The Linear-Non-Linear Substitution Monad

OWLS, 15 July 2020

Christine TASSON (tasson@irif.fr)

Institut de Recherche en Informatique Fondamentale
Roadmap: The linear-non-linear substitution monad

Motivation:
- Differential $\lambda$-calculus

Goal:
- Axiomatisation using generalized multicategories.

Tool:
- A colimit construction applied to combine 2-monads on $\text{Cat}$

Results:
- The colimit is a 2-monad.
- Characterization of its algebras.
Linear-non-linear substitution

Substitutions in differential $\lambda$-calculus
**Differential $\lambda$-Calculus**

(Ehrhard-Regnier 2003, Ehrhard 2018)

**Semantical observation:** in quantitative models of Linear Logic, programs are interpreted by smooth functions, hence differentiation.

<table>
<thead>
<tr>
<th>Programs</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M, N$</td>
<td>$f, g$</td>
</tr>
<tr>
<td>Variable</td>
<td>$x$</td>
</tr>
<tr>
<td>Abstraction</td>
<td>$\lambda x. M$</td>
</tr>
<tr>
<td>Application</td>
<td>$(\lambda x. M)N$</td>
</tr>
<tr>
<td>Differentiation</td>
<td>$D\lambda x. M \cdot N$</td>
</tr>
</tbody>
</table>
Linear and Non-Linear substitutions in Differential $\lambda$-calculus

Substitution

$$(\lambda x. M)N \rightarrow M[x \backslash N]$$

$${\text{D}}\lambda x. M \cdot N \rightarrow \lambda x. \left( \frac{\partial M}{\partial x} \cdot N \right)$$

Linear approximation

$$f(x) \rightarrow \text{D}f(x) \cdot u$$

$f(x)$ and $\text{D}f(x) \cdot u$ are depicted in the diagram with $x$ as the input point.
Linear and Non-Linear substitutions in Differential $\lambda$-calculus

Substitution

$$(\lambda x. M)N \rightarrow M[x\backslash N]$$

$$(\delta\lambda x. M) \cdot N \rightarrow \lambda x. \left(\frac{\partial M}{\partial x} \cdot N\right)$$

Linear approximation

$$f(x)$$

D$f_x(u)$
Linear-non-linear substitution

Type system and term calculus
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)

\[ x_1 : a_1, \ldots, x_\ell : a_\ell \quad | \quad y_1 : b_1, \ldots, y_n : b_n \vdash t : c \]

Linear

Non-Linear

Linear rules:

\[
\frac{\Gamma, x : a \vdash t : b}{\Gamma \vdash \lambda x^a.t : a \rightarrow b}
\]

\[
\frac{\Gamma \vdash s : a \rightarrow b \quad \Gamma' \vdash t : a}{\Gamma, \Gamma' \vdash (s)t : b}
\]

Non-linear rules:

\[
\frac{\Gamma \vdash s : a \rightarrow b \quad \cdot \quad \Gamma \vdash t : a}{\Gamma \vdash (s)t : b}
\]

\[
\frac{\Gamma \vdash s : a \rightarrow b \quad \Gamma, x : a \vdash t : b}{\Gamma, x : a \vdash (s)t : b}
\]

Linear-non-linear rule:

\[
\frac{\Gamma, x : a \vdash t : b}{\Gamma \vdash \lambda x^a.t : a \rightarrow b}
\]
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)

$$x_1 : a_1, \ldots, x_\ell : a_\ell \quad \vdash \quad y_1 : b_1, \ldots, y_n : b_n \vdash t : c$$

Linear

Non-Linear

Linear rules:

$$x : a \vdash x : a$$

$$\Gamma, x : a \vdash t : b \quad \Gamma \vdash \lambda x^a.t : a \to b$$

Non-linear rules:

$$\Gamma \vdash s : a \to b \quad \Gamma' \vdash t : a$$

$$\Gamma, \Gamma' \vdash (s)t : b$$

$$\Gamma \vdash s : a \to b \quad \cdot \quad \Gamma \vdash t : a$$

$$\Gamma \vdash (s)t : b$$

Linear-non-linear rule:

$$\Gamma, x : a \vdash t : b$$

$$\Gamma \vdash \Delta, x : a \vdash t : b$$
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)

