } F \text{ in } G) : \beta}
\end{align*}
\]

**Logical Rules:**

\[
\begin{align*}
\text{(T)} & \quad \frac{\Gamma \vdash F : \alpha, \Gamma \vdash G : \beta}{\Gamma \cup \Delta \vdash (F \otimes G) : \alpha \otimes \beta} \\
\text{(\&)} & \quad \frac{\Gamma \vdash \Delta \vdash \alpha, \Gamma \vdash \Delta \vdash \beta}{\Gamma \vdash \Delta \vdash \both(T_1, T_2) \text{ by } S_1, \ldots, S_n \text{ for } X_1, \ldots, X_n} : \alpha \& \beta \\
\text{(+\&)} & \quad \frac{\Gamma \vdash F : \alpha}{\Gamma \vdash \inl(F) : \alpha \oplus \beta} \\
\text{(+\&)} & \quad \frac{\Gamma \vdash F : \beta}{\Gamma \vdash \inr(F) : \alpha \oplus \beta} \\
\text{(!)} & \quad \frac{\Gamma \vdash \Delta \vdash \alpha, \Gamma \vdash \Delta \vdash \beta}{\Gamma \vdash \Delta \vdash (\text{promote } F_1, \ldots, F_n \text{ for } X_1, \ldots, X_n) : \beta} \\
\text{(?)} & \quad \frac{\Gamma \vdash F : \alpha}{\Gamma \vdash \derelict(F) : ?\alpha} \\
\text{(\&)} & \quad \frac{\Gamma \vdash F : \alpha}{\Gamma \vdash (\lambda x.F) : \forall \alpha} \\
\text{(\&)} & \quad \frac{\Gamma \vdash F : \alpha}{\Gamma \vdash (\exists x.F) : \exists \alpha}
\end{align*}
\]

The above formalism can be generalized to many sorted logics, and second-order logics, by assigning a type \( \iota \) to every individual variable and individual term. In this
case, the quantified formulae are written as $\forall x : \alpha$ and $\exists x : \alpha$, and the rule $(\forall)$ is written as follows:

\[
\Gamma \triangleright F : \alpha \\
\Gamma \triangleright (\lambda x : \alpha. F) : \forall \alpha x
\]

Note that the only rule in which an individual variable becomes bound in a Curry--

Howard term is the rule $(\forall)$.

A free term variable $X : \alpha$ in a Curry--Howard term corresponds to an open assump-

tion $\alpha x$ in the corresponding deduction in $\mathcal{N}$. For the conclusions of the rules $(S)$ and

$(\neg \circ)$ the term variable $X$ is bound. For the conclusions of the rules $(\&)$ and $(!)$ the

term variables $X_1, \ldots, X_n$ are all bound.

A term which has no free term variables is called closed and a term which has free

term variables is called open. A closed term may have free individual variables, and a

closed term which has no free individual variables is called totally closed.

Lemma 2. If $X_1 : \alpha_1, \ldots, X_n : \alpha_n \triangleright F : \beta$ is a judgement, then the distinct free term

variables of $F$ are $X_1, \ldots, X_n$.

Theorem 1 (Unique derivation). (1) Given a proof of the sequent $\Gamma \vdash \beta$ in $\mathcal{N}$ there

is a corresponding judgement $\Gamma \triangleright F : \beta$ which is unique up to renaming of variables

and reordering of premisses.

(2) There is an algorithm which given a judgement $\Gamma \triangleright F : \beta$ obtains from it a

proof of the corresponding sequent $\Gamma \vdash \beta$.

Proof. (1) By the correspondence between the rules in $\mathcal{N}$ and the formation rules of

the Curry--Howard terms.

(2) Given $F : \beta$ we can recover the Curry--Howard terms for the premisses of the

rule with $F : \beta$ as the conclusion. From the term tree with $F : \beta$ as root node we can

construct a proof of $\Gamma \vdash \beta$. □

Theorem 2 (Substitution). (a) Suppose that $\Delta, X_1 : \alpha_1, \ldots, X_n : \alpha_n \triangleright G : \beta$, and $\Gamma_i \triangleright T_i : \alpha_i, i = 1, \ldots, n$ are all judgements. Then, if the term variables in $\Delta, \Gamma_1, \ldots, \Gamma_n$ are all

distinct,

\[
\Delta, \Gamma_1, \ldots, \Gamma_n \triangleright G[X_1/T_1, \ldots, X_n/T_n]
\]

is also a judgement.

