ELSEVIER Annals of Pure and Applied Logic 79 (1996) 281–287

ANNALS OF PURE AND APPLIED LOGIC

A note on full intuitionistic linear logic

G.M. Bierman

Gonville and Caius College, Cambridge CB2 1TA, UK

Received 29 May 1995; revised 17 December 1995
Communicated by D. van Dalen

Abstract

This short note considers the formulation of Full Intuitionistic Linear Logic (FILL) given by Hyland and de Paiva (1993). Unfortunately the formulation is not closed under the process of cut elimination. This note proposes an alternative formulation based on the notion of patterns.

1. Introduction

In proof theory, it is often said that an intuitionistic fragment of a classical logic can be given by restricting the succedent in the sequent calculus formulation to at most one formula occurrence. Many proof theorists have considered relaxing this rather heavy restriction. For example, Takeuti [7] gives such a system in his book.

One might ask whether a similar enterprise exists for linear logic. Intuitionistic Linear Logic (ILL), which arises from restricting Classical Linear Logic to one succedent, is studied at length in my thesis [1]. A more relaxed system was given by de Paiva [3] in her thesis and tightened in a subsequent paper by Hyland and de Paiva [4]. The resulting logic, *Full* Intuitionistic Linear Logic (FILL), has two notable features. Firstly it contains a connective, the multiplicative disjunction ('par' or '$\invamp$'), which does not ordinarily exist in ILL. Secondly Hyland and de Paiva's formulation is given with respect to a term assignment system as it critically needs a notion of formula dependency.

Hyland and de Paiva's formulation is given in Fig. 1. Formulae are represented by $\phi, \psi, \ldots$ and $\Gamma, \Delta, \ldots$ represent multisets of formulae.

The $\multimap_{\mathcal{R}}$ rule needs some explanation. Normally, for example in Takeuti's system, we make the restriction that the right-hand side contains at most one formula. This is relaxed in FILL to requiring that the variable being abstracted is *not* free in any of the terms except the one it is being abstracted over. This idea is formulated in terms of a notion of *free variable*, given below.

PII S0168-0072(96)00004-8

$$\frac{}{x : \phi \rhd x : \phi}\,\textit{Identity}$$

$$\frac{\Gamma \rhd M : \phi | \vec{N} : \Delta \qquad x : \phi, \Gamma' \rhd \vec{P} : \Delta'}{\Gamma, \Gamma' \rhd \vec{N} : \Delta | \vec{P}[x{:=}M] : \Delta'}\,\textit{Cut}$$

$$\frac{\Gamma \rhd \vec{M} : \Delta}{\Gamma, x : I \rhd \mathrm{let}\, x\, \mathrm{be} * \,\mathrm{in}\, \vec{M} : \Delta}(I_{\mathcal{L}}) \qquad \frac{}{\rhd * : I}(I_{\mathcal{R}})$$

$$\frac{\Gamma, x : \phi, y : \psi \rhd \vec{M} : \Delta}{\Gamma, z : \phi \otimes \psi \rhd \mathrm{let}\, z\, \mathrm{be} x \otimes y\, \mathrm{in}\, \vec{M} : \Delta}(\otimes_{\mathcal{L}}) \quad \frac{\Gamma \rhd M : \phi | \vec{P} : \Delta \qquad \Gamma' \rhd N : \psi | \vec{Q} : \Delta'}{\Gamma, \Gamma' \rhd M \otimes N : \phi \otimes \psi | \vec{P} : \Delta | \vec{Q} : \Delta'}(\otimes_{\mathcal{R}})$$

$$\frac{}{x : \bot \rhd}(\bot_{\mathcal{L}}) \qquad \frac{\Gamma \rhd \vec{M} : \Delta}{\Gamma \rhd \vec{M} : \Delta | \circ : \bot}(\bot_{\mathcal{R}})$$