\[ \begin{align*}
\text{Linear} & : \quad \frac{}{x_1 : a_1, \ldots, x_{\ell} : a_{\ell} \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c} \\
\text{Non-Linear} & : \\
\text{Linear rules:} & : \quad \frac{\Delta \vdash x : a}{\Gamma, x : a \vdash \Delta \vdash t : b} \quad \frac{\Delta \vdash t : b}{\Gamma \vdash \Delta \vdash \lambda x^a.t : a \rightarrow b} \\
& \quad \frac{\Gamma \vdash \Delta \vdash s : a \rightarrow b}{\Gamma, \Gamma' \vdash \Delta \vdash \langle s \rangle t : b} \\
\text{Non-linear rules:} & : \quad \frac{\Delta \vdash s : a \rightarrow b}{\Gamma \vdash \Delta \vdash \Gamma' \vdash \Delta \vdash t : a} \quad \frac{\Gamma \vdash \Delta, x : a \vdash t : b}{\Gamma \vdash \Delta \vdash \lambda x^a.t : a \rightarrow b} \\
& \quad \frac{\Gamma \vdash \Delta \vdash s : a \rightarrow b}{\Gamma \vdash \Delta \vdash (s)t : b} \\
\text{Linear-non-linear rule:} & : \quad \frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta, x : a \vdash t : b}
\end{align*} \]
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)

\[
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\underbrace{x_1 : a_1, \ldots, x_\ell : a_\ell}
\end{array} \\
\text{Linear}
\end{array}
\quad & \quad \\
\begin{array}{c}
\begin{array}{c}
\underbrace{y_1 : b_1, \ldots, y_n : b_n} \vdash t : c
\end{array} \\
\text{Non-Linear}
\end{array}
\end{align*}
\]

Linear rules:

- \( \Gamma, x : a \mid \Delta \vdash x : a \)
- \( \Gamma, x : a \mid \Delta \vdash t : b \)
- \( \Gamma \mid \Delta \vdash \lambda x^a.t : a \rightarrow b \)
- \( \Gamma \mid \Delta \vdash s : a \rightarrow b \)
- \( \Gamma' \mid \Delta \vdash t : a \)
- \( \Gamma, \Gamma' \mid \Delta \vdash (s)t : b \)

Non-linear rules:

- \( \Gamma \mid \Delta \vdash s : a \rightarrow b \)
- \( \Gamma \mid \Delta \vdash t : a \)
- \( \Gamma, x : a \mid \Delta \vdash t : b \)
- \( \Gamma \mid \Delta \vdash \lambda x^a.t : a \rightarrow b \)

Linear-non-linear rule:

- \( \Gamma, x : a \mid \Delta \vdash t : b \)
- \( \Gamma, x : a \mid \Delta \vdash t : b \)
- \( \Gamma \mid \Delta, x : a \vdash t : b \)
- \( \Gamma \mid \Delta, x : a \vdash t : b \)
A term calculus for Linear-non-linear Logic

(Benton-Bierman-de Paiva-Hyland 1993, Barber 1996)

\[
\begin{array}{ccc}
\frac{x_1 : a_1, \ldots, x_\ell : a_\ell \quad y_1 : b_1, \ldots, y_n : b_n \vdash t : c}{\text{Linear}}\\
\frac{\text{Non-Linear}}{}
\end{array}
\]

\[
\begin{align*}
\text{MLL} & \quad & \text{λ-calculus} \\
\frac{x : a \vdash x : a}{\Gamma, x : a \vdash t : b} & \quad & \frac{\Delta, x : a \vdash t : b}{\Delta \vdash \lambda x^a.t : a \to b} \\
\frac{\Gamma \vdash s : a \to b \quad \Gamma' \vdash t : a}{\Gamma, \Gamma' \vdash \langle s \rangle t : b} & \quad & \frac{\Delta \vdash s : a \to b \quad \Delta \vdash t : a}{\Delta \vdash (s) t : b}
\end{align*}
\]
What is a model of substitution?

combining linearity and non-linearity
**Axiomatic using Categories**

In a category $X$, equipped with the right structure (SMCC/ CCC)

- **Types** are interpreted as objects
- **Contexts** are interpreted as objects (products/tensors)
- **Terms** are interpreted as morphisms
- **Substitution** is interpreted as composition

In **Multiplicative Linear Logic**, a proof is interpreted as a morphism

$$x_1: a_1, \ldots, x_\ell : a_\ell \vdash t : c \quad \text{as} \quad a_1 \otimes \cdots \otimes a_\ell \leadsto c.$$
Axiomatic using Categories

In a category $X$, equipped with the right structure (SMCC/ CCC)

- **Types** are interpreted as objects
- **Contexts** are interpreted as objects (products/tensors)
- **Terms** are interpreted as morphisms
- **Substitution** is interpreted as composition

In **Multiplicative Linear Logic**, a proof is interpreted as a morphism

$$x_1 : a_1, \ldots, x_\ell : a_\ell \vdash t : c \quad \text{as} \quad a_1 \otimes \cdots \otimes a_\ell \to c.$$
Axiomatic using Categories

