A Functorial model of Differential Linear Logic

SCALP 2023

Marie Kerjean (UP13), Valentin 'Richie' Maestracci (AMU), Morgan Rogers (UP13)

November 2023
Let's review: Models of LL

What is a model of linear logic?
Let's review: Models of LL

What is a model of linear logic? A possible categorical answer: Seely Categories

\[(\mathcal{C}, \times) \leftrightarrow (\mathcal{L}, \otimes)\]

A strong monoidal adjunction (\(! \overset{\text{def}}{=} \mathcal{E}' \circ U\))

Between a monoidal category of linear morphisms

And a cartesian category of non-linear morphisms
Let's review: Models of LL

How do we interpret a proof?

A proof $\pi$ of conclusion $\Gamma \vdash A$ is interpreted as a morphism $⟦\Gamma⟧ \to ⟦A⟧$ compositionally:

$$
\begin{array}{c}
\pi \\
\hline
A \vdash \Gamma \\
\hline \\

!A \vdash \Gamma
\end{array}
$$

Use a natural transformation $d_A : !A \to A$

$$
\begin{array}{ccc}
!A & \xrightarrow{d} & A \\
\pi & \rightarrow & \Gamma
\end{array}
$$
Let's review: Models of LL

This is a compact version of requiring to have these natural transformations

<table>
<thead>
<tr>
<th>Operator</th>
<th>Type</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>!A → 1</td>
<td>Create constant function</td>
</tr>
<tr>
<td>c</td>
<td>!A → !A ⊗ !A</td>
<td>From 2 to 1 parameter</td>
</tr>
<tr>
<td>d</td>
<td>!A → A</td>
<td>Forget linearity</td>
</tr>
<tr>
<td>p</td>
<td>!A → !!A</td>
<td>Higher order</td>
</tr>
</tbody>
</table>

Plus some commutative diagrams (to respect cut elimination)
What is Differential Linear Logic (DiLL)?
DiLL is a variant of linear logic that was discovered by Ehrhard & Regnier in 2004 with a notion of differentiation:
What is Differential Linear Logic (DiLL)?
DiLL is a variant of linear logic that was discovered by Ehrhard & Regnier in 2004 with a notion of differentiation:

This amounts to adding these operator:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Type</th>
<th>Operator</th>
<th>Type</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>!A \rightarrow 1</td>
<td>\overline{w}</td>
<td>1 \rightarrow !A</td>
<td>Evaluation at 0</td>
</tr>
<tr>
<td>c</td>
<td>!A \rightarrow !A \otimes !A</td>
<td>\overline{c}</td>
<td>!A \otimes !A \rightarrow !A</td>
<td>Convolution</td>
</tr>
<tr>
<td>d</td>
<td>!A \rightarrow A</td>
<td>\overline{d}</td>
<td>A \rightarrow !A</td>
<td>Differentiation</td>
</tr>
<tr>
<td>p</td>
<td>!A \rightarrow !!A</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Dbar diagrams
Dbar diagrams 2

\[
\begin{align*}
A & \xrightarrow{\bar{d}} !A \xrightarrow{p} !!A \\
A \otimes I & \xrightarrow{\bar{d} \otimes w} !A \otimes !A \xrightarrow{\bar{d}_! \otimes p} !!A \otimes !!A \\
A \otimes !B & \xrightarrow{\bar{d} \otimes 1} !A \otimes !B \xrightarrow{\varphi} !(A \otimes B)
\end{align*}
\]
What is Differential Linear Logic (DiLL) ?

There is a reformulation of such models with a biproduct $\boxdot$ which automatically gives most operators.

$$\mathcal{E}' \approx p$$

$$\left(\mathcal{E}, \times\right) \leftrightarrow \left(\mathcal{L}, \otimes, \boxdot\right)$$

$$U \approx d$$

<table>
<thead>
<tr>
<th>Type</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$!A \multimap A$</td>
<td>$\bar{d}$</td>
</tr>
</tbody>
</table>

But where would $\bar{d}$ fit in such a setting?
Our contribution: A model where $\bar{d}$ is expressed as a functor.

(Purely functorial)
Why would we want that?