$$\frac{\Gamma, x : \phi \rhd \vec{M} : \Delta \qquad \Gamma', y : \psi \rhd \vec{N} : \Delta'}{\Gamma, \Gamma', z : \phi \parr \psi \rhd \mathrm{let}\, z\, \mathrm{be} x \parr - \;\mathrm{in}\, \vec{M} : \Delta | \mathrm{let}\, z\, \mathrm{be} - \parr\, y\, \mathrm{in}\, \vec{N} : \Delta'}(\parr_{\mathcal{L}})$$

$$\frac{\Gamma \rhd M : \phi | N : \psi | \vec{P} : \Delta}{\Gamma \rhd M \parr N : \phi \parr \psi | \vec{P} : \Delta}(\parr_{\mathcal{R}})$$

$$\frac{\Gamma \rhd M : \phi | \vec{N} : \Delta \qquad \Gamma', x : \psi \rhd \vec{P} : \Delta'}{\Gamma, \Gamma', y : \phi \multimap \psi \rhd \vec{N} : \Delta | (\vec{P}[x := (yM)]) : \Delta'}(\multimap_{\mathcal{L}})$$

$$\frac{\Gamma, x : \phi \rhd M : \psi | \vec{N} : \Delta}{\Gamma \rhd (\lambda x : \phi . M) : \phi \multimap \psi | \vec{N} : \Delta}(\multimap_{\mathcal{R}}) \text{ where } x \notin FV(\vec{N})$$

Fig. 1. Hyland and de Paiva's formulation of FILL.

Definition 1. Given a term, M, its set of free variables, $FV(M)$, is given by the following definition.

$$\begin{aligned}
FV(x) &\stackrel{\text{def}}{=} \{x\} \\
FV(MN) &\stackrel{\text{def}}{=} FV(M) \cup FV(N) \\
FV(\lambda x : \phi . M) &\stackrel{\text{def}}{=} FV(M) - \{x\} \\
FV(M \otimes N) &\stackrel{\text{def}}{=} FV(M) \cup FV(N) \\
FV(\mathrm{let}\, M\, \mathrm{be}\, x \otimes y\, \mathrm{in}\, N) &\stackrel{\text{def}}{=} FV(M) \cup (FV(M) - \{x, y\}) \\
FV(\mathrm{let}\, M\, \mathrm{be} * \mathrm{in}\, N) &\stackrel{\text{def}}{=} FV(M) \cup FV(N) \\
FV(M \parr N) &\stackrel{\text{def}}{=} FV(M) \cup FV(N) \\
FV(\mathrm{let}\, M\, \mathrm{be} - \parr x\, \mathrm{in}\, N) &\stackrel{\text{def}}{=} FV(M) \cup (FV(N) - \{x\}) \\
FV(\mathrm{let}\, M\, \mathrm{be} y \parr - \,\mathrm{in}\, N) &\stackrel{\text{def}}{=} FV(M) \cup (FV(N) - \{y\}) \\
FV(*), FV(\circ) &\stackrel{\text{def}}{=} \emptyset.
\end{aligned}$$

Unfortunately as it stands, this system is not closed under the standard process of cut elimination (contrary to Theorem 6 of the original paper [4, p. 289]). The problematic rule is $(\parr_{\mathcal{L}})$. The problem is that it introduces unnecessary formula bindings. In the process of cut elimination, there are a number of so-called commuting cuts, i.e. those where the cut formula is a minor premise. One such example is the $(-,\parr_{\mathcal{L}})$-cut, viz.

$$\dfrac{\Gamma \rhd M{:}\phi|\vec{N}{:}\Delta \qquad \dfrac{x{:}\phi,\ w{:}\psi,\Gamma'\rhd\vec{P}{:}\Delta' \qquad z{:}\theta,\Gamma''\rhd\vec{Q}{:}\Delta''}{x{:}\phi,y{:}\psi\parr\theta,\Gamma',\Gamma''\rhd \text{let } y \text{ be } w\parr- \text{ in } \vec{P}{:}\Delta'|\text{let } y \text{ be } -\parr z \text{ in } \vec{Q}{:}\Delta''}\,(\parr_{\mathcal{L}})}{\Gamma,y:\psi\parr\theta,\Gamma',\Gamma''\rhd(\text{let } y \text{ be } w\parr- \text{ in } \vec{P})[x{:=}M]{:}\ \Delta' \mid (\text{let } y \text{ be } -\parr z \text{ in } \vec{Q})[x{:=}M]{:}\Delta''|\vec{N}{:}\Delta}\,\mathit{Cut}$$

