A Mixed Linear and Non-Linear Logic: Proofs, Terms and Models (Preliminary Report)*

P. N. Benton†
University of Cambridge

Abstract

Intuitionistic linear logic regains the expressive power of intuitionistic logic through the ! ('of course') modality. Benton, Bierman, Hyland and de Paiva have given a term assignment system for ILL and an associated notion of categorical model in which the ! modality is modelled by a comonad satisfying certain extra conditions. Ordinary intuitionistic logic is then modelled in a cartesian closed category which arises as a full subcategory of the category of coalgebras for the comonad.

This paper attempts to explain the connection between ILL and IL more directly and symmetrically by giving a logic, term calculus and categorical model for a system in which the linear and non-linear worlds exist on an equal footing, with operations allowing one to pass in both directions. We start from the categorical model of ILL given by Benton, Bierman, Hyland and de Paiva and show that this is equivalent to having a symmetric monoidal adjunction between a symmetric monoidal closed category and a cartesian closed category. We then derive both a sequent calculus and a natural deduction presentation of the logic corresponding to the new notion of model.

*A shorter version of this paper is to be presented at, and submitted to the proceedings of, the 1994 Annual Conference of the European Association for Computer Science Logic, Kazimierz, Poland.

†Author's address: University of Cambridge, Computer Laboratory, New Museums Site, Pembroke Street, Cambridge CB2 3QG, United Kingdom. Email: Nick.Benton@cl.cam.ac.uk. Research supported by a SERC Fellowship and the EU Esprit project LOMAPS.

Contents

1 Introduction

1.1 Background

This paper concerns a variant of the intuitionistic fragment of Girard's linear logic [Gir87]. As is well-known, linear logic does not contain the structural rules of weakening and contraction, but these are reintroduced in a controlled way via a unary operator, written ! and pronounced 'of course', 'bang' or 'shriek'. The sequent calculus rules for ! are the following:

$$\frac{!\Gamma \vdash A}{!\Gamma \vdash !A}\ \text{Promotion} \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ \text{Dereliction}$$

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ \text{Contraction} \qquad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ \text{Weakening}$$

The rules above allow ordinary intuitionistic logic to be interpreted within intuitionisitic linear logic via (for example) the so-called 'Girard translation'. In [BBHdP92, BBHdP93b, BBHdP93a], Benton, Bierman, Hyland and de Paiva formulated a natural deduction presentation of the multiplicative/exponential fragment of ILL, together with a term calculus (extending the propositions as types analogy to linear logic) and a categorical model (a *linear category*). In that work, the multiplicative (i.e. $\otimes, \multimap$) part of the logic is modelled in a symmetric monoidal closed category (SMCC). That much is standard and well-understood. The ! modality is then modelled by a monoidal comonad on the SMCC which is required to satisfy certain extra (and non-trivial) conditions. These extra conditions are sufficient to ensure that the category of coalgebras for the comonad contains a full subcategory which is cartesian closed and thus models the interpretation of IL in ILL.

Whilst the view that linear logic is primary and that ordinary logic is merely a part of linear logic is appealing (particularly if one takes seriously the claims of linear logic to be "the logic behind logic"), it is not necessarily always the best way of seeing the situation. This paper tries to present a more symmetric view of the relationship between IL and ILL, starting from a model-theoretic perspective, and it seems worth trying to give some motivation for why this might be worth doing.

1.2 Motivation

1.2.1 Functional Programming

From a practical point of view, there are a number of reasons why the standard linear term calculus (LTC) of [BBHdP92] might be considered unsuitable as the basis of a linear functional programming language. Firstly, linear functional programming is verbose and unnatural – whilst the linear term calculus might well be a useful intermediate language for a compiler, it is not very appropriate as a language for everyday programming. If linearity is to be made visible to the programmer at all, it appears preferable to have some extension of a traditional non-linear language in which one could write the occasional linear function in order to deal with input/output, in-place update or whatever.

A second, more fundamental, problem is that, despite considerable research effort, the *precise* way in which a linear language can help with what we have deliberately referred to rather vaguely as 'input/output, in-place update or whatever' is still not clear. Most published proposals for using linear types to control or describe intensional features of functional programs are either unconvincing or use type systems which are only loosely inspired by linear logic. Systems in the last category can, pragmatically, be extremely successful; the most obvious example being the language CLEAN. The type system of CLEAN [BS93] incorporates a 'uniqueness' operator for (roughly) making non-linear types linear. This is in some sense dual to the ! of linear logic, which allows linear types to be treated non-linearly. Unique types in CLEAN are used to add destructive updates and I/O to the language in a clean (referentially transparent) way.

One (currently somewhat speculative) aim of the work described here is to provide a sound mathematical and logical basis for a type system like that of CLEAN. We are motivated and encouraged not only by the similarities between CLEAN and the calculus to be presented here (the LNL term calculus), but also by the fact that other researchers looking at practical implementations of linear languages have come up with systems which include aspects of the LNL term calculus. For example, Lincoln and Mitchell's linear variant [LM92] of Fairbairn and Wray's 'three instruction machine' [FW87] divides memory into two spaces corresponding to linear and non-linear objects. Similarly, Wadler's 'active and passive' type system [Wad92] separates linear from non-linear types in an interesting way. It should also be mentioned that some of Wadler's earliest attempts to define a linear type system for a functional language flagged linear types as the exception, rather than the rule [Wad90], although he later reverted to 'belling the cat' by annotating non-linear types.

Jacobs [Jac93] has independently described how a sequent calculus inspired by CLEAN's uniqueness types may be interpreted using the linear categories of [BBHdP92] under some extra simplifying assumptions which are sufficient to make the whole Eilenberg-Moore category of coalgebras be cartesian closed. Jacobs's logic turns out to be essentially the same as LNL logic, and we will discuss his work further in Section 3.1.7.

The logic described here is, in a fairly strong sense, equivalent to ordinary ILL. How then could such a system possibly lead to a better linear programming language? The first answer is that we *refine* ILL: there are distinct LNL terms which correspond to the same LTC term. The second answer is that logical systems which are denotationally equivalent may still have very different *dynamic* (proof-theoretic) behaviours. However, such speculations should only be viewed as motivation for studying the logic. We do not yet have any formal results concerning, for example, the memory graphs of programs written in a language based on the LNL term calculus.

1.2.2 Logic

From a more logical point of view, there has recently been much interest in Girard's system LU [Gir93] and related systems in which the (multi)sets of formulae occuring in sequents are split into different zones. Formulae in some zones are treated classically, whilst those in other zones are treated linearly.

Intuitionistic logics inspired by LU have been proposed by Plotkin [Plo93] and Wadler [Wad93]. It is desirable to study the proof and model theory of such systems directly, rather than treating them as syntactic sugar for, for example, ordinary linear logic (if only to verify that it is possible to treat them as such syntactic sugar). The logic of this paper should turn out to be equivalent to a subsystem of LU, though there are some superficial differences of presentation such as the fact that LNL logic has no zones – the formulae themselves are either linear or conventional.[1]

[1] Though we, perhaps unwisely, abuse notation by writing a semicolon between formulae of different

1.2.3 Semantics

From the categorical perspective, it seems natural to explore the more symmetric situation where one starts from an SMCC and a CCC with (adjoint) functors between them, rather than an SMCC with sufficient extra structure to ensure the existence of such a CCC. This is particularly true in the light of the fact that the definition of a linear category in [BBHdP92] was arrived at mostly from the proof theory of linear logic, but also (and this was something of a 'hidden agenda') from a desire to have enough structure to be able to derive an appropriate CCC from the model.[2] It is also fair to say that the definition of a linear category is surprisingly complicated, so looking for simpler models, or simpler presentations of the same models, is a good idea. Pratt has also suggested that the comonad modelling ! might be less fundamental than the adjunctions from which it arises [Pra92].

1.3 Overview

The initial motivation for the present work comes from the categorical picture sketched in the previous section, and it is this which is explored first in Section 2. Once the definition has been made a little more precise, we shall show that such a situation leads to a comonad on the linear part of the model which automatically satisfies all the extra conditions required of a linear category, and thus gives a sound model of ILL including the ! operator. Furthermore, the converse holds – every linear category gives rise to such a pair of categories. Thus we have an alternative, simpler, definition of what constitutes a model for ILL. This can be seen as giving a purely category-theoretic reconstruction of !, in that a linear category (a model for ILL with !) is exactly what one obtains if one attempts directly to model an interpretation of IL in ILL without the !.

Another interesting feature of the model is that it gives rise to a strong monad on the CCC part. Thus one obtains a model not just of the lambda calculus, but of Moggi's computational lambda calculus [Mog89, Mog91]. This may shed further light on the 'monads versus comonads' debate which has occasionally arisen in programming language theory. As we shall see, however, not all strong monads arise in this way, so the connection is not quite as neat as one might hope.

Section 3 then looks at the logic and term calculus which are associated with our new notion of model. After a brief description of two unsatisfactory versions of the logic, we formulate a sequent calculus presentation which satisfies cut-elimination and then give an equivalent natural deduction system. This then gives, by the Curry-Howard correspondence, an interesting term calculus which combines the usual simply-typed lambda calculus with a linear lambda calculus. We also consider translations in both directions between this new term calculus and the linear calculus introduced in [BBHdP92].

This paper is fairly self-contained and assumes only a basic knowledge of category theory (up to, say, adjunctions), some familiarity with linear logic and an understanding of typed lambda calculus and the Curry-Howard correspondence. A nodding acquaintance with previous work on the linear term calculus and categorical models of ILL is also desirable.

---

kinds.

[2] This is not to say that there is anything in the model which is not justifiable in terms of the proof theory (given a proper proof-theoretic account of η-rules), but merely that, given that a translation of IL proofs into ILL proofs exists, any correct model for ILL must be able to reflect the translation semantically.

This is a preliminary report, and doubtless contains errors and omissions. It certainly leaves plenty of obvious questions unanswered. Comments, questions and suggestions for improvement are welcome.

2 The Categorical Picture

Our aim is to present a logic/terms/categories correspondence, similar to that between intuitionistic logic, simply-typed lambda calculus and cartesian closed categories, in which the categorical vertex of the triangle consists of (essentially) the following:

1. a cartesian closed category $(\mathcal{C}, 1, \times, \to)$;
2. a symmetric monoidal closed category $(\mathcal{L}, I, \otimes, \multimap)$ and
3. a pair of functors $G : \mathcal{L} \to \mathcal{C}$ and $F : \mathcal{C} \to \mathcal{L}$ between them with $F \dashv G$ (i.e. F is the left adjoint to G).

Intuitively, the requirement that the two functors be adjoint should be understood as saying that there is an interpretation of IL (the CCC) into ILL (the SMCC).

We will, however, need our categorical model to satisfy some extra conditions before we can have any hope of it modelling a logic or term calculus. It is necessary for the two functors and the unit and counit of the adjunction to behave well with respect to the monoidal structures of the two categories. The reason for this is that we have to handle contexts correctly, and the multicategorical structure implied by the comma in a context will be represented by the appropriate tensor product. The need for such extra structure also arises in, for example, models of the computational lambda calculus (the monad must be strong) and linear categories (the comonad must be symmetric monoidal). The extra conditions which we shall impose are not ad hoc, but are just what is required to ensure coherence.[3] Although the present paper gives all the definitions and proofs in an elementary form, it should be noted that morally we should regard everything as taking place in the 2-category of symmetric monoidal categories, in which context the extra monoidal conditions arise more naturally. Indeed, this view is an instance of a general principle concerning the categorical modelling of programming languages expressed by Moggi in [Mog91][4]

> when studying a complex language the 2-category $\mathcal{C}at$ of small categories, functors and natural transformations may not be adequate; however, one may replace $\mathcal{C}at$ with a different 2-category, whose objects capture better some *fundamental structure* of the language, while *less fundamental structure* can be modelled by *2-categorical concepts.*

Definition 1 *A* monoidal category *is a category $\mathcal{M}$ equipped with a bifunctor $\otimes : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$, and object I of $\mathcal{M}$, and natural isomorphisms*

$$\alpha_{X,Y,Z} : (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z)$$

$$l_X : I \otimes X \to X$$

$$r_X : X \otimes I \to X$$

---

[3] As has become traditional, however, we shall say very little about this important issue...

[4] Thanks to Ian Stark for bringing this quote to my attention.

which satisfy the following pair of coherence diagrams:

$$\begin{tikzcd}
((W \otimes X) \otimes Y) \otimes Z \arrow[rr, "\alpha \otimes 1"] \arrow[d, "\alpha"'] & & (W \otimes (X \otimes Y)) \otimes Z \arrow[dd, "\alpha"] \\
(W \otimes X) \otimes (Y \otimes Z) \arrow[d, "\alpha"'] & & \\
W \otimes (X \otimes (Y \otimes Z)) & & W \otimes ((X \otimes Y) \otimes Z) \arrow[ll, "1 \otimes \alpha"]
\end{tikzcd}$$

$$\begin{tikzcd}
(X \otimes I) \otimes Y \arrow[rr, "\alpha"] \arrow[dr, "r \otimes 1"'] & & X \otimes (I \otimes Y) \arrow[dl, "1 \otimes l"] \\
& X \otimes Y &
\end{tikzcd}$$

and for which $l_I = r_I$.

Definition 2 *A* symmetric monoidal category *(SMC) is a monoidal category* $(\mathcal{M}, \otimes, I, \alpha, l, r)$ *together with a natural transformation* $\sigma_{X,Y} : X \otimes Y \to Y \otimes X$ *satisfying the following three coherence conditions:*

$$\begin{tikzcd}
(X \otimes Y) \otimes Z \arrow[r, "\alpha"] \arrow[d, "\sigma \otimes 1"'] & X \otimes (Y \otimes Z) \arrow[r, "\sigma"] & (Y \otimes Z) \otimes X \arrow[d, "\alpha"] \\
(Y \otimes X) \otimes Z \arrow[r, "\alpha"'] & Y \otimes (X \otimes Z) \arrow[r, "1 \otimes \sigma"'] & Y \otimes (Z \otimes X)
\end{tikzcd}$$

$$\begin{tikzcd}
X \otimes Y \arrow[d, "\sigma"'] \arrow[dr, "1"] & \\
Y \otimes X \arrow[r, "\sigma"'] & X \otimes Y
\end{tikzcd}
\qquad
\begin{tikzcd}
I \otimes X \arrow[rr, "\sigma"] \arrow[dr, "l"'] & & X \otimes I \arrow[dl, "r"] \\
& X &
\end{tikzcd}$$

Note that every cartesian category (i.e. with finite products) is an SMC.

Definition 3 *A* symmetric monoidal closed category *(SMCC) is a symmetric monoidal category* $(\mathcal{M}, \otimes, I, \alpha, l, r, \sigma)$ *such for each* $B \in \mathcal{M}_0$ *the functor* $- \otimes B : \mathcal{M} \to \mathcal{M}$ *has a (specified) right adjoint. Thus there is for every* $A, C \in \mathcal{M}_0$ *an object* $(B \multimap C)$ *and a natural bijection*

$$\mathcal{M}(A \otimes B, C) \cong \mathcal{M}(A, B \multimap C)$$

Symmetric monoidal closed categories are also sometimes called *autonomous* categories.

Definition 4 *A* cartesian closed category *(CCC) is an SMCC for which the tensor product is cartesian.*

Whilst one might wish to consider functors between SMCs which preserve the structure on the nose or up to natural isomorphism, we shall take the class of functors between SMCs to be those preserving the structure up to a comparsion. We thus make the following definitions.

Definition 5 *Given monoidal categories* $(\mathcal{M}, \otimes, I, \alpha, l, r)$ *and* $(\mathcal{M}', \otimes', I', \alpha', l', r')$, *a* monoidal functor $F : \mathcal{M} \to \mathcal{M}'$ *is a functor from* $\mathcal{M}$ *to* $\mathcal{M}'$ *equipped with a map* $m_I : I' \to F(I)$ *in* $\mathcal{M}'$ *and a natural transformation* $m_{X,Y} : F(X) \otimes' F(Y) \to F(X \otimes Y)$ *which satisfy the following coherence conditions:*

$$\begin{tikzcd}
(F(X) \otimes' F(Y)) \otimes' F(Z) \arrow[r, "\alpha'"] \arrow[d, "m \otimes' 1"'] & F(X) \otimes' (F(Y) \otimes' F(Z)) \arrow[d, "1 \otimes' m"] \\
F(X \otimes Y) \otimes' F(Z) \arrow[d, "m"'] & F(X) \otimes' F(Y \otimes Z) \arrow[d, "m"] \\
F((X \otimes Y) \otimes Z) \arrow[r, "F(\alpha)"'] & F(X \otimes (Y \otimes Z))
\end{tikzcd}$$

$$\begin{tikzcd}
I' \otimes' F(X) \arrow[r, "l'"] \arrow[d, "m \otimes' 1"'] & F(X) \\
F(I) \otimes' F(X) \arrow[r, "m"'] & F(I \otimes X) \arrow[u, "F(l)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
F(X) \otimes' I' \arrow[r, "r'"] \arrow[d, "1 \otimes' m"'] & F(X) \\
F(X) \otimes' F(I) \arrow[r, "m"'] & F(X \otimes I) \arrow[u, "F(r)"']
\end{tikzcd}$$

Definition 6 *If* $\mathcal{M}$ *and* $\mathcal{M}'$ *above are* symmetric *monoidal, then* F *is a* symmetric monoidal functor *if it is monoidal and in addition satisfies the following coherence condition:*

$$\begin{tikzcd}
F(X) \otimes' F(Y) \arrow[r, "\sigma'"] \arrow[d, "m"'] & F(Y) \otimes' F(X) \arrow[d, "m"] \\
F(X \otimes Y) \arrow[r, "F(\sigma)"'] & F(Y \otimes X)
\end{tikzcd}$$

In the definition of a symmetric monoidal functor, one of the coherence diagrams for l and r is redundant, as it follows from the other and the diagram for σ. Note also that the identity functor is (symmetric) monoidal and that (symmetric) monoidal functors can be composed in an obvious way – if $(F, m) : \mathcal{M} \to \mathcal{M}'$ and $(G, n) : \mathcal{M}' \to \mathcal{M}''$ then their composite is given by the usual composition of functors together with the comparison natural transformation $p_{X,Y} : GFX \otimes'' GFY \to GF(X \otimes Y)$ where $p_{X,Y} = G(m_{X,Y}) \circ n_{FX,FY}$ (and similarly for the nullary version). It is then routine to check that (GF, p) is indeed a (symmetric) monoidal functor, and that (symmetric) monoidal categories and (symmetric) monoidal functors form a category.

Definition 7 *If (F, m) and (G, n) are monoidal functors from an MC $\mathcal{M}$ to an MC $\mathcal{M}'$, then a* monoidal natural transformation *from (F, m) to (G, n) is a natural transformation f_X from F to G which is compatible with the comparison maps in the sense that the following two diagrams commute:*

$$\begin{tikzcd}
F(X) \otimes' F(Y) \arrow[r, "m"] \arrow[d, "f_X \otimes' f_Y"'] & F(X \otimes Y) \arrow[d, "f_{X \otimes Y}"] \\
G(X) \otimes' G(Y) \arrow[r, "n"'] & G(X \otimes Y)
\end{tikzcd}$$

$$\begin{tikzcd}
F(I) \arrow[rr, "f_I"] & & G(I) \\
& I' \arrow[ul, "m"] \arrow[ur, "n"'] &
\end{tikzcd}$$

Definition 8 *If $\mathcal{M}$ and $\mathcal{M}'$ are (symmetric) monoidal categories then a* (symmetric) monoidal adjunction *between them is an ordinary adjunction in which both of the functors are (symmetric) monoidal functors and both the unit and the counit of the adjunction are monoidal natural transformations (with respect to the natural monoidal structure on the two composite functors, as defined above).*

Having made the basic definitions, we are now in a position to define more precisely the categorical model sketched earlier.

