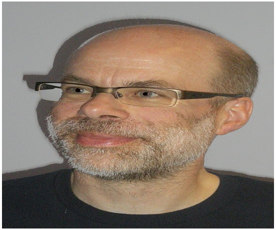
Going Without: a linear modality and FP

Valeria de Paiva

TOPOS INSTITUTE

MSFP 2022
(joint work with Eike Ritter, Paola Maneggia and Milly Maietti)

Thanks for the invitation Max and Jeremy!



Today

Categorical Models for Linear and Intuitionistic Type Theory (FOSSACS, LNCS Springer, vol 1784, 2000).

Relating Categorical Semantics for Intuitionistic Linear Logic (Applied Categorical Structures, volume 13(1):1–36, 2005)

why are these important?

Linear Type Theories

why I love them

I worked on Gödel's Dialectica interpretation for my thesis with Martin Hyland.

I realized that the basic internal categorical model for the Dialectica corresponds to **linear logic**, not intuitionistic logic.

Linear Type Theories

why I love them



Together with Martin Hyland, Nick Benton, and Gavin Bierman I worked on the first linear type theory, 1993.

Nick then worked on his own version, the LNL calculus, 1995.

Andrew Barber (Plotkin's student) came up with what I thought was the **best** formulation, the DILL calculus, 1997.

xSLAM: eXplicit Substitutions Linear Abstract Machine EPSRC 1997–2000

- Ritter's PhD thesis on categorical combinators for the Calculus of Constructions (Cambridge 1992, TCS 1994)
- de Paiva's PhD thesis on models of Linear Logic & linear lambda-calculus (Cambridge 1990)
- Put the two together for Categorical Abstract Machines for Linear Functional Programming
- Project xSLAM Explicit Substitutions for Linear Abstract Machines
- 'categorical structured syntax' in Hyland's turn of phrase

xSLAM: eXplicit Substitutions Linear Abstract Machine

Twenty years later:

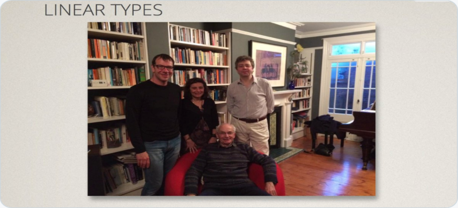
- More than 10 papers only submitted to conferences
- Only two in journals: Linear Explicit Substitutions. (Ghani, de Paiva, Ritter, 2000) and Relating Categorical Semantics for Intuitionistic Linear Logic (Maietti, Maneggia, de Paiva, Ritter 2005)
- Master's thesis: An Abstract Machine based on Linear Logic and Explicit Substitutions. Francisco Alberti, 1997
- Two international workshops: Logical Abstract Machines (Saarbruecken, 1998), Logical Abstract Machines (Birmingham, 1999)
- International Dagstuhl meeting: Linear Logic and Applications (1999)

xSLAM Project Work

1. Maietti, de Paiva, Ritter. Linear primitive recursion. manuscript 2000.
2. Maietti, de Paiva, Ritter. Categorical Models for Intuitionistic and Linear Type Theory. FoSSaCS 2000.
3. Cervesato, de Paiva, Ritter. Explicit Substitutions for Linear Logical Frameworks. LFM 1999
4. Ghani, de Paiva, Ritter. Explicit Substitutions for Constructive Necessity. ICALP 1998
5. Ghani, de Paiva, Ritter. Categorical Models for Explicit Substitutions. FoSSaCS 1999.
6. de Paiva, E. Ritter. On Explicit Substitutions and Names. ICALP 1997.
7. Nesi, de Paiva, Ritter. Rewriting Properties of Combinators for Intuitionistic Linear Logic. Higher Order Algebra, Logic and Term Rewriting, HOA'1993.
8. Maietti, de Paiva and Ritter. Normalization Bounds in Rudimentary Linear Logic ICC, 2002.
9. de Paiva and Eike Ritter. Variations on Linear PCF. WESTAPP, 1999.

Many years later...

Valeria dePaiva @valeriadepaiva · Jul 13, 2016
Gang of Four paper, 25 years later in `**Rust** for Semanticists' at LOLA, yay! thanks @asajeffrey



Tweag @tweagio · Oct 23, 2018
Proposal: add **linear** types to GHC #haskell. Status: (conditionally) accepted!

