Theory and Applications of Categories, Vol. 17, No. 3, 2006, pp. 30–48.

A PARIGOT-STYLE LINEAR λ-CALCULUS FOR FULL INTUITIONISTIC LINEAR LOGIC

VALERIA DE PAIVA AND EIKE RITTER

Abstract. This paper describes a natural deduction formulation for Full Intuitionistic Linear Logic (FILL), an intriguing variation of multiplicative linear logic, due to Hyland and de Paiva. The system FILL resembles intuitionistic logic, in that all its connectives are independent, but resembles classical logic in that its sequent-calculus formulation has intrinsic multiple conclusions. From the intrinsic multiple conclusions comes the inspiration to modify Parigot's natural deduction systems for classical logic, to produce a natural deduction formulation and a term assignment system for FILL.

1. Introduction

This paper describes a natural deduction formulation for Full Intuitionistic Linear Logic (FILL), a variant of Linear Logic, first described by Hyland and de Paiva [HdP93]. The system FILL has all the multiplicative connectives of classical linear logic [Gir95], but it is *not* involutive: the double-negation $A^{\perp\perp}$ is not the same as A, as is the case in classical linear logic. The system FILL has arisen from its categorical model, the Dialectica construction [dP89], which can be seen as a variant of the Chu construction [Bar96].

Weakly distributive categories [CS97] with a right adjoint to the tensor product but without negation provide another sound and complete model for FILL. Furthermore, for every weakly distributive category modelling FILL one can identify a *-autonomous subcategory which models classical linear logic. This subcategory corresponds to all objects for which $\neg\neg A$ and A are isomorphic. This shows that FILL can be seen as classical linear logic without involutive negation.

A sequent-style calculus for FILL has been presented in [BdP98]. Our natural deduction formulation of FILL is based on Pym and Ritter's extension [RP01] of Parigot's $\lambda\mu$ natural deduction system [Par92] for (propositional) classical logic. Their extension, called $\lambda\mu\nu$, was used to provide a term calculus for a multiplicative version of disjunction in classical logic. We provide a term-calculus for FILL based on $\lambda\mu\nu$, which we call FILL_μ and prove its basic proof-theoretical properties.

We first recall the system FILL, in its sequent-style formulation due to Braüner and de Paiva [BdP98]. Then we recap the main ideas of Parigot's system $\lambda\mu$, as well as Pym and Ritter's extension $\lambda\mu\nu$. In the next section we introduce our natural deduction term calculus for FILL, based on $\lambda\mu$, called FILL_μ and prove its main proof theoretical

Received by the editors 2003-10-31 and, in revised form, 2005-02-28.
Published on 2006-12-16 in the volume *Chu spaces: theory and applications.*
2000 Mathematics Subject Classification: 03B20.
Key words and phrases: linear logic, $\lambda\mu$-calculus, Curry-Howard isomorphism.

properties. Finally we show that the relationship between FILL_μ and FILL is as expected: all derivations of the sequent calculus FILL can be seen as terms of the FILL_μ calculus. Conversely all decorations can be removed from FILL_μ-terms to yield sequent-calculus FILL derivations.

2. The system FILL

The system FILL is a variant of (multiplicative) linear logic proposed in [HdP93] whose logical connectives are all independent, that is, they are not interderivable, as they are in (multiplicative) classical linear logic. The relationship between FILL and classical linear logic is somewhat analogous to the relationship between intuitionistic logic and classical logic. In classical logic all the connectives can be expressed in terms of implication and negation, whereas in intuitionistic logic, conjunction, disjunction and implication are all independent connectives. Only negation is defined in terms of implication and the constant $\perp$ for *falsum*. In FILL the linear negation $A^\perp$ is defined as $A \multimap \perp$ and it is not an involution, thus $A \vdash A^{\perp\perp}$ but not vice-versa, as is the case in classical linear logic.

Hyland and de Paiva only deal with the multiplicative fragment of FILL. We will do the same here, but following the dependency-style formulation of Braüner and de Paiva [BdP98]. Formulae of FILL are defined by the grammar:

$$S \quad ::= \quad S \otimes S \mid I \mid S \invamp S \mid \perp \mid S \multimap S$$

The sequent calculus rules for FILL are as follows, where the asterisk $*$ in the implication right rule indicates the side-condition that A does not depend on any formula in Δ (see below).

$$\frac{}{A \vdash A}$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\otimes_L \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\otimes_R$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, I \vdash \Delta} I_L \qquad \frac{}{\vdash I} I_R$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \invamp B \vdash \Delta, \Delta'}\invamp_L \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \invamp B, \Delta}\invamp_R$$

$$\frac{}{\perp \vdash}\perp_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta}\perp_R$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'}\multimap_L \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta}\multimap_R{}^*$$

To explain the side-condition on the implication right rule, we need a notion of *dependency*, relating formulae occurrences. Basically we must define, given a proof of a sequent $\Gamma, B \vdash A, \Delta$, in classical linear logic when the succedent formula occurrence A depends on the antecedent formula occurrence B. This notion of dependency between formulae occurrences in the sequent, when properly defined, allows us to express the constructive property that characterizes FILL. Intuitively we can say that "genuine" dependencies start in axioms, constants do not introduce dependencies and dependencies "percolate" through a proof as expected. The formal definition of the set of dependencies $Dep_\tau(A)$ in a proof [BdP98] is recalled in the appendix.

Summing up: the implication right rule in classical linear logic (like the one in classical logic) allows any (linear) implications whatsoever. The implication right rule in intuitionistic linear logic enforces the existence of a single conclusion on the sequent. The implication right rule for FILL is more liberal than a single formula in the consequent, but more restricted than the classical linear logic rule.

3. The system $\lambda\mu\nu$

The $\lambda\mu$-calculus was introduced by Parigot [Par92] and extended to deal with multiplicative disjunction by Pym and Ritter [RPW00a, RPW00b]. In this section, we briefly recap both $\lambda\mu$ and Pym and Ritter's extension $\lambda\mu\nu$.

The original $\lambda\mu$-calculus provides a term calculus for the implicational fragment of classical propositional natural deduction: *i.e.*, realizers for a calculus in which multiple conclusioned sequents can be derived. The typing judgements in the $\lambda\mu$-calculus are of the form $\Gamma \vdash t\colon A\,,\,\Delta$, where Γ is a context familiar from the typed λ-calculus and Δ is a context containing types indexed by names, $\alpha, \beta, \ldots$, distinct from variables. These types are written as A^α, *etc.*. The relationship of this typing judgement with classical logic, the "propositions-as-types correspondence", is simply stated as follows: there is a term t such that the judgement

$$x_1\colon A_1, \ldots, x_m\colon A_m \vdash t\colon A, B_1^{\beta_1}, \ldots, B_n^{\beta_n}$$

is provable in the $\lambda\mu$-calculus if and only if

$$A_1 \wedge \ldots \wedge A_m \wedge \neg B_1 \wedge \ldots \wedge \neg B_n \to A$$

is a classical propositional tautology.