Definition 9 *A linear/non-linear model (LNL model) consists of*

1. *a cartesian closed category $(\mathcal{C}, 1, \times, \to)$;*
2. *a symmetric monoidal closed category $(\mathcal{L}, I, \otimes, \multimap)$ and*
3. *a pair of symmetric monoidal functors $(G, n) : \mathcal{L} \to \mathcal{C}$ and $(F, m) : \mathcal{C} \to \mathcal{L}$ between them which form a symmetric monoidal adjunction with $F \dashv G$.*

We shall usually use A, B, C to range over objects of $\mathcal{L}$ and X, Y, Z for objects of $\mathcal{C}$. Spelling the definition out in a bit more detail, this means that we have a pair of natural transformations $\eta : 1_{\mathcal{C}} \dot{\rightarrow} GF$ and $\varepsilon : FG \dot{\rightarrow} 1_{\mathcal{L}}$ which satisfy the triangle laws for an adjunction:

$$
\begin{tikzcd}
GA \arrow[r, "\eta_{GA}"] \arrow[dr, "1_{GA}"'] & GFGA \arrow[d, "G\varepsilon_A"] \\
& GA
\end{tikzcd}
\qquad
\begin{tikzcd}
FX \arrow[r, "F\eta_X"] \arrow[dr, "1_{FX}"'] & FGFX \arrow[d, "\varepsilon_{FX}"] \\
& FX
\end{tikzcd}
$$

That η and ε are *monoidal* natural transformations means that the following four diagrams commute:

$$
\begin{tikzcd}
X \times Y \arrow[r, "\eta_X \times \eta_Y"] \arrow[d, "\eta_{X \times Y}"'] & GFX \times GFY \arrow[d, "n_{FX,FY}"] \\
GF(X \times Y) & G(FX \otimes FY) \arrow[l, "G(m_{X,Y})"]
\end{tikzcd}
\qquad
\begin{tikzcd}
FGA \otimes FGB \arrow[r, "m_{GA,GB}"] \arrow[d, "\varepsilon_A \otimes \varepsilon_B"'] & F(GA \times GB) \arrow[d, "F(n_{A,B})"] \\
A \otimes B & FG(A \otimes B) \arrow[l, "\varepsilon_{A \otimes B}"]
\end{tikzcd}
$$

$$
\begin{tikzcd}
1 \arrow[rr, "\eta_1"] & & GF(1) \\
& 1 \arrow[ul, "1"] \arrow[ur, "G(m) \circ n"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
FG(I) \arrow[rr, "\varepsilon_I"] & & I \\
& I \arrow[ul, "F(n) \circ m"] \arrow[ur, "1"'] &
\end{tikzcd}
$$

2.1 An Isomorphism

An important consequence of the definition of an LNL model is that as well as the natural transformations

$$m_{X,Y} : FX \otimes FY \to F(X \times Y)$$
$$n_{A,B} : GA \times GB \to G(A \otimes B)$$

and their nullary versions, the maps

$$m : I \to F1$$

$$n : 1 \to GI$$

we have a family of maps

$$p_{X,Y} : F(X \times Y) \to FX \otimes FY$$

given by the transpose of $n_{FX,FY} \circ \eta_X \times \eta_Y$:

$$
\begin{tikzcd}
F(X \times Y) \arrow[r, "F(\eta_X \times \eta_Y)"] & F(GFX \times GFY) \arrow[r, "F(n_{FX,FY})"] & FG(FX \otimes FY) \arrow[d, "\varepsilon_{FX \otimes FY}"] \\
& & FX \otimes FY
\end{tikzcd}
$$

and a map $p : F1 \to I$ given by

$$F1 \xrightarrow{Fn} FGI \xrightarrow{\varepsilon_I} I$$

It is straightforward to check that the $p_{X,Y}$ are the components of a natural transformation. We do not, however, get a collection of maps in the other possible direction, *viz.* from $G(A \otimes B)$ to $GA \times GB$.

Proposition 1 *In an LNL model (in fact for any monoidal adjunction), the maps $m_{X,Y}$ are the components of a natural isomorphism with inverses $p_{X,Y}$ and, furthermore, the map m is an isomorphism with inverse p:*

$$F(X) \otimes F(Y) \;\cong\; F(X \times Y)$$
$$I \;\cong\; F(1)$$

Proof. We shall just prove the first of the isomorphisms above as the second is very similar. Firstly, we need to show that $m_{X,Y} \circ p_{X,Y} = 1_{F(X \times Y)}$:

$$\begin{tikzcd}
F(GFX \times GFY) \arrow[rr, "F(n_{FX,FY})"] & & FG(FX \otimes FY) \arrow[rr, "\varepsilon_{FX \otimes FY}"] \arrow[d, "FG(m_{X,Y})"] & & FX \otimes FY \arrow[dd, "m_{X,Y}"] \\
& & FGF(X \times Y) \arrow[drr, "\varepsilon_{F(X \times Y)}"'] & & \\
F(X \times Y) \arrow[uu, "F(\eta_X \times \eta_Y)"] \arrow[urr, "F(\eta_{X \times Y})"] \arrow[rrrr, "1"'] & & & & F(X \times Y)
\end{tikzcd}$$

The square on the right commutes by naturality of ε, whilst that on the left commutes as it is F applied to the earlier square which says that η is monoidal. The triangle on the bottom is one of the triangles for an adjunction and so the path up the left hand side, along the top and down the right hand side is equal to that along the bottom, as required.

Secondly, we claim that $p_{X,Y} \circ m_{X,Y} = 1_{FX \otimes FY}$, which follows from a similar diagram:

$$\begin{tikzcd}
F(X \times Y) \arrow[rr, "F(\eta_X \times \eta_Y)"] & & F(GFX \times GFY) \arrow[rr, "F(n_{FX,FY})"] & & FG(FX \otimes FY) \arrow[dd, "\varepsilon_{FX \otimes FY}"] \\
& & FGFX \otimes FGFY \arrow[u, "m_{GFX,GFY}"] \arrow[drr, "\varepsilon_{FX} \otimes \varepsilon_{FY}"'] & & \\
FX \otimes FY \arrow[uu, "m_{X,Y}"] \arrow[urr, "F(\eta_X) \otimes F(\eta_Y)"] \arrow[rrrr, "1 \otimes 1"'] & & & & FX \otimes FY
\end{tikzcd}$$

The square on the left commutes by naturality of m and that on the right because ε is monoidal. The triangle on the bottom commutes by two applications of one of the triangle laws for an adjunction and so the outer path is equal to that along the bottom, which is trivially equal to the identity. $\square$

So F preserves the monoidal structure up to an isomorphism rather than merely up to a comparison. That is to say, F is a *strong* functor. This has a converse – given a strong functor with an adjoint, the adjoint (in fact the whole adjunction) has a unique monoidal structure. In our case, this means that instead of taking n as part of the definition of an LNL model and deriving p, we could equally well have started with p and defined $n_{A,B}$ to be the composite

$$G(\varepsilon_A \otimes \varepsilon_B) \circ G(p_{GA,GB}) \circ \eta_{GA \times GB}$$

This fact will crop up again in Section 3.1.2.

There is, of course, a lot more interesting structure in an LNL model. To begin with, the adjunction induces a comonad on $\mathcal{L}$ and a monad on $\mathcal{C}$. We discuss each of these below. Given one of the categories and the appropriate monad (triple) or comonad (cotriple), the other category and the adjunction arise as a *resolution* of the triple (cotriple). In contrast with some other proposed models of intuitionistic linear logic, we do not assume that this is initial or terminal in the category of all resolutions.

2.2 The Comonad and Comparison with Linear Categories

The comonad on $\mathcal{L}$ is $(FG, \varepsilon : FG \to 1, \delta : FG \to FGFG)$ where ε is the counit of the adjunction and δ is the natural transformation with components $\delta_A : FG(A) \to FGFG(A)$ given by $\delta_A = F(\eta_{G(A)})$. Writing ! for FG, we obtain the usual comonad diagrams:

$$\begin{tikzcd}
!A & !!A \arrow[l, "\varepsilon_{!A}"'] \arrow[r, "!(\varepsilon_A)"] & !A \\
& !A \arrow[ul, "1"] \arrow[u, "\delta_A"'] \arrow[ur, "1"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
!!!A & !!A \arrow[l, "!(\delta_A)"'] \\
!!A \arrow[u, "\delta_{!A}"] & !A \arrow[l, "\delta_A"] \arrow[u, "\delta_A"']
\end{tikzcd}$$

Lemma 2 *The comonad $(!, \varepsilon, \delta)$ is symmetric monoidal, i.e. ! is a symmetric monoidal functor and ε and δ are monoidal natural transformations.*

Proof. Clearly ! is a symmetric monoidal functor. The monoidal structure is given by a natural transformation q with components $q_{A,B} : !A \otimes !B \to !(A \otimes B)$ and a map $q : I \to !I$ whose definitions are

$$q_{A,B} = F(n_{A,B}) \circ m_{GA,GB}$$

$$q = F(n) \circ m$$

That ε is monoidal is part of the definition of an LNL model. The case of δ requires some easy checking. $\square$

In [BBHdP92], we defined a model of the multiplicative/exponential fragment of intuitionistic linear logic as follows:

Definition 10 *A* linear category *is specified by the following data:*

1. *A symmetric monoidal closed category* $(\mathcal{L}, \otimes, I, \multimap)$.
2. *A symmetric monoidal comonad* $(!, \varepsilon, \delta, q)$ *on* $\mathcal{L}$.
3. *Monoidal natural transformations*[5] *with components*

$$e_A : !A \to I$$

$$d_A : !A \to !A \otimes !A$$

such that

(a) *each* $(!A, e_A, d_A)$ *is a commutative comonoid,*

(b) e_A *and* d_A *are coalgebra maps*[6], *and*

(c) *all coalgebra maps between free coalgebras preserve the comonoid structure.*

2.2.1 LNL model Implies Linear Category

Now, any LNL model includes, by definition, part 1 of Definition 10, and we have just seen (Lemma 2) that it also satisfies part 2. Furthermore, there are plausible candidates for e_A and d_A:

$$e_A \stackrel{\text{def}}{=} p \circ F(*_{GA})$$

where $*_{GA}$ is the unique map from GA to the terminal object 1 of $\mathcal{C}$, and

$$d_A \stackrel{\text{def}}{=} p_{GA,GA} \circ F(\Delta_{GA})$$

where Δ_{GA} is the diagonal map from GA to $GA \times GA$ in $\mathcal{C}$. We now embark on showing that these satisfy all the conditions which ensure that we have a linear category. The reader who is prepared to take this on trust may prefer to skip straight to Corollary 8.

Lemma 3 e_A *and* d_A *as defined above are the components of natural transformations.*

Proof. This is obvious as a result of general facts about composition of, and application of functors to, natural transformations. For example, we have to check that for any $f : A \to B$,

$$\begin{tikzcd} !A \arrow[r, "d_A"] \arrow[d, "!f"'] & !A \otimes !A \arrow[d, "!f \otimes !f"] \\ !B \arrow[r, "d_B"'] & !B \otimes !B \end{tikzcd}$$

---

[5]Note that this only makes sense because the functors $A \mapsto I$ and $A \mapsto !A \otimes !A$ are themselves (symmetric) monoidal, but this is easily seen to be true. See the proof of Lemma 4 below for the details.

[6]Exactly what this means is spelled out in the proof of Lemma 6.

which expands to give

$$\begin{tikzcd}
FGA \arrow[r, "F(\Delta_{GA})"] \arrow[d, "FGf"'] & F(GA \times GA) \arrow[r, "p_{GA,GA}"] \arrow[d, "F(Gf \times Gf)"'] & FGA \otimes FGA \arrow[d, "FGf \otimes FGf"] \\
FGB \arrow[r, "F(\Delta_{GB})"'] & F(GB \times GB) \arrow[r, "p_{GB,GB}"'] & FGB \otimes FGB
\end{tikzcd}$$

The left-hand square commutes by F applied to naturality of Δ and the other by naturality of p. Naturality of e is similar. $\square$

Lemma 4 *e and d are* monoidal *natural transformations.*

Proof. We first have to make explicit the symmetric monoidal structure on the functors $K : A \mapsto I$ and $D : A \mapsto !A \otimes !A$. For K we require a natural transformation $s_{A,B} : K(A) \otimes K(B) \to K(A \otimes B)$ and a map $s : I \to K(I)$. Clearly we can take $s_{A,B} = l_I$ and $s = 1_I$ and then verification of the coherence conditions showing that (K, s) is symmetric monoidal is trivial.

For D we need a natural transformation with components

$$t_{A,B} : (!A \otimes !A) \otimes (!B \otimes !B) \to !(A \otimes B) \otimes !(A \otimes B)$$

together with a map $t : I \to !I \otimes !I$. We take $t_{A,B}$ to be the composite

$$(!A \otimes !A) \otimes (!B \otimes !B) \xrightarrow{iso} (!A \otimes !B) \otimes (!A \otimes !B) \xrightarrow{q_{A,B} \otimes q_{A,B}} !(A \otimes B) \otimes !(A \otimes B)$$

where *iso* represents a combination of natural isomorphisms[7] and t to be $(q \otimes q) \circ l^{-1}$. That $t_{A,B}$ is natural and that the coherence conditions making D a symmetric monoidal functor are satisfied is trivial.

The lemma is thus the statement that the following four diagrams commute:

$$\begin{tikzcd}
!A \otimes !B \arrow[r, "q_{A,B}"] \arrow[d, "e_A \otimes e_B"'] & !(A \otimes B) \arrow[d, "e_{A \otimes B}"] \\
I \otimes I \arrow[r, "l_I"'] & I
\end{tikzcd}
\qquad
\begin{tikzcd}
!I \arrow[rr, "e_I"] & & I \\
& I \arrow[ul, "q"] \arrow[ur, "1"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
!A \otimes !B \arrow[r, "q_{A,B}"] \arrow[d, "d_A \otimes d_B"'] & !(A \otimes B) \arrow[d, "d_{A \otimes B}"] \\
(!A \otimes !A) \otimes (!B \otimes !B) \arrow[r, "t_{A,B}"'] & !(A \otimes B) \otimes !(A \otimes B)
\end{tikzcd}$$

[7] There is actually a choice here, but it doesn't matter which *iso* we pick.

$$\begin{tikzcd}
!I \arrow[rr, "d_I"] & & !I\otimes !I \\
& I \arrow[ul, "q"] \arrow[ur, "t"'] &
\end{tikzcd}$$

We will verify a couple of these. Firstly, the triangle for e expands and fills in as follows:

$$\begin{tikzcd}
I \arrow[d, "m"'] \arrow[ddrr, "1"] & & \\
F1 \arrow[d, "F(n)"'] \arrow[dr, "F(1)"] & & \\
FGI \arrow[r, "F(*_{GI})"'] & F1 \arrow[r, "p"'] & I
\end{tikzcd}$$

The triangle commutes because it is F applied to a triangle which commutes by the uniqueness of maps into 1. The square commutes because F is a functor and $m = p^{-1}$ (Proposition 1).

The square for e expands and fills in like this (omitting subscripts on natural transformations):

$$\begin{tikzcd}
FGA\otimes FGB \arrow[r, "F(*)\otimes F(*)"] \arrow[dd, "m"'] & F1\otimes F1 \arrow[rr, "p\otimes p"] \arrow[d, "1"'] & & I\otimes I \arrow[dl, "1\otimes m"'] \arrow[dd, "l"] \\
& F1\otimes F1 \arrow[d, "m"'] & I\otimes F1 \arrow[l, "m\otimes 1"'] \arrow[ddl, "l"] & \\
F(GA\times GB) \arrow[r, "F(*\times *)"] \arrow[d, "F(n)"'] & F(1\times 1) \arrow[d, "F(l)"'] & & I \arrow[dll, "m"'] \arrow[d, "1"] \\
FG(A\otimes B) \arrow[r, "F(*)"'] & F1 \arrow[rr, "p"'] & & I
\end{tikzcd}$$

The top left square commutes by naturality of m, and the bottom left one by the fact that 1 is terminal. The triangle at the bottom of the right hand side and the quadrilateral at the top of the right hand side both commute because $p = m^{-1}$. The triangle in the middle at the left of the right hand side of the diagram commutes because F is a monoidal functor, and the remaining quadrilateral by naturality of l.

Filling in the two diagrams for d is left as an exercise in diagram chasing for the reader. They are rather larger, but fundamentally similar to those for e. □

Lemma 5 *For any A, $(!A, e_A, d_A)$ is a commutative comonoid.*

Proof. This requires the following three diagrams to commute:

$$
\begin{tikzcd}
& !A \arrow[dl, "r^{-1}"'] \arrow[d, "d_A"] \\
!A \otimes I & !A\otimes !A \arrow[l, "1_{!A} \otimes e_A"]
\end{tikzcd}
\qquad
\begin{tikzcd}
!A\otimes !A \arrow[rr, "\sigma"] & & !A\otimes !A \\
& !A \arrow[ul, "d_A"] \arrow[ur, "d_A"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
!A \arrow[rr, "d_A"] \arrow[d, "d_A"'] & & !A\otimes !A \arrow[d, "1 \otimes d_A"] \\
!A\otimes !A \arrow[r, "d_A \otimes 1"'] & (!A\otimes !A)\otimes !A \arrow[r, "\alpha"'] & !A \otimes (!A\otimes !A)
\end{tikzcd}
$$

These are all fairly straightforward. For example, the first diagram can be expanded and filled in as follows:

$$
\begin{tikzcd}
FGA \arrow[rrr, "1"] \arrow[d, "F(\Delta)"'] & & & FGA \\
F(GA \times GA) \arrow[r, "1"] \arrow[dd, "p"'] & F(GA \times GA) \arrow[r, "F(1 \times *)"] & F(GA \times 1) \arrow[ur, "F(\pi_1)"] & \\
& & FGA \otimes F1 \arrow[u, "m"'] & \\
FGA \otimes FGA \arrow[uur, "m"'] \arrow[rr, "1 \otimes F(*)"'] & & FGA \otimes F1 \arrow[u, "1"'] \arrow[r, "1 \otimes p"'] & FGA \otimes I \arrow[ul, "1 \otimes m"'] \arrow[uuu, "r"']
\end{tikzcd}
$$

Taking the regions clockwise from the top, the first is a consequence of obvious facts about cartesian products (in fact, that they give a commutative comonoid structure in $\mathcal{C}$). The second commutes because F is a monoidal functor. The third and fifth because $p = m^{-1}$, and the fourth because m is a natural transformation.

The other two diagrams commute by similar reasoning. □

Lemma 6 *e_A and d_A are coalgebra morphisms (with respect to the canonical coalgebra structures on I,$!A$ and $!A\otimes!A$, see Section 2.2.2).*

Proof. We need the following pair of diagrams:

$$
\begin{tikzcd}
!A \arrow[r, "e_A"] \arrow[d, "\delta_A"'] & I \arrow[d, "q"] \\
!!A \arrow[r, "!e_A"'] & !I
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[rr, "\delta_A"] \arrow[d, "d_A"'] & & !!A \arrow[d, "!d_A"] \\
!A \otimes !A \arrow[r, "\delta_A \otimes \delta_A"'] & !!A \otimes !!A \arrow[r, "q_{!A,!A}"'] & !(!A \otimes !A)
\end{tikzcd}
$$

The second of these can be dealt with like this:

$$
\begin{tikzcd}
FGA \arrow[r, "F(\Delta)"] \arrow[dddd, "F(\eta)"'] & F(GA \times GA) \arrow[rr, "p"] \arrow[dddd, "F(\eta)"'] \arrow[dr, "1"'] & & FGA \otimes FGA \arrow[dl, "m"] \arrow[d, "F(\eta) \otimes F(\eta)"] \\
& & F(GA \times GA) \arrow[dr, "F(\eta \times \eta)"'] & FGFGA \otimes FGFGA \arrow[d, "m"] \\
& & & F(GFGA \times GFGA) \arrow[d, "F(n)"] \\
& & & FG(FGA \otimes FGA) \arrow[d, "1"] \arrow[dll, "FG(m)"'] \\
FGFGA \arrow[r, "FGF(\Delta)"'] & FGF(GA \times GA) \arrow[rr, "FG(p)"'] & & FG(FGA \otimes FGA)
\end{tikzcd}
$$

The large square on the left commutes by naturality of η. The two triangles both commute because $p = m^{-1}$. The region on the far right commutes by naturality of m, and that in the middle because η is monoidal.

