Multiple Conclusion Intuitionistic Linear Logic and Cut Elimination

Harley Eades III and Valeria de Paiva

Abstract

Full Intuitionistic Linear Logic (FILL) was first introduced by Hyland and de Paiva as one of the results of their investigation into a categorical understanding of Gödel's dialectica interpretation. FILL went against current beliefs that it was not possible to incorporate all of the linear connectives, e.g. tensor, par, and implication, into an intuitionistic linear logic. They showed that it is natural to support all of the connectives given sequents that have multiple hypotheses and multiple conclusions. To enforce intuitionism de Paiva's original formalization of FILL used the well-known Dragalin restriction, forcing the implication right rule to have only a single conclusion in its premise, but Schellinx showed that this results in a failure of cut-elimination. To overcome this failure Hyland and de Paiva introduced a term assignment for FILL that eliminated the need for the strong restriction. The main idea was to first relax the restriction by assigning variables to each hypothesis and terms to each conclusion. Then when introducing an implication on the right enforcing that the variable annotating the hypothesis being discharged is only free in the term annotating the conclusion of the implication. Bierman showed that this formalization of FILL still did not enjoy cut-elimination, because of a flaw in the left rule for par. However, Bellin proposed an alternate left rule for par and gave an indirect proof showing that cut-elimination is restored. In this note we adopt Bellin's proposed rule and give a direct proof of cut-elimination. Additionally, we show that a categorical model of FILL in the basic dialectica category is also a LNL model of Benton.

1 Introduction

A commonly held belief during the early history of linear logic was that the linear-connective par could not be incorporated into an intuitionistic linear logic. This belief was challenged when de Paiva gave a categorical understanding of Gödel's dialectica interpretation in terms of dialectica categories [8, 7]. Upon setting out on her investigation she initially believed that dialectica categories would end up being a model of intuitionistic logic, but to her surprise they are actually models of intuitionistic linear logic, containing the linear connectives: tensor, implication, additives, and exponentials. She then improved her models to capture both FILL and CLL. Furthermore, unlike other models at that time the units did not collapse into a single object.

Armed with this semantic insight de Paiva gave the first formalization of Full Intuitionistic Linear Logic (FILL) [7]. FILL is a sequent calculus with multiple conclusions in addition to multiple hypotheses. Logics of this type go back to Gentzen's work on the sequent calculi LK and LJ, and Maehara's work on LJ' [14, 21]. The sequents in these types of logics usually have the form $\Gamma \vdash \Delta$ where Γ and Δ are multisets of formulas. Sequents such as these are read as "the conjunction of the formulas in Γ imply the disjunction of the formulas in Δ". For a brief, but more complete history of logics with multiple conclusions see the introduction to [9].

Gentzen showed that to obtain intuitionistic logic one could start with the logic LK and then place a cardinality restriction on the right-hand side of sequents, however, this is not the only means of enforcing intuitionism. Maehara showed that in the propositional case one could simply place the cardinality restriction on the premise of the implication right rule, and leave all of the other rules of LK unrestricted. This restriction is sometimes called the Dragalin restriction, as it appeared in his AMS textbook [10]. The classical implication right rule has the form:

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta}\ \textsc{impR}$$

By placing the Dragalin restriction on the previous rule we obtain:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \textsc{impR}$$

de Paiva's first formalization of FILL used the Dragalin restriction, see [7] p. 58, but Schellinx showed that this restriction has the unfortunate consequence of breaking cut-elimination [19].

Later, Hyland and de Paiva gave an alternative formalization of FILL with the intention of regaining cut-elimination [11]. This new formalization lifted the Dragalin restriction by decorating sequents with a term assignment. Hypotheses were assigned variables, and the conclusions were assigned terms. Then using these terms one can track the use of hypotheses throughout a derivation. They proposed a new implication right rule:

$$\frac{\Gamma, x : A \vdash t : B, \Delta \qquad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.t : A \multimap B, \Delta}\ \textsc{impR}$$

Intuitionism is enforced in this rule by requiring that the variable being dischanged, x, is potentially free in only one term annotating a conclusion. Unfortunately, this formalization did not enjoy cut-elimination either.