The intuition for this term calculus is that each $\lambda\mu$-sequent has exactly one *active*, or principal, formula, A, on the right-hand side, *i.e.*, the leftmost one, which is the formula upon which all introduction and elimination rules operate. This formula is the type of the term t. The basic grammar of $\lambda\mu$ terms is as follows:

$$t \;::=\; x \mid \lambda x\colon A\,.\,t \mid tt \mid [\alpha]t \mid \mu\alpha\,.\,t.$$

The corresponding inference rules of the $\lambda\mu$-calculus are given below.

$$\frac{}{\Gamma, x\colon A \vdash x\colon A, \Delta}\ Ax$$

$$\frac{\Gamma, x\colon A \vdash t\colon B, \Delta}{\Gamma \vdash \lambda x\colon A.t\colon A \to B, \Delta} \to I \qquad \frac{\Gamma \vdash t\colon A \to B, \Delta \qquad \Gamma \vdash s\colon A, \Delta}{\Gamma \vdash ts\colon B, \Delta} \to E$$

$$\frac{\Gamma \vdash t\colon A, \Delta}{\Gamma \vdash [\alpha]t\colon \bot, A^\alpha, \Delta}\ \mathit{freeze} \qquad \frac{\Gamma \vdash t\colon \bot, A^\alpha, \Delta}{\Gamma \vdash \mu\alpha.t\colon A, \Delta}\ \mathit{unfreeze}$$

In the rules *freeze* and *unfreeze* the type-name pair A^α may already be contained in Δ. If it is, the rule *freeze* models contraction and the rule *unfreeze* models weakening on the right hand side of the sequent. A context Γ is a set of pairs $x\colon A$, where x is a variable and A is a type, and a context Δ is a set of pairs A^α, where A is a type and α a name. We require that no name and no variable occurs twice in any given context.

The term $[\alpha]t$ realizes the introduction of a name. The term $\mu\alpha.[\beta]t$ realizes the exchange operation:

$$\frac{\Gamma \vdash t\colon B, A^\alpha, \Delta}{\Gamma \vdash \mu\alpha.[\beta]t\colon A, B^\beta, \Delta}\quad \text{Exchange}$$

i.e., if A^α was part of "side-context" of the succedent before the exchange, then A is the principal formula of the succedent after the exchange. Taken together, these terms also provide a notation for the realizers of contractions and weakenings on the right of a multiple-conclusioned calculus. It is easy to detect whether a formula B^β in the right-hand side of the sequent is, in fact, superfluous, *i.e.*, there is a derivation of $\Gamma \vdash t\colon A, \Delta'$ where Δ' does not contain B. The formula B is superfluous if β is not a free name in t. The negation of a formula A is modelled in the $\lambda\mu$-calculus as $A \to \bot$, and the two rules for $\bot$ express the fact that $\bot$ can be added and removed to the succedent at any time.

Pym and Ritter's variation on $\lambda\mu$ presented below has two aspects. Firstly, in addition to implicational types, they include both *conjunctive* (product) and *disjunctive* (disjoint sum) types. The addition of the conjunctive types follows the standard method for adding products to the simply-typed λ-calculus and is omitted. The addition of disjunctive types requires a more subtle approach. The key point in the addition of disjunctive types is naturally explained in the setting of the multiple-conclusioned sequent calculus. Their formulation, based on that found in Gentzen's classical sequent calculus LK [Gen34] and also in Dummett's multiple-conclusioned intuitionistic sequent calculus [Dum80] exploits the presence of multiple conclusions via the introduction rule for disjunctions.

$$\frac{\Gamma \vdash A_1, A_2, \Delta}{\Gamma \vdash A_1 \vee A_2, \Delta}.$$

For the $\lambda\mu$-calculus, the latter formulation presents a new difficulty. Suppose the $\lambda\mu$-sequent

$$\Gamma \vdash t\colon A, B^\beta, \Delta$$

is to be the premiss of the ∨I rule. In forming the disjunctive active formula $A \vee B$, we move the named formula B^β from the context to the active position. Consequently, ∨I is formulated as a binding operation and we introduce the following additional constructs, to form the grammar of $\lambda\mu\nu$-terms:

$$t \;::=\; \langle\beta\rangle t \mid \nu\beta \,.\, t.$$

The term $\nu\beta \,.\, t$ introduces a disjunction and the term $\langle\beta\rangle t$ eliminates one. The associated inference rules are given by

$$\frac{\Gamma \vdash t\colon A, B^\beta, \Delta}{\Gamma \vdash \nu\beta.t\colon A \vee B, \Delta} \vee I \qquad \frac{\Gamma \vdash t\colon A \vee B, \Delta}{\Gamma \vdash \langle\beta\rangle t\colon A, B^\beta, \Delta} \vee E$$

The definition of the reduction rules requires not only the standard substitution $t[s/x]$, but also a *substitution for names* $t[s/[\alpha]u]$, which intuitively indicates the term t with all occurrences of a subterm of the form $[\alpha]u$ replaced by s.

Parigot gives only reduction rules for β-reduction. Pym and Ritter also provide η-rules as expansions, meaning that each term of functional type is transformed into a λ-abstraction, each term of product type into a pair and each term of sum type into a term $\nu\beta.t'$. These η-rules generate critical pairs which give rise to additional reduction rules. All the reduction rules are collected in the appendix.

The system described by Pym and Ritter satisfy two key properties of reduction systems: namely *confluence* and *strong normalization. Local* confluence is the property that any two one-step reducts of a term have a common reduct and *confluence* is the property that any two reducts of a term have a common reduct. *Normalization* is the property that any term has a terminating reduction sequence and *strong normalization* is the property that all reduction sequences for any given term terminate. Both local confluence and strong normalization were proved by Pym and Ritter in the paper cited.

4. A natural deduction system for FILL

We use the $\lambda\mu$-calculus as a blueprint to provide a natural deduction version for FILL, which we call FILL_μ. Since the $\lambda\mu$-calculus calculus was originally developed for classical logic, we need a modification to be able to use it for FILL. FILL arises from classical linear logic via a restriction, that is similar to the one turning intuitionistic logic into a subsystem of classical logic, discussed in [RPW00a]. The main difference between classical and intuitionistic (propositional) systems is the implication right rule, where the intuitionistic restriction is that the right-hand side consists of a single formula and the classical rule has no restrictions. Hence a classical derivation is intuitionistic if for all instances of the implication right rule all side formulae on the right-hand side of the sequent arise by weakening. In Pym and Ritter [RPW00a] a syntactic criterion for a $\lambda\mu$-term to satisfy this condition is given. They define a notion of weakening occurrence of a $\lambda\mu$-name, with the intention that if a name has got only weakening occurrences in a term, the corresponding formula in the right-hand side arises by weakening. For FILL_μ we have to capture dependencies rather than terms which arise by weakening or not. Hence we decorate types with the list of variables which they depend on and restate the linear implication right rule accordingly.

As a consequence FILL_μ has judgements of the form

$$\Gamma \vdash M\colon A^{\sigma}, \alpha_1\colon A_1^{\sigma_1}, \ldots, \alpha_n\colon A_n^{\sigma_n}$$

where $\sigma, \sigma_1, \ldots$ stand for decorations on types and $\alpha \ldots$ for names. As we need to keep track of how dependencies are propagated, the decorations are not only variables but are given by the grammar

$$\sigma ::= x \mid \lambda x.\sigma \mid \sigma\sigma \mid \sigma \otimes \sigma \mid \mathsf{Fst}(\sigma) \mid \mathsf{Snd}(\sigma) \mid \sigma \parr \sigma \mid l(\sigma) \mid r(\sigma) \mid \mathtt{nodep}$$

We have the following reduction relation on decorations:

$$\begin{array}{rcl} (\lambda x.\tau)\sigma & \rightsquigarrow & \tau[\sigma/x] \\ \mathsf{Fst}(\sigma\otimes\tau) & \rightsquigarrow & \sigma \\ \mathsf{Snd}(\sigma\otimes\tau) & \rightsquigarrow & \tau \\ l(\sigma\parr\tau) & \rightsquigarrow & \sigma \\ r(\sigma\parr\tau) & \rightsquigarrow & \tau \end{array}$$

Decorations can be seen as terms in a suitable simply-typed λ-calculus. Hence there exists a unique normal form for decorations. ¿From now on, we assume that all decorations are in normal form.