The other diagram is similar. □

Lemma 7 *Any coalgebra map $f : (!A, \delta_A) \to (!B, \delta_B)$ between free coalgebras preserves the comonoid structure given by e and d.*

Proof. This means that for any such f, the following pair of diagrams commute:

$$
\begin{tikzcd}
!A \arrow[rr, "f"] \arrow[dr, "e_A"'] & & !B \arrow[dl, "e_B"] \\
& I &
\end{tikzcd}
\qquad
\begin{tikzcd}
!A \arrow[r, "d_A"] \arrow[d, "f"'] & !A \otimes !B \arrow[d, "f \otimes f"] \\
!B \arrow[r, "d_B"'] & !B \otimes !B
\end{tikzcd}
$$

The second of these can be expanded out like this:

$$
\begin{tikzcd}
!B \arrow[dr, "\delta"] \arrow[dddd, "d"'] & & & !A \arrow[lll, "f"'] \arrow[dl, "\delta"'] \arrow[dddd, "d"] \\
& !!B \arrow[d, "d"'] & !!A \arrow[l, "!f"'] \arrow[d, "d"] & \\
& !!B \otimes !!B \arrow[d, "q"'] & !!A \otimes !!A \arrow[l, "!f \otimes !f"'] \arrow[d, "q"] & \\
& !(!B \otimes !B) \arrow[dl, "\varepsilon"'] & !(!A \otimes !A) \arrow[l, "!(f \otimes f)"'] \arrow[dr, "\varepsilon"] & \\
!B \otimes !B & & & !A \otimes !A \arrow[lll, "f \otimes f"]
\end{tikzcd}
$$

Taking the regions in the middle from the top, the first commutes by assumption (that f is a coalgebra morphism) and the second, third and fourth by naturality of d,q and ε respectively. The remaining two regions are both easily seen to commute, since they expand as follows:

$$
\begin{tikzcd}
FGFGA \arrow[rr, "F(\Delta)"] & & F(GFGA \times GFGA) \arrow[dd, "1"'] \arrow[dr, "p"] & \\
FGA \arrow[u, "F(\eta)"] \arrow[d, "F(\Delta)"'] & & & FGFGA \otimes FGFGA \arrow[dl, "m"] \\
F(GA \times GA) \arrow[uurr, "F(\eta \times \eta)"] \arrow[d, "p"'] & & F(GFGA \times GFGA) \arrow[d, "F(n)"] & \\
FGA \otimes FGA & & FG(FGA \otimes FGA) \arrow[ll, "\varepsilon"] &
\end{tikzcd}
$$

The triangle on the right commutes because $p = m^{-1}$ and that on the top left by naturality of Δ. The middle region commutes simply because it is the definition of p. $\square$

Taking the previous lemmas together, we have shown

Corollary 8 *Any LNL model is a linear category.* □

2.2.2 Linear Category Implies LNL model

In this section we sketch the proof of the converse to Corollary 8. Whilst this is largely a matter of recalling results which were proved in [BBHdP92] and [Bie94a], by doing this carefully we obtain a slightly better understanding of the situation.

Assume that $\mathcal{L}$ is a linear category as in Definition 10. We need to show that this gives rise to a CCC $\mathcal{C}$ and a symmetric monoidal adjunction between $\mathcal{L}$ and $\mathcal{C}$ as in Definition 9. Recall that the comonad on $\mathcal{L}$ gives rise to two categories of algebras:

- The Eilenberg-Moore category $\mathcal{L}^!$. This has as objects all the !-coalgebras $(A, h_A : A \to !A)$ and as morphisms all the coalgebra morphisms.
- The (co-)Kleisli category $\mathcal{L}_!$. This is the full subcategory of $\mathcal{L}^!$ which has as objects all the *free* !-coalgebras $(!A, \delta_A : !A \to !!A)$. (This is not quite the most common definition of $\mathcal{L}_!$, but the two definitions are equivalent.)

Each of these categories comes with a pair of adjoint functors $F \dashv G$ where $G : A \mapsto (!A, \delta_A)$ and $F : (A, h_A) \mapsto A$, thus (note that we are overloading F and G):

$$\begin{tikzcd}
 & \mathcal{L}^! \arrow[dl, shift right, "F"'] \\
\mathcal{L} \arrow[ur, shift right, "G"'] \arrow[dr, shift left, "G"] & \\
 & \mathcal{L}_! \arrow[uu, "i"'] \arrow[ul, shift left, "F"]
\end{tikzcd}$$

where $i : \mathcal{L}_! \hookrightarrow \mathcal{L}^!$ is the inclusion functor.

Lemma 9 *If $\mathcal{L}$ is a linear category then $\mathcal{L}^!$ has finite products.*

Proof. The terminal object is $(I, q : I \to !I)$. The unique map from (A, h_A) to the terminal object is $e_A \circ h_A$. The product of (A, h_A) and (B, h_B) is $(A \otimes B, q_{A,B} \circ (h_A \otimes h_B))$. Projections and diagonals are given by the following composites

$$\begin{aligned}
\pi_1 &= A \otimes B \xrightarrow{1 \otimes h_B} A \otimes !B \xrightarrow{1 \otimes e_B} A \otimes I \xrightarrow{r} A \\
\Delta_A &= A \xrightarrow{h_A} !A \xrightarrow{d_A} !A \otimes !A \xrightarrow{\varepsilon_A \otimes \varepsilon_A} A \otimes A
\end{aligned}$$

These are easily checked to satisfy the relevant conditions. □

In general, there is no reason why the Eilenberg-Moore category should be cartesian closed, since there is no reason why it should have an internal hom for arbitrary pairs of coalgebras. There *are* extra conditions which are sufficient to ensure that this does happen, such as requiring that $\mathcal{L}^!$ have equalisers of coreflexive pairs [Bie94a] or simply all equalisers [Jac93]. Although there are non-trivial examples in which such conditions hold, we shall not consider them further since we can find an appropriate CCC without them.

Lemma 10 *In $\mathcal{L}^!$, all the free coalgebras are exponentiable. That is, there is an internal hom into any* free *coalgebra $(!B, \delta_B)$. Furthermore, the internal hom is itself a free coalgebra.*

Proof. We claim that

$$[(A, h_A), (!B, \delta_B)] \stackrel{\mathrm{def}}{=} (!(A \multimap B), \delta_{A \multimap B})$$

is an internal hom. This follows from the adjunction between F and G and from the closed structure on $\mathcal{L}$, since for any coalgebra (C, h_C) there are bijections:

$$\frac{\dfrac{\dfrac{\mathcal{L}^!((C, h_C),\ (!(A \multimap B), \delta_{A \multimap B}))}{\mathcal{L}(C,\ A \multimap B)}}{\mathcal{L}(C \otimes A,\ B)}}{\mathcal{L}^!((C \otimes A, h_{C \otimes A}),\ (!B, \delta_B))}$$

for any $h_{C \otimes A}$ giving $C \otimes A$ a coalgebra structure, in particular that arising from the product on $\mathcal{L}^!$. So an instance of the last line is

$$\mathcal{L}^!((C, h_C) \times (A, h_A),\ (!B, \delta_B))$$

as required. □

Now, notice that in any cartesian category, if an object X is exponentiable then so is $[Y, X]$ for any Y, since we can take $[Z, [Y, X]]$ to be $[Z \times Y, X]$. Furthermore, the product of two exponentiable objects X and Y is exponentiable since we can take $[Z, X \times Y]$ to be $[Z, X] \times [Z, Y]$. Taking this together with the previous lemma, we have:

Lemma 11 *The full subcategory $\mathcal{E}xp(\mathcal{L}^!)$ of the Eilenberg-Moore category having as objects the exponentiable coalgebras is cartesian closed and contains the Kleisli category $\mathcal{L}_!$.* □

Note that the Kleisli category is not, in general, cartesian closed, since the product of two free coalgebras is not necessarily free. We shall consider a case in which this does happen in Section 2.2.3. In the general case, we do have the following, however:

Lemma 12 *The full subcategory $\mathcal{L}_!^*$ of $\mathcal{E}xp(\mathcal{L}^!)$ consisting of finite products of free coalgebras is cartesian closed.* □

The situation can be pictured thus:

$$
\begin{tikzcd}
 & \mathcal{L}^{!} \arrow[ddl, shift left, "G"] \\
 & \mathcal{E}xp(\mathcal{L}^{!}) \arrow[u, "i"'] \\
\mathcal{L} \arrow[uur, shift left, "F"] \arrow[ddr, shift left, "G"] & \\
 & \mathcal{L}^{*}_{!} \arrow[uu, "i"'] \\
 & \mathcal{L}_{!} \arrow[u, "i"'] \arrow[uul, shift left, "F"]
\end{tikzcd}
$$

We claim that either of these two CCCs will give rise to an LNL model.[8] In what follows we let $\mathcal{C}$ stand for either $\mathcal{E}xp(\mathcal{L}^{!})$ or $\mathcal{L}^{*}_{!}$.

It is easy to see that F and G are still adjoint functors when regarded as going between $\mathcal{C}$ and $\mathcal{L}$, so it merely remains to show that this is a symmetric monoidal adjunction.

Lemma 13 *The forgetful functor $F : \mathcal{C} \to \mathcal{L}$ is symmetric monoidal.*

Proof. We need a natural transformation with components $m_{X,Y} : F(X) \otimes F(Y) \to F(X \times Y)$ and a map $m : I \to F(1)$ satisfying certain conditions. But if X and Y are (A, h_A) and (B, h_B) respectively, this amounts to $m_{X,Y} : A \otimes B \to A \otimes B$ and $m : I \to I$. Taking $m_{X,Y} = 1_{A\otimes B}$ and $m = 1_I$ is then easily seen to work. □

Lemma 14 *The free functor $G : \mathcal{L} \to \mathcal{C}$ is symmetric monoidal.*

Proof. We need a natural transformation with components $n_{A,B} : GA \times GB \to G(A \otimes B)$ and a map $n : 1 \to GI$ satisfying some conditions. Spelling this out a bit, $n_{A,B}$ is a coalgebra map:

$$n_{A,B} : (!A \otimes !B, q_{A,B} \circ (\delta_A \otimes \delta_B)) \to (!(A \otimes B), \delta_{A\otimes B})$$

Now the symmetric monoidal structure on ! gives a map between the underlying objects of these two coalgebras $q_{A,B} : !A \otimes !B \to !(A \otimes B)$, and that this is a coalgebra map follows immediately from the fact that δ is a monoidal natural transformation. The nullary case is similar. That this definition of n satisfies the conditions making (G, n) symmetric monoidal is then immediate from the fact that $(!, q)$ is symmetric monoidal. □

Lemma 15 *The unit of the adjunction $\eta : 1_{\mathcal{C}} \dot{\to} GF$ is a monoidal natural transformation.*

[8]It may well be that there is a sensible definition of a category of 'linear resolutions' in which $\mathcal{L}^{*}_{!}$ is initial and $\mathcal{E}xp(\mathcal{L}^{!})$ is terminal, but this idea has not yet been followed up.

Proof. This is also straighforward, though we have not so far made explicit what the definition of $\eta_{(A,h_A)} : (A, h_A) \to (!A, \delta_A)$ is. The answer is that it is just h_A, which is readily seen to be a coalgebra morphism by the definition of coalgebra and to be natural by the definition of coalgebra map. That h_A is monoidal is then completely trivial from the definition of the product of coalgebras. □

Lemma 16 *The counit $\varepsilon : FG \dot{\to} 1_{\mathcal{L}}$ of the adjunction is a monoidal natural transformation.*

Proof. By assumption. □

Taking the preceding results together, we have:

Corollary 17 *Any linear category gives rise to an LNL model, though it is not in general unique.* □

Of course, given a linear category $\mathcal{L}$, there may be many choices of $\mathcal{C}$ which lead to an LNL model other than the two given above. One could start with an arbitrary LNL model comprising some $\mathcal{L}$ and $\mathcal{C}$ together with the associated data, and then construct the linear category $(\mathcal{L}, !)$. In general, there is then no reason why $\mathcal{C}$ should be equivalent to either of $\mathcal{L}^*_!$ or $\mathcal{E}xp(\mathcal{L}^!)$, although in particular cases the distinction between some or all of these CCCs can collapse.

2.2.3 Additives and the Seely Isomorphisms

So far, we have concentrated on the relationship between the multiplicative $\otimes, \multimap$ fragment of ILL and the $\times, \to$ fragment of IL. We now consider briefly what happens when an LNL model (or, equivalently, a linear category) also has the extra structure required to model the additive linear connectives $\&, \oplus$ and the non-linear sum $+$.

The simplest case is when the SMCC part $\mathcal{L}$ of an LNL model also has finite products, modelling the additive connective 'with' (&). The functor G preserves limits because it is a right adjoint, and in particular

$$\begin{array}{rcl} G(A\&B) & \cong & GA \times GB \\ G1 & \cong & 1 \end{array}$$

(note that we use 1 for the terminal object in both $\mathcal{L}$ and $\mathcal{C}$). Taking this together with Proposition 1, we obtain the following natural isomorphisms:

$$\begin{array}{rcl} !A \otimes !B & \cong & !(A\&B) \\ I & \cong & !1 \end{array}$$

These isomorphisms were central to Seely's proposed model of ILL [See80], which also proposed interpreting IL in the Kleisli category. See [Bie94a] or [Bie94b] for a critique of Seely's semantics; here we shall merely show that a linear category with products does indeed have a Kleisli category which is cartesian closed.

The isomorphisms $\phi_{A,B} : !(A\&B) \to !A \otimes !B$ and $\phi : !1 \to I$ can be given explicit definitions in terms of the data determining a linear category thus:

$$\begin{array}{rcl} \phi_{A,B} & \stackrel{\text{def}}{=} & !(A\&B) \xrightarrow{d_{A\&B}} !(A\&B) \otimes !(A\&B) \xrightarrow{!\pi_1 \otimes !\pi_2} !A \otimes !B \\ \phi & \stackrel{\text{def}}{=} & !1 \xrightarrow{e_1} I \end{array}$$

(Explicit definitions of the inverses are left as an exercise for the reader.)

Lemma 18 *If a linear category has products then the product of two free !-coalgebras is a free coalgebra.*

Proof. This amounts to checking the following diagram:

$$\begin{tikzcd}
!(A\&B) \arrow[r, "\phi_{A,B}"] \arrow[dd, "\delta_{A\&B}"'] & !A\otimes !B \arrow[d, "\delta_A \otimes \delta_B"] \\
& !!A\otimes !!B \arrow[d, "q_{!A,!B}"] \\
!!(A\&B) \arrow[r, "!(\phi_{A,B})"'] & !(!A\otimes !B)
\end{tikzcd}$$

which is an easy consequence of naturality and the fact that d is a coalgebra morphism. □

Corollary 19 *If a linear category has products then the Kleisli category $\mathcal{L}_!$ is cartesian closed.*

Proof. Lemma 18 says that $\mathcal{L}_!$ coincides with $\mathcal{L}_!^*$, which is cartesian closed by Lemma 12. □

Products were relatively easy to deal with – the correspondence between linear categories and LNL models extends trivially to one between linear categories with finite products and LNL models with products on the SMCC part. Coproducts are slightly more problematic. Whilst the appropriate extension of an LNL model seems obvious (just require both $\mathcal{L}$ and $\mathcal{C}$ to have finite coproducts), this does not correspond quite as simply as one might hope to linear categories with coproducts.

The difficulty is that, whilst an LNL model with coproducts certainly gives rise to a linear category with coproducts, the converse does not appear necessarily to be true. Assume $\mathcal{L}$ is a linear category with finite coproducts, then $\mathcal{L}^!$ also has finite coproducts as we can define the coproduct of (A, h_A) and (B, h_B) to be

$$(A + B,\ [!\mathrm{inl} \circ h_A, !\mathrm{inr} \circ h_B])$$

and this is easily checked to satisfy the appropriate conditions. There seems no general reason, however, why either of the two CCCs which we have already identified as arising from $\mathcal{L}$ should be closed under this coproduct.

Fortunately, something can be salvaged. There are *weak* finite coproducts $\oplus$ in the Kleisli category, obtained by defining

$$(!A, \delta_A) \oplus (!B, \delta_B) \stackrel{\mathrm{def}}{=} (!(!A + !B), \delta_{!A+!B})$$

with, for example, the left injection given by $!\mathrm{inl} \circ \delta_A$. That this is a weak coproduct is easy to check.

2.3 The Monad and Comparison with Let-CCCs

The monad on $\mathcal{C}$ is $(GF, \eta : 1 \to GF, \mu : GFGF \to GF)$ where η is the unit of the adjunction and μ is the natural transformation with components $\mu_X : GFGF(X) \to GF(X)$ given by $\mu_X = G(\varepsilon_{FX})$. Writing T for GF, we obtain the usual monad diagrams:

$$\begin{tikzcd}
TX \arrow[r, "\eta_{TX}"] \arrow[dr, "1"'] & T^2X \arrow[d, "\mu_X"] & TX \arrow[l, "T(\eta_X)"'] \arrow[dl, "1"] \\
& TX &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3X \arrow[r, "T(\mu_X)"] \arrow[d, "\mu_{TX}"'] & T^2X \arrow[d, "\mu_X"] \\
T^2X \arrow[r, "\mu_X"'] & TX
\end{tikzcd}$$

It is then easy to see that (T, η, μ) is a symmetric monoidal monad, in that T is a symmetric monoidal functor and both η and μ are monoidal natural transformations (this is simply a monad in the 2-category of SMCs, SM functors and monoidal natural transformations [Str72]). Cartesian closed categories with (not necessarily symmetric) monoidal monads have recently been the focus of some interest, as they are the models for Moggi's computational lambda calculus [Mog89, Mog91, BBdP93]. The definition is, however, more commonly given in terms of *strong* monads, for which we now make a brief digression. Most of the definitions and results about the various kinds of monads on various kinds of monoidal categories are due to Anders Kock; the interested reader should see [Koc71] and the further references cited there.

2.3.1 Strong Monads

Definition 11 *If $(\mathcal{M}, \otimes, I, \alpha, l, r)$ is a monoidal category, and (T, η, μ) is a monad on $\mathcal{M}$, then T is a* strong monad *if there is a natural transformation τ (called the* tensorial strength*) with components*

$$\tau_{A,B} : A \otimes TB \to T(A \otimes B)$$

such that the following four diagrams commute:

$$\begin{tikzcd}
I \otimes TA \arrow[r, "\tau"] \arrow[dr, "l"'] & T(I \otimes A) \arrow[d, "T(l)"] \\
& TA
\end{tikzcd}
\qquad
\begin{tikzcd}
A \otimes B \arrow[r, "1 \otimes \eta"] \arrow[dr, "\eta"'] & A \otimes TB \arrow[d, "\tau"] \\
& T(A \otimes B)
\end{tikzcd}$$

$$\begin{tikzcd}
(A \otimes B) \otimes TC \arrow[rr, "\tau"] \arrow[d, "\alpha"'] & & T((A \otimes B) \otimes C) \arrow[d, "T(\alpha)"] \\
A \otimes (B \otimes TC) \arrow[r, "1 \otimes \tau"'] & A \otimes T(B \otimes C) \arrow[r, "\tau"'] & T(A \otimes (B \otimes C))
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes T^2 B \arrow[r, "\tau"] \arrow[d, "1 \otimes \mu"'] & T(A \otimes TB) \arrow[r, "T(\tau)"] & T^2(A \otimes B) \arrow[d, "\mu"] \\
A \otimes TB \arrow[rr, "\tau"'] & & T(A \otimes B)
\end{tikzcd}$$

If $\mathcal{M}$ above is *symmetric* monoidal (with symmetry σ), then there is a 'twisted' tensorial strength

$$\tau'_{A,B} : TA \otimes B \to T(A \otimes B)$$

given by

$$\tau'_{A,B} = T(\sigma) \circ \tau_{B,A} \circ \sigma$$

In this case we can also construct a pair of natural transformations Φ, Φ' which have components

$$\Phi_{A,B}, \Phi'_{A,B} : TA \otimes TB \to T(A \otimes B)$$

given by

$$\Phi_{A,B} = \mu_{A \otimes B} \circ T(\tau'_{A,B}) \circ \tau_{TA,B}$$

$$\Phi'_{A,B} = \mu_{A \otimes B} \circ T(\tau_{A,B}) \circ \tau'_{A,TB}$$

The monad is said to be *commutative* if $\Phi = \Phi'$.