To eliminate this commuting cut we float the application of the *Cut* above the application of the $(\parr_{\mathcal{L}})$ rule, i.e.

$$\dfrac{\dfrac{\Gamma\rhd M{:}\phi|\vec{N}{:}\Delta \qquad x{:}\phi,\ w{:}\psi,\Gamma'\rhd\vec{P}{:}\Delta'}{\Gamma,w{:}\psi,\Gamma'\rhd\vec{P}[x{:=}M]{:}\Delta'|\vec{N}{:}\Delta}\,\mathit{Cut} \qquad z{:}\theta,\Gamma''\rhd\vec{Q}{:}\ \Delta''}{\Gamma,\ y{:}\psi\parr\theta,\Gamma',\Gamma''\rhd\text{let } y \text{ be } w\parr- \text{ in } (\vec{P}[x{:=}M]){:}\ \Delta' \mid \text{let } y \text{ be } w\parr- \text{ in } \vec{N}{:}\Delta \mid \text{let } y \text{ be } -\parr z \text{ in } \vec{Q}{:}\Delta'}\,(\parr_{\mathcal{L}})$$

Ordinarily we should expect that the result of a commutative cut is an identity at the level of terms, but the reduction above hides a more sinister behaviour. It has introduced a new free variable to part of the term (term $\vec{N}$ has become ($\text{let } y \text{ be } w\parr- \text{ in } \vec{N}$) with the *new* free variable y). The role of free variables is paramount to the success of this formulation and the problem above means that cut elimination does not hold for the logic.

An example may illuminate this point. The deduction below contains an application of the *Cut* rule given above.

$$\dfrac{\dfrac{\dfrac{\dfrac{v{:}\phi\rhd v{:}\phi}{v{:}\phi\rhd v{:}\phi|\circ{:}\bot}\,(\bot_{\mathcal{R}}) \qquad \dfrac{\dfrac{x{:}\phi\rhd x{:}\phi \qquad y{:}\psi\rhd y{:}\psi}{x{:}\phi,y{:}\psi\rhd x\otimes y{:}\phi\otimes\psi}\,(\otimes_{\mathcal{R}}) \qquad w{:}\theta\rhd w{:}\theta}{x{:}\phi,\ z{:}\psi\parr\theta\rhd(\text{let } z \text{ be } y\parr- \text{ in } x\otimes y){:}\phi\otimes\psi|(\text{let } z \text{ be} -\parr w \text{ in } w){:}\theta}\,(\parr_{\mathcal{L}})}{v{:}\phi,z{:}\psi\parr\theta\rhd(\text{let } z \text{ be } y\parr- \text{ in } v\otimes y){:}\phi\otimes\psi|(\text{let } z \text{ be} -\parr w \text{ in } w){:}\theta|\circ{:}\bot}\,\mathit{Cut}}{v{:}\phi,z{:}\psi\parr\theta\rhd(\text{let } z \text{ be } y\parr- \text{ in } v\otimes y)\parr(\text{let } z \text{ be} -\parr w \text{ in } w){:}(\phi\otimes\psi)\parr\theta|\circ{:}\bot}\,(\parr_{\mathcal{R}})}{v{:}\phi\rhd\lambda z{:}\psi\parr\theta.((\text{let } z \text{ be } y\parr- \text{ in } v\otimes y)\parr(\text{let } z \text{ be} -\parr w \text{ in } w)){:}(\psi\parr\theta)\multimap(\phi\otimes\psi)\parr\theta|\circ{:}\bot}\,(\multimap_{\mathcal{R}})$$