The typing judgements of FILL_μ are the following ones:

$$\frac{}{x\colon A \vdash x\colon A^x}$$

$$\frac{\Gamma, x\colon A \vdash M\colon B^{\tau}, \Delta}{\Gamma \vdash \lambda x\colon A.M\colon (A \multimap B)^{\lambda x.\tau}, \Delta}\; x \notin NF(\Delta)$$

$$\frac{\Gamma_1 \vdash M_1\colon (A \multimap B)^{\sigma} \qquad \Gamma_2 \vdash M_2\colon B^{\tau}, \Delta_1}{\Gamma_1, \Gamma_2 \vdash MN\colon B^{\sigma\tau}, \Delta_1, \Delta_2}$$

$$\frac{\Gamma_1 \vdash M_1\colon A^{\sigma}, \Delta_1 \qquad \Gamma_2 \vdash M_2\colon B^{\tau}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash M \otimes N\colon (A \otimes B)^{\sigma\otimes\tau}, \Delta_1, \Delta_2}$$

$$\frac{\Gamma_1 \vdash M_1\colon A \otimes B^{\sigma}, \Delta_1 \qquad \Gamma_2, x\colon A, y\colon B \vdash M_2\colon C^{\tau}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \mathtt{let}\ M\ \mathtt{be}\ x \otimes y\ \mathtt{in}\ N\colon (C, \Delta_2)^{[\mathsf{Fst}(\sigma)/x, \mathsf{Snd}(\sigma)/y]}, \Delta_1}$$

$$\frac{}{\vdash *\colon I^{\mathtt{nodep}}}$$

$$\frac{\Gamma_1 \vdash M\colon I^{\sigma}, \Delta_1 \qquad \Gamma_2 \vdash N\colon A^{\sigma}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \mathtt{let}\ M\ \mathtt{be}\ *\ \mathtt{in}\ N\colon A^{\sigma}, \Delta_1, \Delta_2}$$

$$\frac{\Gamma_1 \vdash M\colon A^\sigma, \alpha\colon B^\tau, \Delta}{\Gamma \vdash \nu\alpha.M\colon (A \parr B)^{\sigma \parr \tau}, \Delta}$$

$$\frac{\Gamma \vdash M\colon (A \parr B)^\sigma, \Delta}{\Gamma \vdash \langle\alpha\rangle M\colon A^{l(\sigma)}, \alpha\colon B^{r(\sigma)}, \Delta}\ \alpha \notin \Delta$$

$$\frac{\Gamma \vdash M\colon A^\sigma, \Delta}{\Gamma \vdash [\alpha]M\colon \bot^{\texttt{nodep}}, \alpha\colon A^\sigma, \Delta}$$

$$\frac{\Gamma \vdash M\colon \bot^\sigma, \alpha\colon A^\tau, \Delta}{\Gamma \vdash \mu\alpha.M\colon A^\tau, \Delta\colon}$$

The important restriction in the typing rules is the side condition $x \notin NF(\Delta)$ in the linear implication right rule, meaning that the variable x does not appear in the decorations of Δ. This condition captures the essence of FILL: the formulae in Δ do not depend on the formula A.

The definition of the reduction rules requires not only the standard substitution $t[s/x]$, but also a substitution for names $t[s/[\alpha]u]$, which intuitively indicates the term t with all occurrences of a subterm of the form $[\alpha]u$ replaced by s. To define this notion, we need the notion of a term with holes. Such a term $\mathcal{C}$ with holes of type A is a FILL_μ-term which may have also the additional term constructor $\_$ with the rule $\Gamma \vdash \_\colon A, \Delta$. The term $\mathcal{C}(u)$ denotes the term $\mathcal{C}$ with the holes textually (with possible variable capture) replaced by u. Then we define $t[\mathcal{C}(u)/[\alpha]u]$, where α is a name and u is a metavariable, by

$$\begin{array}{rcl} x[\mathcal{C}(u)/[\alpha]u] & = & x \\ ([\alpha]t)[\mathcal{C}(u)/[\alpha]u] & = & \mathcal{C}(t[\mathcal{C}(u)/[\alpha]u]) \\ (\langle\alpha\rangle t)[[\mathcal{C}(u)/[\alpha]u]] & = & \mu\gamma.\mathcal{C}(\mu\alpha.[\gamma]\langle\alpha\rangle t[\mathcal{C}(u)/[\alpha]u]) \end{array}$$

and define $t[\mathcal{C}(u)/[\alpha]u]$ on all other expressions t by pushing the replacement inside.

We need the usual substitution lemma for the FILL_μ-calculus. As we have not only substitution for variables, but also for names, we obtain several cases for the lemma, in the same way as in the $\lambda\mu\nu$-calculus.

4.1. LEMMA.

(i) Assume $\Gamma_1, x\colon A \vdash M\colon B, \Delta_1$ *and* $\Gamma_2 \vdash N\colon A^\sigma, \Delta_2$. *Then we have also* $\Gamma_1, \Gamma_2 \vdash M[N/x]\colon (B, \Delta_1, \Delta_2)[\sigma/x]$.

(ii) Assume $\Gamma_1 \vdash M\colon A, \alpha\colon (B \multimap C)^\sigma, \Delta_1$ *and* $\Gamma_2 \vdash N\colon B^\tau, \Delta_2$. *Then we have also* $\Gamma_1, \Gamma_2 \vdash M[[\beta]RN/[\alpha]R]\colon A, \beta\colon C^{\sigma\tau}, \Delta_1, \Delta_2)$.

(iii) Assume $\Gamma_1 \vdash \mu\alpha.M\colon (A \otimes B)^\sigma, \Delta_1$ *and*

$$\Gamma_2, x\colon A, y\colon B \vdash N\colon (C, \Delta_2)[\mathsf{Fst}(\sigma)/x, \mathsf{Snd}(\sigma)/y]\ .$$

Then also $\Gamma_1, \Gamma_2 \vdash \mu\beta.M[[\beta]\texttt{let}\ R\ \texttt{be}\ x \otimes y\ \texttt{in}\ N/[\alpha]R]\colon C, \Delta_1, \Delta_2$.