(b) If $t$ is an individual term and each of the following are judgements:

\[
\Delta, X_1 : \alpha_1, \ldots, X_n : \alpha_n \triangleright G : \beta,
\]

\[
\Gamma_i \triangleright X_i : \alpha_i[x/t] \quad \text{for } i = 1, \ldots, n,
\]

and the term variables in $\Delta, \Gamma_1, \ldots, \Gamma_n$ are all distinct, then

\[
\Delta', \Gamma_1, \ldots, \Gamma_n \triangleright G : \beta[x/t]
\]
is also a judgement, where \( \Delta' \) denotes the substitution of \( t \) for \( x \) in the type of each term variable in \( \Delta \).

**Proof.** The proof is by induction on the construction of the judgement \( \Delta, X_1 : \alpha_1, \ldots, X_n : \alpha_n \vdash G : \beta \), in both cases.

4. Reductions

Reductions of proofs play a dual rôle. On the one hand, they eliminate unnecessary steps in a proof, e.g. writing \( \rightarrow \) for "reduces":

\[
\frac{\gamma \vdash [\gamma] (S)}{\gamma \vdash [\gamma] (S)} \rightarrow
\]

On the other hand, they provide the dynamics of the terms, i.e. they describe how the computation should proceed. For the original Curry–Howard terms (the typed lambda terms) the reductions correspond to \( \beta \)-reductions. For our term system, many of the reductions correspond to "contractions" as defined in Girard [7]. The reduction above gives the rule

\[\text{swap } Z \text{ and } Z \text{ using } F \rightarrow F.\]

A reduction involving the pair \((\&, \oplus_L)\) is indicated by

\[
\begin{array}{cccccccc}
\Gamma & \Gamma & \Delta_1 & \Delta_n & \delta & \Delta_1 & \Delta_n \\
\vdash \pi_a & \vdash \pi_\beta & \vdots & \vdots & \vdots & \vdots & \\
\alpha & \beta & \gamma & \ldots & \eta & \alpha_\perp \\

\hline
\alpha \& \beta & a_\perp \oplus \beta_\perp & (\&L) & \vdash \pi_\alpha & \vdots & \\
\delta_\perp & (S) & \\
\end{array}
\]

where \( \Gamma = \{\gamma \ldots \eta\} \). In term form we have the reduction

\[\text{swap } Z \text{ and inl}(H) \text{ using both } (T_1, T_2 \text{ by } S_1, \ldots, S_n \text{ for } X_1, \ldots, X_n) \rightarrow \text{swap } Z \text{ and } H \text{ using } T_1[X_1/S_1, \ldots, X_n/S_n].\]

The full list of term reductions is as follows:

\( (-o, \otimes) \) swap \( Z \) and \( (F(Z) \otimes G) \) using \( (\lambda Y.H) \rightarrow \text{swap } Z \) and \( H[Y/F] \) using \( G \)

\( \text{swap } Z \) and \( (F \otimes G(Z)) \) using \( (\lambda Y.H) \rightarrow \text{swap } Z \) and \( G \) using \( H[Y/F] \)

\( \text{swap } Z \) and \( (\lambda Y.H)(Z) \) using \( (F \otimes G) \rightarrow \text{swap } Z \) and \( H[Y/F] \) using \( G \)

\( (\&, \oplus_I) \) swap \( Z \) and \( \text{both } (T_1, T_2 \text{ by } S_1, \ldots, S_n \text{ for } X_1, \ldots, X_n) \) using \( \text{inl } (H) \)

\[ \rightarrow \text{swap } Z \) and \( T_1[X_1/S_1, \ldots, X_n/S_n] \) using \( H \)
\[\text{swap } Z \text{ and } \text{inl } (H) \text{ using both } (T_1, T_2 \text{ by } S_1, \ldots, S_n \text{ for } X_1, \ldots, X_n)\]
\[\rightarrow \text{swap } Z \text{ and } H \text{ using } T_1[X_1/X_1, \ldots, X_n/X_n].\]

\((\&, \oplus, \odot)\) similar to the above, using \text{inr} instead of \text{inl} and \(T_2\) instead of \(T_1\).