In a category $X$, equipped with the right structure (SMCC/ CCC)

- **Types** are interpreted as objects
- **Contexts** are interpreted as objects (products/tensors)
- **Terms** are interpreted as morphisms
- **Substitution** is interpreted as composition

In $\lambda$-calculus, a term is interpreted as a morphism

$$x_1 : b_1, \ldots, x_n : b_n \vdash t : c \quad \text{as} \quad b_1 \times \cdots \times b_n \to c.$$
In a category $X$, equipped with the right structure (SMCC/ CCC)

- **Types** are interpreted as objects
- **Contexts** are interpreted as objects (products/tensors)
- **Terms** are interpreted as morphisms
- **Substitution** is interpreted as composition

In $\lambda$-calculus, a term is interpreted as a morphism

$$x_1 : b_1, \ldots, x_n : b_n \vdash t : c \quad \text{as} \quad b_1 \times \cdots \times b_n \to c.$$
Axiomatic using Categories

In a category $X$, equipped with the right structure (SMCC/ CCC)

- **Types** are interpreted as objects
- **Contexts** are interpreted as objects (products/tensors)
- **Terms** are interpreted as morphisms
- **Substitution** is interpreted as composition

In $\lambda$-calculus, a term is interpreted as a morphism

\[
\begin{align*}
\mathcal{A}_1 : b_1, \ldots, \mathcal{A}_n : b_n \vdash t : c \quad &\text{as} \quad b_1 \times \cdots \times b_n \rightarrow c. \\
&\text{as} \quad b_1 \otimes \cdots \otimes b_n \rightarrow c.
\end{align*}
\]
Axiomatic using Categories

In a category $X$, equipped with the right structure (SMCC/ CCC)

**Types** are interpreted as objects

**Contexts** are interpreted as objects (products/tensors)

**Terms** are interpreted as morphisms

**Substitution** is interpreted as composition

In \( \lambda \)-calculus, a term is interpreted as a morphism

\[
\begin{array}{c}
\begin{array}{c}
\vdash x_1 : b_1, \ldots, x_n : b_n \vdash t : c
\end{array}
\end{array}
\]

as

\[
\begin{array}{c}
\begin{array}{c}
\bigotimes b_1 \times \cdots \times b_n \rightarrow c.
\end{array}
\end{array}
\]

In \textbf{Inl} \( \lambda \)-calculus, \( x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c \) as

\[
\begin{array}{c}
\begin{array}{c}
a_1 \otimes \cdots \otimes a_\ell \otimes! b_1 \otimes \cdots \otimes! b_n \rightarrow c.
\end{array}
\end{array}
\]
Axiomatic using generalized multicategories

A multicategory is a set of operations:

Together with identity and multicomposition:
Axiomatic using generalized Multicategories

**In a multicategory**

- **Types** are interpreted as **objects**
- **Terms** are interpreted as **multimorphisms**
- **Substitution** is interpreted as **multicomposition**.
Axiomatic using generalized Multicategories

In a multicategory

Types are interpreted as objects
Terms are interpreted as multimorphisms
Substitution is interpreted as multicomposition.

In Multiplicative Linear Logic,
a term is interpreted as a multimorphism in a symmetric multicategory:

$$x_1 : a_1, \ldots, x_\ell : a_\ell \vdash t : c$$

denoted as

$$a_1, \ldots, a_\ell \rightarrow c.$$
Axiomatic using generalized Multicategories

In a multicategory

**Types** are interpreted as objects

**Terms** are interpreted as **multi**morphisms

**Substitution** is interpreted as **multi**composition.

In \( \lambda \)-calculus, a term is interpreted as a multimorphism in a cartesian multicategory:

\[
y_1 : b_1, \ldots, y_n : b_n \vdash t : c
\]

denoted as

\[
 b_1, \ldots, b_n \rightarrow c.
\]
Axiomatic using generalized Multicategories

In a multicategory

- **Types** are interpreted as objects
- **Terms** are interpreted as **multi**morphisms
- **Substitution** is interpreted as **multi**composition.

What are the operations for the **Linear-non-linear calculus**, a term is interpreted as a multimorphism in a generalized multicategory:

\[ x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c \]

denoted as

\[ a_1, \ldots, a_\ell, b_1, \ldots, b_n \to c. \]
Axiomatic using generalized Multicategories

In a multicategory

**Types** are interpreted as objects

**Terms** are interpreted as **multi**morphisms

**Substitution** is interpreted as **multi**composition.

What are the operations for the **Linear-non-linear calculus**, a term is interpreted as a multimorphism in a generalized multicategory:

\[ x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c \]

denoted as

\[ a_1, \ldots, a_\ell, b_1, \ldots, b_n \to c. \]

Multicategories can be seen as profunctors combined with a monad.