GHC Steering Committee

Linear types by aspiwack · Pull Request #111 · ghc-...
The proposal has been accepted; the following discussion is mostly of historic interest. This ...
github.com

2 41 108

time to try again?

Many years later...

Linear Haskell

Practical Linearity in a Higher-Order Polymorphic Language

JEAN-PHILIPPE BERNARDY, University of Gothenburg, Sweden
MATHIEU BOESPFLUG, Tweag I/O, France
RYAN R. NEWTON, Indiana University, USA
SIMON PEYTON JONES, Microsoft Research, UK
ARNAUD SPIWACK, Tweag I/O, France

v 2017

"Despite their obvious promise, and a huge research literature, linear type systems have not made it into mainstream programming languages,[...]"

Linear Haskell, Today and Tomorrow I: Jean-Philippe Bernardy University of Gothenburg, Sweden, ICFP 2021
Haskell 2021 - Why Functional Programming with Linear Types Matters, Sep 2021, Mathieu Boespflug Tweag I/O

Categorical Proof Theory

- Model derivations/proofs, not whether theorems are true or not
- Proofs definitely first-class citizens
- How? Uses extended Curry-Howard correspondence
- Why is it good? Modeling derivations useful where you need proofs themselves, in linguistics, functional programming, compilers, etc.
- Why is it important? Widespread use of logic/algebra in CS means new important problems to solve with our favorite tools.

Why so little impact on maths, CS or logic?

Linear Type Theories with Categorical Models

- Linear λ-calculus (Benton, Bierman, de Paiva, Hyland, 1992) https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-262.html
- DILL system (Plotkin and Barber, 1996) http://www.lfcs.inf.ed.ac.uk/reports/96/ECS-LFCS-96-347/
- LNL system (Benton, 1995) `https://www.researchgate.net/publication/221558077_A_Mixed_Linear_and_Non-Linear_Logic_Proofs_Terms_and_Models`

Many more without explicit categorical models...

Linear λ-calculus

$$x : A \vdash x : A$$

$$\frac{\Gamma \vdash e : A \qquad \Delta, x : A \vdash f : B}{\Gamma, \Delta \vdash f[e/x] : B}\ \mathit{Cut}$$

$$\frac{\Gamma \vdash e : A \qquad \Delta, x : B \vdash f : C}{\Gamma, g : A \multimap B, \Delta \vdash f[(ge)/x] : C}\ (\multimap_{\mathcal{L}}) \qquad \frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x.e : A \multimap B}\ (\multimap_{\mathcal{R}})$$

$$\frac{\Gamma \vdash e : A}{\Gamma, x : I \vdash \text{let } x \text{ be } * \text{ in } e : A}\ (I_{\mathcal{L}}) \qquad \frac{}{\vdash * : I}\ (I_{\mathcal{R}})$$

$$\frac{\Delta, x : A, y : B \vdash f : C}{\Delta, z : A \otimes B \vdash \text{let } z \text{ be } x \otimes y \text{ in } f : C}\ (\otimes_{\mathcal{L}}) \qquad \frac{\Gamma \vdash e : A \qquad \Delta \vdash f : B}{\Gamma, \Delta \vdash e \otimes f : A \otimes B}\ (\otimes_{\mathcal{R}})$$

$$\frac{\Gamma \vdash e : B}{\Gamma, z :!A \vdash \text{discard } z \text{ in } e : B}\ \mathit{Weakening} \qquad \frac{\Gamma, x :!A, y :!A \vdash e : B}{\Gamma, z :!A \vdash \text{copy } z \text{ as } x, y \text{ in } e : B}\ \mathit{Contraction}$$

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma, z :!A \vdash e[\text{derelict}(z)/x] : B}\ \mathit{Dereliction}$$

$$\frac{\overline{x} :!\Gamma \vdash e : A}{\overline{y} :!\Gamma \vdash \text{promote } \overline{y} \text{ for } \overline{x} \text{ in } e :!A}\ \mathit{Promotion}$$