\((!, ?)\)
\[\text{swap } Z \text{ and derelict}(H) \text{ using } (\text{promote } F_1, \ldots, F_n \text{ for } Y_1, \ldots, Y_n \text{ in } G)\]
\[\rightarrow \text{swap } Z \text{ and } H \text{ using } G[Y_1/F_1, \ldots, Y_n/F_n]\]
\[\text{swap } Z \text{ and } (\text{promote } F_1, \ldots, F_n \text{ for } Y_1, \ldots, Y_n \text{ in } G) \text{ using derelict}(H)\]
\[\rightarrow \text{swap } Z \text{ and } G[Y_1/F_1, \ldots, Y_n/F_n] \text{ using } H\]

\((\forall, \exists)\)
\[\text{swap } Z \text{ and } (t, F) \text{ using } \lambda x. G \rightarrow \text{swap } Z \text{ and } F \text{ using } G[x/t]\]
\[\text{swap } Z \text{ and } \lambda x. G \text{ using } (t, F) \rightarrow \text{swap } Z \text{ and } G[x/t] \text{ using } F\]

\((S, S')\)
\[\text{swap } Z \text{ and } F \text{ using } (\text{swap } Y \text{ and } G \text{ using } H) \rightarrow \text{swap } Z \text{ and } G[Y/F] \text{ using } H\]

Now suppose that the last rule that formed \(F\) was not \((S)\). Then
\[\text{swap } Z \text{ and } (\text{swap } Y \text{ and } G(Z) \text{ using } H) \rightarrow \text{swap } Z \text{ and } G[Y/F] \text{ using } H\]
\[\text{swap } Z \text{ and } (\text{swap } Y \text{ and } G \text{ using } H(Z)) \text{ using } F \rightarrow \text{swap } Z \text{ and } H \text{ using } G[Y/F]\]

Lemma 3. Suppose \(F : \beta\) is a term with the free individual variables \(x = x_1, \ldots, x_n\), the free term variables \(X = X_1 : X_1, \ldots, X_m : X_m\), and \(F \rightarrow G\). Then \(G\) is a term of type \(\beta\) with the free individual variables \(x\) and the free term variables \(X\).

5. Normalization and the Church–Rosser property

The reductions in Section 4 correspond to conversions in an \(\mathcal{N}\) deduction. However, suppose a deduction \(\pi\) contains \(k\) \((k > 1)\) occurrences of a label \(X\) for an open assumption, so that there are \(k\) isomorphic copies of some deduction \(\pi'\). If we carry out a reduction step inside just one of these copies then the isomorphism is destroyed, and we no longer have an \(\mathcal{N}\) deduction. Thus any reduction inside a copy of \(\pi'\) must be carried out simultaneously in all such copies. We write \(\pi \rightarrow^* \sigma\) if \(\sigma\) is obtained from \(\pi\) by a single reduction step, possibly applied repeatedly, as just described. Note that if \(\pi\) contains no open assumptions (it may still contain some isomorphic sub-deductions) then the above restriction is no longer applicable. A deduction is in normal form if no reduction step can be applied.

Analogous restrictions and definitions to the above apply to our Curry–Howard terms, so we will also write \(T \rightarrow S\) if a term \(T\) reduces to term \(S\) via an application of one of the reductions in Section 4. A term \(T\) is strongly normalizable if all reduction sequences \(T \rightarrow^* S \rightarrow^* \ldots\) are finite. We write \(T \Rightarrow W\) if there is a finite sequence of \(\rightarrow^*\) reductions from \(T\) to \(W\).

Theorem 3. Every term is strongly normalizable.
Proof. Define the degree of a swap application in $\mathcal{N}$, say

\[
\beta \frac{\pi_{\beta}}{\gamma} (\Sigma),
\]

by $1 + \deg(\beta)$ where $\deg(\beta)$ is the number of logical symbols in $\beta$. Define the degree of a deduction in $\mathcal{N}$ by $\text{degree}(\pi) = \text{sum of the degrees of the swaps in } \pi$, counted modulo isomorphic sub-deductions in $\pi$, that is to say that, if a label $X$ for an open assumption $x$ in $\pi$ occurs $k$ times ($k > 1$), then the contribution to $\text{degree}(\pi)$ of the associated isomorphic deductions $\pi_i$ (see Section 2.4) is calculated by considering only one of the $\pi_i$. Then in each of the reductions involving dual symbols, $(\neg, \otimes)$, etc., the degree of the swap decreases.