(Fiore-Plotkin-Turi 1999, Tanaka-Power 2006)
Generalized Multicategories and Context Monads

A **multicategory** is a set of operations: \( M : \mathcal{T} X^{\text{op}} \times X \rightarrow \text{Set} \)

The **context** is represented as a sequence of objects via a **monad** \( \mathcal{T} \) on **Cat**.

Together with unit and **multicomposition**: \( M \circ \mathcal{T} M \Rightarrow M \)
Axiomatization using Multicategories via Profunctors

In a multicategory \( M : \mathcal{T} X^{op} \times X \to \text{Set} \):

- **Types** are interpreted as objects in \( X \)
- **Terms** are interpreted as elements of \( M \)
- **Substitution** is interpreted by the monadic structure \( M \circ \mathcal{T} M \Rightarrow M \)
Axiomatization using Multicategories via Profunctors

In a multicategory \( M : T X^{op} \times X \to \text{Set} \)

- **Types** are interpreted as objects in \( X \)
- **Terms** are interpreted as elements of \( M \)
- **Substitution** is interpreted by the monadic structure \( M \circ T M \Rightarrow M \)

In **Multiplicative Linear Logic**, a term is interpreted in a symmetric multicategory: \( M : L X^{op} \times X \to \text{Set} \)

\( x_1 : a_1, \ldots, x_\ell : a_\ell \vdash t : c \) as \( a_1, \ldots, a_\ell \rightharpoonup c \) in \( M(\langle a_1, \ldots, a_\ell \rangle; c) \)

Algebras of \( L \) are symmetric strict monoidal categories
\( L X \) is the free one on \( X \)

(Fiore-Gambino-Hyland-Winskel 2007)
Axiomatization using Multicategories via Profunctors

In a multicategory $M : \mathcal{T} X \text{op} \times X \to \text{Set}$

- **Types** are interpreted as objects in $X$
- **Terms** are interpreted as elements of $M$
- **Substitution** is interpreted by the monadic structure $M \circ \mathcal{T} M \Rightarrow M$

In $\lambda$-calculus,
a term is interpreted in a cartesian multicategory $M : MX \text{op} \times X \to \text{Set}$

$y_1 : b_1, \ldots, y_n : b_n \vdash t : c$ as $b_1, \ldots, b_n \to c$ in $M(\langle b_1, \ldots, b_n \rangle; c)$

Algebras of $M$ are the categories with product

$MX$ is the free one over $X$

Axiomatization using Multicategories via Profunctors

In a multicategory \( M : \mathcal{T} X^{op} \times X \rightarrow \text{Set} \):

- **Types** are interpreted as objects in \( X \)
- **Terms** are interpreted as elements of \( M \)
- **Substitution** is interpreted by the monadic structure \( M \circ \mathcal{T} M \Rightarrow M \)

What is \( Q \) for a **Mixed Linear-Non-Linear calculus**?

A term is interpreted in a generalized multicategory \( M : Q X^{op} \times X \rightarrow \text{Set} : \)

\[
x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c \text{ in } M(\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle; c).
\]

\( QX \) is the category whose objects are mixed LNL sequences.
In a multicategory $M : T X^{op} \times X \to \text{Set}$

Types are interpreted as objects in $X$

Terms are interpreted as elements of $M$

Substitution is interpreted by the monadic structure $M \circ TM \Rightarrow M$

What is $Q$ for a Mixed Linear-Non-Linear calculus

a term is interpreted in a generalized multicategory $M : Q X^{op} \times X \to \text{Set}$:

$x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : b_1, \ldots, y_n : b_n \vdash t : c$ in $M(\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle; c)$.

$QX$ is the category whose objects are mixed LNL sequences.

Is $Q$ a monad?

What are $Q$-algebras?
A Colimit construction

To build the Linear-non-linear monad
colimits induced by a map in a category $\mathcal{K}$

If $\lambda : A \to B$ is a map in $\mathcal{K}$, then the induced colimit is

$$
\begin{array}{c}
\begin{array}{ccc}
A & \xrightarrow{k} & C \\
\downarrow{\lambda} & & \\
B & \xrightarrow{\ell} & C
\end{array}
\end{array}
$$

- for any

$$
\begin{array}{c}
\begin{array}{ccc}
A & \xrightarrow{f} & D \\
\downarrow{\lambda} & & \\
B & \xrightarrow{g} & D
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{ccc}
A & \xrightarrow{k} & C \\
\downarrow{\lambda} & & \\
B & \xrightarrow{\ell} & C \\
\downarrow{r} & & \\
D & & \\
\end{array}
\end{array}
$$
Colax colimits induced by a map in a 2-category $\mathcal{K}$