- Express a modular transformation of program
  → (This is a key point of Differentiable Programming)

- Compactify definitions: Proving that something is a model becomes easier

- Makes link easier with Chiralities
Let's look at Differentiation

We want to capture the chain-rule in a functorial way:

\[ D_a(g \circ f) = D_{f(a)}(g) \circ D_a(f) \]
Let's look at Differentiation

We want to capture the chain-rule in a functorial way:

\[ D_a(g \circ f) = D_{f(a)}(g) \circ D_a(f) \]

\[ D_a(g \circ f) = D_{f(a)}(g : B \to C) \circ D_a(f : A \to B) \]
Let's look at Differentiation

We want to capture the chain-rule in a functorial way:

\[ D_a(g \circ f) = D_{f(a)}(g) \circ D_a(f) \]

\[ D_a(g \circ f) = D_{f(a)}(g : B \to C) \circ D_a(f : A \to B) \]

Let \( D \) depend on the point of differentiation!
Let's look at Differentiation

We want to capture the chain-rule in a functorial way:

\[ D_a(g \circ f) = D_{f(a)}(g) \circ D_a(f) \]

\[ D_a(g \circ f) = D_{f(a)}(g : B \to C) \circ D_a(f : A \to B) \]

Let \( D \) depend on the point of differentiation!

\[
\begin{array}{ccc}
(A, a) & \xrightarrow{f} & (B, f(a)) \\
& \xrightarrow{g} & (C, g(f(a)))
\end{array}
\]
Let's look at Differentiation

We want to capture the chain-rule in a functorial way:

\[ D_a(g \circ f) = D_{f(a)}(g) \circ D_a(f) \]

\[ D_a(g \circ f) = D_{f(a)}(g : B \to C) \circ D_a(f : A \to B) \]

Let \( D \) depend on the point of differentiation!

\[ \begin{array}{ccc}
(A, a) & \xrightarrow{f} & (B, f(a)) \\
& \xrightarrow{g} & (C, g(f(a))) \\
\end{array} \]

\( \tilde{D} \) should be a functor, with morally \( \tilde{D}(f) = D_a(f) \)
The co-Slice Category

The category $I \downarrow \mathcal{C}$, the co-Slice of $\mathcal{C}$ is defined as follows:

- **Objects**: $(A, a)$ with $a : I \to A$, intuitively, an element of $A$
- **Arrows**: $f : (A, a) \to (B, b)$ such that $f(a) = b$
The co-Slice Category
The category $I \downarrow C$, the co-Slice of $C$ is defined as follows:

- **Objects**: $(A, a)$ with $a : I \rightarrow A$, intuitively, an element of $A$
- **Arrows**: $f : (A, a) \rightarrow (B, b)$ such that $f(a) = b$

\[
\begin{array}{ccc}
I & \xrightarrow{a} & A \\
\downarrow & & \downarrow f \\
& B & \\
\end{array}
\]

Now $b$:

\[D_a(g \circ f) = D(g) \circ D(f) = D_{f(a)}(g) \circ D_a(f)\]
Second thing

The differential (i.e., best linear approximation) of a linear is itself. Hence $\mathcal{D}$ should preserve linear morphisms: $\forall a, l : D_a(l) = l$
But...

We cannot go up from $\mathcal{C}$ to $I \downarrow \mathcal{C}$!

→ Would require to choose a point
The category of Generalized Elements

Given a functor $U : \mathcal{L} \to \mathcal{C}$, the category $I \downarrow U$ of generalized elements over $U$ is defined as:

- **Objects:** $(A, a)$ with $a : I \to U(A)$
- **Arrows:** $l : A \to B$ such that $U(l)(a) = b$

In a sense, the linear part of the co-Slice.
Definition of Functorial DiLL Model

A pre-model of DiLL,

\[
\begin{align*}
I & \downarrow U & \rightarrow & & I & \downarrow \mathcal{C} \\
\Pi & \downarrow & & \downarrow & \Pi \\
(\mathcal{L}, \otimes, \Diamond) & \rightarrow & (\mathcal{C}, \times) & \text{over} & \mathcal{U}
\end{align*}
\]
Definition of Functorial DiLL Model

A pre-model of DiLL, plus a functor $\vec{D}$

$$
\begin{align*}
I \downarrow U & \quad \xrightarrow{\vec{U}} \quad I \downarrow C \\
\Pi \downarrow & \quad \xrightarrow{\vec{D}} \quad \Pi \\
(\mathcal{L}, \diamond) & \quad \xrightarrow{U} \quad (\mathcal{C}, \times)
\end{align*}
$$
Definition of Functorial DiLL Model

A pre-model of DiLL, plus a functor \( \vec{D} \)

\[
\begin{array}{ccc}
I \downarrow U & \overset{\vec{U}}{\longrightarrow} & I \downarrow C \\
\Pi \downarrow & \vec{D} & \downarrow \Pi \\
(\mathcal{L}, \Diamond) & \overset{\vec{U}}{\longrightarrow} & (C, \times)
\end{array}
\]