DILL

$$(Int - Ax)\ \Gamma, x : A; \_ \vdash x : A \qquad\qquad (Lin - Ax)\ \Gamma; x : A \vdash x : A$$

$$(Graph)\ \frac{\Gamma; \Delta \vdash t : A}{\Gamma; \_ \vdash f(t) : B}\ (f \in G(A, B))$$

$$(I - I)\ \Gamma; \_ \vdash * : I \qquad\qquad (I - E)\ \frac{\Gamma; \Delta_1 \vdash t : I \quad \Gamma; \Delta_2 \vdash u : A}{\Gamma; \Delta_1, \Delta_2 \vdash \text{let } * \text{ be } t \text{ in } u : A}$$

$$(\otimes - I)\ \frac{\Gamma; \Delta_1 \vdash t : A \quad \Gamma; \Delta_2 \vdash u : B}{\Gamma; \Delta_1, \Delta_2 \vdash t \otimes u : A \otimes B} \qquad (\otimes - E)\ \frac{\Gamma; \Delta_1 \vdash u : A \otimes B \quad \Gamma; \Delta_2, x : A, y : B \vdash t : C}{\Gamma; \Delta_1, \Delta_2 \vdash \text{let } x \otimes y : A \otimes B \text{ be } u \text{ in } t : C}$$

$$(\multimap I)\ \frac{\Gamma; \Delta, x : A \vdash t : B}{\Gamma; \Delta \vdash (\lambda x : A.t) : (A \multimap B)} \qquad\qquad (\multimap E)\ \frac{\Gamma; \Delta_1 \vdash u : A \multimap B \quad \Gamma; \Delta_2 \vdash t : A}{\Gamma; \Delta_1, \Delta_2 \vdash (ut) : B}$$

$$(! - I)\ \frac{\Gamma; \_ \vdash t : A}{\Gamma; \_ \vdash !t : !A} \qquad\qquad (! - E)\ \frac{\Gamma; \Delta_1 \vdash u : !A \quad \Gamma, x : A; \Delta_2 \vdash t : B}{\Gamma; \Delta_1, \Delta_2 \vdash \text{let } !x : A \text{ be } u \text{ in } t : B}$$

LNL

$$\Theta; a\colon A \vdash_{\mathcal{L}} a\colon A \qquad \Theta, x\colon X \vdash_{\mathcal{C}} x\colon X$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X \qquad \Theta \vdash_{\mathcal{C}} t\colon Y}{\Theta \vdash_{\mathcal{C}} (s,t)\colon X \times Y} \qquad \frac{}{\Theta \vdash_{\mathcal{C}} ()\colon 1}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{fst}(s)\colon X} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{snd}(s)\colon Y}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e \otimes f\colon A \otimes B} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \otimes B \qquad \Theta; \Delta, a\colon A, b\colon B \vdash_{\mathcal{L}} f\colon C}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f\colon C}$$

$$\frac{}{\Theta \vdash_{\mathcal{L}} *\colon I} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon I \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ * = e\ \mathsf{in}\ f\colon A}$$

$$\frac{\Theta, x\colon X \vdash_{\mathcal{C}} s\colon Y}{\Theta \vdash_{\mathcal{C}} (\lambda x\colon X.s)\colon X \to Y} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon X \to Y \qquad \Theta \vdash_{\mathcal{C}} t\colon X}{\Theta \vdash_{\mathcal{C}} s\ t\colon Y}$$

$$\frac{\Theta; \Gamma, a\colon A \vdash_{\mathcal{L}} e\colon B}{\Theta; \Gamma \vdash_{\mathcal{L}} (\lambda a\colon A.e)\colon A \multimap B} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \multimap B \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e\ f\colon B}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X}{\Theta \vdash_{\mathcal{L}} \mathsf{F}(s)\colon FX} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon FX \qquad \Theta, x\colon X; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ \mathsf{F}(x) = e\ \mathsf{in}\ f\colon A}$$

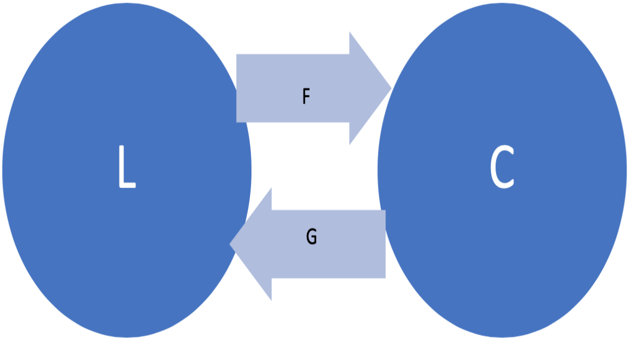
$$\frac{\Theta \vdash_{\mathcal{L}} e\colon A}{\Theta \vdash_{\mathcal{C}} \mathsf{G}(e)\colon GA} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon GA}{\Theta \vdash_{\mathcal{L}} \mathsf{derelict}(s)\colon A}$$

