Multiple Conclusion Intuitionistic Linear Logic and Cut Elimination

Harley Eades III and Valeria de Paiva

February 18, 2015

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 in a short note 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 conjectured that by adopting his rule cut-elimination is restored. In this note we show that adopting Bellin's proposed rule one does obtain cut-elimination for FILL, as suggested. Additionally, we show that this new formalization can be modeled by a new form of dialectica category called order-enriched dialectica category, and discuss future work giving FILL a semantics in terms of Lorenzen games.

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 [4, 3]. Upon

1

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, par, implication, and their units. 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) [3]. 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' [8, 11]. The sequents in these types of logics usually have the form $\Gamma \vdash \Delta$ where Γ and Δ are multisets of formulas. Sequents of this form 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 [6].

Gentzen showed that to obtain intuitionistic logic one could start with the logic LK and then place a cardinality restriction on the righthand side of sequents, however, this is not the only means of enforcing intuitionism. Maehara showed that 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. The classical implication right rule has the form:

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

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

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

de Paiva's first formalization of FILL used the Dragalin restriction, see [3] p. 58, but Schellinx showed that this restriction has the unfortunate consequence of breaking cut-elimination [10].

Later, Hyland and de Paiva gave an alternate formalization of FILL in the hopes to regain cut-elimination [7]. 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:

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

Intuitionism is enforced in this rule by requiring that the variable being discharged, 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 [1]. As Bierman explains the problem was with the left rule for par. The original rule was as follows:

$$\frac{\Gamma, x : A \vdash \Delta \qquad \Gamma', y : B \vdash \Delta'}{\Gamma, \Gamma', z : A \,\mathbin{⅋}\, B \vdash \mathsf{let}\, z \,\mathsf{be}\, (x \,\mathbin{⅋}\, -)\, \mathsf{in}\, \Delta \mid \mathsf{let}\, z \,\mathsf{be}\, (- \,\mathbin{⅋}\, y)\, \mathsf{in}\, \Delta'} \;\; \text{\small PARL}$$

In this rule the pattern variables x and y are bound in each term of L and L' respectively. Notice that the variable z becomes free in every term in L and L'. 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:

$$\frac{\Gamma, x : A \vdash \Delta \quad \Gamma', y : B \vdash \Delta'}{\Gamma, \Gamma', z : A \,\mathbin{⅋}\, B \vdash \mathsf{let\text{-}pat}\, z \,(x \,\mathbin{⅋}\, -)\, \Delta \mid \mathsf{let\text{-}pat}\, z \,(- \,\mathbin{⅋}\, y)\, \Delta'} \;\; \text{\small PARL}$$

In this rule $\mathsf{let\text{-}pat}\, z \,(x \,\mathbin{⅋}\, -)\, t$ and $\mathsf{let\text{-}pat}\, z \,(- \,\mathbin{⅋}\, y)\, t'$ 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 conjectured that adopting this rule results in FILL regaining cut-elimination. However, no proof has been given.

Contributions. In this paper our main contribution is to confirm Bellin's conjecture by adopting his proposed rule (Section 2) and prove cut-elimination (Section 3). In addition, we show that this new formalization can be modeled by a new form of dialectica category called order-enriched dialectica category (Section 4).

Related Work. The first formalization of FILL with cut-elimination was due to Brauner and de Paiva [2]. 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 a what they call the FILL property:

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

They were able to show that their formalization is sound, complete, and enjoys cut-elimination. The one point 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, we do not wish to detract from the importance of their work.

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 [6]. They annotate hypothesis with natural number indices, and conclusions with finite sets of indices. The sets on conclusions correspond to the collection of the hypotheses that the conclusion depends on. Then they have a similar property to that of the FILL property from Brauner 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.