(And well pointedness relative to \( I \) ...)
Theorems

**Theorem:** Our functorial model is a model of DiLL.

**Theorem:** The converse is true for well pointed models.
What if my model isn't well pointed?

A pre-model of DiLL, plus a functor $\vec{D}$

\[
\begin{align*}
    U & \downarrow U & \xrightarrow{\vec{U}} & U \downarrow C \\
    \Pi & \downarrow & \xrightarrow{\vec{D}} & \downarrow \Pi \\
    (\mathcal{L}, \diamond) & \rightarrow & (\mathcal{C}, \times) & \rightarrow
\end{align*}
\]
What if my model isn't well pointed?

A pre-model of DiLL, plus a functor $\vec{D}$

$$
\begin{align*}
U \downarrow U & \quad \xrightarrow{\vec{U}} \quad U \downarrow \mathcal{C} \\
\Pi \downarrow & \quad \xleftarrow{\vec{D}} \quad \Pi \\
(\mathcal{L}, \diamond) & \quad \xrightarrow{} \quad (\mathcal{C}, \times) \\
U &
\end{align*}
$$

But we have no model/intuition of what is going on here...
What are Chiralities?

A categorical framework by Melliès which refines \(*\)-autonomous categories into models of Polarized MLL.

\[
\begin{align*}
(P, \otimes, 1) & \quad \xRightarrow{\sim} \quad (N^{\text{op}}, \Rightarrow, \perp) \\
(-)^1_P & \quad \mapsfrom \quad \uparrow \\
(-)^1_N & \quad \mapsfrom \quad \downarrow
\end{align*}
\]

With the left being strong monoidal, and \(\downarrow \circ \uparrow = \text{Id}\)

Plus some extra conditions on the adjunction

Appears a lot for “smooth” models of DiLL (in Functional Analysis)
A parallel to be made
In Chiralities: Positive vs Negative  |  In DiLL: Linear vs Non-Linear

\[(\mathcal{P}, \otimes) \overset{(-)}{\longrightarrow} (\mathcal{N}^{\text{op}}, \otimes)\]

\[(\mathcal{P}, \otimes) \overset{(-)}{\longrightarrow} (\mathcal{N}^{\text{op}}, \otimes)\]

\[\mathcal{E} \overset{U'}{\longrightarrow} (\mathcal{L}^{\text{op}}, \otimes)\]

\[\mathcal{E} \overset{U'}{\longrightarrow} (\mathcal{L}^{\text{op}}, \otimes)\]

\[\mathcal{P} \overset{\Pi}{\longrightarrow} \mathcal{N} \overset{\Pi}{\longrightarrow} \mathcal{N} \]

\[\mathcal{P} \overset{\Pi}{\longrightarrow} \mathcal{N} \overset{\Pi}{\longrightarrow} \mathcal{N} \]

\[\mathcal{L} \overset{\Pi}{\longrightarrow} \mathcal{C} \overset{\Pi}{\longrightarrow} \mathcal{C}\]

\[\mathcal{L} \overset{\Pi}{\longrightarrow} \mathcal{C} \overset{\Pi}{\longrightarrow} \mathcal{C}\]
Last but not least: a funny remark

When $\mathcal{L}$ is a calculus category (so with integration on top), we have an relative $\otimes \text{Id}$-adjunction:

\begin{align*}
\begin{array}{c}
\mathcal{D} \\
\Upsilon \\
\end{array}
\end{align*}

$I \downarrow \mathcal{C} \quad \overset{\sim}{\leftrightarrow} \quad (\mathcal{L}, \otimes)$

$(I \downarrow \mathcal{C})(((a, A), (b, B)) \simeq \mathcal{L}(!A \otimes A, B)$

With $\Upsilon(l) = (U(l), U(u_A))$

This adjunction corresponds mathematically to the fundamental theorem of calculus!
Future Work

- Pursue similarities and extending our framework to Polarized LL with Mellie’s Chiralities
- New model of Kerjean & Pacaud-Lemay with co-promotion should have a symmetric behavior: a functorial co-chain rule?
- Investigate the dependent flavour
Future Work

- Pursue similarities and extending our framework to Polarized LL with Mellie’s Chiralities
- New model of Kerjean & Pacaud-Lemay with co-promotion should have a symmetric behavior: a functorial co-chain rule?
- Investigate the dependent flavour

Thank you for listening!

Any questions? 😊