We have already seen an example of a $(\&$, $\oplus_1)$ reduction in $\mathcal{N}$ in Section 4. Some typical cases for $(\neg, \otimes)$, $(1, ?)$ and $(\lor, \exists)$ reductions are shown below. In each case the degree of the swap is decreased by one.

\[
\frac{\prod \beta (\neg) \quad \alpha \frac{\pi_{\beta}}{\gamma} \otimes \beta \oplus \gamma}{\prod \beta (\alpha/\gamma)} (S) \quad \frac{\pi_{\beta}}{\gamma} (S)
\]

\[
\frac{\pi_1 \quad \pi_n}{\prod \beta (1)} \quad \frac{\beta \quad \beta \frac{\pi_{\beta}}{\gamma}}{\beta \frac{\pi_{\beta}}{\gamma} (S) \quad \beta \frac{\pi_{\beta}}{\gamma} (S)} \quad \frac{\pi_{\beta}}{\gamma} (S)
\]

For the cases of $(\lor, \lor)$, and also the first of the listed reductions, a swap is removed from $\pi$. In all cases, if $\pi \rightarrow \sigma$, then $\text{degree}(\sigma) < \text{degree}(\pi)$, so we have strong normalization. □

Define $\rightarrow^*$ as the reflexive closure of the relation $\rightarrow$. Then $\rightarrow^*$ satisfies the diamond property that if $T \rightarrow^* T_1$ and $T \rightarrow^* T_2$, then there exists a $T_3$ such that $T_1 \rightarrow^* T_3$.
and $T_2 \overset{!\ast}{\rightarrow} T_3$. This result can be established by induction on the complexity of the term $T$. Unlike the analogous case for the pure lambda calculus, we are able to carry out identical simultaneous reductions (or substitutions) in isomorphic sub-terms and the resultant term formed is an immediate reduct of $T$. As an example, in our system consider a term $T$ of the form

$$\text{swap } Z \text{ and } (F(Z) \otimes G) \text{ using } (\lambda Y \cdot H)$$

where $H$ contains two occurrences of $Y$. Then we have the reduction $T \rightsquigarrow \text{swap } Z$ and $H[Y/F]$ using $G$, where $F$ has been substituted twice. We also have that $T \rightsquigarrow \text{swap } Z$ and $(F'(Z) \otimes G)$ using $(\lambda Y \cdot H)$, where $F(Z) \rightsquigarrow F'(Z)$. But now $\text{swap } Z$ and $H[Y/F]$ using $G$, by our rule for conversions in isomorphic sub-terms.

As a corollary of the diamond property of $\overset{!\ast}{\rightarrow}$, we obtain the Church–Rosser property for the relation $\rightarrow$ by the usual argument.

6. Program extraction

**Theorem 4** (Normalized form). Let $T : \gamma$ be a closed normalized term, which is not of the form (discard $F$ in $G$).

1. If $\gamma = T$, then $T = \text{true}(F_1, \ldots, F_n)$, for some closed terms $F_1, \ldots, F_n$.
2. If $\gamma = \alpha \otimes \beta$, then $T = F \otimes G$, for some closed terms $F : \alpha$ and $G : \beta$.
3. If $\gamma = \alpha \cdot \circ \beta$, then $T = \lambda X : \alpha.F$, for some term $F : \beta$, where the only free term variable in $F$ is $X : \alpha$.
4. If $\gamma = \alpha \& \beta$, then $T = F \& G$, for some closed terms $F : \alpha$ and $G : \beta$.
5. If $\gamma = \alpha \oplus \beta$, then either $T = \text{inl}(F)$, for some closed term $F : \alpha$, or $T = \text{inr}(G)$, for some closed term $G : \beta$.
6. If $\gamma = !\alpha$, then either $T = \text{promote}(F$ for $X_1, \ldots, X_n$ in $G$), for some closed terms $F = F_1 : !\alpha_1, \ldots, F_n : !\alpha_n$, or $T$ is computable from $F$.
7. If $\gamma = \forall x \cdot \alpha$, then $T = \lambda x.F$, for some closed term $F : \alpha$.
8. If $\gamma = \exists x \cdot \alpha$, then $T = (t, F)$, for some closed term $F : \alpha(x/t)$.

**Note.** If $T : \gamma$ is of the form (discard $F$ in $G$) then we can consider instead the term $G : \gamma$.

**Theorem 5.** Suppose $\mathcal{N} \vdash \forall x \exists y \cdot \alpha(x, y)$. Then, for each individual term $t$ there exists an individual term $u$ such that $\vdash \alpha(t, u)$ and $u$ is computable from $t$.