Bierman was able to give a counterexample to cut-elimination [4]. As Bierman explains the problem was with the left rule for par. The original rule was as follows:

$$\frac{\Gamma, x : A \vdash \Delta \qquad \Gamma', y : B \vdash \Delta'}{\Gamma, \Gamma', z : A \parr B \vdash \mathsf{let}\, z\, \mathsf{be}\, (x \parr -)\, \mathsf{in}\, \Delta \mid \mathsf{let}\, z\, \mathsf{be}\, (- \parr y)\, \mathsf{in}\, \Delta'}\ \textsc{parL}$$

In this rule the pattern variables x and y are bound in each term of Δ and Δ' respectively. Notice that the variable z becomes free in every term in Δ and Δ'. Bierman showed that this rule mixed with the restriction on implication right prevents the usual cut-elimination step that commutes cut with the left rule for par. The main idea behind the counterexample is that in the derivation before commuting the cut it is possible to discharge z using implication right, but after the cut is commuted past the left rule for par, the variable z becomes free in more than one conclusion, and thus, can no longer be discharged.

In the conclusion of Bierman's note he gives an alternate left rule for par that he attributes to Bellin. This new left-rule is as follows:

$$\frac{\Gamma, x : A \vdash \Delta \quad \Gamma', y : B \vdash \Delta'}{\Gamma, \Gamma', z : A \parr B \vdash \text{let-pat}\, z\, (x \parr -)\, \Delta \mid \text{let-pat}\, z\, (- \parr y)\, \Delta'}\ \textsc{Parl}$$

In this rule let-pat $z\,(x \parr -)\, t$ and let-pat $z\,(- \parr y)\, t'$ only let-bind z in t or t' if $x \in FV(t)$ or $y \in FV(t')$. Otherwise the terms are left unaltered. Bellin showed that by adopting this rule cut-elimination can be proven by reduction to the cut-elimination procedure for proof nets for multiplicative linear logic with the mix rule [1]. However, this is an indirect proof that requires the adoption of proof nets.

Contributions. In this paper our main contribution is to give a direct proof of cut-elimination for FILL with Bellin's proposed par-left rule (Section 3). In addition, we show that the categorical model of FILL called $\mathsf{Dial_2(Sets)}$, the basic dialectica category, is also a linear/non-linear model of Benton (Section 4).

Related Work. The first formalization of FILL with cut-elimination was due to Braüner and de Paiva [5]. Their formalization can be seen as a linear version of LK with a sophisticated meta-level dependency tracking system. A proof of a FILL sequent in their formalization amounts to a classical derivation, π, invariant in what they call the FILL property:

- The hypothesis discharged by an application of the implication right rule in π is a dependency of the conclusion of the implication being introduced.

They were able to show that their formalization is sound, complete, and enjoys cut-elimination. In favor of the term assignment formalization given here over Brauner and de Paiva's formalization is that the dependency tracking system complicates both the definition of the logic and its use. However, one might conjecture that their system is more fundamental and hence more generalizable.

de Paiva and Pereira used annotations on the sequents of LK to arrive at full intuitionistic logic (FIL) with multiple conclusion that enjoys cut-elimination [9]. They annotate hypothesis with natural number indices, and conclusions with finite sets of indices. The sets of indices on conclusions correspond to the collection of the hypotheses that the conclusion depends on. Then they have a similar property to that of Braüner and de Paiva's formalization. In fact, the dependency tracking system is very similar to this formalization, but the dependency tracking has been collapsed into the object language instead of being at the meta-level.

Clouston et al. give both a deep inference calculus and a display calculus for FILL that admits cut-elimination [6]. Both of these systems are refinements of a larger one called bi-intuitionistic linear logic (BiLL). This logic contains every logical connective of FILL with the addition of the exclusion (or subtraction) connective. This connective can be defined categorically as the left-adjoint to par. Thus, exclusion is the dual to implication. A positive aspect to this work is that the resulting systems are annotation free, but at a price of obscurity. Deep inference and display calculi are harder to understand, and their system requires FILL to be defined as a refinement of a system with additional connectives. We show in this paper that such a refinement is unnecessary. In addition, a term assignment system is closer to traditional logic than deep inference and display calculi, and it is closer, through the lens of the Curry-Howard-Lambek correspondence, to a type theoretic understanding of FILL.

2 Full Intuitionistic Linear Logic (FILL)

In this section we give a brief description of FILL. We first give the syntax of formulas, patterns, terms, and contexts. Following the syntax we define several meta-functions that will be used when defining the inference rules of the logic.