Categorical Models

- Linear λ-calculus (ILL)
  A linear category is a symmetric monoidal closed (smc) category equipped with a linear monoidal comonad, such that the co-Kleisli category of the comonad is a Cartesian closed Category (CCC). (loads of conditions)
- DILL system
  A DILL-category is a pair, a symmetric monoidal closed category and a cartesian category, related by a monoidal adjunction.
- LNL system
  An LNL-model is a (symmetric) monoidal adjunction between a smcc and a ccc.

In pictures

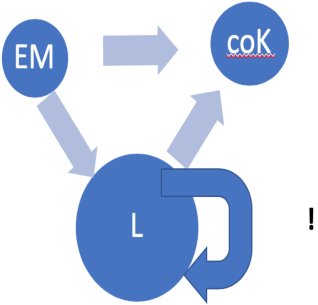
Monoidal Adjunctions:
L is smcc, C is ccc, F-|G, both monoidal



In pictures

Linear Categories

L smcc, !=linear exponential comonad



Wait!

Are these models equivalent?
How do I choose one?
How do I choose the model morphisms?
the solution: Internal Language Theorems!

Explaining Internal Languages

Thanks Milly Maietti for slides!

for a given calculus τ and a class of complete models

- **Organize models of τ into a categories $\mathrm{Mod}(\tau)$**
  (find suitable model morphisms)
- **Organize theories of τ into a category $\mathrm{Th}(\tau)$**
  (find suitable theory morphisms)
- **Check validity of Internal language theorem:**

  $\mathrm{Mod}(\tau)$ is equivalent to $\mathrm{Th}(\tau)$

  if so, the category of best models of τ are determined up to equivalences

Explaining Internal Languages

Thanks Milly Maietti for slides!

Idea of an internal language theorem

- Define internal language out of a model of τ :

$$Int\colon \mathrm{Mod}(\tau) \longrightarrow \mathrm{Th}(\tau)$$

- Define syntactic category out of a theory of τ :

$$Syn\colon \mathrm{Th}(\tau) \longrightarrow \mathrm{Mod}(\tau)$$

- such that

for any model M $\quad$ & $\quad$ for any theory T

$$\mathsf{M} \simeq Syn(Int(\mathsf{M})) \qquad \& \qquad \mathsf{T} \simeq Int(Syn(\mathsf{T}))$$

$\simeq$ is isomorphism
(works for simple type theories)

Explaining Internal Languages

Thanks Milly Maietti for slides!

- Define internal language out of a model of τ :

$$Int\colon \mathrm{Mod}(\tau) \to \mathrm{Th}(\tau)$$

- Define syntactic category out of a theory of τ :

$$Syn\colon \mathrm{Th}(\tau) \to \mathrm{Mod}(\tau)$$

- such that

for any model M		for any theory T
$\mathsf{M} \simeq Syn(Int(\mathsf{M}))$	&	$\mathsf{T} \simeq Int(Syn(\mathsf{T}))$

$\simeq$ is equivalence of categories
(only this works for dependent type theories)

Explaining Internal Languages

Thanks Milly Maietti for slides!

Soundness and completeness may just give

- an internal language functor:

$$Int\colon \mathrm{Mod}(\tau) \to \mathrm{Th}(\tau)$$

- a syntactic category functor:

$$Syn\colon \mathrm{Th}(\tau) \to \mathrm{Mod}(\tau)$$

- sometimes a monic natural trasformation for any theory T

$$\mathsf{T} \hookrightarrow Int(Syn(\mathsf{T}))$$

Explaining Internal Languages

Thanks Milly Maietti for slides!

various formulations for Intuitionistic Linear Logic

- Benton-Bierman-de Paiva-Hyland '93 ILL type calculus : based on usual sequents $\Gamma \vdash A$
- Barber-Plotkin '97 Dual Intuitionistic Linear Logic (DILL) : based on double-context sequents $\Gamma \mid \Delta \vdash A$

  such that

  Γ has intuitionistic assumptions
  Δ has linear assumptions

$$\Gamma, B \mid \Delta \vdash A \quad \text{iff} \quad \Gamma \mid !B, \Delta \vdash A$$