**Proof.** Suppose $\vdash \forall x \exists y \cdot \alpha(x, y)$ and $t$ is an individual term. Then

$$
\frac{
\vdash \forall x \exists y \cdot \alpha(x, y) \\
\forall y \cdot \alpha^-(t, y) \vdash \forall y \cdot \alpha^-(t, y)}{
\forall y \cdot \alpha^-(t, y) \vdash \exists y \forall y \cdot \alpha^-(x, y)} \quad \text{(3)}
$$

$$
\vdash \exists y \cdot \alpha(t, y) \quad \text{(S)}
$$
So, by Theorem 1 and Lemma 2, there exists a unique closed term, \( F: \exists y \alpha(t, y) \) corresponding to the proof of \( \exists y \alpha(t, y) \). Then, by Lemma 3 and Theorem 3, there exists a reduct of \( F \), say \( G: \exists y \alpha(t, y) \), which is a closed normal term. Therefore, by Theorem 4, \( G = (u, \Pi) \), where \( u \) is an individual term and \( \Pi \) is a closed normal term. As each of these steps is computable, \( u \) is computable from \( t \). \( \square \)

Theorem 5 follows the usual approach of extracting programs from proofs and therefore the usual benefits, and costs, in using this method to obtain programs which are correct (in the sense that they satisfy their specifications) accrue here too. In addition, since linear logic is a resource logic (see [8]), the terms will indicate what resources are required in running the program.

The use of the term calculus to construct practical programming systems is obviously of interest. There is a problem here. The development of an analogue of, say, Coquand and Huet's [5] calculus of constructions would require a very significant additional development of the techniques for proving theorems in the formal system of linear logic. Such techniques are not yet found in the literature. Nevertheless, the methodology of linear logic together with its emphasis on resources and parallelism should lead to a more sophisticated analysis of the process of developing programs.

Acknowledgements

We wish to thank F.A. Bäuerle for his important contributions in the early stages of this project. Also, thanks to J.C. Shepherdson and G. Bierman for their many comments on our earlier paper with F.A. Bäuerle, which were very useful in the writing of this paper.

Appendix. The equivalence of system \( \mathcal{N} \) with Girard's sequent calculus

Girard presents a sequent calculus for linear logic in which all sequents have no left-hand side. The linear negation operation on formulae allows for such an economical presentation of the sequent calculus. In Girard's system, the connective \( \otimes \) is taken as primitive and linear implication is defined by taking \( \alpha \circ \beta \) as standing for \( \alpha \perp \otimes \beta \). In our system \( \circ \) is primitive (see Section 2.1). A version of Girard's sequent calculus using \( \circ \) as primitive is given by the following axioms and rules, where upper-case Greek letters denote sequences of formulae.

**Axioms**

\( \vdash \alpha, \alpha \perp \vdash 1 \)
\( \vdash \top, \Delta \)

**Rules**

\[
\frac{\vdash \alpha, \Delta, \alpha \perp, \Sigma}{\vdash \Delta, \Sigma} \quad (\text{cut})
\]
where \( \Sigma \) is a permutation of \( \Delta \):

\[
\frac{\vdash \Delta}{\vdash \Sigma} \quad \text{(ex)}
\]

\( \vdash \Delta \) is a permutation of \( \Delta \):

\[
\frac{\vdash \Delta}{\vdash \bot} \quad \text{(\( \bot \))}
\]

\[
\frac{\vdash \alpha, \Delta \quad \vdash \beta, \Sigma}{\vdash \alpha \otimes \beta, \Delta, \Sigma} \quad \text{(?)}
\]

\[
\frac{\vdash \alpha, \beta, \Delta}{\vdash \alpha \rightarrow \beta, \Delta} \quad \text{(-\( \rightarrow \))}
\]

\[
\frac{\vdash \alpha, \Delta}{\vdash \alpha \oplus \beta, \Delta} \quad \text{(?\( L \))}
\]

\[
\frac{\vdash \alpha, \Delta}{\vdash \alpha \& \beta, \Delta} \quad \text{(?\( \& \))}
\]

\[
\frac{\vdash \beta, A}{\vdash \alpha \oplus \beta, A} \quad \text{(?\( R \))}
\]

\[
\frac{\vdash ? \alpha, A}{\vdash \forall \alpha \Delta} \quad \text{(?\( \forall \))}
\]

\[
\frac{\vdash A}{\vdash \exists \alpha \Delta} \quad \text{(\( \exists \))}
\]

\( x \) not free in \( \Delta \)

\textbf{Lemma A.1.} If the sequent \( \vdash \alpha_1, \ldots, \alpha_2 \) is provable in Girard’s system, then we have in system \( \mathcal{N} \) the deductions \( \alpha_1^\perp, \ldots, \alpha_2^\perp, \alpha_{1'}^\perp, \ldots, \alpha_{n'}^\perp \vdash \alpha_i \) for \( i = 1, \ldots, n \).