Proposition 20 *If $\mathcal{M}$ is a symmetric monoidal category and T is a strong monad on $\mathcal{M}$, then*

1. *either of Φ or Φ', together with the map $\eta_I : I \to TI$, makes T into a monoidal functor;*

2. *both η and μ are monoidal natural transformations with respect to either of these monoidal structures on T;*

3. *T is a symmetric monoidal functor iff it is commutative.*

□

Now, a model of the computational lambda calculus (what Crole and Pitts call a *let-ccc* [Cro92, CP90]) is a cartesian closed category with a strong monad. The above implies that an LNL model always has a strong monad on the CCC part of the model and thus includes a let-ccc. The monad is, however, always commutative (because T is a *symmetric* monoidal functor). It is not the case that all strong monads on CCCs are commutative; indeed, some very important monads arising in computer science are non-commutative, for example the free monoid monad $(list, [\cdot], flatten)$ on the category of sets. Thus it is certainly the case that not all, or even all interesting, let-ccc's will arise from LNL models. Having said that, many of the most important monads arising in semantics, such as lifting and various flavours of powerset/powerdomain, *are* commutative, so the theory of commutative strong monads on CCCs is not without independent interest.

2.4 Examples

The preceding material is all rather abstract, so we now give a couple of concrete examples of LNL models. The first example is important from a computer science perspective and was a major motivation for the present work. The second arises from one of the most common (or at least, most commonly cited) 'mathematical' examples of a symmetric monoidal closed category.

2.4.1 ω-complete Partial Orders

Let $\mathcal{L}$ be the category of pointed ωcpos (ω-cocomplete partial orders with a least element) and strict (bottom preserving) continuous maps. This is a symmetric monoidal closed category with tensor product given by the so-called smash product, the identity for the tensor by the one-point space (which is also a biterminator) and internal hom by the strict continuous function space. In fact, $\mathcal{L}$ also has binary products and coproducts, given by cartesian product and coalesced sum respectively.

Given this choice of $\mathcal{L}$, there are a couple of obvious choices for the CCC $\mathcal{C}$ which give an LNL model. One is to take $\mathcal{C}$ to be the category of pointed ω-cpos and continuous (not necessarily strict) maps, G to be the inclusion functor and F to be the lifting functor $F : X \to X_\perp$. The monoidal structure m on F is given by the evident isomorphism $X_\perp \otimes Y_\perp \cong (X \times Y)_\perp$. In this case, $\mathcal{C}$ is (equivalent to) the Kleisli category of the lifting comonad on $\mathcal{L}$. Note that the cartesian closure of the Kleisli category follows from the fact that $\mathcal{L}$ has products. There are strong coproducts in $\mathcal{L}$ but only weak ones in $\mathcal{C}$.

An alternative choice of $\mathcal{C}$ is the category of (not necessarily pointed) ω-cpos (these are sometimes called *predomains*) and continuous maps, again with inclusion and lifting functors. This is equivalent to the Eilenberg-Moore category of the lift comonad on $\mathcal{L}$, so it has products and coproducts by our previous general arguments, but it also turns out to be cartesian closed.

2.4.2 Abelian Groups

Let $\mathcal{L}$ be the category of Abelian groups and group homomorphisms. This is symmetric monoidal closed with $A \otimes B$ the Abelian group generated by the set of tokens $\{a \otimes b \mid a \in A, b \in B\}$ subject to the relations

$$\begin{aligned}(a_1 + a_2) \otimes b &= a_1 \otimes b + a_2 \otimes b \\ a \otimes (b_1 + b_2) &= a \otimes b_1 + a \otimes b_2\end{aligned}$$

(More categorically, $A \otimes B$ can be defined by a homomorphism $A \times B \to A \otimes B$ which is universal amongst bilinear maps into Abelian groups.) The unit for $\otimes$ is the group of integers under addition, Z, and the internal hom $A \multimap B$ is the group of homomorphisms from A to B with the multiplication inherited from B. In fact $\mathcal{L}$ also has biproducts – the direct sum $A \oplus B$ is both a product and a coproduct and the trivial group is a biterminator.

Now let $\mathcal{C}$ be the category of sets, which is the prototypical example of a cartesian closed category, and F and G be the free and forgetful functors respectively. This gives an LNL model with the monoidal structures on the functors given (in what should be comprehensible notation) by

$$\begin{aligned}n_{A,B} &: GA \times GB \to G(A \otimes B) \\ &: (a, b) \mapsto [a \otimes b]\end{aligned}$$

$$
\begin{array}{rcl}
n & : & 1 \to GZ \\
 & : & * \mapsto 1 \\
m_{X,Y} & : & FX \otimes FY \to F(X \times Y) \\
 & : & [\Sigma_i n_i x_i \ \otimes \ \Sigma_j m_j y_j] \mapsto \Sigma_{i,j} n_i.m_j \, (x_i, y_j) \\
m & : & I \to F1 \\
 & : & n \mapsto n.*
\end{array}
$$

It is fairly straightforward to check that this does indeed give an LNL model. The comonad on $\mathcal{L}$ takes an Abelian group to the free group on its underlying set. ε is 'evaluation' and η is the insertion of generators. This is another example of the situation described in Section 2.2.3, since $\mathcal{C}$ is equivalent to the Kleisli category of the comonad on $\mathcal{L}$.

3 LNL Logic

LNL-models are, of course, supposed to be models of a logical system. Corollary 8 says that they are models for intuitionistic linear logic as defined by Girard, but the form of the definition of LNL-model suggests an interesting alternative presentation of the logic. The basic idea is that one starts with two independent logics, corresponding to the categories $\mathcal{L}$ and $\mathcal{C}$ and then adds operators which correspond in some way to the adjunction between the two categories. To do this and obtain a logic with a good proof theory is, however, not *entirely* straightforward.

Before attempting to be more precise about the rules of LNL-logic, we should perhaps say a little about what we are aiming for. Different researchers approach logic from many different backgrounds and with many different motivations, as is at least partly reflected by the question of whether one describes one's work as logic, type theory or proof theory (or even categorical logic, or categorical proof theory). Of course, the very fact that such a confusion is possible is at the heart of what makes constructive logics exciting objects of study, but it does seem to lead to a certain lack of consensus about what constitutes a 'good' or 'well-behaved' system, and about which results are important.

We take propositional intuitionistic logic as our touchstone and the following properties of that system as our goals: Gentzen-style sequent calculus presentation with (preferably local) cut-elimination and subformula property; equivalent natural deduction system and term calculus with strong normalisation; natural class of categorical models which reflects accurately not just provability, but the equalities on proofs given by cut-elimination and proof normalisation. On the minus side, we are prepared to accept certain infelicities of syntax, such as commuting conversions in natural deduction, and we shall, at least in this paper, ignore Hilbert-style axiomatic presentations entirely. Furthermore we want a logic which contains both linear and non-linear propositions, treated in a way which reflects the symmetric presentation of the intended categorical models.

In keeping with our earlier conventions for naming objects of $\mathcal{L}$ and $\mathcal{C}$, we will use A, B, C to range over linear propositions and X, Y, Z for conventional ones. We shall use Γ and Δ to range over linear contexts (finite multisets of linear propositions) and Θ and Φ for non-linear ones. We also decorate turnstiles with $\mathcal{L}$ or $\mathcal{C}$ to indicate which subsystem they belong to. Finally, if Θ is $X_1, \ldots, X_n$ then $F\Theta$ means $FX_1, \ldots, FX_n$, and similarly for $G\Gamma$. The two classes of propositions with which we shall be dealing are defined by the following grammar:

$$\begin{array}{rcl} A, B & := & A_0 \mid I \mid A \otimes B \mid A \multimap B \mid FX \\ X, Y & := & X_0 \mid 1 \mid X \times Y \mid X \to Y \mid GA \end{array}$$

where A_0 (resp. X_0) ranges over some unspecified set of atomic linear (resp. non-linear) propositions.

3.1 Sequent Calculus

Sequent calculus rules may be divided into three main classes: structural rules, such as weakening or exchange; the cut rule, which allows proofs to be composed, and logical rules. The logical rules are further divided into left and right rules for each connective. In a well-behaved sequent system there should be a certain symmetry between the left and right rules which leads to a cut elimination theorem. Furthermore, in many logics the cut rule is the only rule which can have a formula in the premises which is not a subformula of a formula in the conclusion. For such systems, a cut elimination theorem means that any provable sequent has a proof which only mentions subformulae of the conclusion, which has important implications for, for example, proof search.

The two logics with which we start are very familiar *viz.* the exponential-free, multiplicative fragment of propositional intuitionistic linear logic and the $\times, \rightarrow$ fragment of ordinary intuitionistic logic. These both have sequent presentations with all the properties we desire. How should the systems be enriched and combined to give LNL-logic? We shall approach this question by first outlining two unsatisfactory answers.

3.1.1 The First Wrong Way

The most obvious answer is to take the two familiar sequent calculi and add rules for the two functors and the unit and counit of the adjunction. Thus we have all the usual linear rules (including cut) for deducing sequents of the form $\Gamma \vdash_{\mathcal{L}} A$ and all the usual non-linear rules (including contraction, weakening and another cut rule) for deducing things of the form $\Theta \vdash_{\mathcal{C}} X$, together with the following four new rules:

$$\frac{\Gamma, A \vdash_{\mathcal{L}} B}{\Gamma, FGA \vdash_{\mathcal{L}} B}\ \textit{FG-left} \qquad \frac{\Theta \vdash_{\mathcal{C}} X}{F\Theta \vdash_{\mathcal{L}} FX}\ F$$

$$\frac{\Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{C}} GFX}\ \textit{GF-right} \qquad \frac{\Gamma \vdash_{\mathcal{L}} A}{G\Gamma \vdash_{\mathcal{C}} GA}\ G$$

Categorically we interpret proofs of conventional sequents

$$\begin{array}{c}\Pi \\ X_1, \ldots, X_n \vdash_{\mathcal{C}} Y\end{array}$$

as maps

$$[\![\Pi]\!] : [\![X_1]\!] \times \cdots \times [\![X_n]\!] \rightarrow [\![Y]\!]$$

in $\mathcal{C}$, and proofs of linear sequents

$$\begin{array}{c}\Sigma \\ A_1, \ldots, A_m \vdash_{\mathcal{L}} B\end{array}$$

as maps

$$[\![\Sigma]\!] : [\![A_1]\!] \otimes \cdots \otimes [\![A_m]\!] \rightarrow [\![B]\!]$$

in $\mathcal{L}$, where each of the logical connectives is interpreted as the obviously corresponding piece of categorical structure. (Thus $[\![A \otimes B]\!]$ is $[\![A]\!] \otimes [\![B]\!]$ and so on. Henceforth we will omit semantic brackets whenever we think we can get away with it.)

The interpretations of the four new rules are as follows:

$$\frac{\Gamma \otimes A \xrightarrow{e} B}{\Gamma \otimes FGA \xrightarrow{1 \otimes \varepsilon_A} \Gamma \otimes A \xrightarrow{e} B}\ \textit{FG-left}$$

$$\frac{X_1 \times \cdots \times X_n \xrightarrow{e} Y}{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{m} F(X_1 \times \cdots \times X_n) \xrightarrow{Fe} FY}\ F$$

$$\frac{\Gamma \xrightarrow{e} X}{\Gamma \xrightarrow{e} X \xrightarrow{\eta_X} GFX}\ \textit{GF-right}$$

$$\frac{A_1 \otimes \cdots \otimes A_n \xrightarrow{e} B}{GA_1 \times \cdots GA_n \xrightarrow{n} G(A_1 \otimes \cdots \otimes A_n) \xrightarrow{Ge} GB}\ G$$

Notice how the monoidal structure of the model is used to interpret the two functor rules. The coherence conditions on the model are sufficient to ensure that what we have rather glibly written as m and n above are in fact determined up to isomorphism, and we will in general be rather sloppy about including all the natural isomorphisms which should strictly be included in the categorical interpretations of logical rules. The interpretations of the remaining rules are completely standard, so we omit them for the moment, but note that the two cut rules are interpreted by composition in the two categories.

From the point of view of provability, this collection of rules is fine – it proves exactly the sequents we intend. From the point of view of *proofs*, however, things are not so good. Whilst the logic allows us to express each of the different intended proofs of a given sequent (i.e. morphisms in the free LNL-model), the equality of morphisms is not reflected by a good proof theory. This shows up most obviously in the fact that cut elimination fails for this formulation of the logic. We should not be too surprised that these rules are unsatisfactory, as their form is rather strange – the functor rules introduce a connective on *both* sides of the turnstile whilst the two other rules introduce two connectives at once. The failure of cut elimination can be seen by considering the sequent $FX \vdash_{\mathcal{L}} FX \otimes FX$. This sequent is certainly provable, but there is no rule which could be the last rule of a cut-free proof. This particular problem could be fixed in a slightly *ad hoc* way by adding contraction for linear assumptions of the form FX, but there are other problems, such as the following cut:

$$\frac{\dfrac{\Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{C}} GFX}\ GF\text{-}right \qquad \dfrac{\Gamma, FX \vdash_{\mathcal{L}} A}{G\Gamma, GFX \vdash_{\mathcal{C}} GA}\ G}{\Theta, G\Gamma \vdash_{\mathcal{C}} GA}\ \mathcal{C}\text{-}cut$$

This cut cannot, in general, be removed. (There is a rewrite which replaces the cut with a simpler $\mathcal{L}$-cut, but it also introduces a new cut of the original form for each formula in Θ.)

3.1.2 The Second Wrong Way

The second set of rules which we shall consider looks even odder than the first, so the fact that it too fails to have a good proof theory is no surprise at all. The system is worth mentioning, however, because it is very simple and has a certain appeal from a categorical point of view. Like the first system, we start with the two separate logics but now we add just two rules, each of which is the inverse of the other:

$$\frac{F\Theta \vdash_{\mathcal{L}} A}{\Theta \vdash_{\mathcal{C}} GA}\ G\text{-}right \qquad \frac{\Theta \vdash_{\mathcal{C}} GA}{F\Theta \vdash_{\mathcal{L}} A}\ F\text{-}left$$

These rules are, of course, syntax for the alternative presentation of the adjunction in the model in terms of a natural bijection between hom sets

$$\frac{\mathcal{L}(FX, A)}{\overline{\mathcal{C}(X, GA)}}$$

and indeed it turns out that this system proves exactly the same sequents as the previous one. This is because each rule of one system is derivable (or admissible) in the other. For example, the G rule of the first system is admissible in the second system:

$$\dfrac{\dfrac{GA_n \vdash_{\mathcal{C}} GA_n}{FGA_n \vdash_{\mathcal{L}} A_n}\,F\text{-}left \qquad \begin{array}{c} \dfrac{\dfrac{GA_1 \vdash_{\mathcal{C}} GA_1}{FGA_1 \vdash_{\mathcal{L}} A_1}\,F\text{-}left \qquad A_1,\ldots,A_n \vdash_{\mathcal{L}} B}{FGA_1, A_2, \ldots, A_n \vdash_{\mathcal{L}} B}\,\mathcal{L}\text{-}cut \\ \vdots \\ FGA_1,\ldots,FGA_{n-1},A_n \vdash_{\mathcal{L}} B \end{array}}{\dfrac{FGA_1,\ldots,FGA_n \vdash_{\mathcal{L}} B}{GA_1,\ldots,GA_n \vdash_{\mathcal{C}} GB}\,G\text{-}right}\,\mathcal{L}\text{-}cut$$

The reader may be surprised by the equivalence of these two systems. In particular, the way in which the monoidal structures on the two functors arise in the first presentation is clear, but the second system does not mention G on the left at all. Where, then, does the monoidal structure on G come from? The answer is in the remark made at the end of Section 2.1 – the categorical interpretation of the second system uses both m and m^{-1} (which we called p earlier), and this, together with the adjunction, is sufficient to ensure that G is monoidal too.

$$\dfrac{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{e} A}{X_1 \times \cdots \times X_n \xrightarrow{\eta} GF(X_1 \times \cdots \times X_n) \xrightarrow{Gm^{-1}} G(FX_1 \otimes \cdots \otimes FX_n) \xrightarrow{Ge} GA}\;G\text{-}right$$

$$\dfrac{X_1 \times \cdots \times X_n \xrightarrow{e} GA}{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{m} F(X_1 \times \cdots \times X_n) \xrightarrow{Fe} FGA \xrightarrow{\varepsilon} A}\;F\text{-}left$$

That the second systems fails to have cut-elimination may be seen by considering the following situation:

$$\dfrac{\dfrac{F\Theta \vdash_{\mathcal{L}} A}{\Theta \vdash_{\mathcal{C}} GA}\,G\text{-}right \qquad \dfrac{F\Theta', FGA \vdash_{\mathcal{L}} B}{\Theta', GA \vdash_{\mathcal{C}} GB}\,G\text{-}right}{\Theta, \Theta' \vdash_{\mathcal{C}} GB}\,\mathcal{C}\text{-}cut$$

or by trying to find a cut-free proof of $FX \vdash_{\mathcal{L}} F(X \times X)$. Another reason for rejecting this second system is that even cut-free proofs do not have the subformula property.

3.1.3 A Well-Behaved Sequent Calculus

Fortunately, there *is* a way to present the logic which has a good proof theory. The trick is to allow conventional non-linear formulae to appear in the assumptions of a linear sequent. A typical linear sequent looks, therefore, like this:

$$X_1, \ldots, X_m, A_1, \ldots, A_n \vdash_{\mathcal{L}} B$$

which is interpreted as a morphism in $\mathcal{L}$ of the form

$$FX_1 \otimes \cdots \otimes FX_m \otimes A_1 \otimes \cdots \otimes A_n \longrightarrow B$$

Non-linear sequents are still constrained to have purely non-linear antecedents and are interpreted as morphisms in $\mathcal{C}$ as before.[9]

We will usually abuse notation by writing linear sequents in the form $\Theta; \Gamma \vdash_{\mathcal{L}} A$, even though there is no need at all for the ';' since linear and non-linear formulae can never be confused. It is important to understand that there is really just one kind of comma in the antecedent, and that the exchange rule (which we will supress) really allows linear and non-linear formulae to be mingled. Once this is understood, however, our potentially misleading notation seems rather less confusing than the alternative (to which we shall return) of introducing new metavariables ranging over arbitrary propositions and contexts. The sequent rules for LNL logic are shown in Figures 1 and 2.

There are several points to be noted about the rules. There are three cut rules according to the type of the cut formula and of the ultimate conclusion (there is no $\mathcal{LC}$-cut rule because a linear formula cannot be cut into a non-linear sequent). Each of the non-linear left rules (including contraction and weakening) splits into two versions according to the type of the overall sequent. The rules for F and G look much pleasant than in the two unsatisfactory systems – each has one left and one right rule, neither of which affect the rest of the sequent. The annotations on the turnstiles are, strictly speaking, redundant as they are implicit in the consequent. The following is easy to verify:

Proposition 21 *The sequent rules of LNL logic are equivalent in terms of provability to the two systems presented earlier. To be precise:*

- $\Theta \vdash_{\mathcal{C}} X$ *in LNL logic iff* $\Theta \vdash_{\mathcal{C}} X$ *in either of the earlier systems.*
- $\Theta; \Gamma \vdash_{\mathcal{L}} A$ *in LNL logic iff* $F\Theta, \Gamma \vdash_{\mathcal{L}} A$ *in either of the earlier systems.*

□

The interpretation of LNL logic in an LNL-model is fairly straightforward, given what has gone before. We assume that the reader is familiar with the interpretation of the standard logical connectives and just give details of the interpretation of one of the cut rules and the four rules for F and G in Figure 3.