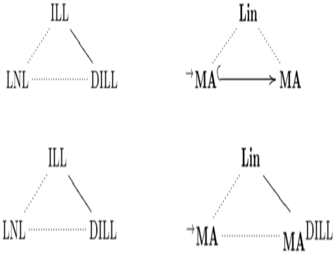
ILL is not equivalent to DILL via a bijective translation of proofs:
ILL-sequents correspond only to DILL-sequents of the form $\_ \mid \Delta \vdash A$

but

$$\mathrm{Th}(DILL) \simeq \mathrm{Th}(ILL)$$

Linear Curry-Howard Isos

"Relating Categorical Semantics for Intuitionistic Linear Logic"(with P. Maneggia, M. Maietti and E. Ritter), Applied Categorical Structures, vol 13(1):1–36, 2005.



Take home: Categorical models need to be more than sound and complete. They need to provide *internal languages* for the theories they model.

Why a new calculus?

(ILT, Fossacs 2000)

Choosing between the three type theories:

- DILL is best, less verbose than ILL, but closer to what we want to do than LNL.
- Easy formulation of the promotion rule
- Contains the usual lambda-calculus as a subsystem
- However, most FPers would prefer to use a variant of DILL where instead of !, one has two function spaces, $\rightarrow$ and $\multimap$
- But then: what's the categorical model?

Intuitionistic and Linear Type Theory

(ILT, Fossacs 2000)

If we have two function spaces, but no modality !, how can we model it?
All the models discussed before have a notion of !, created by the adjunction.
Well, we need to use deeper mathematics, i.e. **fibrations** or indexed categories.

Calculus ILT
Categorical models for intuitionistic and linear type theory (Maietti et al, 2000)

$$\Gamma \mid a : A \vdash a : A \qquad \Gamma, x : A \mid \_ \vdash x : A$$

$$\frac{\Gamma \mid \Delta, a : A \vdash M : B}{\Gamma \mid \Delta \vdash \lambda a^A . M : A \multimap B} \qquad \frac{\Gamma, x : A \mid \Delta \vdash M : B}{\Gamma \mid \Delta \vdash \lambda x^A . M : A \to B}$$

$$\frac{\Gamma \mid \Delta_1 \vdash M : A \multimap B \quad \Gamma \mid \Delta_2 \vdash N : A}{\Gamma \mid \Delta \vdash M\,{}_l N : B} \qquad \frac{\Gamma \mid \Delta \vdash M : A \to B \quad \Gamma \mid \_ \vdash N : A}{\Gamma \mid \Delta \vdash M\,{}_i N : B}$$

ILT MODELS? (Maietti et al, 2000)

Idea goes back to Lawvere's hyperdoctrines satisfying the comprehension axiom.

Model of ILT should modify this setting to capture the separation between intuitionistic and linear variables.

A base category B, which models the intuitionistic contexts of ILT, objects in B model contexts $(\Gamma|\_)$.

Each fibre over an object in B modelling a context models terms $\Gamma|\Delta \vdash M : A$ for any context Δ.

The fibres are now symmetric monoidal closed categories with finite products and model the linear constructions of ILT

ILT MODELS
have internal language theorems

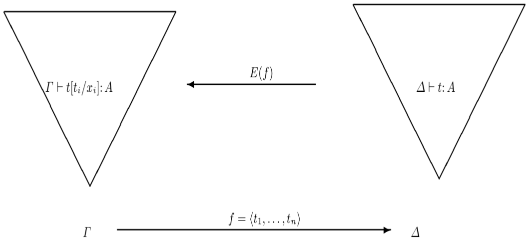


Fig. 1. Modelling the simply-typed λ-calculus in a D-category

Missing an understanding of what is essential, what can be changed, and limitations of the methods.

Speculation

Can we do the same for Modal Logic S4?
Can we do the same for Relevant Logic?
I hope so. But have not managed yet.
Meanwhile pictures do help to remember the main messages.

Conclusions

- Linear type theories are interesting both from implementation and theory perspectives
- Internal language theorems are essential for categorical proof theory: soundness&completeness not enough
- Paying attention to morphisms is necessary
- Paying attention to implementors is necessary
- Finding right abstractions depends on checking other logical systems. more experimentation is needed!

Definitely a good time to sharpen our theoretical tools

Thanks!