If $\lambda : A \to B$ is a map in $\mathcal{K}$, then the induced colax colimit is

$$A \xrightarrow{k} C$$

There are two universal aspects for 1-cells and 2-cells:

1. For any $f : A \to B$ and $g : B \to D$
   $$A \xleftarrow{\lambda} B \xrightarrow{\phi} D$$
   $$\exists ! r : A \xleftarrow{\lambda} B \xrightarrow{\phi} D$$

2. For any $f : A \to B$ and $g : B \to C$
   $$A \xrightarrow{\lambda} B \xrightarrow{\ell} C$$
   $$\exists ! r : A \xrightarrow{\lambda} B \xrightarrow{\ell} C$$

$$A \xrightarrow{f} B \xrightarrow{g} D$$

$$A \xrightarrow{k} C \xrightarrow{\lambda} D$$
**Colax colimits induced by a map in a 2-category \( \mathcal{K} \)**

If \( \lambda : A \to B \) is a map in \( \mathcal{K} \), then the induced colax colimit is

\[
A \quad \xrightarrow{k} \quad B \xrightarrow{\alpha} C
\]

There are two universal aspects for 1-cells and 2-cells

- for any
  \[
  \lambda \downarrow \xrightarrow{\phi} \phantom{f} \\
  B \quad \xrightarrow{g} D
  \]
  \[
  \lambda \downarrow \xrightarrow{k} \\
  B \xrightarrow{\ell} C \quad \exists ! r \\
  D
  \]
- for any
  \[
  \lambda \downarrow \xrightarrow{\phi} \phantom{f} \\
  B \quad \xrightarrow{f} D
  \]
  \[
  \lambda \downarrow \xrightarrow{\alpha} \\
  B \xrightarrow{\ell} C \quad \exists ! r \xRightarrow{\tau} r' \text{ s.t.}
  D
  \]

\[
A \xRightarrow{\rho} D = A \xrightarrow{k} C \xRightarrow{\tau} D
\]

\[
B \xRightarrow{\sigma} D = B \xrightarrow{\ell} C \xRightarrow{\tau} D
\]
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

\[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle a_1, \ldots, a_\ell \rangle \in \mathcal{M}X \]
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category
  \[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in L X \mapsto \langle a_1, \ldots, a_\ell \rangle \in M X \]

Wanted

- \( Q X \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle \)
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

\[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in L X \mapsto \langle a_1, \ldots, a_\ell \rangle \in M X \]

Wanted

- \( Q X \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle \)
- \( Q X \) contains Linear objects

\[ k_X : \langle a_1, \ldots, a_\ell \rangle \in L X \mapsto \langle a_1, \ldots, a_\ell \mid \cdot \rangle \in Q X \]
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category
  \[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \leftrightarrow \langle a_1, \ldots, a_\ell \rangle \in \mathcal{M}X \]

Wanted

- \( QX \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle \)
- \( QX \) contains Linear objects
  \[ k_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \leftrightarrow \langle a_1, \ldots, a_\ell \mid \cdot \rangle \in QX \]
- \( QX \) contains Non-Linear ones
  \[ \ell_X : \langle b_1, \ldots, b_n \rangle \in \mathcal{M}X \leftrightarrow \langle \cdot \mid b_1, \ldots, b_n \rangle \in QX \]
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category
  \[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in L_X \mapsto \langle a_1, \ldots, a_\ell \rangle \in M_X \]

Wanted

- \( Q_X \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle \)
- \( Q_X \) contains Linear objects
  \[ k_X : \langle a_1, \ldots, a_\ell \rangle \in L_X \mapsto \langle a_1, \ldots, a_\ell | \cdot \rangle \in Q_X \]
- \( Q_X \) contains Non-Linear ones
  \[ \ell_X : \langle b_1, \ldots, b_n \rangle \in M_X \mapsto \langle \cdot | b_1, \ldots, b_n \rangle \in Q_X \]

Solution: Colax Colimit over \( \lambda \) in the 2-category of SymStMonCat
Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

\[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle a_1, \ldots, a_\ell \rangle \in \mathcal{M}X \]

Wanted

- \( \mathcal{Q}X \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle \)
- \( \mathcal{Q}X \) contains Linear objects

\[ k_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle a_1, \ldots, a_\ell \mid \cdot \rangle \in \mathcal{Q}X \]