(iv) Assume $\Gamma \vdash M\colon A, \alpha\colon (B \parr C)^\sigma, \Delta_1$. *Then we have also*

$$\Gamma \vdash M[[\beta]\langle\gamma\rangle N/[\alpha]N]\colon A, \beta\colon B^{l(\sigma)}, \gamma\colon C^{r(\sigma)}, \Delta\ .$$

Proof. Induction over the structure of M. ■

We have the following reduction rules in addition to the standard β-reduction rules:

$$\begin{array}{rcl}
(\mu\alpha.M)N & \leadsto & M[[\beta]UN/[\alpha]U] \\
\langle\beta\rangle\mu\alpha.M & \leadsto & M[[\gamma]\langle\beta\rangle N/[\alpha]N] \\
(\mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ N)R & \leadsto & \mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ NR \\
\langle\alpha\rangle(\mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ N) & \leadsto & \mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ \langle\alpha\rangle N \\
\mathtt{let}\ \mathtt{let}\ M_1\ \mathtt{be}\ x_1{\otimes}y_1\ \mathtt{in}\ N_1\ \mathtt{be}\ x_2{\otimes}y_2\ \mathtt{in}\ N_2 & \leadsto & \mathtt{let}\ M_1\ \mathtt{be}\ x_1{\otimes}y_1 \\
 & & \quad \mathtt{in}\ \mathtt{let}\ N_1\ \mathtt{be}\ x_2{\otimes}y_2\ \mathtt{in}\ N_2 \\
\mu\alpha.\mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ N & \leadsto & \mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ \mu\alpha.N \\
 & & \quad \text{if } \alpha \notin FN(M)
\end{array}$$

These rules correspond to commuting conversions in the sequent calculus which are necessary to obtain cut-elimination. We will call these rules ζ-rules, and we will use the term $\beta\zeta$-reduction for a β-or a ζ-reduction.

Now we can show the property usually called *subject reduction.*

4.2. Proposition. *Assume M is a FILL_μ-term, and $M \leadsto N$. Then N is also a FILL_μ-term.*

Proof. Consider each reduction in turn. All β-rules are done via the appropriate substitution lemma. ■

Now we turn to the proof that FILL_μ with the $\beta\zeta$-reductions is strongly normalising and confluent. The strong normalisation and confluence of $\lambda\mu$-calculus is instrumental in this proof, as is the subject reduction for FILL_μ.

4.3. Theorem. *The calculus FILL_μ with $\beta\zeta$-reductions is strongly normalising and confluent.*

Proof. We start by showing strong normalisation. Firstly, we define an embedding of FILL_μ into the $\lambda\mu\nu$-calculus by replacing a tensor term $M{\otimes}N$ with the product term $\langle M, N\rangle$ and the term $\mathtt{let}\ M\ \mathtt{be}\ x{\otimes}y\ \mathtt{in}\ N$ with the substitution $N[\pi(M)/x, \pi'(M)/y]$. This embedding only creates more β-redexes, and two terms which are equal modulo ζ-redexes are mapped to equal $\lambda\mu\nu$-terms. Hence any infinite sequence of $\beta\zeta$-reductions contains only a finite number of β-reductions, and there is a term M such that all further reductions are ζ-reductions. Secondly, we define a weight of a syntax tree by assigning 2 to all nodes except a let-node, which is assigned the weight 1. The weight of a tree where the left-hand side has weight w_1, the right-hand side has weight w_2 and the node has weight w, is $w + 2w_1 + w_2$. It is now easy to see that the weight of the left-hand side of any ζ-rule is bigger than the weight of the right-hand side. The fact that the weight of the left-hand side is higher than the weight of the right-hand side ensures that the left-hand side of rules which push `let`-constructors from the left-hand side of the syntax tree to the right-hand side of the syntax tree has a higher weight than the right-hand side of such a rule. The lower weight of the `let`-constructors ensures that also the left-hand side of rules which push `let`-constructors down the syntax tree has a higher weight than the right-hand side. Hence there is no infinite sequence of ζ-reductions. Hence there is also no infinite sequence of $\beta\zeta$-reductions.

Now we turn to confluence. Local confluence can be shown by considering all critical pairs, and strong normalization then implies confluence. ∎

5. Relating FILL and FILL_μ

Next we show that FILL_μ proves exactly the same theorems as FILL. For this, we show how to translate FILL_μ-derivations to $\lambda\mu$-calculus-terms. We do this by induction over the structure of FILL_μ-derivations. Note that as FILL is given by a sequent calculus, this translation does a translation from a sequent calculus into natural deduction as well.

In the translation below the decorations are ignored. The reason is that they are not necessary for the definition of the translation but only to ensure that the result of the translation of a FILL-derivation is a FILL_μ-term.

We shall use the following notation: if ϕ is a derivation whose last rule is R applied to the derivations $\phi_1, \ldots, \phi_n$, we write $(\phi_1, \ldots, \phi_n); R$ for ϕ.

5.1. Definition. *Let $\phi\colon \Gamma \vdash A, \Delta$ be a FILL sequent derivation and suppose that each occurrence of a formula in Γ and Δ has a label, i.e., the contexts Γ and Δ satisfy $\Gamma = x_1\colon A_1, \ldots, x_n\colon A_n$ and $\Delta = B_1^{\beta_1}, \ldots, B_m^{\beta_m}$. (These labels turn into variables and names in the FILL_μ-calculus, hence we also use them for the derivations.) We define a FILL_μ-term ϕ^{fl} satisfying $\Gamma \vdash \phi^{fl}\colon A, \Delta$ by induction over the structure of ϕ as follows (note the clause for the exchange rule):*

Axiom: *Suppose $\phi : x\colon A \vdash A, \Delta$ is an axiom, then $\phi^{fl} \stackrel{\text{def}}{=} x$;*

Exchange: *Suppose $\phi\colon \Gamma \vdash A, B^\beta, \Delta$, and*

$$\phi' = \phi; \mathtt{exc}\colon \Gamma \vdash B, A^\alpha, \Delta.$$

We define ϕ'^{fl} to be $\mu\beta.[\alpha]\phi^{fl}$, where α is a name which does not occur freely in ϕ^{fl};

$\otimes L$**:** *Suppose we have the derivation*

$$\frac{\phi\colon \Gamma, x\colon A, y\colon B \vdash A, \Delta}{\phi; \otimes L\colon \Gamma, z\colon A \otimes B \vdash A, \Delta}\otimes L,$$

then the corresponding $\lambda\mu$-term is

$$\phi; \otimes L^{fl} \stackrel{\text{def}}{=} \mathtt{let}\ x \otimes y\ \mathtt{be}\ z\ \mathtt{in}\ \phi^{fl};$$

$\otimes R$**:** *Suppose we have the derivation*

$$\frac{\phi\colon \Gamma_1 \vdash A, \Delta_1 \qquad \psi\colon \Gamma_2 \vdash B, \Delta_2}{(\phi, \psi);\ \ R\colon \Gamma_1, \Gamma_2 \vdash A \otimes B, \Delta_1, \Delta_2} \otimes R,$$

then we define

$$(\phi, \psi); \otimes R^{fl} \stackrel{\text{def}}{=} \phi^{fl} \otimes \psi^{fl};$$

$I\ L$**:** *Suppose we have the derivation*

$$\frac{\Gamma \vdash \Delta}{\Gamma, I \vdash \Delta}\, I\ L$$

then we define

$$\phi; I\ L^{fl} \stackrel{\text{def}}{=} \texttt{let}\ x\ \texttt{be}\ *\ \texttt{in}\ \phi^{fl}\ ;$$

$I\ R$**:** *Suppose we have the derivation*

$$\frac{}{\vdash I}\, I\ R$$

then we define

$$I\ R^{fl} \stackrel{\text{def}}{=} *\ ;$$

$\multimap L$**:** *Suppose we have the derivation*

$$\frac{\phi\colon \Gamma_1 \vdash A, \Delta_1 \qquad \psi\colon \Gamma_2, w\colon B \vdash C, \Delta_2}{(\phi, \psi); \multimap L\colon \Gamma_1, \Gamma_2, x\colon A \multimap B \vdash C, \Delta_1, \Delta_2} \multimap L$$

then we define $(\phi, \psi); \multimap L^{fl}$ *to be* $\psi^{fl}[x\phi^{fl}/w]$*;*