Definition 1. *The syntax for FILL is as follows:*

$$\begin{array}{rl}
\textit{(Formulas)} & A, B, C, D, E ::= \top \mid \perp \mid A \multimap B \mid A \otimes B \mid A \parr B \\
\textit{(Patterns)} & p ::= * \mid - \mid x \mid p_1 \otimes p_2 \mid p_1 \parr p_2 \\
\textit{(Terms)} & t, e ::= x \mid * \mid \circ \mid t_1 \otimes t_2 \mid t_1 \parr t_2 \mid \lambda x.t \mid \mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ e \mid t_1\ t_2 \\
\textit{(Left Contexts)} & \Gamma ::= \cdot \mid x : A \mid \Gamma_1, \Gamma_2 \\
\textit{(Right Contexts)} & \Delta ::= \cdot \mid t : A \mid \Delta_1, \Delta_2
\end{array}$$

The formulas of FILL are standard, but we denote the unit of tensor as $\top$ and the unit of par as $\perp$. Patterns are used to distinguish between the various let-expressions for tensor, par, and their units. There are three different let-expressions:

Tensor:	Par:	Tensor Unit:
$\mathsf{let}\ t\ \mathsf{be}\ p_1 \otimes p_2\ \mathsf{in}\ e$	$\mathsf{let}\ t\ \mathsf{be}\ p_1 \parr p_2\ \mathsf{in}\ e$	$\mathsf{let}\ t\ \mathsf{be}\ *\ \mathsf{in}\ e$

In addition, each of these will have their own equational rules, see Figure 2. The role each term plays in the overall logic will become clear after we introduce the inference rules.

At this point we introduce some syntax and meta-level functions that will be used in the definition of the inference rules for FILL. Left contexts are multisets of formulas labeled with a variable, and right contexts are multisets of formulas labeled with a term. We will often write $\Delta_1 \mid \Delta_2$ as syntactic sugar for Δ_1, Δ_2. The former should be read as "Δ_1 or Δ_2." We denote the usual capture-avoiding substitution by $[t/x]t'$, and its straightforward extension to right contexts as $[t/x]\Delta$. Similarly, we find it convenient to be able to do this style of extension for the let-binding as well.

Definition 2. *We extend let-binding terms to right contexts as follows:*

$$\begin{array}{l}
\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in} \cdot = \cdot \\
\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ (t' : A) = (\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ t') : A \\
\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ (\Delta_1 \mid \Delta_2) = (\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ \Delta_1) \mid (\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ \Delta_2)
\end{array}$$

$$\frac{}{x : A \vdash x : A}\ \textsc{Ax} \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma', y : A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta \mid [t/y]\Delta'}\ \textsc{Cut} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, x : \top \vdash \mathsf{let}\ x\ \mathsf{be}\ *\ \mathsf{in}\ \Delta}\ \textsc{Tl} \qquad \frac{}{\cdot \vdash * : \top}\ \textsc{Tr}$$

$$\frac{\Gamma, x : A, y : B \vdash \Delta}{\Gamma, z : A \otimes B \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ \Delta}\ \textsc{TenL} \qquad \frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma' \vdash f : B \mid \Delta'}{\Gamma, \Gamma' \vdash e \otimes f : A \otimes B \mid \Delta \mid \Delta'}\ \textsc{TenR} \qquad \frac{}{x : \perp \vdash \cdot}\ \textsc{Pl} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \circ : \perp \mid \Delta}\ \textsc{Pr}$$

$$\frac{\Gamma, x : A \vdash \Delta \quad \Gamma', y : B \vdash \Delta'}{\Gamma, \Gamma', z : A \parr B \vdash \text{let-pat}\ z\ (x \parr -)\ \Delta \mid \text{let-pat}\ z\ (- \parr y)\ \Delta'}\ \textsc{ParL} \qquad \frac{\Gamma \vdash \Delta \mid e : A \mid f : B \mid \Delta'}{\Gamma \vdash \Delta \mid e \parr f : A \parr B \mid \Delta'}\ \textsc{ParR}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma', x : B \vdash \Delta'}{\Gamma, y : A \multimap B, \Gamma' \vdash \Delta \mid [y\ e/x]\Delta'}\ \textsc{ImpL} \qquad \frac{\Gamma, x : A \vdash e : B \mid \Delta \quad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.e : A \multimap B \mid \Delta}\ \textsc{ImpR} \qquad \frac{\Gamma, x : A, y : B \vdash \Delta}{\Gamma, y : B, x : A \vdash \Delta}\ \textsc{ExL}$$

$$\frac{\Gamma \vdash \Delta_1 \mid t_1 : A \mid t_2 : B \mid \Delta_2}{\Gamma \vdash \Delta_1 \mid t_2 : B \mid t_1 : A \mid \Delta_2}\ \textsc{ExR}$$