This proof reduces to the following (illegal) proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{v{:}\phi\rhd v{:}\phi}{v{:}\phi\rhd v{:}\phi|\circ{:}\bot}\,(\bot_{\mathcal{R}}) \qquad \dfrac{x{:}\phi\rhd x{:}\phi \qquad y{:}\psi\rhd y{:}\psi}{x{:}\phi,y{:}\psi\rhd x\otimes y{:}\phi\otimes\psi}\,(\otimes_{\mathcal{R}})}{y{:}\psi,v{:}\phi\rhd v\otimes y{:}\phi\otimes\psi|\circ{:}\bot}\,\mathit{Cut} \qquad w{:}\theta\rhd w{:}\theta}{v{:}\phi,z{:}\psi\parr\theta\rhd(\text{let } z \text{ be } y\parr- \text{ in } v\otimes y){:}\phi\otimes\psi \mid (\text{let } z \text{ be} -\parr w \text{ in } w){:}\theta \mid \text{let } z \text{ be } y\parr- \text{ in } \circ{:}\bot}\,(\parr_{\mathcal{L}})}{v{:}\phi,z{:}\psi\parr\theta\rhd(\text{let } z \text{ be } y\parr- \text{ in } v\otimes y)\parr(\text{let } z \text{ be} -\parr w \text{ in } w){:}(\phi\otimes\psi)\parr\theta \mid \text{let } z \text{ be } y\parr- \text{ in } \circ{:}\bot}\,(\parr_{\mathcal{R}})}{v{:}\phi\rhd\lambda z{:}\psi\parr\theta.((\text{let } z \text{ be } y\parr- \text{ in } v\otimes y)\parr(\text{let } z \text{ be} -\parr w \text{ in } w)){:}(\psi\parr\theta)\multimap(\phi\otimes\psi)\parr\theta \mid \text{let } z \text{ be } y\parr- \text{ in } \circ{:}\bot}\,(\multimap_{\mathcal{R}})$$

Clearly the final application of the $\multimap_{\mathcal{R}}$ rule is invalid as z is a free variable of the rightmost term.

Hence the standard process of cut elimination fails but one may wonder if another method might succeed. Schellinx [6] showed that this was the case for Takeuti's multiple-conclusion formulation of Intuitionistic Logic (IL). There the traditional cut elimination technique fails but the system does have the property that every valid formula has a cut-free proof. However, this is *not* the case for Hyland and de Paiva's formulation of FILL. An example is the formula

$$((\sigma \parr \tau) \parr \upsilon) \multimap (\sigma \parr ((\tau \parr \upsilon \multimap \psi) \parr \phi) \multimap (\psi \parr \phi)).$$

Although this is a valid formula (in the sense that there is a corresponding morphism in the free full multiplicative category), Hyland and de Paiva's term assignment formulation does not permit any cut-free derivation.

2. A pattern calculus for FILL

To circumvent these problems I shall use a pattern syntax for the sequents. Traditionally term assignments for various logics are of the form

$$\vec{x} : \Gamma \rhd M : \phi.$$

Thus assumptions are always labelled with *variables* and applications of rules are recorded with some form of syntax on the right-hand side of the turnstile. Whilst this is clearly correct for formulations in natural deduction, it seems less convincing for sequent calculus formulations. A feature of the sequent calculus is its symmetry, in that rules appear on both the left and the right of the turnstile. The pattern calculus arises from taking this symmetry seriously; thus only right rules introduce syntax on the right and left rules now introduce syntax on the left, called *patterns.* [1]

Patterns, p, and terms, M, are given by the mutually recursive grammars

$$p ::= x \mid * \mid \circ \mid p \otimes p \mid p \parr p \mid M \multimap p$$

and

$$M ::= x \mid * \mid \circ \mid \lambda p : \phi . M \mid M \otimes M \mid \text{cut } M \text{ for } p \text{ in } M \mid M \parr M.$$

The proof system is given in Fig. 2. It is easy to see how the set of variables contained in a pattern, p, denoted $Var(p)$, can be defined.

It is worth pointing out that applications of the *Cut* rule appear in the pattern calculus as an explicit term constructor. This is strongly connected to the *explicit substitution* operator sometimes used in the λ-calculus. Cut elimination, at the level of terms, yields a set of reduction rules which shows how this term constructor is eliminable.