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:*

(Formulas)	$A, B, C, D, E ::= \top \mid \bot \mid A \multimap B \mid A \otimes B \mid A \mathbin{\rotatebox[origin=c]{180}{\&}} B$
(Patterns)	$p ::= * \mid - \mid x \mid p_1 \otimes p_2 \mid p_1 \mathbin{\rotatebox[origin=c]{180}{\&}} p_2$
(Terms)	$t, e ::= x \mid * \mid \circ \mid t_1 \otimes t_2 \mid t_1 \mathbin{\rotatebox[origin=c]{180}{\&}} t_2 \mid \lambda x.t \mid \mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ e \mid t_1\ t_2$
(Left Contexts)	$\Gamma ::= \cdot \mid x : A \mid \Gamma_1, \Gamma_2$
(Right Contexts)	$\Delta ::= \cdot \mid t : A \mid \Delta_1, \Delta_2$

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

$$
\begin{array}{rl}
\text{Tensor:} & \mathsf{let}\ t\ \mathsf{be}\ p_1 \otimes p_2\ \mathsf{in}\ e \\
\text{Par:} & \mathsf{let}\ t\ \mathsf{be}\ p_1 \mathbin{\rotatebox[origin=c]{180}{\&}} p_2\ \mathsf{in}\ e \\
\text{Tensor Unit:} & \mathsf{let}\ t\ \mathsf{be}\ *\ \mathsf{in}\ e
\end{array}
$$

In addition, each of these will have their own equational rules, see Definition 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 mete-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:*

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

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 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:*

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

$$\frac{}{\cdot \vdash * : \top}\ \text{IR} \qquad \frac{\Gamma, x : A, y : B \vdash \Delta}{\Gamma, z : A \otimes B \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ \Delta}\ \text{TL}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma' \vdash f : B \mid \Delta'}{\Gamma, \Gamma' \vdash e \otimes f : A \otimes B \mid \Delta \mid \Delta'}\ \text{TR} \qquad \frac{}{x : \bot \vdash \cdot}\ \text{PL} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \circ : \bot \mid \Delta}\ \text{PR}$$

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

$$\frac{\Gamma \vdash \Delta \mid e : A \mid f : B \mid \Delta'}{\Gamma \vdash \Delta \mid e\ ⅋\ f : A\ ⅋\ B \mid \Delta'}\ \text{ParR} \qquad \frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma', x : B \vdash \Delta'}{\Gamma, y : A \multimap B, \Gamma' \vdash \Delta \mid [y\ e/x]\Delta'}\ \text{ImpL}$$

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

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

Figure 1: Inference rules for FILL

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

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

The motivation behind this function is that it only binds the pattern variables in $x\ ⅋\ -$ and $-\ ⅋\ y$ if and only if those pattern variables are free in the body of the let. This over comes the counterexample given by Bierman in [1]. Throughout the sequel we will denote derivations of the previous rules by π.

Similarly to λ-calculi the terms of FILL are equipped with an equivalence relation. This equivalence on terms is defined in Figure 2. However, these rules should not be considered as computational rules, but rather are only necessary for the cut-elimination procedure. The rules are highly motivated by the semantic interpretation of FILL into symmetric monodial categories. 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 [7].

3   Cut-elimination

The usual proof of cut-elimination for intuitionistic and classical linear logic should suffice for FILL. Thus, in this section we simply give the cut-elimination procedure for FILL following the development in [9]. However, there is one invariant that must be verified across each derivation transformation. The invariant is that if a derivation π is transformed into a derivation π', then the

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

$$\frac{}{\mathsf{let}\ * \ \mathsf{be}\ *\ \mathsf{in}\ e = e} \;\; \textsc{Eta1I} \qquad \frac{}{\mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ [*/z]f = [u/z]f} \;\; \textsc{BetaI}$$

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

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

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [x \otimes y/z]f = [u/z]f} \;\; \textsc{Beta2Ten}$$

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

$$\frac{}{(\mathsf{let}\ u\ \mathsf{be}\ x\ \mathbin{\invamp}\ -\ \mathsf{in}\ x)\ \mathbin{\invamp}\ (\mathsf{let}\ u\ \mathsf{be}\ -\ \mathbin{\invamp} y\ \mathsf{in}\ y) = u} \;\; \textsc{EtaPar}$$

$$\frac{}{\mathsf{let}\ u\ \mathbin{\invamp}\ t\ \mathsf{be}\ x\ \mathbin{\invamp}\ -\ \mathsf{in}\ e = [u/x]e} \;\; \textsc{Beta1Par} \qquad \frac{}{\mathsf{let}\ u\ \mathbin{\invamp}\ t\ \mathsf{be}\ -\ \mathbin{\invamp} y\ \mathsf{in}\ e = [t/y]e} \;\; \textsc{Beta2Par}$$

$$\frac{}{\mathsf{let}\ t\ \mathsf{be}\ x\ \mathbin{\invamp}\ -\ \mathsf{in}\ [u/x]f = [\mathsf{let}\ t\ \mathsf{be}\ x\ \mathbin{\invamp}\ -\ \mathsf{in}\ u/x]f} \;\; \textsc{Nat1Par}$$

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

Figure 2: Equivalence on terms

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', but using what notion of equivalence?