Figure 1: Inference rules for FILL

$$\frac{y \notin \mathsf{FV}(t)}{t = [y/x]t}\ \textsc{Alpha} \qquad \frac{x \notin \mathsf{FV}(f)}{(\lambda x.f\ x) = f}\ \textsc{EtaFun} \qquad \frac{}{(\lambda x.e)\ e' = [e'/x]e}\ \textsc{BetaFun} \qquad \frac{}{\mathsf{let}\ *\ \mathsf{be}\ *\ \mathsf{in}\ e = e}\ \textsc{Eta1I}$$

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ [*/z]f = [u/z]f}\ \textsc{BetaI} \qquad \frac{}{[\mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ e/y]f = \mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ [e/y]f}\ \textsc{NatI}$$

$$\frac{}{\mathsf{let}\ e \otimes t\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ u = [e/x, t/y]u}\ \textsc{Beta1Ten} \qquad \frac{}{\mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [x \otimes y/z]f = [u/z]f}\ \textsc{Beta2Ten}$$

$$\frac{}{[\mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ g/w]f = \mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [g/w]f}\ \textsc{NatTen} \qquad \frac{}{u = \circ}\ \textsc{EtaParU}$$

$$\frac{}{(\mathsf{let}\ u\ \mathsf{be}\ x \parr -\ \mathsf{in}\ x) \parr (\mathsf{let}\ u\ \mathsf{be}\ - \parr y\ \mathsf{in}\ y) = u}\ \textsc{EtaPar} \qquad \frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ x \parr -\ \mathsf{in}\ e = [u/x]e}\ \textsc{Beta1Par}$$

$$\frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ - \parr y\ \mathsf{in}\ e = [t/y]e}\ \textsc{Beta2Par} \qquad \frac{}{\mathsf{let}\ t\ \mathsf{be}\ x \parr -\ \mathsf{in}\ [u/x]f = [\mathsf{let}\ t\ \mathsf{be}\ x \parr -\ \mathsf{in}\ u/x]f}\ \textsc{Nat1Par}$$

$$\frac{}{\mathsf{let}\ t\ \mathsf{be}\ - \parr y\ \mathsf{in}\ [v/y]f = [\mathsf{let}\ t\ \mathsf{be}\ - \parr y\ \mathsf{in}\ v/y]f}\ \textsc{Nat2Par}$$

Figure 2: Equivalence on terms

Lastly, we denote the usual function that computes the set of free variables in a term by $\mathsf{FV}(t)$, and its straightforward extension to right contexts as $\mathsf{FV}(\Delta)$.

The inference rules for FILL are defined in Figure 1. The $\textsc{Parl}$ rule depends on the function let-pat $z\ p\ \Delta$ which we define next.

Definition 3. *The function* let-pat $z\ p\ t$ *is defined as follows:*

$$\text{let-pat}\ z\ (x \parr -)\ t = t \quad \textit{where } x \notin \mathsf{FV}(t) \qquad \text{let-pat}\ z\ (- \parr y)\ t = t \quad \textit{where } y \notin \mathsf{FV}(t) \qquad \text{let-pat}\ z\ p\ t = \mathsf{let}\ z\ \mathsf{be}\ p\ \mathsf{in}\ t$$

It is straightforward to extend the previous definition to right-contexts, and we denote this extension by let-pat $z\ p\ \Delta$.

The motivation behind this function is that it only binds the pattern variables in $x \parr -$ and $- \parr y$ if and only if those pattern variables are free in the body of the let. This overcomes the counterexample given by Bierman in [4].

The terms of FILL are equipped with an equivalence relation defined in Figure 2. There are a number of α, β, and η like rules as well as several rules we call naturality rules. These rules are similar to the rules presented in [11].

3 Cut-elimination

FILL can be viewed from two different angles: i. as an intuitionistic linear logic with par, or ii. as a restricted form of classical linear logic. Thus, to prove cut-elimination of FILL one only needs to start with the cut-elimination procedure for intuitionistic linear logic, and then dualize all of the steps in the procedure for tensor and its unit to obtain the steps for par and its unit. Similarly, one could just as easily start with the cut-elimination procedure for classical linear logic, and then apply the restriction on the implication right rule producing a cut-elimination procedure for FILL.

The major difference between proving cut-elimination of FILL from classical or intuitionistic linear logic is that we must prove an invariant across each step in the procedure. The invariant is that if a derivation π is transformed into a derivation π', then the terms in the conclusion of the final rule applied in π must be equivalent to the terms in the conclusion of the final rule applied in π' using the rules from Figure 2.

We finally arrive at cut-elimination.