$\multimap R$**:** *Suppose we have the derivation*

$$\frac{\phi\colon \Gamma, x\colon A \vdash B, \Delta}{\phi; \multimap R\colon \Gamma \vdash A \multimap B, \Delta} \multimap R$$

then we define $[\![\phi; \multimap R]\!]$ *to be* $\lambda x\colon A.[\![\phi]\!]$*;*

$⅋R$**:** *Assume we have a derivation*

$$\frac{\phi\colon \Gamma \vdash A, B^{\beta}, \Delta}{\phi; ⅋R\colon \Gamma \vdash A ⅋ B, \Delta}$$

then we define $\phi; ⅋R^{fl} = \nu\beta.[\![\phi]\!]$*;*

$\invamp L$: *Assume we have a derivation*

$$\frac{\phi_1 : \Gamma_1, x : A \vdash \Delta_1 \qquad \phi_2 : \Gamma_2, y : B \vdash \Delta_2}{(\phi_1, \phi_2); \invamp L : \Gamma_1, z : A \invamp B \vdash \Delta_1, \Delta_2} \invamp L$$

Then $(\phi_1, \phi_2; \invamp L)^{fl}$ *is the* FILL_μ*-term*

$$\phi_1^{fl}[\mu\alpha.[\gamma][\mu\beta.[\alpha]\langle\beta\rangle z/y]/x] \ ;$$

$\perp L$: *Suppose we have the derivation*

$$\frac{}{\perp \vdash} \perp L$$

then we define

$$\perp L^{fl} \stackrel{\text{def}}{=} \mu\alpha.x \ ;$$

$\perp R$: *Suppose we have the derivation*

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta} \perp R$$

then we define

$$\phi; \perp R^{fl} \stackrel{\text{def}}{=} [\alpha]\phi^{fl} \ .$$

So far we have not used the decorations at all, neither for FILL nor for FILL_μ. They are important when we show that the translation of a FILL-derivation is a FILL_μ-term. In particular, the restriction of the $\multimap R$-rule is captured by the side condition of the rule for λ-abstraction of FILL_μ.

5.2. Theorem. *Assume* $\phi : \Gamma \vdash \Delta$ *is a* FILL*-derivation. Then* ϕ^{fl} *is a* FILL_μ*-term.*

Proof. Induction over the derivation. ∎

As an example for a FILL-derivation which is not an intuitionistic derivation consider the derivation

$$\frac{\dfrac{\dfrac{u : A \vdash A^u \qquad v : B \vdash B^v}{z : A \invamp B \vdash A^{l(z)}, B^{r(z)}} \invamp L \qquad x : C \vdash C^x}{z : A \invamp B, x : C \vdash (A \otimes C)^{l(z) \otimes x}, B^{r(z)}} \otimes R}{z : A \invamp B \vdash (C \multimap (A \otimes C))^{\lambda x. l(z) \otimes x}, B^{r(z)}} \vdash R$$

The corresponding FILL_μ-term is

$$z : A \invamp B \vdash \lambda x.(\langle\beta\rangle z) \otimes x : (C \multimap (A \otimes C))^{\lambda x. l(z) \otimes x}, \beta : B^{r(z)}$$

The crucial point is that the name β depends only on z and not on x. This ensures that this term is a FILL_μ-term. This derivation is not an intuitionistic derivation, which can be derived syntactically from the fact that the operator $\langle\beta\rangle$ is applied to a variable. This means that there is a second formula on the RHS when the $\multimap R$-rule is applied.

In the other direction, we define a translation from FILL_μ-terms to FILL-derivation by induction over the structure of FILL_μ-terms. Again, the decorations are not used in the definition of the translation, only in the proof that the translation transforms FILL_μ-terms to FILL-derivations.

x: Assume $x{:}\,A \vdash x{:}\,A$. Then the derivation x^{lf} is given by

$$A \vdash A\;;$$

$\mu\alpha.M$: Assume $M^{lf} = \phi{:}\,\Gamma \vdash \bot, A, \Delta$. Then $\mu\alpha.M^{lf}$ is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma \vdash \bot, A, \Delta} \qquad \dfrac{}{\bot \vdash}\,\bot L}{\Gamma \vdash A, \Delta}\,\mathit{cut}\;;$$

$[\alpha]M$: Assume $M^{lf} = \phi{:}\,\Gamma \vdash A, \Delta$. Then $[\alpha]M^{lf}$ is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma \vdash A, \Delta}}{\Gamma \vdash \bot, A, \Delta}\,\bot R\;;$$

$\lambda x{:}\,A.M$: Assume $M^{lf} = \phi{:}\,\Gamma, A \vdash B, \Delta$. Then $\lambda x{:}\,A.M^{lf}$ is the derivation

$$\dfrac{\phi}{\Gamma \vdash A \multimap B, \Delta}\,\multimap R;$$

MN: Assume $M^{lf} = \phi$, $N^{lf} = \psi$, $\Gamma_1 \vdash M{:}\,A \multimap B, \Delta_1$ and $\Gamma_2 \vdash N{:}\,A, \Delta_2$. Then MN^{lf} is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma_1 \vdash A \multimap B, \Delta_1} \qquad \dfrac{\Gamma_2 \vdash A, \Delta_2 \qquad \dfrac{A \vdash A \qquad B \vdash B}{A \multimap B, A \vdash B}\,\multimap L}{\Gamma_2, A \multimap B \vdash B, \Delta_2}}{\Gamma_1, \Gamma_2 \vdash B, \Delta_1, \Delta_2}\,\mathit{cut}\;;$$

$M \otimes N$: Assume $M^{lf} = \phi{:}\,\Gamma_1 \vdash A, \Delta_1$, and $N^{lf} = \psi{:}\,\Gamma_2 \vdash B, \Delta_2$. Then $M \otimes N^{lf}$ is the derivation

$$\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad \Gamma_2 \vdash B, \Delta_2}{\Gamma_1, \Gamma_2 \vdash A \otimes B, \Delta_1, \Delta_2}\,\otimes R\;;$$

$\mathtt{let}\ M\ \mathtt{be}\ x\otimes y\ \mathtt{in}\ N$: Assume that M and N satisfy $M^{lf} = \phi\colon \Gamma_1 \vdash A\otimes B, \Delta_1$, and $N^{lf} = \psi\colon \Gamma_2, A, B \vdash C, \Delta_2$. Then $\mathtt{let}\ M\ \mathtt{be}\ x\otimes y\ \mathtt{in}\ N^{lf}$ is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma_1 \vdash A\otimes B, \Delta_1} \qquad \dfrac{\Gamma_2, A, B \vdash C, \Delta_2}{\Gamma_2, A\otimes B \vdash C, \Delta_2}\otimes L}{\Gamma_1, \Gamma_2 \vdash C, \Delta_1, \Delta_2}\, cut\ ;$$

$*$: $*^{lf} = I\ R$;

$\mathtt{let}\ M\ \mathtt{be}\ *\ \mathtt{in}\ N$: Assume $M^{lf} = \phi\colon \Gamma_1 \vdash I, \Delta_1$, and $N^{lf} = \psi\colon \Gamma_2 \vdash A, \Delta_2$. Then $\mathtt{let}\ M\ \mathtt{be}\ *\ \mathtt{in}\ N^{lf}$ is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma_1 \vdash I, \Delta_1} \qquad \dfrac{\Gamma_2 \vdash C, \Delta_2}{\Gamma_2, I \vdash C, \Delta_2}\, I\ L}{\Gamma_1, \Gamma_2 \vdash C, \Delta_1, \Delta_2}\, cut\ ;$$

