Syntactic linearity and Linear hyperdoctrines for second-order intuitionistic Linear Logic (work in progress)

Alejandro Díaz-Caro[1,2], Malena Ivnisky[3,2], and Octavio Malherbe[2]

[1]Université de Lorraine, Inria/LORIA, France
[2]Universidad de la República, FIng, Uruguay
[3]Universidad de Buenos Aires, DC/ICC, Argentina

Abstract

We present a polymorphic Linear lambda-calculus as a proof language for second-order intuitionistic Linear Logic. The calculus is extended with scalar multiplication and term addition, which enable the proof of a linearity result at the syntactic level. The syntactic linearity yields a correspondence between proof-terms and linear functions. We develop a sound and adequate denotational semantics based on hyperdoctrines valued in semimodule-enriched Linear categories.

Introduction

The design of proof languages plays a central role in both logic and programming languages, particularly when reasoning about resource usage and algebraic computation. Linear Logic [9] provides a framework for such reasoning, but its standard proof-term calculi do not make linear properties explicit at the syntactic level, such as distributivity over addition or scalar multiplication. A linear calculus with syntactic linear properties can be useful in settings like quantum computing, where linearity is a fundamental constraint. In such a framework, syntactic linearity can be used to represent matrices and vectors directly as proof-terms.

This increase in expressiveness requires operations such as addition and scalar multiplication, which are typically absent from the proof language. In [4, 7] this challenge is addressed by introducing the $\mathcal{L}^{\mathcal{S}}$-calculus, a proof language for the non-exponential fragment of intuitionistic Linear Logic (IMALL) extended with syntactic constructs for addition ($+$) and scalar multiplication ($\bullet$). This calculus has been extended to second-order Linear Logic [5], and a categorical model for its exponential fragment has been developed [6] and presented in a talk at TLLA25. The present abstract focuses on a categorical model for the full polymorphic calculus.

Related work

There have been other works on algebraic calculi, such as Arrighi and Dowek's Lineal [1] and, even more closely related, Vaux's Algebraic lambda-calculus [11], which originates from

$$
\begin{array}{lll}
t = x \mid t \mathbin{+\!\!+} u \mid a \bullet t & & \\
\quad \mid a.\star & \mid \delta_{\mathbb{1}}(t,u) & (\mathbb{1}) \\
\quad \mid \lambda x^A.t & \mid t\ u & (\multimap) \\
\quad \mid t \otimes u & \mid \delta_{\otimes}(t, x^A y^B.u) & (\otimes) \\
\quad \mid \langle\rangle & & (\top) \\
 & \mid \delta_{\mathbb{0}}(t) & (\mathbb{0}) \\
\quad \mid \langle t,u\rangle & \mid \delta^1_{\&}(t,x^A.u) \mid \delta^2_{\&}(t,x^B.u) & (\&) \\
\quad \mid \mathit{inl}(t) \mid \mathit{inr}(t) & \mid \delta_{\oplus}(t, x^A.u, y^B.v) & (\oplus) \\
\quad \mid !t & \mid \delta_{!}(t, x^A.u) & (!) \\
\quad \mid \Lambda X.t & \mid t\ A & (\forall) \\
\text{Introductions} & \text{Eliminations} & \text{Connective}
\end{array}
$$

Figure 1: The proof-terms of the $\mathcal{L}_2^{\mathcal{S}}$-calculus.

Ehrhard's Differential lambda-calculus [8].

The $\mathcal{L}_2^{\mathcal{S}}$-calculus

The $\mathcal{L}_2^{\mathcal{S}}$-calculus [5] is a proof language for second-order intuitionistic Linear Logic. We follow the presentation of F_{DILL} [10], extended with the additive linear connectives. Propositions are generated by the following grammar:

$$A = X \mid \mathbb{1} \mid A \multimap A \mid A \otimes A \mid \top \mid \mathbb{0} \mid A \,\&\, A \mid A \oplus A \mid !A \mid \forall X.A$$

Let $\mathcal{S}$ be a fixed semiring, such as $\{\star\}$, $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$. The $\mathcal{L}_2^{\mathcal{S}}$-calculus extends the proof language of Linear Logic with addition and scalar multiplication over $\mathcal{S}$. Its proof-terms are given in Figure 1,

A judgement has the form $\Upsilon;\Gamma \vdash t : A$, where t is a proof-term, A is a proposition, Υ is an intuitionistic context, and Γ is a linear context.