- \( \mathcal{Q}X \) contains Non-Linear ones

\[ \ell_X : \langle b_1, \ldots, b_n \rangle \in \mathcal{M}X \mapsto \langle \cdot \mid b_1, \ldots, b_n \rangle \in \mathcal{Q}X \]

Solution: Colax Colimit over \( \lambda \) in the 2-category of SymStMonCat

\[ \alpha_X, \langle a_1, \ldots, a_\ell \rangle \in \mathcal{Q}X(\langle \cdot \mid a_1, \ldots, a_\ell \rangle, \langle a_1, \ldots, a_\ell \mid \cdot \rangle) \]

\[ x_1 : a_1, \ldots, x_\ell : a_\ell \mid \cdot \vdash t : b \]

\[ \cdot \mid x_1 : a_1, \ldots, x_\ell : a_\ell \vdash t : b \]
Mixing Linear and Non-Linear contexts via a colimit

Remark: \( \lambda : \mathcal{L} \to \mathcal{M} \) is a map of 2-monads

- Every category with products is a symmetric strict monoidal category
  \[ \lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L} X \mapsto \langle \overline{a}_1, \ldots, \overline{a}_\ell \rangle \in \mathcal{M} X \]

Wanted

- \( QX \) is in SymStMonCat and objects are \( \langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle \)
- \( QX \) contains Linear objects
  \[ k_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L} X \mapsto \langle a_1, \ldots, a_\ell \mid \cdot \rangle \in QX \]
- \( QX \) contains Non-Linear ones
  \[ \ell_X : \langle b_1, \ldots, b_n \rangle \in \mathcal{M} X \mapsto \langle \cdot \mid b_1, \ldots, b_n \rangle \in QX \]

Solution: Colax Colimit over \( \lambda \) in the 2-category of SymStMonCat

\[
\begin{array}{ccc}
\mathcal{L} X & \xrightarrow{k_X} & QX \\
\xrightarrow{\lambda_X} & & \xleftarrow{\alpha_X} \\
\mathcal{M} X & \xrightarrow{\ell_X} & QX
\end{array}
\]

\[ \alpha_X, \langle a_1, \ldots, a_\ell \rangle \in QX(\langle \cdot \mid a_1, \ldots, a_\ell \rangle, \langle a_1, \ldots, a_\ell \mid \cdot \rangle) \]

\[
\begin{array}{c}
\frac{x_1 : a_1, \ldots, x_\ell : a_\ell \mid \cdot \vdash t : b}{\cdot \mid x_1 : \overline{a}_1, \ldots, x_\ell : \overline{a}_\ell \vdash t : b}
\end{array}
\]
A Colimit construction

\( Q \) is a 2-monad on \( \text{Cat} \) and \( Q \)-algebras
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$
- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$
- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra

and coherences
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $\mathcal{L}X$ the free symmetric strict monoidal category
- $\mathcal{M}X$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$
- $QX$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $QX$ is equipped with a strictly idempotent comonad

\[
f : QX \xrightarrow{hX} MX \xrightarrow{kX} QX
\]
Properties of the $QX$ from universality for 1-cell and 2-cell Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle$

- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
- $QX$ is equipped with a strictly idempotent comonad $f : QX \xrightarrow{hX} MX \xrightarrow{kX} QX$

\[ \langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle \mapsto \langle a_1, \ldots, a_\ell, b_1, \ldots b_n \rangle \mapsto \langle \cdot | a_1, \ldots, a_\ell, b_1, \ldots, b_n \rangle \]
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric strict monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle$

- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
- $QX$ is equipped with a strictly idempotent comonad

$$f : QX \xrightarrow{h_X} MX \xrightarrow{k_X} QX$$

$$\langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle \mapsto \langle a_1, \ldots, a_\ell, b_1, \ldots b_n \rangle \mapsto \langle \cdot | a_1, \ldots, a_\ell, b_1, \ldots, b_n \rangle$$

$$f \Rightarrow \beta \Rightarrow \text{id}_{QX}$$

$$\frac{x_1 : a_1, \ldots, x_\ell : a_\ell | \Delta \vdash t : b}{\cdot | x_1 : a_1, \ldots, x_\ell : a_\ell, \Delta \vdash t : b}$$

$$\beta \langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle : \langle \cdot | a_1, \ldots, a_\ell, b_1, \ldots, b_n \rangle \rightarrow \langle a_1, \ldots, a_\ell | b_1, \ldots, b_n \rangle$$
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$
- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
- $QX$ is equipped with a strictly idempotent comonad
  \[ f : QX \xrightarrow{h_X} MX \xrightarrow{k_X} QX \]
- $QX$ is almost with products, i.e. a left-semi $M$-algebra:
  \[ MQX \xrightarrow{z} QX \]
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$

- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
- $QX$ is equipped with a strictly idempotent comonad
  
  $f : QX \xrightarrow{h_X} MX \xrightarrow{k_X} QX$

- $QX$ is almost with products, i.e. a left-semi $M$-algebra:
  
  $MQX \xrightarrow{z} QX$

and coherences
Properties of the $QX$ from universality for 1-cell and 2-cell

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$

- $QX$ is a symmetric strict monoidal category, i.e. an $L$-algebra
- $QX$ is equipped with a strictly idempotent comonad
  \[ f : QX \xrightarrow{hX} MX \xrightarrow{kX} QX \]
- $QX$ is almost with products, i.e. a left-semi $M$-algebra:
  \[ MQX \xrightarrow{z} QX \]
- The induced left-semi $L$-algebras are equal:
  \[ LQX \xrightarrow{\lambda_{QX}} MQX \xrightarrow{z} QX \]
  \[ LQX \xrightarrow{w} QX \xrightarrow{f} QX \]
Structure category of $Q$

Colimit in the 2-category of Symmetric Strict Monoidal Categories.

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$

Structure Category

- $Z$ is a symm. str. monoidal category, i.e. an $\mathcal{L}$-algebra: $\mathcal{L}Z \xrightarrow{w} Z$
- $Z$ is almost with products, i.e. a left-semi $\mathcal{M}$-algebra: $\mathcal{M}Z \xrightarrow{z} Z$
- $Z$ is equipped with a strictly idempotent comonad $f : Z \xrightarrow{\eta_z} \mathcal{M}Z \xrightarrow{z} Z$
- The induced left-semi $\mathcal{L}$-algebras are equal:
  $$\mathcal{L}Z \xrightarrow{\lambda_z} \mathcal{M}Z \xrightarrow{z} Z = \mathcal{L}Z \xrightarrow{w} Z \xrightarrow{f} Z$$
**Structure category of $Q$**

**Colimit** in the 2-category of Symmetric Strict Monoidal Categories.

- $LX$ the free symmetric st. monoidal category
- $MX$ the free category with products
- $QX$ objects are $\langle a_1, \ldots, a_\ell \mid b_1, \ldots, b_n \rangle$

---

**Structure Category**

- $Z$ is a symm. str. monoidal category, i.e. an $L$-algebra: $LZ \xrightarrow{w} Z$
- $Z$ is almost with products, i.e. a left-semi $M$-algebra: $MZ \xrightarrow{z} Z$
- $Z$ is equipped with a strictly idempotent comonad
  $$f : Z \xrightarrow{\eta_z} MZ \xrightarrow{z} Z$$
- The induced left-semi $L$-algebras are equal:
  $$LZ \xrightarrow{\lambda_Z} MZ \xrightarrow{z} Z \quad \quad \quad LZ \xrightarrow{w} Z \xrightarrow{f} Z$$

$QX$ has this structure!
A general colax colimit construction on 2-monads

Let $\lambda : \mathcal{L} \to \mathcal{M}$ a map of 2-monad on $\mathbf{Cat}$. If $\mathcal{L}$-algebras has colimits, then the colimit is:

$$
\begin{array}{c}
\mathcal{L}X \\
\lambda_X \\
\mathcal{M}X \\
\alpha_X \\
\mathcal{Q}X
\end{array}
\xymatrix{
\mathcal{L}X \\
\mathcal{M}X \\
\mathcal{Q}X
}
$$

Theorem

- $\mathcal{Q}$-algebras are objects in the Structure Category and
- $\mathcal{Q}$ is a 2-monad on $\mathbf{Cat}$.

The proof uses universality of the colimit.
It is not the end of the story

Does $Q$ lifts from Cat to Prof?
A **multicategory** can be seen as a **profunctor** in the Kleisli bicat of $\mathcal{T}$:

$$M : X \mapsto \mathcal{T}X \quad M : \mathcal{T}X^{\text{op}} \times X \to \text{Set}$$

Together with unit and **multicomposition**: $M \circ \mathcal{T}M \Rightarrow M$

$Q$ on $\text{Cat}$ has to **extend** to $\text{Prof}$ as $L$ and $M$. Equivalently,

Does the presheaf pseudomonad $\mathbf{Psh}$ lifts to pseudo $Q$-algebras?

Does $Q$ extends from $\text{Cat}$ to $\text{Prof}$?