3.1.4 Cut Elimination

We now turn to the question of cut elimination in LNL logic. As usual, the proof describes a procedure in which the cuts in a proof are locally rewritten (making the proof, in general, much larger) so that they percolate up towards the leaves, where they eventually disappear. As is also usual, the fine details of making the induction go through are slightly delicate. In particular, we start by replacing the $\mathcal{CL}$-cut and $\mathcal{CC}$-cut rules with the following n-ary ($n \geq 0$) variants, yielding an equivalent system which we call LNL$^+$:

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad X^n, \Phi; \Gamma \vdash_{\mathcal{L}} A}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}cut_n \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad X^n, \Phi \vdash_{\mathcal{C}} Y}{\Theta, \Phi \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\text{-}cut_n$$

where

$$X^n \stackrel{\text{def}}{=} \overbrace{X, \ldots, X}^{n}$$

[9]The attempt to make a more symmetric system by allowing linear assumptions in conventional sequents gives yet another system without cut-elimination.

Axioms

$$A \vdash_{\mathcal{L}} A \ \mathcal{L}\textit{-axiom} \qquad\qquad X \vdash_{\mathcal{C}} X \ \mathcal{C}\textit{-axiom}$$

Structural Rules

$$\frac{\Theta, X, X; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\textit{-contraction} \qquad \frac{\Theta, X, X \vdash_{\mathcal{C}} Y}{\Theta, X \vdash_{\mathcal{C}} Y}\ \mathcal{C}\textit{-contraction}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\textit{-weakening} \qquad \frac{\Theta \vdash_{\mathcal{C}} Y}{\Theta, X \vdash_{\mathcal{C}} Y}\ \mathcal{C}\textit{-weakening}$$

Cut Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Phi; \Gamma \vdash_{\mathcal{L}} A}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\textit{-cut} \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Phi \vdash_{\mathcal{C}} Y}{\Theta, \Phi \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\textit{-cut}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Phi; A, \Delta \vdash_{\mathcal{L}} B}{\Theta, \Phi; \Gamma, \Delta \vdash_{\mathcal{L}} B}\ \mathcal{LL}\textit{-cut}$$

$\times/1$ Rules

$$\frac{\Theta, X \vdash_{\mathcal{C}} Z}{\Theta, X \times Y \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\times\textit{-left1} \qquad \frac{\Theta, Y \vdash_{\mathcal{C}} Z}{\Theta, X \times Y \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\times\textit{-left2}$$

$$\frac{\Theta, X; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X \times Y; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\text{-}\times\textit{-left1} \qquad \frac{\Theta, Y; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X \times Y; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\text{-}\times\textit{-left2}$$

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad \Phi \vdash_{\mathcal{C}} Y}{\Theta, \Phi \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-right} \qquad \frac{}{\vdash_{\mathcal{C}} 1}\ 1\textit{-right}$$

$\otimes/I$ Rules

$$\frac{\Theta; \Gamma, A, B \vdash_{\mathcal{L}} C}{\Theta; \Gamma, A \otimes B \vdash_{\mathcal{L}} C}\ \otimes\textit{-left} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Phi; \Delta \vdash_{\mathcal{L}} B}{\Theta, \Phi; \Gamma, \Delta \vdash_{\mathcal{L}} A \otimes B}\ \otimes\textit{-right}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A}{\Theta; \Gamma, I \vdash_{\mathcal{L}} A}\ I\textit{-left} \qquad \frac{}{\vdash_{\mathcal{L}} I}\ I\textit{-right}$$

Figure 1: Sequent calculus presentation of LNL logic (I)

$\rightarrow$ Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad Y, \Phi \vdash_{\mathcal{C}} Z}{\Theta, X \rightarrow Y, \Phi \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}{\rightarrow}\text{-}\mathit{left} \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad Y, \Phi; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X \rightarrow Y, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\text{-}{\rightarrow}\text{-}\mathit{left}$$

$$\frac{\Theta, X \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} X \rightarrow Y}\ {\rightarrow}\text{-}\mathit{right}$$

$\multimap$ Rules

$$\frac{\Theta; \Gamma, A \vdash_{\mathcal{L}} B}{\Theta; \Gamma \vdash_{\mathcal{L}} A \multimap B}\ {\multimap}\text{-}\mathit{right} \qquad \frac{\Theta, \Gamma \vdash_{\mathcal{L}} A \qquad \Phi; \Delta, B \vdash_{\mathcal{L}} C}{\Theta, \Phi; \Gamma, A \multimap B, \Delta \vdash_{\mathcal{L}} C}\ {\multimap}\text{-}\mathit{left}$$

F Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{L}} FX}\ F\text{-}\mathit{right} \qquad \frac{\Theta, X; \Gamma \vdash_{\mathcal{L}} A}{\Theta; FX, \Gamma \vdash_{\mathcal{L}} A}\ F\text{-}\mathit{left}$$

G Rules

$$\frac{\Theta; B, \Gamma \vdash_{\mathcal{L}} A}{\Theta, GB; \Gamma \vdash_{\mathcal{L}} A}\ G\text{-}\mathit{left} \qquad \frac{\Theta \vdash_{\mathcal{L}} A}{\Theta \vdash_{\mathcal{C}} GA}\ G\text{-}\mathit{right}$$

Figure 2: Sequent calculus presentation of LNL logic (II)

$$\frac{Y_1 \times \cdots \times Y_n \xrightarrow{e} X \qquad FX \otimes F\Phi \otimes \Gamma \xrightarrow{f} A}{\left(\bigotimes_i FY_i\right) \otimes F\Phi \otimes \Gamma \xrightarrow{m \otimes 1 \otimes 1} F\left(\prod_i Y_i\right) \otimes F\Phi \otimes \Gamma \xrightarrow{Fe \otimes 1 \otimes 1} FX \otimes F\Phi \otimes \Gamma \xrightarrow{f} A}\ \mathcal{CL}\text{-}\mathit{cut}$$

$$\frac{X_1 \times \cdots \times X_n \xrightarrow{e} X}{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{m} F(X_1 \times \cdots \times X_n) \xrightarrow{Fe} FX}\ F\text{-}\mathit{right}$$

$$\frac{(F\Theta \otimes FX) \otimes \Gamma \xrightarrow{e} A}{F\Theta \otimes (FX \otimes \Gamma) \xrightarrow{e} A}\ F\text{-}\mathit{left}$$

$$\frac{F\Theta \otimes B \otimes \Gamma \xrightarrow{e} A}{F\Theta \otimes FGB \otimes \Gamma \xrightarrow{1 \otimes \varepsilon \otimes 1} F\Theta \otimes B \otimes \Gamma \xrightarrow{e} A}\ G\text{-}\mathit{left}$$

$$\frac{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{e} A}{\prod_i X_i \xrightarrow{\eta} GF\left(\prod_i X_i\right) \xrightarrow{Gm^{-1}} G\left(\bigotimes_i FX_i\right) \xrightarrow{Ge} GA}\ G\text{-}\mathit{right}$$

Figure 3: Categorical interpretation of LNL logic (sketch)

These rules are easily seen to be admissible in LNL, and it is also clear that cut-elimination for LNL$^+$ implies cut-elimination for LNL.

Define the *rank* $|A|$ (resp. $|X|$) of a linear (resp. non-linear) proposition to be the number of logical connectives in the proposition (so in particular, the rank of atomic propositions is 0). The *cut rank* $c(\Pi)$ of a proof Π is one more than the maximum of the ranks of all the cut formulae in Π, and 0 if Π is cut-free. The *depth* $d(\Pi)$ of a proof Π is the length of the longest path in the proof tree (so the depth of an axiom is 0). The key to the proof is the following lemma, which shows how to transform a single cut, either by removing it or by replacing it with one or more simpler cuts:

Lemma 22 (Cut Reduction)

1. *If* Π_1 *is an* LNL$^+$ *proof of* $\Theta \vdash_{\mathcal{C}} X$ *and* Π_2 *is an* LNL$^+$ *proof of* $X^n, \Phi \vdash_{\mathcal{C}} Y$ *with* $c(\Pi_1), c(\Pi_2) \leq |X|$ *then there exists a proof* Π *of* $\Theta, \Phi \vdash_{\mathcal{C}} Y$ *with* $c(\Pi) \leq |X|$*;*
2. *If* Π_1 *is an* LNL$^+$ *proof of* $\Theta \vdash_{\mathcal{C}} X$ *and* Π_2 *is an* LNL$^+$ *proof of* $X^n, \Phi; \Gamma \vdash_{\mathcal{L}} A$ *with* $c(\Pi_1), c(\Pi_2) \leq |X|$ *then there exists a proof* Π *of* $\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A$ *with* $c(\Pi) \leq |X|$*;*
3. *If* Π_1 *is an* LNL$^+$ *proof of* $\Theta; \Gamma \vdash_{\mathcal{L}} A$ *and* Π_2 *is an* LNL$^+$ *proof* $\Phi; A, \Delta \vdash_{\mathcal{L}} B$ *with* $c(\Pi_1), c(\Pi_2) \leq |A|$ *then there exists a proof* Π *of* $\Theta, \Phi; \Gamma, \Delta \vdash_{\mathcal{L}} B$ *with* $c(\Pi) \leq |A|$.

Proof. The three parts are proved simultaneously by induction on $d(\Pi_1) + d(\Pi_2)$. We consider cases according to the classes of the last rules used in each of the two proofs:

1. Both proofs end in logical rules which introduce the cut formula (so Π_1 ends in a right rule and Π_2 in a corresponding left rule). This is the most interesting case, and we consider each subcase in turn:

   F-right/ F-left In this case we have

$$\Pi_1 = \dfrac{\begin{array}{c}\pi_1\\ \Theta \vdash_{\mathcal{C}} X\end{array}}{\Theta \vdash_{\mathcal{L}} FX}\ F\text{-}right \qquad \Pi_2 = \dfrac{\begin{array}{c}\pi_2\\ \Phi, X; FX^n, \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Phi; FX^{n+1}, \Gamma \vdash_{\mathcal{L}} A}\ F\text{-}left$$

   By the induction hypothesis applied to the proofs Π_1 and π_2 there exists a proof Π' of $\Theta, \Phi, X; \Gamma \vdash_{\mathcal{L}} A$ with $c(\Pi') \leq |FX| = |X| + 1$. Then let Π be the following proof:

$$\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Pi'\\ \Theta, \Phi, X; \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta, \Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}cut_1}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A}$$

   where the double line stands for a number of contractions. Π has cut rank $\max(|X| + 1, c(\pi_1), c(\Pi'))$ which is equal to $|X| + 1 = |FX|$ as required.
   Note that there is an obvious simplification in the case that $n = 0$ as we can then avoid an appeal to the induction hypothesis altogether by letting Π be simply

$$\dfrac{\begin{array}{c}\pi_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\pi_2\\ \Phi, X; \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}cut_1$$

G-right/ G-left We have

$$\Pi_1 = \cfrac{\overset{\pi_1}{\Theta \vdash_{\mathcal{L}} A}}{\Theta \vdash_{\mathcal{C}} GA}\ G\text{-}right \qquad \Pi_2 = \cfrac{\overset{\pi_2}{\Phi, GA^n; A, \Gamma \vdash_{\mathcal{L}} B}}{\Phi, GA^{n+1}; \Gamma \vdash_{\mathcal{L}} B}\ G\text{-}left$$

By applying the induction hypothesis to Π_1 and π_2 we obtain a proof Π' of $\Theta, \Phi; A, \Gamma \vdash_{\mathcal{L}} B$ with $c(\Pi') \leq |GA| = |A| + 1$. Now let Π be

$$\cfrac{\cfrac{\overset{\pi_1}{\Theta \vdash_{\mathcal{L}} A} \qquad \overset{\Pi'}{\Theta, \Phi; A, \Gamma \vdash_{\mathcal{L}} B}}{\Theta, \Theta, \Phi; \Gamma \vdash_{\mathcal{L}} B}\ \mathcal{LL}\text{-}cut}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} B}$$

The cut rank of Π is $\max(|A|+1, c(\pi_1), c(\Pi')) = |A|+1$ so we are done. Again, there is an obvious simplification when $n = 0$.

×-right/ C-×-left1 We have

$$\Pi_1 = \cfrac{\overset{\pi_1}{\Theta_1 \vdash_{\mathcal{C}} X} \qquad \overset{\pi_2}{\Theta_2 \vdash_{\mathcal{C}} Y}}{\Theta_1, \Theta_2 \vdash_{\mathcal{C}} X \times Y}\ \times\text{-}right$$

and

$$\Pi_2 = \cfrac{\overset{\pi_3}{\Phi, X, (X \times Y)^n \vdash_{\mathcal{C}} Z}}{\Phi, (X \times Y)^{n+1} \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\times\text{-}left1$$

Let Π' be the result of applying the induction hypothesis to Π_1 and π_3, so Π' is a proof of $\Theta_1, \Theta_2, \Phi, X \vdash_{\mathcal{C}} Z$ with $c(\Pi') \leq |X \times Y| = |X| + |Y| + 1$. Now let Π be

$$\cfrac{\cfrac{\overset{\pi_1}{\Theta_1 \vdash_{\mathcal{C}} X} \qquad \overset{\Pi'}{\Theta_1, \Theta_2, \Phi, X \vdash_{\mathcal{C}} Z}}{\Theta_1, \Theta_1, \Theta_2, \Phi \vdash_{\mathcal{C}} Z}\ \mathcal{CC}\text{-}cut_1}{\Theta_1, \Theta_2, \Phi \vdash_{\mathcal{C}} Z}$$

which has a cut rank of $\max(|X| + 1, c(\pi_1), c(\Pi')) \leq |X \times Y|$.

- The remaining subcases are similar and left to the reader.

2. The last rule used in Π_1 is *not* a right logical rule. These are dealt with by simple permutations of the rules. We consider each remaining possibility for the last rule in Π_1 and form of conclusion of Π_2 in turn. A few representative cases:

C-contraction/ C sequent The situation is

$$\Pi_1 = \cfrac{\overset{\pi_1}{\Theta, X, X \vdash_{\mathcal{C}} Y}}{\Theta, X \vdash_{\mathcal{C}} Y}\ \mathcal{C}\text{-}contraction \qquad \overset{\Pi_2}{Y^n, \Phi \vdash_{\mathcal{C}} Z}$$

and by induction applied to π_1 and Π_2 there is a proof Π' of $\Theta, X, X, \Phi \vdash_{\mathcal{C}} Z$ with $c(\Pi') \leq |Y|$. Let Π be

$$\dfrac{\begin{array}{c}\Pi' \\ \Theta, X, X, \Phi \vdash_{\mathcal{C}} Z\end{array}}{\Theta, X, \Phi \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}contraction$$

Clearly, $c(\Pi) \leq |Y|$ so we are done.

$\mathcal{CC}$-cut$_n$/ $\mathcal{L}$ sequent

$$\Pi_1 = \dfrac{\begin{array}{c}\pi_1 \\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\pi_2 \\ X^n, \Phi \vdash_{\mathcal{C}} Y\end{array}}{\Theta, \Phi \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\text{-}cut_n \qquad \begin{array}{c}\Pi_2 \\ Y^m, \Phi'; \Gamma \vdash_{\mathcal{L}} A\end{array}$$

By induction applied to π_2 and Π_2 we can form Π' proving $X^n, \Phi, \Phi'; \Gamma \vdash_{\mathcal{L}} A$ with $c(\Pi') \leq |Y|$. Now let Π be

$$\dfrac{\begin{array}{c}\pi_1 \\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Pi' \\ X^n, \Phi, \Phi'; \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta, \Phi, \Phi'; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}cut_n$$

By assumption, $c(\Pi_1) \leq |Y|$, so in particular $|X| + 1 \leq |Y|$. This means $c(\Pi) = \max(|X| + 1, c(\pi_1), c(\Pi')) \leq |Y|$ as required.

$\multimap$-left/ $\mathcal{L}$ sequent

$$\Pi_1 = \dfrac{\begin{array}{c}\pi_1 \\ \Theta; \Gamma \vdash_{\mathcal{L}} A\end{array} \qquad \begin{array}{c}\pi_2 \\ \Phi; B, \Delta \vdash_{\mathcal{L}} C\end{array}}{\Theta, \Phi; \Gamma, A \multimap B, \Delta \vdash_{\mathcal{L}} C}\ \multimap\text{-}left \qquad \begin{array}{c}\Pi_2 \\ \Theta'; C, \Gamma' \vdash_{\mathcal{L}} D\end{array}$$

By induction applied to π_2 and Π_2 there's a proof Π' of $\Phi, \Theta'; B, \Delta, \Gamma' \vdash_{\mathcal{L}} D$ with $c(\Pi') \leq |C|$. Let Π be

$$\dfrac{\begin{array}{c}\pi_1 \\ \Theta; \Gamma \vdash_{\mathcal{L}} A\end{array} \qquad \begin{array}{c}\Pi' \\ \Phi, \Theta'; B, \Delta, \Gamma' \vdash_{\mathcal{L}} D\end{array}}{\Theta, \Phi, \Theta'; \Gamma, A \multimap B, \Delta, \Gamma' \vdash_{\mathcal{L}} D}\ \multimap\text{-}left$$

and $c(\Pi) = \max(c(\pi_1), c(\Pi')) \leq |C|$ as required.

$\mathcal{L}$-axiom/ $\mathcal{L}$ sequent This is one of the base cases for the induction. We have

$$\Pi_1 = A \vdash_{\mathcal{L}} A \qquad \begin{array}{c}\Pi_2 \\ \Theta; A, \Gamma \vdash_{\mathcal{L}} B\end{array}$$

and we simply let Π be Π_2 (recall that $c(\Pi_2) \leq |A|$ by assumption).

3. The cut formula is a minor formula of the last rule in Π_2. These cases are also dealt with by fairly straightforward permutations and we omit them.

4. The last rule in Π_2 is contraction on the cut formula. This is why we have the n-ary cut rules.

$\mathcal{C}$-contraction

$$\begin{array}{c}\Pi_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \Pi_2 = \dfrac{\begin{array}{c}\pi_1\\ X^{n+2}, \Phi \vdash_{\mathcal{C}} Y\end{array}}{X^{n+1}, \Phi \vdash_{\mathcal{C}} Y}\ \mathcal{C}\text{-}contraction$$

By induction on Π_1 and π_1 there is a Π proving $\Theta, \Phi \vdash_{\mathcal{C}} Y$ with $c(\Pi) \leq |X|$ as required.

$\mathcal{L}$-contraction Similar.

5. The last rule in Π_2 is weakening introducing the cut formula.

$\mathcal{L}$-weakening

$$\begin{array}{c}\Pi_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \dfrac{\begin{array}{c}\pi_1\\ X^{n}, \Phi; \Gamma \vdash_{\mathcal{L}} A\end{array}}{X^{n+1}, \Phi; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\text{-}weakening$$

By induction on Π_1 and π_1 there is a Π proving $\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A$ with $c(\Pi) \leq |X|$ as required. There is a simplification if $n = 0$, in which case Π is just

$$\begin{array}{c}\pi_1\\ \Phi; \Gamma \vdash_{\mathcal{L}} A\\ \hline\hline \Theta, \Phi; \Gamma \vdash_{\mathcal{L}} A\end{array}$$

where $c(\Pi) = c(\pi_1) \leq |X|$ by assumption.

$\mathcal{C}$-weakening Similar.

6. Π_2 is an axiom on the cut formula. Trivial.