This extension of Linear Logic replaces the proof-term $\star$ of proposition $\mathbb{1}$ with a family of proof-terms $a.\star$, one for each scalar a in $\mathcal{S}$:

$$\frac{}{\Upsilon;\varnothing \vdash a.\star : \mathbb{1}}\ \mathbb{1}_i(a)$$

It also introduces inference rules for proof-terms of the form $t \mathbin{+\!\!+} u$ and $a \bullet t$, without altering provability:

$$\frac{\Upsilon;\Gamma \vdash t : A \quad \Upsilon;\Gamma \vdash u : A}{\Upsilon;\Gamma \vdash t \mathbin{+\!\!+} u : A}\ \text{sum} \qquad \frac{\Upsilon;\Gamma \vdash t : A}{\Upsilon;\Gamma \vdash a \bullet t : A}\ \text{prod}(a)$$

Some of the reduction rules are shown in Figure 2. The first group of rules correspond to the reduction of cuts on the connectives $\mathbb{1}$, &, !, and $\forall$. The next two groups allow the rules sum and prod(a) to commute with the introduction rules of the connectives $\mathbb{1}$, &, and $\forall$. For instance, the rule $\langle t,u\rangle \mathbin{+\!\!+} \langle v,w\rangle \longrightarrow \langle t \mathbin{+\!\!+} v, u \mathbin{+\!\!+} w\rangle$ pushes the symbol $\mathbin{+\!\!+}$ inside the pair. The zero-ary commutation rules correspond to scalar addition and multiplication: $a.\star \mathbin{+\!\!+} b.\star \longrightarrow (a+b).\star$, $a \bullet b.\star \longrightarrow (a \times b).\star$.

Second-order quantification adds parametric polymorphism to the calculus, allowing inductive types to be defined as in System F.

$$\begin{aligned}
\delta_{\mathbf{1}}(a.\star, t) &\longrightarrow a \bullet t \\
\delta^1_{\&}(\langle t_1, t_2\rangle, x^A.v) &\longrightarrow (t_1/x)v \\
\delta^2_{\&}(\langle t_1, t_2\rangle, x^A.v) &\longrightarrow (t_2/x)v \\
(\Lambda X.t)\ A &\longrightarrow (A/X)t
\end{aligned}$$

$$\begin{aligned}
a.\star \mathbin{+\!\!+} b.\star &\longrightarrow (a+b).\star & \qquad a \bullet b.\star &\longrightarrow (a \times b).\star \\
\langle t, u\rangle \mathbin{+\!\!+} \langle v, w\rangle &\longrightarrow \langle t \mathbin{+\!\!+} v, u \mathbin{+\!\!+} w\rangle & a \bullet \langle t, u\rangle &\longrightarrow \langle a \bullet t, a \bullet u\rangle \\
(\Lambda X.t) \mathbin{+\!\!+} (\Lambda X.u) &\longrightarrow \Lambda X.t \mathbin{+\!\!+} u & a \bullet \Lambda X.t &\longrightarrow \Lambda X.a \bullet t
\end{aligned}$$

Figure 2: Some of the reduction rules of the $\mathcal{L}_2^{\mathcal{S}}$-calculus.

Example 1. In the $\mathcal{L}_2^{\mathcal{S}}$-calculus, we can represent the natural numbers:

$$\begin{aligned}
\mathsf{Nat} &= \forall X.X \multimap !(X \multimap X) \multimap X \\
\mathsf{zero} &= \Lambda X.\lambda x^X.\lambda f^{!(X \multimap X)}.x \\
\mathsf{succ} &= \lambda n^{\mathsf{Nat}}.\Lambda X.\lambda x^X.\lambda f^{!(X \multimap X)}.\delta_!(f, g^{X \multimap X}.g)\ (n\ X\ x\ f)
\end{aligned}$$

Syntactic linearity

The $\mathcal{L}_2^{\mathcal{S}}$-calculus expresses linearity in a syntactic way: $t\ (u \mathbin{+\!\!+} v)$ is observationally equivalent to $t\ u \mathbin{+\!\!+} t\ v$, for any proof-term t of $A \multimap B$. Similarly, $t\ (a \bullet u)$ is equivalent to $a \bullet t\ u$. This linearity property makes it possible to represent vectors and matrices over $\mathcal{S}$ as proof-terms.