[1] This idea has occurred to many people. As far as I am aware the first person to put it into print (for IL) was Lafont [5]. The idea was rediscovered by Breazu-Tannen et al. [2], again for IL, and I have also employed this idea, in a forthcoming paper, to devise syntax for various fragments of Linear Logic.

$$\frac{}{x : \phi \rhd x : \phi}\,Identity$$

$$\frac{\Gamma \rhd M : \phi | \vec{N} : \Delta \qquad p : \phi, \Gamma' \rhd \vec{P} : \Delta'}{\Gamma, \Gamma' \rhd \vec{N} : \Delta | \text{cut } M \text{ for } p \text{ in} \vec{P} : \Delta'}\,Cut$$

$$\frac{\Gamma \rhd \vec{M} : \Delta}{\Gamma, * : I \rhd \vec{M} : \Delta}(I_{\mathcal{L}}) \qquad \frac{}{\rhd * : I}(I_{\mathcal{R}})$$

$$\frac{\Gamma, p : \phi, q : \psi \rhd \vec{M} : \Delta}{\Gamma, p \otimes q : \rhd \phi \otimes \psi \rhd \vec{M} : \Delta}(\otimes_{\mathcal{L}}) \quad \frac{\Gamma M : \phi | \vec{P} : \Delta \quad \Gamma' \rhd N : \psi | \vec{Q} : \Delta'}{\Gamma, \Gamma' \rhd M \otimes N : \phi \otimes \psi | \vec{P} : \Delta | \vec{Q} : \Delta'}(\otimes_{\mathcal{R}})$$

$$\frac{}{\circ : \bot \rhd}(\bot_{\mathcal{L}}) \qquad \frac{\Gamma \rhd \vec{M} : \Delta}{\Gamma \rhd \vec{M} : \Delta | \circ : \bot}(\bot_{\mathcal{R}})$$

$$\frac{\Gamma, p : \phi \rhd \vec{M} : \Delta \qquad \Gamma', q : \psi \rhd \vec{N} : \Delta'}{\Gamma, \Gamma', p \parr q : \phi \parr \psi \rhd \vec{M} : \Delta | \vec{N} : \Delta'}(\parr_{\mathcal{L}})$$

$$\frac{\Gamma \rhd M : \phi | N : \psi | \vec{P} : \Delta}{\Gamma \rhd M \parr N : \phi \parr \psi | \vec{P} : \Delta}(\parr_{\mathcal{R}})$$

$$\frac{\Gamma \rhd M : \phi | \vec{N} : \Delta \qquad \Gamma', p : \psi \rhd \vec{P} : \Delta'}{\Gamma, \Gamma', M \multimap p : \phi \multimap \psi \rhd \vec{N} : \Delta | \vec{P} : \Delta'}(\multimap_{\mathcal{L}})$$

$$\frac{\Gamma, p : \phi \rhd M : \psi | \vec{N} : \Delta}{\Gamma \rhd (\lambda p : \phi . M) : \phi \multimap \psi | \vec{N} : \Delta}(\multimap_{\mathcal{R}}) \text{ where } Var(p) \cap FV(\vec{N}) = \emptyset$$

Fig. 2. Pattern formulation of FILL.

Consider the previous problematic step of cut elimination with the new pattern calculus. The instance of the commuting cut appears as

$$\frac{\Gamma \rhd M : \phi | \vec{N} : \Delta \qquad \dfrac{p : \phi, q : \psi, \Gamma' \rhd \vec{P} : \Delta' \quad r : \theta, \Gamma'' \rhd \vec{Q} : \Delta''}{p : \phi, q \parr r : \psi \parr \theta, \Gamma', \Gamma'' \rhd \vec{P} : \Delta' | \vec{Q} : \Delta''}(\parr_{\mathcal{L}})}{\Gamma, q \parr r : \psi \parr \theta, \Gamma', \Gamma'' \rhd \vec{N} : \Delta | \text{ cut } M \text{ for } p \text{ in } \vec{P} : \Delta' | \text{ cut } M \text{ for } p \text{ in } \vec{Q} : \Delta''}\,Cut,$$