$\nu\alpha.M$: Assume $M^{lf} = \phi\colon \Gamma \vdash A, B, \Delta$. Then $\nu\alpha.M^{lf}$ is the derivation

$$\dfrac{\phi}{\Gamma \vdash A\parr B, \Delta}\parr R\ ;$$

$\langle\alpha\rangle M$: Assume $M^{lf} = \phi\colon \Gamma \vdash A\parr B, \Delta$. Then $\langle\alpha\rangle M^{lf}$ is the derivation

$$\dfrac{\dfrac{\phi}{\Gamma \vdash A\parr B, \Delta} \qquad \dfrac{A \vdash A \qquad B \vdash B}{A\parr B \vdash A, B}\parr L}{\Gamma \vdash A, B, \Delta}\, cut\ .$$

We have the expected theorem:

5.3. Theorem. *Assume* $\Gamma \vdash M\colon A, \Delta$ *is* FILL_μ*-term. Then the derivation* M^{lf} *is a* FILL*-derivation.*

Proof. Induction over the structure of M. ■

6. A sound semantics for FILL_μ

So far we have only established that FILL and FILL_μ are logically equivalent, *i.e.*, they prove the same theorems. In this section we show that they are not equivalent as far as proofs are concerned. The reason is that the ν-construction in FILL_μ does not provide co-products. In fact, in [RP01] we show that if the ν-construction is a co-product, then the calculus is trivial in the sense that in any semantic model all terms of the same type are interpreted by the same element in the model.

However, we can show that weakly distributive categories with a right adjoint to the tensor product but without negation (which we call *full linear categories*) are a sound model of the FILL_μ-calculus. The reason is that the handling of dependencies which was crucial for the semantics of FILL also works for FILL_μ.

We now present the soundness proof, following closely [HdP93]. We repeat here the notion of independence from [HdP93].

6.1. Definition. *Suppose that we have a map of the form $f\colon A\otimes B \to C\invamp D$ in a weakly distributive category $\mathcal{L}$. We say that the object C is independent of B (for f) if there exists an object E of $\mathcal{L}$ and maps $g\colon A \to C\invamp E$ and $h\colon E\otimes B \to D$ such that the composite $(1_C\invamp h)\circ\delta\circ(g\otimes 1_B)$ is equal to f, where δ is the weak distributivity with domain $(C\invamp E)\otimes B$ and co-domain $C\invamp(E\otimes B)$.*

Now we show that this notion of independence is compatible with linear function spaces.

6.2. Lemma. *Suppose we have a map $f\colon A\otimes B \to C\invamp D$ in a full linear category $\mathcal{L}$. If C is independent of B for f, then there exists a morphism $\overline{f}\colon A \to C\invamp(B\multimap D)$.*

The operation $f \mapsto \overline{f}$ is crucial for defining the categorical semantics of λ-abstraction. However, we need another lemma, which states that this operation preserves other independences.

6.3. Lemma. *Suppose we have a map $f\colon A\otimes B\otimes C \to D\invamp E\invamp F$ in a full linear category $\mathcal{L}$. Suppose also that $D\invamp E$ is independent of C for f so that we have a map $\overline{f}\colon A\otimes B \to D\invamp E\invamp(C\multimap F)$. Then*

(i) If F is independent of B for f, then $C\multimap F$ is independent of B for $\overline{f}$;

(ii) If E is independent of B for f, then E is independent of B for $\overline{f}$.

Now we can define the semantics of $\mathsf{FILL}_\mu$by induction over the structure of derivations. The clause for λ-abstraction requires a lemma to show that this definition is well-formed. We do this immediately afterwards.

6.4. Definition. *By induction over the structure of derivations, we define the semantics of FILL_μin a weakly distributive category as follows:*

Types: *The type constructors of $\mathsf{FILL}_\mu$are modelled by the corresponding categorical constructions;*

x**:** $[\![x]\!] = \mathsf{Id}$*;*

$\otimes$**:** *Assume $[\![t_1]\!] = f\colon \Gamma_1 \to A_1$ and $[\![t_2]\!] = f_2\colon \Gamma_1 \to A_2$. Then define $[\![t_1\otimes t_2]\!]$ to be*

$$\Gamma_1\otimes\Gamma_2 \xrightarrow{f_1\bigotimes f_2} (A_1\invamp\Delta_1)\otimes(A_2\invamp\Delta_2) \xrightarrow{\delta} (A_1\otimes(A_1\invamp\Delta_2))\invamp\Delta_1$$
$$\xrightarrow{\pi\circ(\delta\invamp 1)} (A_1\otimes A_2)\invamp\Delta_1\invamp\Delta_2\ ;$$

$\mathtt{let}\ t\ \mathtt{be}\ a\otimes b\ \mathtt{in}\ s$**:** *Assume* $[\![t]\!]\colon \Gamma_1 \to A\otimes B, \Delta_1$ *and* $[\![s]\!]\colon \Gamma_2\otimes A\otimes B \to C\invamp\Delta_2$*. Then* $[\![\mathtt{let}\ t\ \mathtt{be}\ a\otimes b\ \mathtt{in}\ s]\!]$ *is defined as*

$$\Gamma_1\otimes\Gamma_2 \xrightarrow{[\![t]\!]\otimes 1} ((A\otimes B)\invamp\Delta_1)\otimes\Gamma_2 \xrightarrow{\delta} \Delta_1\invamp(A\otimes B\otimes\Gamma_2) \xrightarrow{1\invamp[\![s]\!]} C\invamp\Delta_1\invamp\Delta_2\ ;$$

λ-abstraction: *Assume* $[\![t]\!]\colon \Gamma\otimes A \to B\invamp\Delta$*. Define* $[\![\lambda a\colon A.t]\!]$ *to be* $\overline{[\![t]\!]}$*;*

ts**:** *Assume* $[\![t]\!]\colon \Gamma_1 \to (A\multimap B)\invamp\Delta_1$ *and* $[\![s]\!]\colon \Gamma_2 \to A\invamp\Delta_2$*. Then define* $[\![ts]\!]$ *to be*

$$\begin{aligned}&\Gamma_1\otimes\Gamma_2 \xrightarrow{[\![t]\!]\otimes[\![s]\!]} (A\multimap B\invamp\Delta_1)\otimes(A\invamp\Delta_2) \xrightarrow{\delta} ((A\multimap B\invamp\Delta_1)\otimes A)\invamp\Delta_2\\ &\xrightarrow{\delta\invamp 1} (A\multimap B\otimes A)\invamp\Delta_1\invamp\Delta_2 \xrightarrow{\mathsf{App}\invamp\mathsf{Id}} B\invamp\Delta_1\invamp\Delta_2\ ;\end{aligned}$$

$\mu\alpha.t$**:** *Assume* $\Gamma\vdash t\colon\bot, \alpha\colon B^{\tau}, \Delta$*. Then* $[\![\mu\alpha.t]\!]$ *is defined to be the morphism* $u\circ[\![t]\!]$*, where* u *is the unit of* $\invamp$*;*

$[\alpha]t$**:** *Assume* $\Gamma\vdash t\colon A, \Delta$*. Then* $[\![[\alpha]t]\!]$ *is defined to by* $u^{-1}\circ[\![M]\!]$*;*

$\nu\alpha.t$**:** $[\![\nu\alpha.t]\!] = [\![t]\!]$*;*

$\langle\alpha\rangle t$**:** $[\![\langle\alpha\rangle t]\!] = [\![t]\!]$*.*

Well-definedness of the semantics requires the following lemma, which shows that the category-theoretic notion of independence captures the syntactic notion of independence in FILL_{μ}.