□

Lemma 23 *Let Π be an* LNL$^+$ *proof of a sequent $\Theta \vdash_{\mathcal{C}} X$ or $\Theta; \Gamma \vdash_{\mathcal{L}} A$ such that $c(\Pi) > 0$. Then there is a proof Π' of the same sequent with $c(\Pi') < c(\Pi)$.*

Proof. Induction on $d(\Pi)$. If the last inference of Π is not a cut then we simply apply the induction hypothesis. Assume then that the last inference is a cut on a formula A (the two cases of cuts on non-linear formulae are treated in just the same way). If $c(\Pi) > |A| + 1$ then we can apply the induction hypothesis. This leaves the case where the last rule is a cut on A and $c(\Pi) = |A| + 1$ so that

$$\Pi = \dfrac{\begin{array}{c}\Pi_1\\ \Theta; \Gamma \vdash_{\mathcal{L}} A\end{array} \qquad \begin{array}{c}\Pi_2\\ \Phi; A, \Delta \vdash_{\mathcal{L}} B\end{array}}{\Theta, \Phi; \Gamma, \Delta \vdash_{\mathcal{L}} B}\ \mathcal{LL}\text{-}cut$$

Clearly $c(\Pi_1), c(\Pi_2) \leq |A| + 1$, so by induction we can construct Π_1' proving $\Theta; \Gamma \vdash_{\mathcal{L}} A$ and Π_2' proving $\Phi; A, \Delta \vdash_{\mathcal{L}} B$ with $c(\Pi_1'), c(\Pi_2') \leq |A|$. Then by Lemma 22, we can construct a Π' proving $\Theta, \Phi; \Gamma, \Delta \vdash_{\mathcal{L}} B$ with $c(\Pi') \leq |A|$ as required. □

Theorem 24 (Cut Elimination) *Let* Π *be a proof of a sequent* $\Theta \vdash_{\mathcal{C}} X$ *or* $\Theta; \Gamma \vdash_{\mathcal{L}} A$ *such that* $c(\Pi) > 0$*. Then there is an algorithm which yields a cut-free proof* Π' *of the same sequent.*

Proof. This follows immediately by induction on $c(\Pi)$ and Lemma 23. □

It is very important to note that the *proof* of the cut elimination theorem says a lot more than that the theorem is true as stated. The proof gives a *procedure* for simplifying proofs by applying successive rewrites until a cut-free proof is reached. These rewriting steps are purely local and cut-free proofs also have the subformula property. Note that the algorithm described by the cut-elimination proof is non-deterministic – there is some freedom in choosing the order in which rewrites should be applied. On the other hand, the order in which transformations are applied is constrained rather more than is strictly necessary in order to make the induction work. In the present work we shall not, however, consider further the question of the extent to which cut elimination is *strongly* normalising.[10]

3.1.5 Cut Elimination and Semantic Equality

The cut elimination process gives a notion of equality on sequent proofs, obtained by extending the one-step proof rewriting relation of the algorithm to a congruence (an equivalence relation which is compositional on proof trees). We intend this syntactic equality to be modelled soundly by equality in LNL models, and this is indeed the case:

Theorem 25 *The cut-elimination procedure described in Section 3.1.4 is modelled soundly in any LNL model.*

Proof. The basic idea is to show that whenever one proof is simplified to another then the interpretations of those two proofs are equal morphisms in the model. This is done by modifying the statement and proof of the cut reduction lemma (Lemma 22) to show that semantic equality is preserved. Rather than go into the tedious details, we just sketch one of the cases:

G-right/ G-left The cut reduction is

$$\dfrac{\dfrac{\overset{\pi_1}{\Theta \vdash_{\mathcal{L}} A}}{\Theta \vdash_{\mathcal{C}} GA}\,G\text{-}right \qquad \dfrac{\overset{\pi_2}{\Phi; A, \Gamma \vdash_{\mathcal{L}} B}}{\Phi, GA; \Gamma \vdash_{\mathcal{L}} B}\,G\text{-}left}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} B}\,\mathcal{CL}\text{-}cut$$

reduces to

$$\dfrac{\Theta \vdash_{\mathcal{L}} A \qquad \Phi; A, \Gamma \vdash_{\mathcal{L}} B}{\Theta, \Phi; \Gamma \vdash_{\mathcal{L}} B}\,\mathcal{LL}\text{-}cut$$

[10] One of the CSL referees asserted that cut elimination *is* strongly normalising, but I don't see how to justify that without a lot more work.

Now, if $\Theta = X_1, \ldots, X_n$, $[\![\pi_1]\!] = f$ and $[\![\pi_2]\!] = g$, then this corresponds categorically to the commutation of

$$
\begin{tikzcd}
 & (FX_1 \otimes \cdots \otimes FX_n) \otimes F\Phi \otimes \Gamma \arrow[dl, "m \otimes 1 \otimes 1"] \arrow[dddr, "f \otimes 1 \otimes 1"] & \\
F(X_1 \times \cdots \times X_n) \otimes F\Phi \otimes \Gamma \arrow[d, "F(\eta) \otimes 1 \otimes 1"] & & \\
FGF(X_1 \times \cdots \times X_n) \otimes F\Phi \otimes \Gamma \arrow[d, "FG(m^{-1}) \otimes 1 \otimes 1"] & & \\
FG(FX_1 \otimes \cdots \otimes FX_n) \otimes F\Phi \otimes \Gamma \arrow[d, "FG(f) \otimes 1 \otimes 1"] & & A \otimes F\Phi \otimes \Gamma \arrow[dddl, "g"] \\
FGA \otimes F\Phi \otimes \Gamma \arrow[d, "\varepsilon \otimes 1 \otimes 1"] & & \\
A \otimes F\Phi \otimes \Gamma \arrow[dr, "g"] & & \\
 & B &
\end{tikzcd}
$$

which is easily seen to follow using one of the triangle laws for the adjunction and naturality of ε.

□

3.1.6 Variations: Introducing Additive Non-Linear Contexts

There are a large number of possible variations on the sequent rules for LNL logic. One of the most natural is to treat the non-linear antecedents as additive rather than multiplicative (though linear antecedents are still multiplicative, of course). This also has the advantage of a closer correspondence to the natural deduction system which we shall introduce in Section 3.2 and is one of the reasons for our notational device of separating the linear and non-linear parts of the antecedents of linear sequents.

The additive variants of those rules which change are shown in Figure 4. The remaining rules remain unchanged. When we wish to distinguish the additively formulated sequent system from the multipliciative, we shall refer to LNL^a or to LNL^m.

Axioms

$$\Theta; A \vdash_{\mathcal{L}} A \ \mathcal{L}\text{-}axiom \qquad\qquad \Theta, X \vdash_{\mathcal{C}} X \ \mathcal{C}\text{-}axiom$$

Cut Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Theta; \Gamma \vdash_{\mathcal{L}} A}{\Theta, \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}cut \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Theta \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\text{-}cut$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Theta; A, \Delta \vdash_{\mathcal{L}} B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} B}\ \mathcal{LL}\text{-}cut$$

$\times/1$ Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad \Theta \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\text{-}right \qquad\qquad \frac{}{\Theta \vdash_{\mathcal{C}} 1}\ 1\text{-}right$$

$\otimes/I$ Rules

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Theta; \Delta \vdash_{\mathcal{L}} B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} A \otimes B}\ \otimes\text{-}right \qquad\qquad \frac{}{\Theta;\ \vdash_{\mathcal{L}} I}\ I\text{-}right$$

$\rightarrow$ Rules

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad Y, \Theta \vdash_{\mathcal{C}} Z}{\Theta, X \rightarrow Y \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\!\rightarrow\!\text{-}left \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad Y, \Theta; \Gamma \vdash_{\mathcal{L}} A}{\Theta, X \rightarrow Y; \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{L}\text{-}\!\rightarrow\!\text{-}left$$

$\multimap$ Rules

$$\frac{\Theta, \Gamma \vdash_{\mathcal{L}} A \qquad \Phi; \Delta, B \vdash_{\mathcal{L}} C}{\Theta, \Phi; \Gamma, A \multimap B, \Delta \vdash_{\mathcal{L}} C}\ \multimap\text{-}left$$

Figure 4: Additive variations on LNL logic

The following facts concerning LNL^a are easily verified:

Proposition 26

1. *The systems* LNL^a *and* LNL^m *are equivalent: each rule in one system is admissible in the other;*

2. *The weakening and contraction rules of* LNL^m *are admissible in* LNL^a *without weakening and contraction.*

3. *Cut elimination holds for* LNL^a*.*

□

3.1.7 Variations: A Parsimonious Presentation

As we have already mentioned, there is another way of presenting the logic by using some new metavariables: let P, Q range over either linear or non-linear propositions and Υ over mixed contexts. We can then present LNL^m in a concise way as shown in Figure 5.

This presentation is equivalent to that shown in Figures 1 and 2. It has the disadvantage of obscuring the fact that there are really two distinct kinds of sequent. These rules are essentially the same as those given by Jacobs in [Jac93], which also contains good accounts of some examples of concrete categorical models. The description of the semantics in that paper is somewhat different from that given here, however. Jacobs starts with a linear category $\mathcal{L}$ satisfying extra conditions which make the category of !-coalgebras be cartesian closed. He then interprets *all* sequents as morphisms in $\mathcal{L}$ by applying F to the interpretation of non-linear formulae (in much the same way that we have interpreted linear sequents). This causes problems as it is not clear how to interpret $\rightarrow$-right, for example. The solution is a mixture of syntax and semantics – one can verify that all *provable* sequents which only mention non-linear formulae satisfy what is called the *conventional witness property*. This means that they are interpreted by morphisms in $\mathcal{L}$ which are (up to m) the image under F of coalgebra morphisms. This property, which is necessary to complete the interpretation, is shown by induction on derivations. Interestingly, the proof given is incomplete unless one uses the following crucial (and easily verified) fact, which is never actually mentioned:

Lemma 27 *Any provable parsimonious sequent with a non-linear consequent has only non-linear formulae in the anetecedent.*

In fact, for the presentation of the logic given by Jacobs, the lemma above is only true because I is treated as a derived formula (it is defined to be $F(1)$, *cf.* our Proposition 1). The left rule for $\otimes$ is given as

$$\frac{\Upsilon, A, B \vdash P}{\Upsilon, A \otimes B \vdash P}$$

but it just so happens that the conclusion P will always be a linear formula C. This would cease to be true if the left rule for I were given explicitly as the nullary version of that for $\otimes$:

$$\frac{\Upsilon \vdash P}{\Upsilon, I \vdash P}$$

for then one could introduce linear antecedents to non-linear formulae and the proof theory would break down. A slightly subtle point is that the above rule for I appears at first

Axiom

$$P \vdash P$$

Structural Rules

$$\frac{\Upsilon, X, X \vdash P}{\Upsilon, X \vdash P}\ \textit{Contraction} \qquad \frac{\Upsilon \vdash P}{\Upsilon, X \vdash P}\ \textit{Weakening}$$

Cut Rule

$$\frac{\Upsilon \vdash P \qquad P, \Upsilon' \vdash Q}{\Upsilon, \Upsilon' \vdash Q}\ \textit{Cut}$$

$\times/1$ Rules

$$\frac{\Upsilon, X \vdash P}{\Upsilon, X \times Y \vdash P}\ \times\textit{-left1} \qquad \frac{\Upsilon, Y \vdash P}{\Upsilon, X \times Y \vdash P}\ \times\textit{-left2}$$

$$\frac{\Theta \vdash X \qquad \Phi \vdash Y}{\Theta, \Phi \vdash X \times Y}\ \times\textit{-right} \qquad \frac{}{\vdash 1}\ 1\textit{-right}$$

$\otimes/I$ Rules

$$\frac{\Upsilon, A, B \vdash C}{\Upsilon, A \otimes B \vdash C}\ \otimes\textit{-left} \qquad \frac{\Upsilon \vdash A \qquad \Upsilon' \vdash B}{\Upsilon, \Upsilon' \vdash A \otimes B}\ \otimes\textit{-right}$$

$$\frac{\Upsilon \vdash A}{\Upsilon, I \vdash A}\ I\textit{-left} \qquad \frac{}{\vdash I}\ I\textit{-right}$$

$\rightarrow$ Rules

$$\frac{\Theta \vdash X \qquad Y, \Upsilon \vdash P}{\Theta, X \rightarrow Y, \Upsilon \vdash P}\ \rightarrow\textit{-left} \qquad \frac{\Theta, X \vdash Y}{\Theta \vdash X \rightarrow Y}\ \rightarrow\textit{-right}$$

$\multimap$ Rules

$$\frac{\Upsilon \vdash A \qquad B, \Upsilon' \vdash C}{\Upsilon, A \multimap B, \Upsilon' \vdash C}\ \multimap\textit{-left} \qquad \frac{\Upsilon, A \vdash B}{\Upsilon \vdash A \multimap B}\ \multimap\textit{-right}$$

F Rules

$$\frac{\Upsilon, X \vdash A}{\Upsilon, FX \vdash A}\ F\textit{-left} \qquad \frac{\Theta \vdash X}{\Theta \vdash FX}\ F\textit{-right}$$

G Rules

$$\frac{\Upsilon, B \vdash A}{\Upsilon, GB \vdash A}\ G\textit{-left} \qquad \frac{\Theta \vdash A}{\Theta \vdash GA}\ G\textit{-right}$$

Figure 5: Parsimonious presentation of LNL logic

sight to be valid in the semantics given by Jacobs, though it would actually cause the conventional witness property to fail and thus prevent the interpretation of $\rightarrow$.

In any case, once one has observed the importance of the previous lemma, it seems rather more natural to interpret sequents with non-linear consequents as morphisms in the cartesian closed category in the first place, as we have done here.

3.2 Natural Deduction and LNL Terms

In this section we will present a natural deduction formulation of LNL logic and a procedure for normalising deductions. By applying the Curry-Howard correspondence, we then derive a term assignment system and a set of reduction rules, i.e. a mixed linear/non-linear lambda calculus.

3.2.1 The Natural Deduction Rules

The usual way to present natural deductions is as trees, each of which has assumptions at the leaves and a conclusion at the root. Whilst such a presentation of LNL logic is possible, we shall just give a 'sequent style' natural deduction system. The reason for this is that, mainly for reasons to do with term assignment, we wish to give the natural deduction analogue of LNL^a (rather than LNL^m), and the mixture of shared and distinct assumption sets which this involves is more clearly shown in the sequent style presentation. The natural deduction system is characterised by having introduction and elimination rules for each logical connective, rather than the left and right rules of the sequent calculus.

The natural deduction rules are shown in Figure 6. We will call this inference system ND.

Note that

- The elimination rule for F, like that for $\otimes$, builds in some substitution.
- The introduction and elimination rules for G are exact inverses.
- The G-introduction rule corresponds to promotion in ordinary linear logic. The restriction that the assumptions in the premiss be all non-linear corresponds to the restriction on the promotion rule. We do not, however, need to build any substitution into the G-introduction rule.
- None of the natural deduction rules split into $\mathcal{L}$ and $\mathcal{C}$ versions, so the natural deduction formulation is automatically 'parsimonious'.

An important fact about the natural deduction system is that it satisfies the *substitution property.* This essentially means that the cut rules from the sequent calculus presentation are admissible in natural deduction:

Lemma 28 *The following three rules are admissible in the natural deduction formulation of LNL logic:*

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Theta; \Gamma \vdash_{\mathcal{L}} A}{\Theta, \Gamma \vdash_{\mathcal{L}} A}\ \mathcal{CL}\textit{-subs} \qquad \frac{\Theta \vdash_{\mathcal{C}} X \qquad X, \Theta \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\textit{-subs}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Theta; A, \Delta \vdash_{\mathcal{L}} B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} B}\ \mathcal{LL}\textit{-subs}$$

$$\Theta; A \vdash_{\mathcal{L}} A \qquad\qquad \Theta, X \vdash_{\mathcal{C}} X$$

$$\frac{\Theta \vdash_{\mathcal{C}} X \qquad \Theta \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-intro} \qquad\qquad \frac{}{\Theta \vdash_{\mathcal{C}} 1}\ 1\textit{-intro}$$

$$\frac{\Theta \vdash_{\mathcal{C}} X \times Y}{\Theta \vdash_{\mathcal{C}} X}\ \times\textit{-elim1} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} X \times Y}{\Theta \vdash_{\mathcal{C}} Y}\ \times\textit{-elim2}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \qquad \Theta; \Delta \vdash_{\mathcal{L}} B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} A \otimes B}\ \otimes\textit{-intro} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \otimes B \qquad \Theta; \Delta, A, B \vdash_{\mathcal{L}} C}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} C}\ \otimes\textit{-elim}$$

$$\frac{}{\Theta \vdash_{\mathcal{L}} I}\ I\textit{-intro} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} I \qquad \Theta; \Delta \vdash_{\mathcal{L}} A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} A}\ I\textit{-elim}$$

$$\frac{\Theta, X \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} X \to Y}\ \to\textit{-intro} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} X \to Y \qquad \Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{C}} Y}\ \to\textit{-elim}$$

$$\frac{\Theta; \Gamma, A \vdash_{\mathcal{L}} B}{\Theta; \Gamma \vdash_{\mathcal{L}} A \multimap B}\ \multimap\textit{-intro} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} A \multimap B \qquad \Theta; \Delta \vdash_{\mathcal{L}} A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} B}\ \multimap\textit{-elim}$$

$$\frac{\Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{L}} FX}\ F\textit{-intro} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} FX \qquad \Theta, X; \Delta \vdash_{\mathcal{L}} A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} A}\ F\textit{-elim}$$

$$\frac{\Theta \vdash_{\mathcal{L}} A}{\Theta \vdash_{\mathcal{C}} GA}\ G\textit{-intro} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} GA}{\Theta \vdash_{\mathcal{L}} A}\ G\textit{-elim}$$

Figure 6: Natural deduction presentation of LNL logic

Proof. Induction on the derivation of the right-hand premiss. □

We will also need:

Lemma 29 *The weakening rules of the sequent calculus are admissible in the natural deduction system.* □

Using the previous lemmas, we can establish a connection between the sequent calculus and natural deduction formulations of the logic:

Proposition 30 *There are functions $\mathcal{S} : ND \to \mathrm{LNL}^a$ and $\mathcal{N} : \mathrm{LNL}^a \to ND$ which map a proof in one system to a proof of the same sequent in the other system. Furthermore, for any natural deduction Σ, $\mathcal{N}\mathcal{S}(\Sigma)$ is equal to Σ.*

Proof. This is all fairly obvious induction. We start by looking at the definition of $\mathcal{S}$:

- The axioms map to axioms.
- Introduction rules become right rules. For example,

$$\dfrac{\begin{array}{c}\Sigma_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta \vdash_{\mathcal{C}} Y\end{array}}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-intro}$$

  maps to

$$\dfrac{\begin{array}{c}\mathcal{S}(\Sigma_1)\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\mathcal{S}(\Sigma_2)\\ \Theta \vdash_{\mathcal{C}} Y\end{array}}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-right}$$

- Elimination rules become combinations of left rules with cuts. For example

$$\dfrac{\begin{array}{c}\Sigma_1\\ \Theta; \Gamma \vdash_{\mathcal{L}} A \otimes B\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta; \Delta, A, B \vdash_{\mathcal{L}} C\end{array}}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} C}\ \otimes\textit{-elim}$$

  maps to

$$\dfrac{\begin{array}{c}\mathcal{S}(\Sigma_1)\\ \Theta; \Gamma \vdash_{\mathcal{L}} A \otimes B\end{array} \qquad \dfrac{\begin{array}{c}\mathcal{S}(\Sigma_2)\\ \Theta; \Delta, A, B \vdash_{\mathcal{L}} C\end{array}}{\Theta; \Delta, A \otimes B \vdash_{\mathcal{L}} C}\ \otimes\textit{-left}}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} C}\ \mathcal{L}\mathcal{L}\textit{-cut}$$

The function $\mathcal{N}$ mapping sequent proofs to natural deductions is also fairly straightforward:

- Axioms are translated by axioms.
- Instances of cut rules are translated by the appropriate admissible substitution rules (Lemma 28).
- Right rules become introductions.