Theorem 4. *If $\Gamma \vdash t_1 : A_1, \ldots, t_i : A_i$ steps to $\Gamma \vdash t'_1 : A_1, \ldots, t'_i : A_i$ using the cut-elimination procedure, then $t_j = t'_j$ for $1 \leq j \leq i$.*

Proof. The cut-elimination procedure given here is the standard cut-elimination procedure for classical linear logic except the cases involving the implication right rule have the FILL restriction. The structure of our procedure follows the structure of the procedure found in [15]. Due to space limitations we only show one of the most interesting cases, but for the entire proof see the companion report [12].

Case: secondary hypothesis: left introduction of par (first case). The proof

$$
\dfrac{
\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash t : A \mid \Delta}
\qquad
\dfrac{
\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Gamma_1, x : A, \Gamma_2, y : B \vdash \Delta_1}
\qquad
\dfrac{\begin{matrix}\pi_3\\ \vdots\end{matrix}}{\Gamma_3, z : C \vdash \Delta_2}
}{\Gamma_1, x : A, \Gamma_2, \Gamma_3, w : B \parr C \vdash \text{let-pat}\, w\, (y \parr -)\, \Delta_1 \mid \text{let-pat}\, w\, (- \parr z)\, \Delta_2}\ \text{ParL}
}{\Gamma_1, \Gamma, \Gamma_2, \Gamma_3, w : B \parr C \vdash \Delta \mid [t/x](\text{let-pat}\, w\, (y \parr -)\, \Delta_1) \mid [t/x](\text{let-pat}\, w\, (- \parr z)\, \Delta_2)}\ \text{Cut}
$$

transforms into the proof

$$
\dfrac{
\dfrac{
\dfrac{\begin{matrix}\pi_1\\ \vdots\end{matrix}}{\Gamma \vdash t : A \mid \Delta}
\qquad
\dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{\Gamma_1, x : A, \Gamma_2, y : B \vdash \Delta_1}
}{\Gamma_1, \Gamma, \Gamma_2, y : B \vdash \Delta \mid [t/x]\Delta_1}\ \text{Cut}
\qquad
\dfrac{\begin{matrix}\pi_3\\ \vdots\end{matrix}}{\Gamma_3, z : C \vdash \Delta_2}
}{\Gamma_1, \Gamma, \Gamma_2, \Gamma_3, w : B \parr C \vdash \text{let-pat}\, w\, (y \parr -)\, \Delta \mid \text{let-pat}\, w\, (y \parr -)\, [t/x]\Delta_1 \mid \text{let-pat}\, w\, (- \parr z)\, \Delta_2}\ \text{ParL}
$$

First, by inspection of the previous proofs we can see that $y \notin \mathsf{FV}(\Delta)$ and $x \notin \mathsf{FV}(\Delta_2)$. Thus, let-pat $w\,(y \parr -)\,\Delta = \Delta$, and $[t/x]$(let-pat $w\,(- \parr z)\,\Delta_2$) $=$ let-pat $w\,(- \parr z)\,\Delta_2$. It suffices to show that $[t/x]$(let-pat $w\,(y \parr -)\,\Delta_1$) $=$ let-pat $w\,(y \parr -)\,[t/x]\Delta_1$ but this follows by distributing the substitution into the let-pat, and then simplifying using the fact that $w \neq x$.

□

Corollary 5 (Cut-Elimination)**.** *Cut-elimination holds for FILL.*

4 Full LNL Models

One of the difficult questions considering the categorical models of linear logic was how to model Girard's exponential, !, which is read "of course". The ! modality can be used to translate intuitionistic logic into intuitionistic linear logic, and so the correct categorical interpretation of ! should involve a relationship between a cartesian closed category, and the model of intuitionistic linear logic.

de Paiva gave some of the first categorical models of both classical and intuitionistic linear logic in her thesis [7]. She showed that a particular dialectica category called $\mathsf{Dial}_2(\mathsf{Sets})$ is a model of FILL where ! is interpreted as a comonad which produces natural comonoids, see page 76 of [7].