6.5. Lemma. *Assume* $\Gamma, x\colon C \vdash t\colon A, \Delta$*, and assume also that* x *does not appear in* Δ*. Then* Δ *is independent for* C *in* $[\![t]\!]$*.*

Proof. Induction over the structure of t. Lemmata 6.2 and 6.3 are used for the λ-abstraction. ∎

Now the soundness proof is routine.

6.6. Theorem.

(i) Assume $\Gamma\vdash t\colon A, \Delta$*. Then* $[\![t]\!]$ *is a morphism from* $[\![\Gamma]\!]$ *to* $[\![A]\!]\invamp[\![\Delta]\!]$*;*

(ii) Assume $t = s$*. Then also* $[\![t]\!] = [\![s]\!]$*.*

Proof. Induction over the derivation. ∎

7. Conclusions

We have shown how to produce a Natural Deduction formulation for FILL, called FILL_μ, using a modification of Pym and Ritter's $\lambda\mu\nu$-calculus, itself an extension of Parigot's $\lambda\mu$-calculus.

The process of transforming Parigot's $\lambda\mu$-calculus into Pym and Ritter's $\lambda\mu\nu$-calculus and finally into FILL_μ-terms is complicated, but it works and provides a Curry-Howard correspondence for full intuitionistic linear logic. We can prove not only the essential properties of subject reduction and strong normalization for the FILL_μ-calculus but also that this really corresponds to the original (sequent-style) formulation of FILL. But given the somewhat convoluted process, we wonder if something simpler might work as well. This is part of our proposal for future work.

The interested reader may wonder how the terms of the FILL_μ-calculus relate to the terms of the classical linear λ-calculus, described by Bierman[Bie99]. The classical linear λ-calculus also arises from considering Parigot's ideas in the context of linear logic. But since Bierman is dealing with classical linear logic, he can adapt Parigot's ideas more directly. We believe, and this must be checked, that restricting FILL_μ to classical linear logic we obtain a calculus equivalent to the classical linear λ-calculus.

Two other question also suggest themselves. Firstly, since the restriction that characterizes FILL was also considered recently by Crolard [Cro04], in the context of modelling co-routines that do not access the local environment of other co-routines, we wonder whether our new term assignment for FILL can be used to model *linear* co-routines with the same non-interfering property. Secondly, Pratt [Pra03] presents a calculus for Chu-spaces which has two kinds of variables with equal status, one for (positive) assumptions and for (negative) evidence against a formula. He shows that this so-called "dialectic calculus" captures proofs of bi-implicational mulplicative linear logic but does not present reduction rules for this calculus. It would be interesting to check how this dialectic calculus relates to our FILL_μ calculus.

Appendix A: Dependencies in FILL

.1. DEFINITION. *Let τ be a proof in CLL whose end-sequent is $\Gamma \vdash \Delta$ and where A is a formula occurrence in Δ. The immediate subproofs of τ are denoted by τ_i. We define the set $Dep_\tau(A)$ (the formulae occurrences of Γ that A depends on, in the proof τ) by induction on τ. The definition is by cases in accordance with the table in Figure 1.*

If the derivation τ ends in the sequent $\Gamma \vdash \Delta$, and B and A are formula occurrences in Γ and Δ respectively, then we say that A *depends on* B *in* τ iff $B \in Dep_\tau(A)$. If τ and τ' are two derivations ending in the sequent $\Gamma \vdash \Delta$ in CLL, we say that the end-sequent of τ *has the same dependencies as* the end-sequent of τ' iff we for any formula occurrence C in Γ and any formula occurrence A in Δ, we have that A depends on C in τ iff A depends on C in τ'. Similarly say that the end-sequent of τ *has fewer dependencies than* the end-sequent of τ' iff we for any formula occurrence C in Γ and any formula occurrence

$$\frac{}{A \vdash A}\,Ax \qquad Dep_\tau(A) = A$$

$$\frac{\Gamma \vdash B, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma', \Gamma \vdash \Delta', \Delta}\,Cut \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A) & \text{if } A \in \Delta \\ Dep_{\tau_2}(A)[\{B\} \mapsto Dep_{\tau_1}(B)] & \text{if } A \in \Delta' \end{cases}$$

$$\frac{\Gamma, B, C \vdash \Delta}{\Gamma, B \otimes C \vdash \Delta}\,\otimes_L \qquad Dep_\tau(A) = Dep_{\tau_1}(A)[\{B, C\} \mapsto \{B \otimes C\}]$$

$$\frac{\Gamma \vdash B, \Delta \quad \Gamma' \vdash C, \Delta'}{\Gamma, \Gamma' \vdash B \otimes C, \Delta, \Delta'}\,\otimes_R \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A) & \text{if } A \in \Delta \\ Dep_{\tau_2}(A) & \text{if } A \in \Delta' \\ Dep_{\tau_1}(B) \cup Dep_{\tau_2}(C) & \text{if } A = B \otimes C \end{cases}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, I \vdash \Delta}\,I_L \qquad Dep_\tau(A) = Dep_{\tau_1}(A)$$

$$\frac{}{\vdash I}\,I_R \qquad Dep_\tau(I) = \emptyset$$

$$\frac{\Gamma, B \vdash \Delta \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma', B \parr C \vdash \Delta, \Delta'}\,\parr_L \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A)[\{B\} \mapsto \{B \parr C\}] & \text{if } A \in \Delta \\ Dep_{\tau_2}(A)[\{C\} \mapsto \{B \parr C\}] & \text{if } A \in \Delta' \end{cases}$$

$$\frac{\Gamma \vdash B, C, \Delta}{\Gamma \vdash B \parr C, \Delta}\,\parr_R \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A) & \text{if } A \in \Delta \\ Dep_{\tau_1}(B) \cup Dep_{\tau_1}(C) & \text{if } A = B \parr C \end{cases}$$

$$\frac{}{\bot \vdash}\,\bot_L \qquad \text{nothing to define}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}\,\bot_R \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A) & \text{if } A \in \Delta \\ \emptyset & \text{if } A = \bot \end{cases}$$

$$\frac{\Gamma \vdash B, \Delta \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma', B \multimap C \vdash \Delta, \Delta'}\,\multimap_L \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_2}(A)[\{C\} \mapsto \{B \multimap C\} \cup Dep_{\tau_1}(B)] \\ \qquad \text{if } A \in \Delta' \\ Dep_{\tau_1}(A) \\ \qquad \text{if } A \in \Delta \end{cases}$$

$$\frac{\Gamma, B \vdash C, \Delta}{\Gamma \vdash B \multimap C, \Delta}\,\multimap_R \qquad Dep_\tau(A) = \begin{cases} Dep_{\tau_1}(A)[\{B\} \mapsto \emptyset] & \text{if } A \in \Delta \\ Dep_{\tau_1}(C)[\{B\} \mapsto \emptyset] & \text{if } A = B \multimap C \end{cases}$$

Figure 1: Definition of dependencies $Dep_\tau(A)$

A in Δ have that A depends on C in τ entails that A depends on C in τ'.