Example 2. We can express the iterated application of a square matrix to a vector as follows, where A is a vector type.

$$\mathsf{Miter} = \lambda n^{\mathsf{Nat}}.\lambda m^{!(A \multimap A)}.\lambda v^A.n\ A\ v\ m$$

Let $\underline{i}$ be the encoding of $i \in \mathbb{N}$, let $\underline{M}$ be the encoding of a matrix $M \in \mathcal{S}^{p \times p}$, and let $\underline{w}$ be the encoding of a vector $w \in \mathcal{S}^p$. Then the normal form of the term $\mathsf{Miter}\ \underline{i}\ !\underline{M}\ \underline{w}$ is the encoding of the vector $M^i(w) \in \mathcal{S}^p$.

Denotational semantics

A sound and adequate model for the fragment of the $\mathcal{L}_2^{\mathcal{S}}$-calculus without second-order was defined [6] using a Linear category $(\mathcal{C}, \otimes, I, !)$ [2] with biproduct $\oplus$, together with a semiring monomorphism from $\mathcal{S}$ to $Hom(I, I)$. The monomorphism condition ensures completeness for the type $\mathbf{1}$.

An alternative formulation can be given in terms of $\mathcal{S}$-semimodule-enriched Linear categories, which leads to more compact proofs:

Definition 3. *$(\mathsf{SM}_{\mathcal{S}}, \otimes, \mathcal{S})$ is the monoidal category where objects are semimodules over the semiring $\mathcal{S}$, arrows are semimodule homomorphisms, and $\otimes$ is the tensor product.*

Definition 4. *An abstract $\mathbf{C}_!^{\mathcal{S}}$ category is a $\mathsf{SM}_{\mathcal{S}}$-enriched Linear category, where the map in $\mathcal{S} \to Hom(I, I)$ given by $a \mapsto a \cdot id_I$ is a semimodule monomorphism.*

$$\begin{aligned}
[\![\mathtt{1}]\!] &= I & [\![\mathtt{o}]\!] &= \mathbf{0} \\
[\![A \multimap B]\!] &= [[\![A]\!], [\![B]\!]] & [\![A \,\&\, B]\!] &= [\![A]\!] \times [\![B]\!] \\
[\![A \otimes B]\!] &= [\![A]\!] \otimes [\![B]\!] & [\![A \oplus B]\!] &= [\![A]\!] + [\![B]\!] \\
[\![\top]\!] &= \mathbf{1} & [\![!A]\!] &= ![\![A]\!]
\end{aligned}$$

Figure 3: Interpretation of propositions in the fragment without second-order. The functor ! is the monoidal comonad in the Linear category, **0** denotes the initial object and **1** denotes the terminal object.

The fragment of the $\mathcal{L}_2^{\mathcal{S}}$-calculus without second-order can be interpreted in the category $\mathbf{C}_!^{\mathcal{S}}$. The interpretation of propositions as objects in the category is given in Figure 3.

To account for second-order quantification, we define a hyperdoctrine [3] indexing $\mathsf{SM}_{\mathcal{S}}$-enriched Linear categories, extending work by Maneggia [10] on Linear hyperdoctrines:

Definition 5. *The category $\mathbf{C}_!^{\mathcal{S}}\mathcal{AT}$ is defined as follows:*

- *Objects are $\mathbf{C}_!^{\mathcal{S}}$ categories.*
- *Morphisms are given by strict cartesian and cocartesian $\mathsf{SM}_{\mathcal{S}}$-enriched Linear functors.*

We consider a base category $\mathcal{C}$ with a distinguished terminal object U, such that every object is generated by finite products of U.