\textbf{Proof.} The proof is by induction on the length of the proof of \( \vdash \alpha_1, \ldots, \alpha_n \). For the three cases where this proof consists solely of an axiom, we have the following:

(i) \( \vdash \alpha, \alpha^\perp \) corresponds to the trivial deductions \( \alpha^\perp \vdash \alpha^\perp \),

or \( \alpha \vdash \alpha \);

(ii) Recalling that \( 1 = ! \top \) by definition, we have that \( \vdash 1 \) corresponds to the deduction

\[
\frac{\vdash [x]}{\vdash \forall \alpha \perp \pi} \quad \text{(-\( \rightarrow \))}
\]

\[
\frac{\vdash 1}{\vdash \top} \quad \text{(\( \top \))}
\]

\[
\frac{\vdash \top}{\vdash ! \top} \quad \text{(!)}
\]
(iii) \( \vdash T, x_1, \ldots, x_n \) has associated deductions

\[
\frac{x_1^\perp \ldots x_n^\perp}{T} (T), \quad \text{and} \quad \frac{T}{x_i} (S)
\]

for \( i = 1, \ldots, n \).

For the case of non-trivial proofs using one or more rules, we have the following.

1. The proof ends with a \((\text{cut})\),

\[
\frac{\vdash \alpha, \beta_1, \ldots, \beta_n \vdash \gamma_1, \ldots, \gamma_m}{\vdash \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_m} \quad \text{(cut)}
\]

then by the induction hypothesis we have deductions \( \beta_1^\perp, \ldots, \beta_n^\perp \vdash \alpha \) and \( \gamma_1^\perp, \ldots, \gamma_m^\perp \vdash \alpha^\perp \). An application of rule \((S)\) then yields deductions of \( \beta_1^\perp, \ldots, \beta_n^\perp, \gamma_1^\perp, \ldots, \gamma_m^\perp \vdash \beta_i \), for \( i = 1, \ldots, n \). Similarly, we have \( \beta_1^\perp, \ldots, \beta_n^\perp, \gamma_1^\perp, \ldots, \gamma_m^\perp \vdash \alpha^\perp \), for \( j = 1, \ldots, m \).

2. The proof ends with \((\text{ex})\). This case is trivial.

3. The proof ends with a \((\perp)\),

\[
\frac{\vdash \perp, x_1, \ldots, x_n}{\vdash \perp, x_1, \ldots, x_n} \quad (\perp)
\]

From the induction hypothesis we have \( x_1^\perp, \ldots, x_i^\perp, x_{i+1}^\perp, \ldots, x_n^\perp \vdash x_i \) so by rule \((W)\) we can form \( \vdash T, x_1^\perp, \ldots, x_i^\perp, x_{i+1}^\perp, \ldots, x_n^\perp \vdash x_i \) where \( \vdash T \) is \((\perp)^\perp \) in our system. We can now also form a deduction \( x_1^\perp, \ldots, x_i^\perp, \ldots, x_n^\perp \vdash x_i \perp \perp \) as follows:

\[
\frac{T, x_1^\perp, \ldots, x_i^\perp, x_{i+1}^\perp, \ldots, x_n^\perp \vdash x_i}{x_1^\perp, \ldots, x_i^\perp \vdash x_i} (S)
\]

4. The proof ends with \((\otimes)\),

\[
\frac{\vdash \alpha, \gamma_1, \ldots, \gamma_n \vdash \beta, \delta_1, \ldots, \delta_m}{\vdash \alpha \otimes \beta, \gamma_1, \ldots, \gamma_n, \delta_1, \ldots, \delta_m} \quad (\otimes)
\]

We have from the induction hypothesis, \( \gamma_1^\perp, \ldots, \gamma_n^\perp \vdash \alpha \) and \( \delta_1^\perp, \ldots, \delta_m^\perp \vdash \beta \). An application of the rule \((\otimes)\) then yields \( \gamma_1^\perp, \ldots, \gamma_n^\perp, \delta_1^\perp, \ldots, \delta_m^\perp \vdash \alpha \otimes \beta \). An application of rule \((S)\) then yields \( \gamma_1^\perp, \ldots, \gamma_n^\perp, \delta_1^\perp, \ldots, \delta_m^\perp \vdash \gamma_j \), or \( \alpha \otimes \beta^\perp \), \( \gamma_1^\perp, \ldots, \gamma_n^\perp, \delta_1^\perp, \ldots, \delta_m^\perp \vdash \delta_j \).