Appendix B: Reduction rules of the $\lambda\mu\nu$-calculus

$$
\begin{array}{llcl}
\beta & (\lambda x\colon A.t)s & \rightsquigarrow & t[s/x] \\
\eta^{\rightarrow} & t & \rightsquigarrow & \lambda x\colon A.tx \\
\zeta^{\supset} & (\mu\alpha^{A\rightarrow B}.t)s & \rightsquigarrow & \mu\beta^{B}.t[[\beta]us/[\alpha]u] \\
 & (\mu\alpha^{A\rightarrow\perp}.t)s & \rightsquigarrow & t[us/[\alpha]u] \\
\eta^{\mu} & \mu\alpha.[\alpha]s & \rightsquigarrow & s \text{ if } \alpha \text{ not free in } s \\
\beta^{\mu} & [\gamma](\mu\alpha.s) & \rightsquigarrow & s[\gamma/\alpha] \\
\beta^{\wedge} & \pi(\langle t,s\rangle) & \rightsquigarrow & t \\
 & \pi'(\langle t,s\rangle) & \rightsquigarrow & s \\
\eta^{\wedge} & t & \rightsquigarrow & \langle\pi(t),\pi'(t)\rangle \\
\zeta^{\wedge} & \pi(\mu\alpha^{A\wedge B}.s) & \rightsquigarrow & \mu\beta^{A}.s[[\beta]\pi(u)/[\alpha]u] \\
 & \pi(\mu\alpha^{\perp\wedge B}.s) & \rightsquigarrow & s[\pi(u)/[\alpha]u] \\
 & \pi'(\mu\alpha^{A\wedge B}.s) & \rightsquigarrow & \mu\gamma^{B}.s[[\gamma]\pi'(u)/[\alpha]u] \\
 & \pi'(\mu\alpha^{A\wedge\perp}.s) & \rightsquigarrow & s[\pi'(u)/[\alpha]u] \\
\beta^{\vee} & \langle\beta\rangle(\nu\alpha.s) & \rightsquigarrow & s[\beta/\alpha] \\
\eta^{\vee} & t & \rightsquigarrow & \nu\alpha.\langle\alpha\rangle t \\
\zeta^{\vee} & \langle\beta\rangle\mu\gamma.t & \rightsquigarrow & \mu\alpha.t[[\alpha]\langle\beta\rangle s/[\gamma]s] \\
 & \langle\beta\rangle\mu\gamma^{\perp\vee B} & \rightsquigarrow & t[\langle\beta\rangle s/[\gamma]s]
\end{array}
$$

Standard variable-capture are conditions assumed. In the η-rules we assume that t is neither a λ-abstraction nor a product nor a term $\nu\alpha.t'$ and that t does not occur as the first argument of an application, of a projection π or π' or of some term $\langle\beta\rangle\_$. In the $\eta^{\multimap}$-, $\eta^{\wedge}$- and $\eta^{\vee}$-rules. In the $\eta^{\rightarrow}$-, $\eta^{\wedge}$- and $\eta^{\vee}$-rule we also assume that t is of function type, product type and sum type respectively.

References

[Bar96] M. Barr. The Chu construction. *Theory and Applications of Categories*, 2:17–35, 1996.

[BdP98] Torben Braüner and Valeria de Paiva. A formulation of linear logic based on dependency-relations. In *Proc. of Computer Science Logic '97*, Lecture Notes in Computer Science. Springer Verlag, 1998.

[Bie99] G. Bierman. A classical linear lambda calculus. *Theoretical Computer Science*, 227(1–2):43–78, 1999.

[Cro04] T. Crolard. A formulae-as-types interpretation of subtractive logic. *Journal of Logic and Computation*, 14(4):529–570, 2004.

[CS97] J.R.B. Cockett and R.A.G. Seely. Weakly distributive categories. *Journal of Pure and Applied Algebra*, 114(2):133–173, 1997.

[dP89] V.C.V de Paiva. A dialectica-like model of linear logic. In David H. Pitt, David E. Rydeheard, Peter Dybjer, Andrew M. Pitts, and Axel Poigné, editors, *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*. Springer, 1989.

[Dum80] M. Dummett. *Elements of Intuitionism.* OUP, Oxford, 1980.

[Gen34] G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, pages 176–210, 405–431., 1934.

[Gir95] J.-Y. Girard. Linear logic: its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995. London Mathematical Society Lecture Note Series 222, Proceedings of the 1993 Workshop on Linear Logic, Cornell Univesity, Ithaca.

[HdP93] M. Hyland and V. de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64(3):273–291, 1993.

[Par92] M. Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proc. Logic Progr. and Automatic Reasoning, St. Petersburg*, pages 190–201. Lecture Notes in Computer Science No. 624, Berlin, Heidelberg, New York, 1992.

[Pra03] V. Pratt. Chu spaces as a semantic bridge between linear logic and mathematics,. *Theoretical Computer Science*, 294:439–471, 2003.

[RP01] E. Ritter and D. Pym. On the semantics of classical disjunction. *Journal of Pure and Applied Algebra*, 159:315–338, 2001.

[RPW00a] E. Ritter, D. Pym, and L. A. Wallen. On the intuitionistic force of classical search. *Journal of Logic and Computation*, 10(2):173–207, 2000.

[RPW00b] E. Ritter, D. Pym, and L.A. Wallen. Proof-terms for classical and intuitionistic logic. *Theoretical Computer Science*, 232(1–2):299–333, 2000.

Palo Alto Research Center
USA

School of Computer Science
University of Birmingham, UK
Email: `paiva@parc.com`
`E.Ritter@cs.bham.ac.uk`

This article may be accessed via WWW at `http://www.tac.mta.ca/tac/` or by anonymous ftp at `ftp://ftp.tac.mta.ca/pub/tac/html/volumes/17/3/17-03.{dvi,ps,pdf}`

THEORY AND APPLICATIONS OF CATEGORIES (ISSN 1201-561X) will disseminate articles that significantly advance the study of categorical algebra or methods, or that make significant new contributions to mathematical science using categorical methods. The scope of the journal includes: all areas of pure category theory, including higher dimensional categories; applications of category theory to algebra, geometry and topology and other areas of mathematics; applications of category theory to computer science, physics and other mathematical sciences; contributions to scientific knowledge that make use of categorical methods.

Articles appearing in the journal have been carefully and critically refereed under the responsibility of members of the Editorial Board. Only papers judged to be both significant and excellent are accepted for publication.

The method of distribution of the journal is via the Internet tools `WWW/ftp`. The journal is archived electronically and in printed paper format.

Subscription information. Individual subscribers receive (by e-mail) abstracts of articles as they are published. Full text of published articles is available in .dvi, Postscript and PDF. Details will be e-mailed to new subscribers. To subscribe, send e-mail to `tac@mta.ca` including a full name and postal address. For institutional subscription, send enquiries to the Managing Editor, Robert Rosebrugh, `rrosebrugh@mta.ca`.

Information for authors. The typesetting language of the journal is TeX, and LaTeX2e is the preferred flavour. TeX source of articles for publication should be submitted by e-mail directly to an appropriate Editor. They are listed below. Please obtain detailed information on submission format and style files from the journal's WWW server at `http://www.tac.mta.ca/tac/`. You may also write to `tac@mta.ca` to receive details by e-mail.