The cut elimination procedure requires the following two basic results:

Lemma 4 (Substitution Distribution)**.** *For any terms t, t_1, and t_2, $[t_1/x][t_2/y]t = [[t_1/x]t_2/y][t_2/x]t$.*

Proof. This proof holds by straightforward induction on the form of t. □

Lemma 5 (Let-pat Distribution)**.** *For any terms t, t_1, and t_2, and pattern p, let-pat $t\ p\ [t_1/y]t_2 = [\text{let-pat}\ t\ p\ t_1/y]t_2$.*

Proof. This proof holds by case splitting over p, and then using the naturality equations for the respective pattern. □

Throughout the remainder of this section we present a particular step in the cut-elimination procedure, and then give a short proof that equality of terms are preserved across the particular transformation on derivations. Many of the transformations are trivial, and follow directly from the traditional proof. Thus, we only present here the most interesting cases. The full proof can be found in the companion report [**?**].

3.1 Principle formula vs. principle formula

3.1.1 Par

The proof

$$
\text{Parr}\;\frac{\dfrac{\vdots\;\pi_1}{\Gamma_1 \vdash \Delta_1 \mid t_1 : A \mid t_2 : B \mid \Delta_2}}{\dfrac{\Gamma_1 \vdash \Delta_1 \mid t_1 \,⅋\, t_2 : A \,⅋\, B \mid \Delta_2 \quad \dfrac{\dfrac{\vdots\;\pi_2}{\Gamma_2, x : A \vdash \Delta_3} \quad \dfrac{\vdots\;\pi_3}{\Gamma_3, y : B \vdash \Delta_4}}{\Gamma_2, \Gamma_3, z : A \,⅋\, B \vdash \text{let-pat } z\,(x\,⅋\,-)\,\Delta_3 \mid \text{let-pat } z\,(-\,⅋\,y)\,\Delta_4}\;\text{Parl}}{\Gamma_2, \Gamma_3, \Gamma_1 \vdash \Delta_1 \mid \Delta_2 \mid [t_1\,⅋\,t_2/z](\text{let-pat } z\,(x\,⅋\,-)\,\Delta_3) \mid [t_1\,⅋\,t_2/z](\text{let-pat } z\,(-\,⅋\,y)\,\Delta_4)}\;\text{Cut}}
$$

is transformed into the proof

$$
\dfrac{\dfrac{\dfrac{\vdots\;\pi_1}{\Gamma_1 \vdash \Delta_1 \mid t_1 : A \mid t_2 : B \mid \Delta_2} \quad \dfrac{\vdots\;\pi_3}{\Gamma_3, y : B \vdash \Delta_4}}{\Gamma_3, \Gamma_1 \vdash \Delta_1 \mid t_1 : A \mid \Delta_2 \mid [t_2/y]\Delta_4}\;\text{Cut} \quad \dfrac{\vdots\;\pi_2}{\Gamma_2, x : A \vdash \Delta_3}}{\dfrac{\Gamma_2, \Gamma_3, \Gamma_1 \vdash \Delta_1 \mid \Delta_2 \mid [t_2/y]\Delta_4 \mid [t_1/x]\Delta_3}{\Gamma_2, \Gamma_3, \Gamma_1 \vdash \Delta_1 \mid \Delta_2 \mid [t_1/x]\Delta_3 \mid [t_2/y]\Delta_4}\;\text{Series of Exchanges}}\;\text{Cut}
$$