**Wanted:** The presheaf pseudomonad $\text{Psh}$ lifts to pseudo $Q$-algebras.
Does $Q$ extends from Cat to Prof?

**Wanted:** The presheaf pseudomonad $\text{Psh}$ lifts to pseudo $Q$-algebras.

**Conjecture:** Pseudo $Q$-algebras are pseudo Structure Categories.
Does $Q$ extends from Cat to Prof?

**Wanted:** The presheaf pseudomonad $\text{Psh}$ lifts to pseudo $Q$-algebras.

**Conjecture:** Pseudo $Q$-algebras are pseudo Structure Categories.

**Problem:** Pseudo $Q$-algebras are pseudo $\mathcal{L}$-algebras, i.e. symmetric monoidal categories. There is NO COLIMIT in the 2-category of symmetric monoidal categories.
Does $Q$ extends from $\text{Cat}$ to $\text{Prof}$?

**Wanted:** The presheaf pseudomonad $\text{Psh}$ lifts to pseudo $Q$-algebras.

**Conjecture:** Pseudo $Q$-algebras are pseudo Structure Categories.

**Problem:** Pseudo $Q$-algebras are pseudo $\mathcal{L}$-algebras, i.e. symmetric monoidal categories. There is NO COLIMIT in the 2-category of symmetric monoidal categories.

**Work in progress:** Use a strictification to recover colimits.
It is not the end of the story

Categorical axiomatizations?
Can we build a $Q$-multicategory from a LNL adjunction?

Linear-non-linear adjunction

A monoidal adjunction $\xymatrix{X \ar@/^1.5pc/[r]^s & Y \ar@/_1.5pc/[r]_r \ar@<0.5ex>[l]^\tau}$

with $X$ symmetric strict monoidal and $Y$ with products
Can we build a $Q$-multicategory from a LNL adjunction?

**Linear-non-linear adjunction**

A monoidal adjunction $X \xleftarrow{r} \xrightarrow{s} Y$

with $X$ symmetric strict monoidal and $Y$ with products

**Structure Category** from a ?

- $X$ is symmetric strict monoidal, so $\mathcal{L}X \xrightarrow{w} X$
- $Y$ has products, so $\mathcal{M}Y \xrightarrow{y} Y$, we can build $\mathcal{M}X \xrightarrow{r} \mathcal{M}Y \xrightarrow{y} Y \xrightarrow{s} X$
- $f = ! : X \xrightarrow{r} Y \xrightarrow{s} X$ is only lax monoidal
- The two induced left-semi $\mathcal{L}$-algebras are not equal
Can we build a $Q$-multicategory from a LNL adjunction?

**Linear-non-linear adjunction** (Benton 1994)

A monoidal adjunction $X \xleftrightarrow{\tau} Y$

with $X$ symmetric strict monoidal and $Y$ with products

**Structure Category** from a ?

- $X$ is symmetric strict monoidal, so $LX \xrightarrow{w} X$
- $Y$ has products, so $MY \xrightarrow{y} Y$, we can build

$$MX \xrightarrow{r} MY \xrightarrow{y} Y \xrightarrow{s} X$$

- $f =! : X \xrightarrow{r} Y \xrightarrow{s} X$ is only lax monoidal
- The two induced left-semi $L$-algebras are not equal

**Work in progress:** recover a structure category by going through multicategories.
It is not the end of the story

Differential $\lambda$-calculus axiomatization
Towards a multicategorical model of differential $\lambda$-calculus

**Substitution:** linear-non-linear multicategories

**Variable binding** $\lambda x.s$  
(Fiore-Plotkin-Turi 1999, Hyland 2017)

- Closed structure which allows to turn an operation with $n + 1$ inputs to an operation with $n$ inputs.
Towards a multicategorical model of differential $\lambda$-calculus

**Substitution:** linear-non-linear multicategories

Variable binding $\lambda x.s$ (Fiore-Plotkin-Turi 1999, Hyland 2017)

- Closed structure which allows to turn an operation with $n+1$ inputs to an operation with $n$ inputs.

Derivation $u, x \mapsto D_x f(u)$

- Differential interaction nets (Ehrhard-Regnier 2006)
- Differential categories (Blute-Cockett-Seely 2006)
Towards a multicategorical model of differential $\lambda$-calculus

**Substitution:** linear-non-linear multicategories

**Variable binding** $\lambda x.s$  
(Fiore-Plotkin-Turi 1999, Hyland 2017)
- Closed structure which allows to turn an operation with $n+1$ inputs to an operation with $n$ inputs.

**Derivation** $u, x \mapsto D_x f(u)$
- Differential interaction nets  
  (Ehrhard-Regnier 2006)
- Differential categories  
  (Blute-Cockett-Seely 2006)

**Chain rule**
- An additive structure due to the derivation of the contraction.
"The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise".

(E. Dijkstra, The Humble Programmer, ACM Turing Lecture, 1972)