Definition 6. *The category* $\mathsf{Dial}_2(\mathsf{Sets})$ *consists of*

- *objects that are triples,* $A = (U, X, \alpha)$*, where* U *and* X *are sets, and* $\alpha \subseteq U \times X$ *is a relation, and*
- *maps that are pairs* $(f, F) : (U, X, \alpha) \longrightarrow (V, Y, \beta)$ *where* $f : U \longrightarrow V$ *and* $F : Y \longrightarrow X$ *such that*
  - *For any* $u \in U$ *and* $y \in Y$*,* $\alpha(u, F(y))$ *implies* $\beta(f(u), y)$*.*

Suppose $A = (U, X, \alpha)$*,* $B = (V, Y, \beta)$*, and* $C = (W, Z, \gamma)$*. Then identities are given by* $(\mathsf{id}_U, \mathsf{id}_X) : A \longrightarrow A$*. The composition of the maps* $(f, F) : A \longrightarrow B$ *and* $(g, G) : B \longrightarrow C$ *is defined as* $(f;g, G;F) : A \longrightarrow C$*.*

In her thesis de Paiva defines a particular class of dialectica categories called GC over a base category C, see page 41 of [7]. The category $\mathsf{Dial}_2(\mathsf{Sets})$ defined above can be seen as an instantiation of GC by setting C to be the category Sets of sets and functions between them.

Seely gave a different, syntactic categorical model that confirmed that the of-course exponential should be modeled by a comonad [20]. However, Seely's model turned out to be unsound, as pointed out by Bierman [3]. This then prompted Bierman, Hyland, de Paiva, and Benton to define another categorical model called linear categories (Definition 7) that are sound, and also model ! using a monoidal comonad [3].

Definition 7. *A* ***linear category****,* $\mathcal{L}$*, consists of:*

- *A symmetric monoidal closed category* $\mathcal{L}$*,*
- *A symmetric monoidal comonad* $(!, \epsilon, \delta, m_{A,B}, m_I)$ *such that*
  - *For every free !-coalgebra* $(!A, \delta_A)$ *there are two distinguished monoidal natural transformations* $e_A : !A \longrightarrow I$ *and* $d_A : !A \longrightarrow !A \otimes !A$ *which form a commutative comonoid and are coalgebra morphisms.*
  - *If* $f : (!A, \delta_A) \longrightarrow (!B, \delta_B)$ *is a coalgebra morphism between free coalgebras, then it is also a comonoid morphism.*

This definition is the one given by Bierman in his thesis, see [3] for full definitions.

Intuitionistic logic can be interpreted in a linear category as a full subcategory of the category of !-coalgebras for the comonad, see proposition 17 of [3].

Benton gave a more balanced view of linear categories called LNL models.

Definition 8. *A* ***linear/non-linear model (LNL model)*** *consists of*

- *a cartesian closed category* $(\mathcal{C}, 1, \times, \Rightarrow)$*,*
- *a SMCC* $(\mathcal{L}, I, \otimes, \multimap)$*, and*
- *a pair of symmetric monoidal functors* $(G, n) : \mathcal{L} \longrightarrow \mathcal{C}$ *and* $(F, m) : \mathcal{C} \longrightarrow \mathcal{L}$ *between them that form a symmetric monoidal adjunction with* $F \dashv G$*.*

See Benton, [2], for the definitions of symmetric monoidal functors and adjunctions.

A non-trivial consequence of the definition of a LNL model is that the ! modality can indeed be interpreted as a monoidal comonad. Suppose $(\mathcal{L}, \mathcal{C}, F, G)$ is a LNL model. Then the comonad is given by $(!, \epsilon : ! \longrightarrow \mathsf{Id}, \delta : ! \longrightarrow !!)$ where $! = FG$, ϵ is the counit of the adjunction and δ is the natural transformation $\delta_A = F(\eta_{G(A)})$, see page 15 of [2]. We recall the following result from Benton [2]:

Theorem 9 (LNL Models and Linear Categories)**.**

i. (Section 2.2.1 of [2]) Every LNL model is a linear category.

ii. (Section 2.2.2 of [2]) Every linear category is a LNL model.

Proof. The proof of part i. is a matter of checking that each part of the definition of a linear category can be constructed using the definition of a LNL model. See lemmata 3-7 of [2].

As for the proof of part ii. Given a linear category we have a SMCC and so the difficulty of proving this result is constructing the CCC and the adjunction between both parts of the model. Suppose $\mathcal{L}$ is a linear category. Benton constructs the CCC out of the full subcategory of Eilenberg-Moore category $\mathcal{L}^!$ whose objects are exponentiable coalgebras denoted $\mathsf{Exp}(\mathcal{L}^!)$. He shows that this subcategory is cartesian closed, and contains the (co)Kleisli category, $\mathcal{L}_!$, Lemma 11 on page 23 of [2]. As for the adjunction $F : \mathsf{Exp}(\mathcal{L}^!) \longrightarrow L : G$ can be defined using the adjunct functors $F(A, h_A) = A$ and $G(A) = (!A, \delta_A)$, see lemmata 13 - 16 of [2]. $\square$

Next we show that the category $\mathsf{Dial}_2(\mathsf{Sets})$ is a full version of a linear category. First, we extend the definitions of linear categories and LNL models to be equipped with the necessary categorical structure to model par and its unit.