5. The proof ends with \((\&)\),

\[
\frac{\vdash \alpha, \gamma_1, \ldots, \gamma_n \vdash \beta, \gamma_1, \ldots, \gamma_n}{\vdash \alpha \& \beta, \gamma_1, \ldots, \gamma_n} \quad (\&)
\]
We have \( \gamma_1^\perp, \ldots, \gamma_n^\perp \vdash \alpha \) and \( \gamma_1^\perp, \ldots, \gamma_n^\perp \vdash \beta \), and we can form the deduction (suppressing the labels on formulae)

\[
\begin{array}{cccc}
\gamma_1^\perp & \vdots & \cdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\alpha & \beta & \gamma_1^\perp & \gamma_n^\perp
\end{array} \\
\hline
\alpha \& \beta
\]

The use of rule \((S)\) then allows us to form also a deduction \((\alpha \& \beta)^\perp, \gamma_1^\perp, \ldots, \gamma_n^\perp, \gamma_{i+1}^\perp, \ldots, \gamma_m^\perp \vdash \gamma_i^\perp\).

6. The proof ends in either a \((+L)\), \((+R)\), \((-\circ)\), \((?\circ)\) or a \(!\circ)\). These cases are quite routine, using similar arguments to 1–5 above. Similarly for proofs ending in \((\forall)\) or \((\exists)\).

7. The proof ends in a \((W?)\)

\[
\frac{\vdash \alpha_1, \ldots, \alpha_n}{\vdash ?\alpha, \alpha_1, \ldots, \alpha_n} \quad (W?)
\]

We have \( \alpha_1^\perp, \ldots, \alpha_n^\perp \vdash \alpha_1 \) by the induction hypothesis.

An application of rule \((W)\) to \( \alpha_1^\perp \vdash \alpha_1 \) gives us \(!\alpha^\perp, \alpha_1^\perp \vdash \alpha_1^\perp\) and now we form the deduction below.

\[
\begin{array}{cccc}
\alpha_2^\perp & \vdots & \cdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\alpha_1^\perp & \alpha_1^\perp & (S)
\end{array} \\
\hline
?\alpha
\]

8. The final case is \((C?)\),

\[
\frac{\vdash ?\alpha, ?\alpha, \beta_1, \ldots, \beta_n}{\vdash ?\alpha, \beta_1, \ldots, \beta_n} \quad (C?)
\]

Using the induction hypothesis, we have a deduction of

\[
!\alpha^\perp, !\alpha^\perp, \beta_1^\perp, \ldots, \beta_{i-1}^\perp, \beta_{i+1}^\perp, \ldots, \beta_n^\perp \vdash \beta_i.
\]

Thus, we have a labelled deduction tree with open labelled assumptions, say \(!\alpha^\perp\) and \(!\alpha^\perp\), corresponding to the two occurrences of \(!\alpha^\perp\) in the above sequent. Now re-label all open occurrences of \(!\alpha^\perp\) and \(!\alpha^\perp\) to say \(!\alpha^\perp\). Thus yields a labelled deduction tree for the sequent \(!\alpha^\perp, \beta_1^\perp, \ldots, \beta_{i-1}^\perp, \beta_{i+1}^\perp, \ldots, \beta_n^\perp \vdash \beta_i\), and then apply rule \((S)\) to give a deduction \( \beta_1^\perp, \ldots, \beta_{i-1}^\perp, \beta_{i+1}^\perp, \ldots, \beta_n^\perp \vdash ?\alpha \). \(\square\)

**Lemma A.2.** If \( \beta_1, \ldots, \beta_n \vdash \alpha \) holds, then the sequent \( \vdash \alpha, \beta_1^\perp, \ldots, \beta_n^\perp \) is provable in Girard’s system.
Proof. We argue by induction on the length of the deduction in $\mathcal{N}$. For the trivial deduction, $\vdash \alpha$, we have the axiom $\vdash \alpha, \alpha^\perp$ in Girard's system. The non-trivial cases are the following.

1. The deduction ends with rule (S),

$$
\begin{array}{c}
\Delta \\
\vdots \\
\alpha \\
\end{array}
\frac{\beta}{\alpha^\perp} (S)
$$

By the induction hypothesis, both $\vdash \alpha, \Delta^\perp$ and $\vdash \alpha^\perp, \beta^\perp, \Gamma^\perp$ are provable in Girard's system. An application of (cut) then gives a proof of $\vdash \beta^\perp, \Gamma^\perp, \Delta^\perp$.