- Left rules become eliminations modulo some structural fiddling. For example:

$$\dfrac{\begin{array}{c}\Sigma\\ \Theta, X \vdash_{\mathcal{C}} Z\end{array}}{\Theta, X \times Y \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\times\text{-}\mathit{left1}$$

maps to

$$\dfrac{\dfrac{\Theta, X \times Y \vdash_{\mathcal{C}} X \times Y}{\Theta, X \times Y \vdash_{\mathcal{C}} X}\ \times\text{-}\mathit{elim1} \qquad \dfrac{\begin{array}{c}\mathcal{N}(\Sigma)\\ \Theta, X \vdash_{\mathcal{C}} Z\end{array}}{\Theta, X \times Y, X \vdash_{\mathcal{C}} Z}\ \mathcal{C}\text{-}\mathit{weakening}}{\Theta, X \times Y \vdash_{\mathcal{C}} Z}\ \mathcal{CC}\text{-}\mathit{subs}$$

Similarly, the proof

$$\dfrac{\begin{array}{c}\Sigma\\ \Theta, X; \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta; FX, \Gamma \vdash_{\mathcal{L}} A}\ F\text{-}\mathit{left}$$

maps to

$$\dfrac{\Theta; FX \vdash_{\mathcal{L}} FX \qquad \begin{array}{c}\mathcal{N}(\Sigma)\\ \Theta, X; \Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta; FX, \Gamma \vdash_{\mathcal{L}} A}\ F\text{-}\mathit{elim}$$

That $\mathcal{N} \circ \mathcal{S}$ is the identity can then be verified by induction. The proof is most easily obtained with the assistance of the term calculus which we are about to introduce. □

Clearly, there is an categorical interpretation of natural deduction proofs in any LNL model. One way to obtain the interpretation is to apply the $\mathcal{S}$ translation and then the interpretation of sequent proofs which we gave earlier, but it is fairly easy to write down directly (and one does indeed get the same answer!). Some of the clauses of this direct interpretation are shown in Figure 7.

3.2.2 Term Assignment

Just as the simply typed lambda calculus arises as a notation for proofs in a natural deduction system for ordinary intuitionistic propositional logic, we can annotate proofs in our system ND to derive a mixed linear and non-linear term calculus. The term assignment system is shown in Figure 8. We use a, b, c for linear variables, e, f, g, h for linear terms, w, x, y, z for non-linear variables and s, t, u, v for non-linear terms. Distinct linear contexts are assumed to mention disjoint sets of linear variable names.

As should be obvious, the two of the forms of let and the two kinds of λ are variable-binding constructs. We refrain from giving a detailed definition of free and bound variables and capture-avoiding substitution as the reader should be able to work them out without difficulty.

Lemma 31 *Terms encode deductions uniquely – if $\Theta \vdash_{\mathcal{C}} s{:}\, X$ or $\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\, A$ is derivable then the derivation is uniquely determined by the term.* □

Lemma 32 *If $\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\, A$ is derivable then each linear variable in the context (i.e. each variable in Γ) has exactly one free occurence in the term e.*

Non-linear variables in the context may appear any number of times (including zero) in a well-typed term. □

$$\frac{X_1 \times \cdots \times X_n \xrightarrow{s} Y}{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{m} F(X_1 \times \cdots \times X_n) \xrightarrow{F(s)} FY}\ F\textit{-intro}$$

$$\frac{F\Theta \otimes \Gamma \xrightarrow{e} FX \qquad F\Theta \otimes FX \otimes \Delta \xrightarrow{f} A}{F\Theta \otimes \Gamma \otimes \Delta \xrightarrow{dup \otimes 1 \otimes 1} F\Theta \otimes F\Theta \otimes \Gamma \otimes \Delta \xrightarrow{1 \otimes e \otimes 1} F\Theta \otimes FX \otimes \Delta \xrightarrow{f} A}\ F\textit{-elim}$$

where if $\Theta = X_1, \ldots, X_n$, $dup : F\Theta \to F\Theta \otimes F\Theta$ is

$$\bigotimes_i FX_i \xrightarrow{m} F\left(\prod_i X_i\right) \xrightarrow{F(\Delta)} F\left(\left(\prod_i X_i\right) \times \left(\prod_i X_i\right)\right) \xrightarrow{m^{-1}} \left(\bigotimes_i FX_i\right) \otimes \left(\bigotimes_i FX_i\right)$$

$$\frac{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{e} A}{X_1 \times \cdots \times X_n \xrightarrow{\eta} GF(X_1 \times \cdots \times X_n) \xrightarrow{G(m^{-1})} G(FX_1 \otimes \cdots \otimes FX_n) \xrightarrow{G(e)} GA}\ G\textit{-intro}$$

$$\frac{X_1 \times \cdots \times X_n \xrightarrow{s} GB}{FX_1 \otimes \cdots \otimes FX_n \xrightarrow{m} F(X_1 \times \cdots \times X_n) \xrightarrow{F(s)} FGB \xrightarrow{\varepsilon} B}\ G\textit{-elim}$$

Figure 7: Categorical interpretation of natural deductions (sketch)

Lemma 28 can now be restated to include the terms:

Lemma 33 (Substitution) *The following three rules are admissible in the LNL term calculus:*

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X \qquad x\colon X, \Theta; \Gamma \vdash_{\mathcal{L}} e\colon A}{\Theta, \Gamma \vdash_{\mathcal{L}} e[s/x]\colon A}\ \mathcal{CL}\textit{-subs}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s :: X \qquad x\colon X, \Theta \vdash_{\mathcal{C}} t\colon Y}{\Theta \vdash_{\mathcal{C}} t[s/x]\colon Y}\ \mathcal{CC}\textit{-subs}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \qquad \Theta; a\colon A, \Delta \vdash_{\mathcal{L}} f\colon B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} f[e/a]\colon B}\ \mathcal{LL}\textit{-subs}$$

□

It should be noted that the term calculus contains the usual simply typed lambda calculus as a subsystem. Note also that, in contrast to the term assignment system for intuitionistic linear logic, there is no explicit syntax for weakening or contraction in the calculus.

3.2.3 Normalisation and Reduction

We now look at the process of normalisation on natural deduction proofs in our logic, and at the associated reductions on terms. The fundamental kind of normalisation step is the removal of a 'detour' in the deduction, which consists of an introduction rule immediately followed by the corresponding elimination. There is thus a normalisation step for each intro/elim pair, and we consider each of these in turn:

$$\Theta; a{:}\,A \vdash_{\mathcal{L}} a{:}\,A \qquad\qquad \Theta, x{:}\,X \vdash_{\mathcal{C}} x{:}\,X$$

$$\frac{\Theta \vdash_{\mathcal{C}} s{:}\,X \qquad \Theta \vdash_{\mathcal{C}} t{:}\,Y}{\Theta \vdash_{\mathcal{C}} (s,t){:}\,X \times Y} \qquad\qquad \frac{}{\Theta \vdash_{\mathcal{C}} (){:}\,1}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s{:}\,X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{fst}(s){:}\,X} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} s{:}\,X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{snd}(s){:}\,Y}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\,A \qquad \Theta; \Delta \vdash_{\mathcal{L}} f{:}\,B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e \otimes f{:}\,A \otimes B} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\,A \otimes B \qquad \Theta; \Delta, a{:}\,A, b{:}\,B \vdash_{\mathcal{L}} f{:}\,C}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f{:}\,C}$$

$$\frac{}{\Theta \vdash_{\mathcal{L}} *{:}\,I} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\,I \qquad \Theta; \Delta \vdash_{\mathcal{L}} f{:}\,A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ * = e\ \mathsf{in}\ f{:}\,A}$$

$$\frac{\Theta, x{:}\,X \vdash_{\mathcal{C}} s{:}\,Y}{\Theta \vdash_{\mathcal{C}} (\lambda x{:}\,X.s){:}\,X \to Y} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} s{:}\,X \to Y \qquad \Theta \vdash_{\mathcal{C}} t{:}\,X}{\Theta \vdash_{\mathcal{C}} s\,t{:}\,Y}$$

$$\frac{\Theta; \Gamma, a{:}\,A \vdash_{\mathcal{L}} e{:}\,B}{\Theta; \Gamma \vdash_{\mathcal{L}} (\lambda a{:}\,A.e){:}\,A \multimap B} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\,A \multimap B \qquad \Theta; \Delta \vdash_{\mathcal{L}} f{:}\,A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e\,f{:}\,B}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s{:}\,X}{\Theta \vdash_{\mathcal{L}} \mathsf{F}(s){:}\,FX} \qquad\qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e{:}\,FX \qquad \Theta, x{:}\,X; \Delta \vdash_{\mathcal{L}} f{:}\,A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ \mathsf{F}(x) = e\ \mathsf{in}\ f{:}\,A}$$

$$\frac{\Theta \vdash_{\mathcal{L}} e{:}\,A}{\Theta \vdash_{\mathcal{C}} \mathsf{G}(e){:}\,GA} \qquad\qquad \frac{\Theta \vdash_{\mathcal{C}} s{:}\,GA}{\Theta \vdash_{\mathcal{L}} \mathsf{derelict}(s){:}\,A}$$

Figure 8: LNL term assignment system

- The deduction

$$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta \vdash_{\mathcal{C}} Y\end{array}}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-intro}}{\Theta \vdash_{\mathcal{C}} X}\ \times\textit{-elim1}$$

  normalises to

$$\begin{array}{c}\Sigma_1\\ \Theta \vdash_{\mathcal{C}} X\end{array}$$

- The case of $\times$-intro followed by $\times$-elim2 is similar.
- The deduction

$$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \Theta;\Gamma_1 \vdash_{\mathcal{L}} A\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta;\Gamma_2 \vdash_{\mathcal{L}} B\end{array}}{\Theta;\Gamma_1,\Gamma_2 \vdash_{\mathcal{L}} A \otimes B}\ \otimes\textit{-intro} \qquad \begin{array}{c}\Sigma_3\\ \Theta;\Gamma_3, A, B \vdash_{\mathcal{L}} C\end{array}}{\Theta;\Gamma_1,\Gamma_2,\Gamma_3 \vdash_{\mathcal{L}} C}\ \otimes\textit{-elim}$$

  normalises to the deduction denoted by

$$\dfrac{\begin{array}{c}\Sigma_1\\ \Theta;\Gamma_1 \vdash_{\mathcal{L}} A\end{array} \qquad \dfrac{\begin{array}{c}\Sigma_2\\ \Theta;\Gamma_2 \vdash_{\mathcal{L}} B\end{array} \qquad \begin{array}{c}\Sigma_3\\ \Theta;\Gamma_3, A, B \vdash_{\mathcal{L}} C\end{array}}{\Theta;\Gamma_2,\Gamma_3, A \vdash_{\mathcal{L}} C}\ \mathcal{LL}\textit{-subs}}{\Theta;\Gamma_1,\Gamma_2,\Gamma_3 \vdash_{\mathcal{L}} C}\ \mathcal{LL}\textit{-subs}$$

  Note that this is not as asymmetric as it appears – the subs rule is only an *admissible* rule, and the actual deduction intended by the above shorthand is exactly the same as the one obtained by substituting the derivation of A first.

- $$\dfrac{\dfrac{}{\Theta \vdash_{\mathcal{L}} I}\ I\textit{-intro} \qquad \begin{array}{c}\Sigma\\ \Theta;\Gamma \vdash_{\mathcal{L}} A\end{array}}{\Theta;\Gamma \vdash_{\mathcal{L}} A}\ I\textit{-elim}$$

  normalises to

$$\begin{array}{c}\Sigma\\ \Theta;\Gamma \vdash_{\mathcal{L}} A\end{array}$$

- $$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \Theta, X \vdash_{\mathcal{C}} Y\end{array}}{\Theta \vdash_{\mathcal{C}} X \to Y}\ \to\textit{-intro} \qquad \begin{array}{c}\Sigma_2\\ \Theta \vdash_{\mathcal{C}} X\end{array}}{\Theta \vdash_{\mathcal{C}} Y}\ \to\textit{-elim}$$

  normalises to

$$\dfrac{\begin{array}{c}\Sigma_2\\ \Theta \vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Sigma_1\\ \Theta, X \vdash_{\mathcal{C}} Y\end{array}}{\Theta \vdash_{\mathcal{C}} Y}\ \mathcal{CC}\textit{-subs}$$

- $$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \Theta;\Gamma_1,A\vdash_{\mathcal{L}} B\end{array}}{\Theta;\Gamma_1\vdash_{\mathcal{L}} A\multimap B}\ \multimap\text{-}intro \qquad \begin{array}{c}\Sigma_2\\ \Theta;\Gamma_2\vdash_{\mathcal{L}} A\end{array}}{\Theta;\Gamma_1,\Gamma_2\vdash_{\mathcal{L}} B}\ \multimap\text{-}elim$$

  normalises to

  $$\dfrac{\begin{array}{c}\Sigma_2\\ \Theta;\Gamma_2\vdash_{\mathcal{L}} A\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta;\Gamma_1,A\vdash_{\mathcal{L}} B\end{array}}{\Theta;\Gamma_1,\Gamma_2\vdash_{\mathcal{L}} B}\ \mathcal{LL}\text{-}subs$$

- $$\dfrac{\dfrac{\begin{array}{c}\Sigma_1\\ \Theta\vdash_{\mathcal{C}} X\end{array}}{\Theta\vdash_{\mathcal{L}} FX}\ F\text{-}intro \qquad \begin{array}{c}\Sigma_2\\ \Theta,X;\Gamma\vdash_{\mathcal{L}} A\end{array}}{\Theta;\Gamma\vdash_{\mathcal{L}} A}\ F\text{-}elim$$

  normalises to

  $$\dfrac{\begin{array}{c}\Sigma_1\\ \Theta\vdash_{\mathcal{C}} X\end{array} \qquad \begin{array}{c}\Sigma_2\\ \Theta,X;\Gamma\vdash_{\mathcal{L}} A\end{array}}{\Theta;\Gamma\vdash_{\mathcal{L}} A}\ \mathcal{CL}\text{-}subs$$

- $$\dfrac{\dfrac{\begin{array}{c}\Sigma\\ \Theta\vdash_{\mathcal{L}} A\end{array}}{\Theta\vdash_{\mathcal{C}} GA}\ G\text{-}intro}{\Theta\vdash_{\mathcal{L}} A}\ G\text{-}elim$$

  normalises to

  $$\begin{array}{c}\Sigma\\ \Theta\vdash_{\mathcal{L}} A\end{array}$$

The normalisation steps on natural deductions induce β-reductions on the associated terms. These are shown in Figure 9.

As often happpens with natural deduction systems, there is also a secondary class of reductions – the *commuting conversions*, which are caused by rules which have a 'parasitic formula'. In LNL logic there are three such rules, the elimination rules for $\otimes$, I and F. Such a rule can artificially prevent an introduction/elimination pair from reacting unless we explicitly add certain commutations. The basic pattern is that a natural deduction looking like

$$\dfrac{\dfrac{\vdots \qquad \begin{array}{c}\vdots\\ C\end{array}}{C}\ r}{D}\ any\text{-}elim$$

$$
\begin{aligned}
\mathsf{fst}(s,t) &\rightarrow_\beta s \\
\mathsf{snd}(s,t) &\rightarrow_\beta t \\
\mathsf{let}\ a \otimes b = e \otimes f\ \mathsf{in}\ g &\rightarrow_\beta g[e/a, f/b] \\
\mathsf{let}\ * = *\ \mathsf{in}\ e &\rightarrow_\beta e \\
(\lambda x\colon X.s)\ t &\rightarrow_\beta s[t/x] \\
(\lambda a\colon A.e)\ f &\rightarrow_\beta e[f/a] \\
\mathsf{let}\ \mathsf{F}(x) = \mathsf{F}(s)\ \mathsf{in}\ e &\rightarrow_\beta e[s/x] \\
\mathsf{derelict}(\mathsf{G}(e)) &\rightarrow_\beta e
\end{aligned}
$$

Figure 9: Term calculus β-reductions

where r is a rule with parasitic formula C and *any*-elim is any elimination rule, commutes to

$$
\cfrac{\vdots \qquad \cfrac{\begin{array}{c}\vdots\\ C\end{array}}{D}\ \textit{any-elim}}{D}\ r
$$

In the case of LNL logic, the parasitic formula C is always linear, so *any*-elim can only be the elimination of one of the four linear connectives $\otimes$, I, $\multimap$ and F. This means that we have $3 \times 4 = 12$ commuting conversions. Rather than give the conversions explicitly on proofs, we merely list the induced commutations on terms in Figure 10. The proofs are easily reconstructed by Lemma 31.[11]

The reduction relations $\rightarrow_\beta$ and $\rightarrow_c$ are defined as the precongruence closures of the clauses given in Figures 9 and 10 respectively. We write $\rightarrow_{\beta,c}$ for $\rightarrow_\beta \cup \rightarrow_c$. As we have avoided all mention of raw terms (sometimes also known as preterms), the following is almost a complete triviality:

Proposition 34 (Subject Reduction) *Reduction is well-typed:*

- *If* $\Theta;\Gamma \vdash_{\mathcal{L}} e\colon A$ *and* $e \rightarrow_{\beta,c} e'$ *then* $\Theta;\Gamma \vdash_{\mathcal{L}} e'\colon A$.
- *If* $\Theta \vdash_{\mathcal{C}} s\colon X$ *and* $s \rightarrow_{\beta,c} s'$ *then* $\Theta \vdash_{\mathcal{C}} s'\colon X$.

□

Somewhat more interesting is the fact that when a term is reduced its categorical interpretation remains unchanged (cf. Theorem 25).

Theorem 35 *Both the β-reductions and the commuting conversions are soundly modelled by the interpretation of the natural deduction system in any LNL model.*

[11] A small technicality is that the conversion for F-elim against itself is not an entirely local rewrite, but uses the admissible weakening rule. This would not be the case for an entirely multiplicative formulation of the natural deduction system, however.

$$\mathsf{let}\ a \otimes b = (\mathsf{let}\ * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ * = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$

$$\mathsf{let}\ * = (\mathsf{let}\ * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ * = e\ \mathsf{in}\ (\mathsf{let}\ * = f\ \mathsf{in}\ g)$$

$$(\mathsf{let}\ * = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let}\ * = e\ \mathsf{in}\ (f\ g)$$

$$\mathsf{let}\ F(x) = (\mathsf{let}\ * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ * = e\ \mathsf{in}\ (\mathsf{let}\ F(x) = f\ \mathsf{in}\ g)$$

$$\mathsf{let}\ a \otimes b = (\mathsf{let}\ c \otimes d = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ c \otimes d = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$

$$\mathsf{let}\ * = (\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (\mathsf{let}\ * = f\ \mathsf{in}\ g)$$

$$(\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (f\ g)$$

$$\mathsf{let}\ F(x) = (\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (\mathsf{let}\ F(x) = f\ \mathsf{in}\ g)$$

$$\mathsf{let}\ a \otimes b = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$

$$\mathsf{let}\ * = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let}\ * = f\ \mathsf{in}\ g)$$

$$(\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (f\ g)$$

$$\mathsf{let}\ F(y) = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let}\ F(y) = f\ \mathsf{in}\ g)$$

Figure 10: Term calculus commuting conversions

- *If* $\Theta;\Gamma \vdash_{\mathcal{L}} e\colon A$ *and* $e \to_{\beta,c} e'$ *then*
$$[\![\Theta;\Gamma \vdash_{\mathcal{L}} e\colon A]\!] \;=\; [\![\Theta;\Gamma \vdash_{\mathcal{L}} e'\colon A]\!]$$
- *If* $\Theta \vdash_{\mathcal{C}} s\colon X$ *and* $s \to_{\beta,c} s'$ *then*
$$[\![\Theta \vdash_{\mathcal{C}} s\colon X]\!] \;=\; [\![\Theta \vdash_{\mathcal{C}} s'\colon X]\!]$$

□

3.3 Translations

We already know from Section 2 that LNL models and linear categories are equivalent. What we have not yet done is show any direct relationship between provability (or proofs) in LNL logic and in ordinary ILL. Such questions could be approached from the semantic point of view if we had a completeness result for LNL models, but for the moment we shall just argue proof-theoretically.[12] In this section we will relate LNL logic to ILL, restricting attention to the natural deduction formulations (equivalently, the term assignment systems). Comparable translations for the sequent calculus are straightforward to obtain, but omitted.

We begin by recalling in Figure 11 the linear term calculus (LTC) which corresponds to the natural deduction presentation of ILL [BBHdP92].