Definition 10. *A **full linear category**, $\mathcal{L}$, consists of a linear category $(\mathcal{L}, \top, \otimes, \multimap, !A, e_A, d_A)$, a symmetric monoidal structure on L, $(\perp, \parr)$, and distribution natural transformations* $\mathsf{dist}_1 : A \otimes (B \parr C) \longrightarrow (A \otimes B) \parr C$ *and* $\mathsf{dist}_2 : (A \parr B) \otimes C \longrightarrow A \parr (B \otimes C)$.

Definition 11. *A **full linear/non-linear model (full LNL model)** consists of a LNL model $(\mathcal{L}, \mathcal{C}, F, G)$, a symmetric monoidal structure on L, $(\perp, \parr)$, as above.*

Our result is to first prove that $\mathsf{Dial}_2(\mathsf{Sets})$ is a full linear category, and then using the proof by Benton that linear categories are LNL models we obtain that $\mathsf{Dial}_2(\mathsf{Sets})$ is a full LNL model, but in order for this to work we need to know that $\mathsf{Dial}_2(\mathsf{Sets})$ has a symmetric monoidal comonad $(!, \epsilon, \delta, m_{A,B}, m_I)$. However, at the time of de Paiva's thesis it was not known that the comonad modeling the of-course exponential needed to be monoidal. We were able to show that the maps $m_{A,B}$ and m_I exist in the more general setting of dialectica categories, and thus, these maps exist in $\mathsf{Dial}_2(\mathsf{Sets})$. Intuitively, given two objects $A = (X, U, \alpha)$ and $B = (V, Y, \beta)$ of $\mathsf{Dial}_2(\mathsf{Sets})$ the map $m_{A,B}$ is defined as the pair $(\mathsf{id}_{U \times V}, F)$, where $F = (F_1, F_2)$, $F_1 : (U \times V) \Rightarrow (V \Rightarrow X)^* \longrightarrow V \Rightarrow (U \Rightarrow X^*)$ and $F_2 : (U \times V) \Rightarrow (U \Rightarrow Y)^* \longrightarrow U \Rightarrow (V \Rightarrow Y^*)$. The maps F_1 and F_2 build the sequence of all the results of applying each function in the input sequence to the input coordinate.

We can now show our main result of this section.

Lemma 12. *The category* $\mathsf{Dial}_2(\mathsf{Sets})$ *is a full linear category.*

Proof. We only give a sketch of the proof here, but for the full details see that companion report [12]. First, we must show that $\mathsf{Dial}_2(\mathsf{Sets})$ is a linear category. The majority of the linear structure of $\mathsf{Dial}_2(\mathsf{Sets})$ is in de Paiva's thesis [7]. However, we had to extend her definitions to show that the comonad $(!A, \delta, \epsilon)$ is monoidal, however, this is straightforward.

After showing that $\mathsf{Dial}_2(\mathsf{Sets})$ is a linear category one must show that $\mathsf{Dial}_2(\mathsf{Sets})$ is a model of par and its unit. This easily follows from de Paiva's thesis. The bifunctor which models par is given by de Paiva in Definition 10 on page 47 of [7].

Finally, $\mathsf{Dial}_2(\mathsf{Sets})$ must be distributive. The natural transformations $dist_1$ and $dist_2$ can be defined in terms of the maps $k : (A \otimes A') \otimes (B \parr C) \longrightarrow (A \otimes B) \parr (A' \otimes C)$ and $k' : (A \parr B) \otimes (C \otimes C') \longrightarrow (A \otimes C) \parr (B \otimes C')$ given on page 52 of [7]. Set $A' = \top$ in k and $C = \top$ in k' to obtain $dist_1$ and $dist_2$ respectively. $\square$

Corollary 13. *The category* $\mathsf{Dial}_2(\mathsf{Sets})$ *is a full LNL model.*

Proof. This follows directly from the previous lemma and Theorem 9 which shows that linear categories are LNL models. $\square$

Remark: It would appear, from the fact that tensorial logic [16] is a relaxing of linear logic where instead of an involutive negation we have a natural transformation $\eta_A \colon A \longrightarrow \neg\neg A$ that $\mathsf{Dial}_2(\mathsf{Sets})$ would be a model of tensorial logic. After all in $\mathsf{Dial}_2(\mathsf{Sets})$ we do have a natural transformation of the shape described, taking an object (U, X, α) to $(X, U, \neg\alpha)$ and then to $(U, X, \neg\neg\alpha)$ which is "almost" an isomorphism: we use identities in U and X, but unless the predicate α itself is double-negated, we have a morphism $\alpha \longrightarrow \neg\neg\alpha$, but not a converse one. But we have not had the time to check whether the other structure of tensorial logic is present and hence we leave this to future work.