We define a hyperdoctrine via a functor $\mathcal{C}(-,U) : \mathcal{C}^{op} \to \mathbf{C}_!^{\mathcal{S}}\mathcal{AT}$ where, given U^n an object in $\mathcal{C}$, the objects in the category $\mathcal{C}(U^n,U)$ are indexed by the morphisms $U^n \to U$ in $\mathcal{C}$. Given a morphism $f : U^n \to U^m$ in $\mathcal{C}$, $f^* = \mathcal{C}(f,U) : \mathcal{C}(U^m,U) \to \mathcal{C}(U^n,U)$ is the strict cartesian and cocartesian $\mathsf{SM}_{\mathcal{S}}$-enriched Linear functor assigned to f by the functor $\mathcal{C}(-,U)$.

For every n there is a functor $\forall_{U^n} : \mathcal{C}(U^n \times U, U) \to \mathcal{C}(U^n, U)$ which is $\mathsf{SM}_{\mathcal{S}}$-enriched right adjoint to the functor $\pi^*_{U^n} : \mathcal{C}(U^n,U) \to \mathcal{C}(U^n \times U, U)$.

In this setting, a well-formed proposition is interpreted as an object in the category $\mathcal{C}(U^n,U)$, where n is the number of free type variables in it. The interpretation of $\forall X.A$ as an object in the category $\mathcal{C}(U^n,U)$ is then defined as $[\![\forall X.A]\!]^n = \forall_{U^n}([\![A]\!]^{n+1})$, where the application of the functor $\forall_{U^n}$ represents the binding of the variable X which is no longer free in $\forall X.A$. The interpretation of other propositions is essentially defined as in Figure 3, since $\mathcal{C}(U^n,U)$ is a $\mathbf{C}_!^{\mathcal{S}}$ category for every n.

In Figure 4 we show some examples of the interpretations of proof-terms. The sum and scalar product of proof-terms is interpreted using the sum and scalar product of morphisms in the enriched setting, and the introduction of the $\forall$ connective is interpreted applying the bijection corresponding to the adjunction $\pi^*_{U^n} \dashv \forall_{U^n}$.

Soundness of the model follows from the hypotheses on the functors that are morphisms of $\mathbf{C}_!^{\mathcal{S}}\mathcal{AT}$, and from the enrichment hypothesis on the hyperdoctrine adjunctions.

Theorem 6 (Soundness)**.** *Let $\Upsilon;\Gamma \vdash t : A$. If $t \longrightarrow r$, then $[\![\Upsilon;\Gamma \vdash t : A]\!] = [\![\Upsilon;\Gamma \vdash r : A]\!]$.*

As an example, we show the proof for the two commutation rules involving $\forall$:

$$\frac{[\![\Upsilon;\Gamma\vdash t:A]\!]^n = T : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to [\![A]\!]^n \quad [\![\Upsilon;\Gamma\vdash u:A]\!]^n = U : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to [\![A]\!]^n}{[\![\Upsilon;\Gamma\vdash t \mathbin{✚} u:A]\!]^n = T+U : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to [\![A]\!]^n}\ \text{sum}$$

$$\frac{[\![\Upsilon;\Gamma\vdash t:A]\!]^n = T : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to [\![A]\!]^n}{[\![\Upsilon;\Gamma\vdash a \bullet t:A]\!]^n = a\cdot T : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to [\![A]\!]^n}\ \text{prod}(a)$$

$$\frac{[\![\Upsilon;\Gamma\vdash t:A]\!]^{n+1} = T : [\![\Upsilon]\!]^{n+1}_! \otimes [\![\Gamma]\!]^{n+1} \to [\![A]\!]^{n+1}}{[\![\Upsilon;\Gamma\vdash \Lambda X.t:\forall X.A]\!]^n = \Phi(T) : [\![\Upsilon]\!]^n_! \otimes [\![\Gamma]\!]^n \to \forall_{U^n}([\![A]\!]^{n+1})}\ \forall_i$$

Figure 4: Interpretation of some of the proof-terms of the $\mathcal{L}_2^{\mathcal{S}}$-calculus. Φ is the bijection corresponding to the adjunction $\pi^*_{U^n} \dashv \forall_{U^n}$. The rule $\forall_i$ can only be applied if $X \notin \mathsf{fv}(\Upsilon,\Gamma)$. We use the following notation for the interpretation of the typing contexts: $[\![x:A,\Gamma]\!]^n = [\![A]\!]^n \otimes [\![\Gamma]\!]^n$ and $[\![x:A,\Upsilon]\!]^n_! = ![\![A]\!]^n \otimes [\![\Upsilon]\!]^n_!$.