Without loss of generality consider the case when $\Delta_3 = t_3 : C_1 \mid \Delta_3'$ and $\Delta_4 = t_4 : C_2 \mid \Delta_4'$. First, $[t_1 \,⅋\, t_2/z](\text{let-pat } z\,(x\,⅋\,-)\,t_3) = \text{let-pat }(t_1 \,⅋\, t_2)\,(x\,⅋\,-)\,t_3$, and by Eq_Beta1Par we know $\text{let-pat }(t_1 \,⅋\, t_2)\,(x\,⅋\,-)\,t_3 = [t_1/x]t_3$ if $x \in \mathsf{FV}(t_3)$ or $\text{let-pat }(t_1 \,⅋\, t_2)\,(x\,⅋\,-)\,t_3 = t_3$ otherwise. In the latter case we can see that $t_3 = [t_1/x]t_3$, thus, in both cases $\text{let-pat }(t_1 \,⅋\, t_2)\,(x\,⅋\,-)\,t_3 = [t_1/x]t_3$. This argument can be repeated for any terms in Δ_3', and hence $[t_1 \,⅋\, t_2/z](\text{let-pat } z\,(x\,⅋\,-)\,\Delta_3) = \text{let-pat }(t_1 \,⅋\, t_2)\,(x\,⅋\,-)\,\Delta_3 = [t_1/x]\Delta_3$. We can apply a similar argument for $[t_1 \,⅋\, t_2/z](\text{let-pat } z\,(-\,⅋\,y)\,t_4)$ and $[t_1 \,⅋\, t_2/z](\text{let-pat } z\,(-\,⅋\,y)\,\Delta_4)$.

Note that we could have first cut on A, and then on B in the second derivation, but we would have arrived at the same result just with potentially more exchanges on the right.

3.1.2 Implication

The proof

$$
\dfrac{\dfrac{\dfrac{\vdots\;\pi_1}{\Gamma, x : A \vdash t : B \mid \Delta} \quad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.t : A \multimap B \mid \Delta}\;\text{Impr} \quad \dfrac{\dfrac{\vdots\;\pi_2}{\Gamma_1 \vdash t_1 : A \mid \Delta_1} \quad \dfrac{\vdots\;\pi_3}{\Gamma_2, y : B \vdash \Delta_2}}{\Gamma_1, z : A \multimap B, \Gamma_2 \vdash \Delta_1 \mid [z\,t_1/y]\Delta_2}\;\text{Impl}}{\Gamma_1, \Gamma, \Gamma_2 \vdash \Delta \mid [\lambda x.t/z]\Delta_1 \mid [\lambda x.t/z][z\,t_1/y]\Delta_2}\;\text{Cut}
$$

transforms into the proof

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdots\;\pi_2}{\Gamma_1 \vdash t_1 : A \mid \Delta_1} \quad \dfrac{\vdots\;\pi_1}{\Gamma, x : A \vdash t : B \mid \Delta} \quad x \notin \mathsf{FV}(\Delta)}{\Gamma, \Gamma_1 \vdash \Delta_1 \mid [t_1/x]t : B \mid [t_1/x]\Delta}\;\text{Cut} \quad \dfrac{\vdots\;\pi_3}{\Gamma_2, y : B \vdash \Delta_2}}{\Gamma_2, \Gamma, \Gamma_1 \vdash \Delta_1 \mid [t_1/x]\Delta \mid [[t_1/x]t/y]\Delta_2}\;\text{Cut}}{\Gamma_1, \Gamma, \Gamma_2 \vdash [t_1/x]\Delta \mid \Delta_1 \mid [[t_1/x]t/y]\Delta_2}\;\text{Series of Exchanges}
$$

7

Without loss of generality consider the case when $\Delta_2 = t_2 : C \mid \Delta_2'$. First, by hypothesis we know $x \notin \mathsf{FV}(\Delta)$, and so we know $\Delta = [t_1/x]\Delta$. We can see that $[\lambda x.t/z][z\ t_1/y]t_2 = [(\lambda x.t)\ t_1/y]t_2 = [[t_1/x]t/y]t_2$ by using the congruence rules of equality and the rule EQ_BETAFUN. This argument can be repeated for any term in $[\lambda x.t/z][z\ t_1/y]\Delta_2'$, and so $[\lambda x.t/z][z\ t_1/y]\Delta_2 = [[t_1/x]t/y]\Delta_2$. Finally, by inspecting the previous derivations we can see that $z \notin \mathsf{FV}(\Delta_1)$, and thus, $\Delta_1 = [\lambda x.t/z]\Delta_1$.

4 Order-Enriched Dialectica Categories

[5]

5 Conclusion and Future Work

References

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

[2] Torben Brauner 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.

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

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

[5] Valeria de Paiva. Bounded dialectica interpretation: categorically. Under review, 2015.

[6] 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.

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

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

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

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

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