5 Conclusion and Future Work

We first gave the definition of full intuitionistic linear logic using the left rule for par proposed by Bellin in Section 2. We then directly proved cut-elimination of FILL in Section 3 by adapting the well-known cut-elimination procedure for classical linear logic to FILL. Finally, in Section 4 we showed that the category $\mathsf{Dial}_2(\mathsf{Sets})$, a model of FILL, is a full LNL model by showing that it is a full linear category, and then replaying the proof that linear categories are LNL models by Benton.

Future work. Lorenzen games are a particular type of game semantics for various logics developed by Lorenz, Felscher, and Rahman et al. [13, 18].

Rahman showed that Lorenzen games could be defined for classical linear logic [17]. He was able to define a sound and complete semantics in Lorenzen games for classical linear logic, but he does mention that one could adopt a particular structural rule that enforces intuitionism. We plan to show that by adopting this rule we actually obtain a sound and complete semantics in Lorenzen games for FILL.

References

[1] Gianluigi Bellin. Subnets of proof-nets in multiplicative linear logic with mix. *Mathematical Structures in Computer Science*, 7:663–669, 12 1997.

[2] P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models (preliminary report). Technical Report UCAM-CL-TR-352, University of Cambridge Computer Laboratory, 1994.

[3] G. M. Bierman. *On Intuitionistic Linear Logic*. PhD thesis, Wolfson College, Cambridge, December 1993.

[4] Gavin Bierman. A note on full intuitionistic linear logic. *Annals of Pure and Applied Logic*, 79(3):281 – 287, 1996.

[5] Torben Braüner and Valeria Paiva. A formulation of linear logic based on dependency-relations. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin Heidelberg, 1998.

[6] R. Clouston, J. Dawson, R. Gore, and A. Tiu. Annotation-Free Sequent Calculi for Full Intuitionistic Linear Logic – Extended Version. *ArXiv e-prints*, July 2013.

[7] Valeria de Paiva. *The Dialectica Categories*. PhD thesis, University of Cambridge, 1988.

[8] Valeria de Paiva. Dialectica categories. In J. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92, pages 47–62. American Mathematical Society, 1989.

[9] Valeria de Paiva and Luiz Carlos Pereira. A short note on intuitionistic propositional logic with multiple conclusions. *MANUSCRITO - Rev. Int. Fil.*, 28(2):317 – 329, jul-dez 2005.

[10] A. G. Dragalin. *Mathematical Intuitionism. Introduction to Proof Theory*, volume 67 of *Translations of Mathematical Monographs*. Amerian Mathematical Society, 1988.

[11] Martin Hyland and Valeria de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64(3):273 – 291, 1993.

[12] Harley Eades III and Valeria de Paiva. Multiple conclusion intuitionistic linear logic and cut elimination. `http://metatheorem.org/papers/wollic15-report.pdf`.

[13] Laurent Keiff. Dialogical logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2011 edition, 2011.

[14] S. Maehara. Eine darstellung der intuitionistischen logik in der klassischen. *Nagoya Mathematical Journal*, 7:45–64, 1954.

[15] Paul-Andre Mellies. *Categorical Semantics of Linear Logic*. 2009.

[16] Paul-André Melliés and Nicolas Tabareau. Linear continuations and duality. Technical report, Université Paris VII, 2008.

[17] S. Rahman. Un desafio para las teorias cognitivas de la competencia log- ica: los fundamentos pragmaticos de la semantica de la logica linear. *Manuscrito*, XXV(2):381–432, October 2002.

[18] Shahid Rahman and Laurent Keiff. On how to be a dialogician. In Daniel Vanderveken, editor, *Logic, Thought and Action*, volume 2 of *Logic, Epistemology, and the Unity of Science*, pages 359–408. Springer Netherlands, 2005.

[19] Harold Schellinx. Some syntactical observations on linear logic. *Journal of Logic and Computation*, 1(4):537–559, 1991.

[20] R. Seely. Linear logic, *-autonomous categories and cofree coalgebras. In *Computer Science Logic*, 1989.

[21] G. Takeuti. *Proof Theory*. Amsterdam: North-Holland, 1975.