Since the adjunction $\pi^*_{U^n} \dashv \forall_{U^n}$ is $\mathsf{SM}_{\mathcal{S}}$-enriched, the corresponding bijection Φ satisfies, for every $\psi : U^n \times U \to U$, $\phi : U^n \to U$ and $f, g : \pi^*_{U^n}(\psi) \to \phi$ in $\mathcal{C}(U^n \times U, U)$, that:

$$\Phi(f+g) = \Phi(f) + \Phi(g) \tag{1}$$
$$\Phi(a\cdot f) = a\cdot\Phi(f) \tag{2}$$

– *Rule* $(\Lambda X.u) \mathbin{✚} (\Lambda X.v) \longrightarrow \Lambda X.u \mathbin{✚} v$:

$$\begin{aligned}
&[\![\Upsilon;\Gamma\vdash (\Lambda X.u) \mathbin{✚} (\Lambda X.v) : \forall X.B]\!]^n \\
&\quad= \Phi\left([\![\Upsilon;\Gamma\vdash u:B]\!]^{n+1}\right) + \Phi\left([\![\Upsilon;\Gamma\vdash v:B]\!]^{n+1}\right) \\
&\quad= \Phi\left([\![\Upsilon;\Gamma\vdash u:B]\!]^{n+1} + [\![\Upsilon;\Gamma\vdash v:B]\!]^{n+1}\right) && \text{by Equation 1} \\
&\quad= [\![\Upsilon;\Gamma\vdash \Lambda X.u \mathbin{✚} v : \forall X.B]\!]^n
\end{aligned}$$

– *Rule* $a \bullet \Lambda X.u \longrightarrow \Lambda X.a \bullet u$:

$$\begin{aligned}
[\![\Upsilon;\Gamma\vdash a\bullet\Lambda X.u : \forall X.B]\!]^n &= a\cdot\Phi\left([\![\Upsilon;\Gamma\vdash u:B]\!]^{n+1}\right) \\
&= \Phi\left(a\cdot[\![\Upsilon;\Gamma\vdash u:B]\!]^{n+1}\right) && \text{by Equation 2} \\
&= [\![\Upsilon;\Gamma\vdash \Lambda X.a\bullet u : \forall X.B]\!]^n
\end{aligned}$$

Adequacy of the model depends on the semimodule monomorphism condition, which ensures that distinct proofs of $\mathtt{1}$ are interpreted as distinct morphisms.

Work in progress

We are currently working toward a better understanding of the semimodule monomorphism condition required for adequacy, with the goal of obtaining a more conceptual formulation.

We are also studying concrete models for the $\mathcal{L}_2^{\mathcal{S}}$-calculus.

References

[1] P. Arrighi and G. Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1), 2017.

[2] G.M. Bierman. On intuitionistic linear logic. Technical Report UCAM-CL-TR-346, University of Cambridge, Computer Laboratory, August 1994.

[3] Roy L. Crole. *Categories for types*. Cambridge University Press, 1993.

[4] Alejandro Díaz-Caro and Gilles Dowek. A linear linear lambda-calculus. *Mathematical Structures in Computer Science*, 34:1103–1137, 2024.

[5] Alejandro Díaz-Caro, Gilles Dowek, Malena Ivnisky, and Octavio Malherbe. A linear proof language for second-order intuitionistic linear logic. In George Metcalfe, Thomas Studer, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 14672 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2024.

[6] Alejandro Díaz-Caro, Malena Ivnisky, and Octavio Malherbe. An algebraic extension of intuitionistic linear logic: The $\mathcal{L}_!^s$-calculus and its categorical model. *Journal of Logic and Computation*, 35(8):exaf053, 2025.

[7] Alejandro Díaz-Caro and Octavio Malherbe. The sup connective in IMALL: A categorical semantics. *Theoretical Computer Science*, 1072:115845, 2026.

[8] Thomas Ehrhard and Laurent Regnier. The differential λ-calculus. *Theoretical Computer Science*, 309(1-3):1–41, 2003.

[9] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[10] P. Maneggia. *Models of linear polymorphism*. PhD thesis, The University of Birmingham, UK, 2004.

[11] Lionel Vaux. The algebraic lambda-calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.