2. The deduction ends with rule (W),

$$
\frac{\Gamma \Delta}{\alpha \beta} \frac{\alpha \beta^\perp}{\beta^\perp} (W)
$$

The induction hypothesis gives proofs $\vdash \alpha, \Delta^\perp$ and $\vdash \beta, \Delta^\perp$. We construct the following proof of $\vdash \beta, \Delta^\perp, \Gamma^\perp$.

$$
\frac{\vdash \beta, \Delta^\perp}{\vdash \alpha^\perp, \beta, \Delta^\perp (W?)} \frac{\vdash \alpha, \Gamma^\perp}{\vdash \beta, \Delta^\perp, \Gamma^\perp (cut)}
$$

3. For a deduction ending in rule (T),

$$
\begin{array}{c}
\Gamma_1 \\
\vdots \\
\alpha_i \\
\end{array}
\frac{T}{\vdash T, \alpha_i^\perp} (T)
$$

we have that $\vdash \alpha_i, \Gamma_i^\perp$ is provable in Girard's system for each $i = 1, \ldots, n$. Form a proof of $\vdash T, \Gamma_1^\perp, \ldots, \Gamma_n^\perp$ as follows, using the axiom $\vdash T, \alpha_i^\perp, \ldots, \alpha_n^\perp$,

$$
\frac{\vdash T, \alpha_i^\perp, \ldots, \alpha_n^\perp}{\vdash T, \alpha_i^\perp, \Gamma_i^\perp (cut)} \frac{\vdash \alpha_2, \Gamma_2^\perp}{\vdash T, \alpha_3^\perp, \ldots, \alpha_n^\perp, \Gamma_3^\perp, \Gamma_2^\perp (cut)} \frac{\vdash \alpha_n, \Gamma_n^\perp}{\vdash T, \Gamma_1^\perp, \ldots, \Gamma_n^\perp (cut)}
$$

4. For deductions ending with any of the rules ($\otimes$, ($\ominus$), ($\oplus$), ($\ominus$), ($\otimes$), ($\ominus$), ($\otimes$), ($\ominus$), the argument is routine.
5. For a deduction ending with (&),

\[
\begin{array}{cccccc}
\forall_1, \ldots, \forall_n & \forall_1, \ldots, \forall_n & \Delta & \Sigma \\
\vdots & \vdots & \vdots & \vdots \\
\alpha & \beta & \gamma & \ldots & \eta \\
\hline & & & \alpha \& \beta
\end{array}
\]

we have proofs \( \vdash \alpha, \gamma^\perp, \ldots, \eta^\perp, \vdash \beta, \gamma^\perp, \ldots, \eta^\perp \) and \( \vdash \gamma, \Delta^\perp, \ldots, \eta, \Sigma^\perp \). Form the proof \( \vdash \alpha \& \beta, \Delta^\perp, \ldots, \Sigma^\perp \) as shown:

\[
\begin{array}{c}
\vdash \alpha, \gamma^\perp, \ldots, \eta^\perp \\
\vdash \beta, \gamma^\perp, \ldots, \eta^\perp
\end{array}
\]  
\( \text{(cut)} \)

\[
\vdash \alpha \& \beta, \gamma^\perp, \ldots, \eta^\perp \\
\vdash \alpha \& \beta, \Delta^\perp, \ldots, \eta^\perp
\]  
\( \text{cut} \)

6. For a deduction ending in a (!),

\[
\begin{array}{cccc}
\Gamma & \Delta & \\
\vdots & \vdots & \\
[!\gamma & \ldots & !\eta] & \\
\vdots & \\
\beta & \\
\hline & \vdash \beta
\end{array}
\]

we have proofs \( \vdash \beta, \gamma^\perp, \ldots, \eta^\perp \), and also proofs \( \vdash !\gamma, \Gamma^\perp, \ldots, \vdash !\eta, \Delta^\perp \). Form the following proof of \( \vdash !\beta, \Gamma^\perp, \ldots, \Delta^\perp \):

\[
\begin{array}{c}
\vdash \beta, \gamma^\perp, \ldots, \eta^\perp \\
\vdash !\beta, \gamma^\perp, \ldots, \eta^\perp
\end{array}
\]  
\( \text{(cut)} \)

\[
\vdash !\beta, \Gamma^\perp, \ldots, \eta^\perp
\]  
\( \text{cut} \)

\[
\vdash !\beta, \Gamma^\perp, \ldots, \Delta^\perp
\]  
\( \square \)

References

5th Internat. Workshop on Functional and Logic Programming, held at the Castle of Ruischholzhausen,
Lambda Calculi and Applications, International Conf. TLCA ’93, Springer Lecture Notes in Computer