which is reduced to

$$\frac{\dfrac{\Gamma \rhd M : \phi | \vec{N} : \Delta \quad p : \phi, q : \psi, \Gamma' \rhd \vec{P} : \Delta'}{\Gamma, q : \psi, \Gamma' \rhd \vec{N} : \Delta | \text{ cut } M \text{ for } p \text{ in } \vec{P} : \Delta'}\,Cut \qquad r : \theta, \Gamma'' \rhd \vec{Q} : \Delta''}{\Gamma, q \parr r : \psi \parr \theta, \Gamma', \Gamma'' \rhd \vec{N} : \Delta | \text{ cut } M \text{ for } p \text{ in } \vec{P} : \Delta' | \vec{Q} : \Delta''}(\parr_{\mathcal{L}}).$$

It is clear that no false dependencies have been introduced. In fact, a property of pattern calculi is that dependencies are eliminated during the process of cut elimination. Hence we have the following property for the pattern calculus formulation of FILL, where we represent a step in the cut elimination process by the symbol $\rightsquigarrow_{\text{cut}}$.

Theorem 1. *If* $\Gamma \rhd \vec{M} : \Delta$ *and* $\vec{M} \rightsquigarrow_{\mathrm{cut}} \vec{N}$ *then* $\Gamma \rhd \vec{N} : \Delta$.

Corollary 1. *The cut elimination theorem holds for the pattern calculus formulation of* FILL.

3. Conclusions

This short note considered the formulation of FILL given by Hyland and de Paiva and showed that it is not closed under the process of cut elimination. A syntax based on the idea of patterns was introduced which does have this closure property.

In private communication, Gianluigi Bellin has suggested to me that Hyland and de Paiva's formulation of the $⅋_{\mathcal{L}}$ rule could be rewritten to

$$\frac{\Gamma, x : \phi \rhd \vec{M} : \Delta \quad \Gamma', y : \psi \rhd \vec{N} : \Delta'}{\Gamma, \Gamma', z : \phi ⅋ \psi \rhd \vec{M}^* : \Delta | \vec{N}^* : \Delta'} (⅋_{\mathcal{L}}),$$

where $m_i^* \stackrel{\text{def}}{=} \begin{cases} \text{let}\, z\, \text{be}\, x⅋ - \ \text{in}\, m_i & \text{if } x \in FV(m_i), \\ m_i & \text{otherwise;} \end{cases}$

and $n_i^* \stackrel{\text{def}}{=} \begin{cases} \text{let}\, z\, \text{be} - ⅋ y\, \text{in}\, n_i & \text{if } y \in FV(n_i), \\ n_i & \text{otherwise.} \end{cases}$

This also leads to a system which is also closed under the process of cut elimination. I have used the pattern calculus in this paper as it naturally leads to a *proof net* formulation which is possibly the most succinct presentation of FILL, albeit one using graphical notation as opposed to familiar sequents. The details of a proof net presentation and how it relates to the pattern formulation is left to a future paper.

Acknowledgements

I am grateful to Martin Hyland, Valeria de Paiva and Gianluigi Bellin for many useful discussions. This work was originally funded by the EPSRC.

References

[1] G.M. Bierman, On intuitionistic linear logic, Ph.D. Thesis, Computer Laboratory, University of Cambridge, December 1993. Published as Computer Laboratory Technical Report 346, August 1994.

[2] V. Breazu-Tannen, D. Kesner and L. Puel, A typed pattern calculus, in: Proc. Symp. on Logic in Computer Science (1993) 262–274.

[3] V.C.V. de Paiva, The dialetica categories, Ph.D. Thesis, Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, November 1988. Published as Computer Laboratory Technical Report 213, January 1991.

[4] J.M.E. Hyland and V.C.V. de Paiva, Full intuitionistic linear logic, Ann. Pure Appl. Logic 64 (1993) 273–291.

[5] Y. Lafont, Functional programming and linear logic, Lecture Notes from the Glasgow Summer School on Functional Programming and Constructive Logic, September 1989.

[6] H. Schellinx, Some syntactic observations on linear logic, Logic and Comput. 1 (1991) 537–559.

[7] G. Takeuti, Proof Theory (North-Holland, Amsterdam, 1975).