[12] I conjecture that the natural completeness theorem is true, and see no particular reason why the proof should not be by a standard term-model construction – I just haven't done it yet. The first step is to list all the term equalites given by the category theory. These comprise the β, c equalities from the proof theory together with a number of naturality and η (uniqueness) equalities.

$$a : A \vdash a : A \ (\mathcal{A}x)$$

$$\frac{\Gamma, a : A \vdash e : B}{\Gamma \vdash (\lambda a{:}\, A.e) : A \multimap B} \ (\multimap_{\mathcal{I}}) \qquad \frac{\Gamma \vdash e : A \multimap B \qquad \Delta \vdash f : A}{\Gamma, \Delta \vdash ef : B} \ (\multimap_{\mathcal{E}})$$

$$\vdash * : I \ (I_{\mathcal{I}}) \qquad \frac{\Gamma \vdash e : A \qquad \Delta \vdash f : I}{\Gamma, \Delta \vdash \mathsf{let}\ f = * \ \mathsf{in}\ e : A} \ (I_{\mathcal{E}})$$

$$\frac{\Gamma \vdash e : A \qquad \Delta \vdash f : B}{\Gamma, \Delta \vdash e \otimes f : A \otimes B} \ (\otimes_{\mathcal{I}}) \qquad \frac{\Gamma \vdash e : A \otimes B \qquad \Delta, a : A, b : B \vdash f : C}{\Gamma, \Delta \vdash \mathsf{let}\ e = a \otimes b\ \mathsf{in}\ f : C} \ (\otimes_{\mathcal{E}})$$

$$\frac{\Delta_1 \vdash e_1 :!A_1 \ \cdots \ \Delta_n \vdash e_n :!A_n \qquad a_1 :!A_1, \ldots, a_n :!A_n \vdash f : B}{\Delta_1, \ldots, \Delta_n \vdash \mathsf{promote}\ e_1, \ldots, e_n\ \mathsf{for}\ a_1, \ldots, a_n\ \mathsf{in}\ f :!B} \ \mathit{Promotion}$$

$$\frac{\Gamma \vdash e :!A \qquad \Delta, a :!A, b :!A \vdash f : B}{\Gamma, \Delta \vdash \mathsf{copy}\ e\ \mathsf{as}\ a, b\ \mathsf{in}\ f : B} \ \mathit{Contraction}$$

$$\frac{\Gamma \vdash e :!A \qquad \Delta \vdash f : B}{\Gamma, \Delta \vdash \mathsf{discard}\ e\ \mathsf{in}\ f : B} \ \mathit{Weakening} \qquad \frac{\Gamma \vdash e :!A}{\Gamma \vdash \mathsf{derelict}(e) : A} \ \mathit{Dereliction}$$

Figure 11: The linear term calculus (LTC)

3.3.1 ILL to LNL Logic

The translation of ILL into LNL logic is not particularly difficult. If A is an ILL proposition, define the linear LNL proposition A° inductively as follows:

$$\begin{array}{rcl} A_0^\circ & = & A_0 \ (A_0 \text{ atomic}) \\ (A \otimes B)^\circ & = & A^\circ \otimes B^\circ \\ (A \multimap B)^\circ & = & A^\circ \multimap B^\circ \\ I^\circ & = & I \\ (!A)^\circ & = & FG(A^\circ) \end{array}$$

Theorem 36 *If $\Gamma \vdash e{:}\, A$ in ILL, then there is an e° such that $\Gamma^\circ \vdash_{\mathcal{L}} e^\circ{:}\, A^\circ$.*

Proof. This is done by induction on the derivation in ILL (that is, on the structure of the linear term e). The exponential-free rules are completely straightforward, so we just detail the translations of the one introduction and three elimination rules for !. The easiest way to present the translations is just to give the translation from terms to terms, as that determines the translation of proofs.

Promotion The $(-)^\circ$ translation of

$$\mathsf{promote}\ e_1, \ldots, e_n\ \mathsf{for}\ a_1, \ldots, a_n\ \mathsf{in}\ f$$

is

$$\mathsf{let}\ F(y_1) = e_1^\circ\ \mathsf{in}\ \mathsf{let}\ F(y_2) = e_2^\circ\ \mathsf{in}\ \ldots FG(f^\circ[F(y_i)/a_i])$$

where the y_i are fresh. One might be tempted to simplify the translation to

$$FG(f^\circ[e_i/a_i])$$

but a moment's consideration reveals that this latter expression is not well-typed. It is interesting to note how the 'boxing' behaviour of the promotion rule is thus maintained by its translation into LNL logic, even though neither of the introduction rules for F and G themselves involve a change of variable names. Note also that the translation makes use of an admissible substitution rule.

Dereliction

$$(\mathsf{derelict}(e))^\circ \;=\; \mathsf{let}\ F(x) = e^\circ\ \mathsf{in}\ \mathsf{derelict}(x)$$

where x is fresh. (Which version of derelict() is meant should usually be clear from context.)

Weakening

$$(\mathsf{discard}\ e\ \mathsf{in}\ f)^\circ \;=\; \mathsf{let}\ F(x) = e^\circ\ \mathsf{in}\ f^\circ$$

where x is fresh.

Contraction

$$(\mathsf{copy}\ e\ \mathsf{as}\ a, b\ \mathsf{in}\ f)^\circ \;=\; \mathsf{let}\ F(x) = e^\circ\ \mathsf{in}\ f^\circ[F(x)/a, F(x)/b]$$

where x is fresh. Again, note that this is *not* just $f^\circ[e^\circ/a, e^\circ/b]$.

□

So we can translate ILL into the linear-only part of LNL logic in such a way that provability is preserved. That it is also reflected will follow from the translation from LNL logic to ILL which we are about to give.

3.3.2 LNL Logic to ILL

This direction is more interesting. The basic idea is to translate the linear part of LNL logic essentially unchanged and to translate the non-linear part by using a variant of the Girard translation. There is a small technicality concerning atomic propositions, in that LNL logic has both linear and non-linear atoms. We will thus translate into an ILL theory which has an extra atomic proposition A_{X_0} for each non-linear atomic proposition X_0 in the LNL theory. Given this, we can define the ILL proposition A^* or X^* for each LNL proposition A or X inductively as follows

$$\begin{array}{rcl}
A_0^* & = & A_0\ (A_0\ \text{atomic}) \\
(A \otimes B)^* & = & A^* \otimes B^* \\
I^* & = & I \\
(A \multimap B)^* & = & A^* \multimap B^* \\
(FX)^* & = & !(X^*) \\
X_0^* & = & A_{X_0}\ (X_0\ \text{atomic}) \\
(X \times Y)^* & = & !(X^*) \otimes !(Y^*) \\
1^* & = & I \\
(X \to Y)^* & = & !(X^*) \multimap Y^* \\
(GA)^* & = & A^*
\end{array}$$

Note that what is usually called 'the Girard translation' of IL to ILL uses the & connective ('with') to translate conjunction in IL, but we have not done this as we are dealing only with the multiplicative fragment of ILL at the moment. Were we to include additives, then obviously an alternative translation would be possible.

Theorem 37

1. *If* $\Theta \vdash_{\mathcal{C}} s\colon X$ *in LNL logic, then there is an LTC term* s^* *such that*
$$!\Theta^* \vdash s^*\colon X^*$$

2. *If* $\Theta;\Gamma \vdash_{\mathcal{L}} e\colon A$ *in LNL logic, then there is an LTC term* e^* *such that*
$$!\Theta^*, \Gamma^* \vdash e^*\colon A^*$$

Proof. This is proved by induction on the LNL derivation. The translation is slightly more complicated than it might be because of the fact that we have treated the conventional parts of LNL contexts in an additive way, and this does not easily match the purely multiplicative contexts used in ILL. For this reason, as well as the way in which the translation depends upon context, we will present this translation on derivations in the logic rather than on terms. The reader should be able easily to supply the missing term annotations so as to prove the theorem as stated.

- The translation of an axiom
$$\Theta, X \vdash_{\mathcal{C}} X$$
is
$$\dfrac{\dfrac{!X^* \vdash !X^*}{!X^* \vdash X^*}\ \textit{Dereliction}}{!\Theta^*, !X^* \vdash X^*}\ \textit{Weakening*}$$

- The translation of an axiom
$$\Theta; A \vdash_{\mathcal{L}} A$$
is
$$\dfrac{A^* \vdash A^*}{!\Theta^*, A^* \vdash A^*}\ \textit{Weakening*}$$

- If the LNL derivation ends in
$$\dfrac{\Theta \vdash_{\mathcal{C}} X \qquad \Theta \vdash_{\mathcal{C}} Y}{\Theta \vdash_{\mathcal{C}} X \times Y}\ \times\textit{-intro}$$
where $\Theta = Y_1, \ldots, Y_n$, then by induction we have ILL derivations of $!\Theta^* \vdash X^*$ and $!\Theta^* \vdash Y^*$ so that we can form the following (omitting rule names for reasons of space):
$$\dfrac{\dfrac{\dfrac{!Y_1^* \vdash !Y_1^* \cdots !Y_n^* \vdash !Y_n^* \qquad !\Theta^* \vdash X^*}{!Y_1^*, \ldots, !Y_n^* \vdash !X^*} \qquad \dfrac{!Y_1^* \vdash !Y_1^* \cdots !Y_n^* \vdash !Y_n^* \qquad !\Theta^* \vdash Y^*}{!Y_1^*, \ldots, !Y_n^* \vdash !Y^*}}{!\Theta^*, !\Theta^* \vdash !X^* \otimes !Y^*}}{!\Theta^* \vdash !X^* \otimes !Y^*}\ \textit{Contraction*}$$

- If the LNL derivation ends in

$$\frac{\Theta \vdash_{\mathcal{C}} X \to Y \qquad \Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{C}} Y} \to\text{-}elim$$

where $\Theta = Y_1, \ldots, Y_n$,then by induction we have ILL derivations of $!\Theta^* \vdash !X^* \multimap Y^*$ and $!\Theta^* \vdash X^*$ so we can form

$$\dfrac{!\Theta^* \vdash !X^* \multimap Y^* \qquad \dfrac{!Y_1^* \vdash !Y_1^* \quad \cdots \quad !Y_n^* \vdash !Y_n^* \qquad !\Theta^* \vdash X^*}{!Y_1^*, \ldots, !Y_n^* \vdash !X^*}\, Promotion}{\dfrac{!\Theta^*, !\Theta^* \vdash Y^*}{!\Theta^* \vdash Y^*}\, Contraction*} \multimap\text{-}elim$$

- If the LNL derivation ends with

$$\frac{\Theta \vdash_{\mathcal{C}} X}{\Theta \vdash_{\mathcal{L}} FX} \; F\text{-}intro$$

where $\Theta = Y_1, \ldots, Y_n$, then by induction there is a derivation of $!\Theta^* \vdash X^*$ so we can form

$$\frac{!Y_1^* \vdash !Y_1^* \quad \cdots \quad !Y_n^* \vdash !Y_n^* \qquad !\Theta^* \vdash X^*}{!Y_1^*, \ldots, !Y_n^* \vdash !X^*} \; Promotion$$

as required.

- Because of the fact that the G operator of LNL logic translates to nothing in ILL, the translation of both the G-introduction and G-elimination rules is the identity.

- The remaining rules are similar.

□

3.3.3 Further Results on the Translations

We now have translations both ways between LNL logic and ILL which preserve provability. There are probably other translations one could use, but these two seem to be the most natural.

Clearly, if one starts with a judgement of LNL logic and translates it to ILL and then back to LNL logic, one will not, in general, get back to the original judgement. This is because the final judgment will be in the purely linear fragment of LNL logic. Going around the cycle the other way, however, *is* the identity:

Theorem 38 *For any ILL judgement $\Gamma \vdash A$, the result of translating it into LNL logic and then back to ILL, viz.*

$$\Gamma^{\circ *} \vdash A^{\circ *}$$

is equal to the original judgement. As a corollary, the $(-)^\circ$ translation of ILL into LNL logic reflects, as well as preserves, provability in that

$$\Gamma \vdash A$$

is provable in ILL iff

$$\Gamma^\circ \vdash_{\mathcal{L}} A^\circ$$

is provable in LNL logic.

Proof. A trivial induction shows that for all ILL formulae A, $A = A^{\circ *}$, from which the first part of the theorem is immediate. The second part then follows by Theorems 36 and 37. □

A natural question is whether the previous result extends to proofs (rather than just to provability). It is certainly *not* the case that the result of mapping an ILL proof into LNL logic and back again is syntactically identical to the original proof, but it turns out that it is equal to the original proof under the equality on ILL proofs given by linear categories. The easiest way to state and prove this result is by using the linear term calculus:

Theorem 39 *If $\Gamma \vdash e{:}\, A$ in LTC, then not only is $\Gamma \vdash e^{\circ *}{:}\, A$ provable, but $e \approx e^{\circ *}$ where $\approx$ is the categorical equality relation on LTC terms given in [BBHdP92, Figure 11, page 40].*

Proof. This is an induction on the structure of e, but we omit the rather hairy details (which in any case would require the repetition of too much material from the earlier work). One first has to fill in the missing terms in the proof of Theorem 37 and then prove a number of lemmas concerning the way in which the $(-)^*$ translation behaves with respect to the admissible rules of weakening, contraction and substitution in the LNL term calculus (because these rules are used in defining the $(-)^\circ$ translation). The terms arising directly from the composite translation $(-)^{\circ *}$ are in general very large, but, given a certain amount of care over variable names, they simplify fairly easily. □

4 Conclusions and Further Work

We have given a new and intuitively appealing characterisation of categorical models of intuitionistic linear logic. We then used this presentation of the models as the basis for defining a new logic which unifies ordinary intuitionistic logic with intuitionistic linear logic. The natural deduction presentation of the new logic then gave, by the Curry-Howard correspondence, a mixed linear and non-linear lambda calculus.

At first sight, one might be tempted to regard LNL logic as "a logical atrocity without interest" [GLT89]. I hope, however, that I have shown that this is not the case. LNL logic has a very natural class of categorical models and a well-behaved proof theory in both its sequent calculus and natural deduction formulations. Given this, and the links with other research which were mentioned in the introduction, LNL logic certainly seems to merit further study.

On the theoretical side, much remains to be done. We have not proved a completeness theorem, nor have we proved that the LNL term calculus is strong normalising. The strong normalisation proof should be relatively easy to do via a translation argument like that which we have previously used for the linear term calculus [Ben95] and the computational lambda calculus [BBdP93]. It would be nice to have better (that is, less degenerate) examples of concrete models and one might well find such examples by looking at some of the categories arising in game semantics.

The connections between LNL logic and other work on LU and related systems should be looked at more closely. As well as the references cited in the introduction, Schellinx's work [Sch94] on decorating conventional proofs to give linear ones seems particularly interesting in this respect.

It should be noted that although the translations between ILL and LNL logic behave well with respect to equality, we have not claimed anything concerning the translations and reduction. I do not yet have any definitive results on whether, for example, reduction is preserved under either of the translations, but it certainly seems that any positive results will involve commuting conversions as well as the more conventional β rules.

There are also many obvious extensions to the system discussed here. The first of these is to consider the additive connectives on the linear side, and disjunction (coproducts) on the conventional side. We touched briefly on this in Section 2.2.3, but more remains to be done; this should be relatively straightforward, although, as we have already seen, there is some complication regarding coproducts in LNL models compared with coproducts in linear categories. Beyond that, one could consider adding inductive or coinductive datatypes or second-order quantification to the logic. This seems particularly worthwhile in the light of Plotkin's work on parametricity and recursion in a logic rather like ours [Plo93].

On the practical side, we should investigate whether or not the LNL term calculus lends itself more readily to efficient implementation than does the linear term calculus. The hope is that one can arrange an implementation with two memory spaces, corresponding to the two subsystems of LNL logic. The non-linear space would be garbage collected in the usual way, whereas the linear space would contain objects satisfying some useful memory invariant (such as having only one pointer to them at all times) which could be exploited to reduce the space usage of programs. Previous experience, however, shows that turning such intuitively plausible hopes into provably correct implementations is a non-trivial task.

5 Acknowledgements

I should like to thank my collaborators Gavin Bierman, Martin Hyland and Valeria de Paiva for innumerable discussions about logic, terms and categories. Thanks also to Gordon Plotkin for making a remark which led directly to the work described here, and to Barney Hilken for several useful conversations. John Reynolds's diagram macros and Paul Taylor's proof tree macros were used to typeset the paper.

References

[BBdP93] P. N. Benton, G. M. Bierman, and V. C. V. de Paiva. Computational types from a logical perspective. Preprint, 1993.

[BBHdP92] P. N. Benton, G. M. Bierman, J. M. E. Hyland, and V. C. V. de Paiva. Term assignment for intuitonistic linear logic. Technical Report 262, Computer Laboratory, University of Cambridge, August 1992.

[BBHdP93a] P. N. Benton, G. M. Bierman, J. M. E. Hyland, and V. C. V. de Paiva. Linear lambda calculus and categorical models revisited. In E. Börger et al., editor, *Selected Papers from Computer Science Logic '92*, volume 702 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[BBHdP93b] P. N. Benton, G. M. Bierman, J. M. E. Hyland, and V. C. V. de Paiva. A term calculus for intuitionistic linear logic. In M. Bezem and J. F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[Ben95] P. N. Benton. Strong normalisation for the linear term calculus. *Journal of Functional Programming*, 1995. To appear. Also available as Technical Report 305, University of Cambridge Computer Laboratory, July 1993.

[Bie94a] G. M. Bierman. On intuitionistic linear logic (revised version of PhD thesis). Technical Report 346, Computer Laboratory, University of Cambridge, August 1994.

[Bie94b] G. M. Bierman. What is a categorical model of intuitionistic linear logic? Technical Report 333, Computer Laboratory, University of Cambridge, April 1994.

[BS93] E. Barendsen and S. Smetsers. Conventional and uniqueness typing in graph rewrite systems. Technical Report CSI-R9328, Katholieke Universiteit Nijmegen, December 1993.

[CP90] R. L. Crole and A. M. Pitts. New foundations for fixpoint computations: FIX-hyperdoctines and the FIX-logic. Technical Report 204, Computer Laboratory, University of Cambridge, August 1990.

[Cro92] R. L. Crole. *Programming Metalogics with a Fixpoint Type*. PhD thesis, Computer Laboratory, University of Cambridge, February 1992. Available as Technical Report 247.

[FW87] J. Fairbairn and S. Wray. TIM: A simple, lazy abstract machine to execute supercombinators. In G. Kahn, editor, *Proceedings of the 1987 Conference on Functional Langauges and Computer Architecture*. LNCS 274, Springer-Verlag, 1987.

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

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

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[Jac93] B. Jacobs. Conventional and linear types in a logic of coalgebras. Preprint, University of Utrecht, April 1993.

[Koc71] A. Kock. Bilinearity and cartesian closed monads. *Math. Scand.*, 29:161–174, 1971.

[LM92] P. Lincoln and J. C. Mitchell. Operational aspects of linear lambda calculus. In *Proceedings of the 7th Annual Symposium on Logic in Computer Science*. IEEE, 1992.

[Mog89] E. Moggi. Computational lambda-calculus and monads. In *Proceedings of the 4th Annual Symposium on Logic in Computer Science, Asiloomar, CA*, pages 14–23, 1989.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[Plo93] G. D. Plotkin. Type theory and recursion (abstract). In *Proceedings of 8th Conference on Logic in Computer Science*. IEEE Computer Society Press, 1993.

[Pra92] V. Pratt. Message to the 'categories@mta.ca' electronic mailing list 27th November, 1992.

[Sch94] H. Schellinx. *The Noble Art of Linear Decorating*. PhD thesis, University of Amsterdam, 1994.

[See80] R. A. G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. In *Conference on Categories in Computer Science and Logic*, volume 92 of *AMS Contemporary Mathematics*, June 1980.

[Str72] R. Street. The formal theory of monads. *J. Pure and Applied Algebra*, 2:149–168, 1972.

[Wad90] P. Wadler. Linear types can change the world! In M. Broy and C. Jones, editors, *Programming Concepts and Methods*. North Holland, 1990.

[Wad92] P. Wadler. There's no substitute for linear logic (projector slides). In G. Winskel, editor, *Proceedings of the CLICS Workshop (Part I), March 1992, Aarhus, Denmark*, May 1992. Available as DAIMI PB-397-I Computer Science Department, Aarhus University.

[Wad93] P. Wadler. A taste of linear logic. In A. M. Borzyszkowski and S. Sokolowski, editors, *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, number 711 in Lecture Notes in Computer Science, pages 185–210, 1993.