The Linear-Non-Linear Substitution Monad

OWLS, 15 July 2020

Christine Tasson (tasson@irif.fr)
Joint work with Martin Hyland at https://arxiv.org/abs/2005.09559

Institut de Recherche en Informatique Fondamentale

Roadmap: The linear-non-linear substitution monad

Motivation:

- Differential λ-calculus

Goal:

- Axiomatisation using generalized multicategories.

Tool:

- A colimit construction applied to combine 2-monads on **Cat**

Results:

- The colimit is a 2-monad.
- Charaterization of its algebras.

Linear-non-linear substitution

Substitutions in differential λ-calculus

Differential λ-Calculus

(Ehrhard-Regnier 2003, Ehrhard 2018)

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

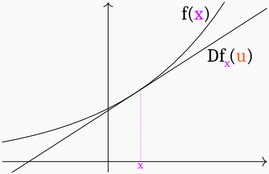
	Programs	Functions	
	M, N	f, g	
Variable	x	x	Variable
Abstraction	$\lambda x.M$	$f : x \mapsto f(x)$	Map
Application	$(\lambda x.M)N$	$f \circ g : x \mapsto f(g(x))$	Composition
Differentiation	$D\lambda x.M \cdot N$	$u, x \mapsto Df_x(u)$	Derivation

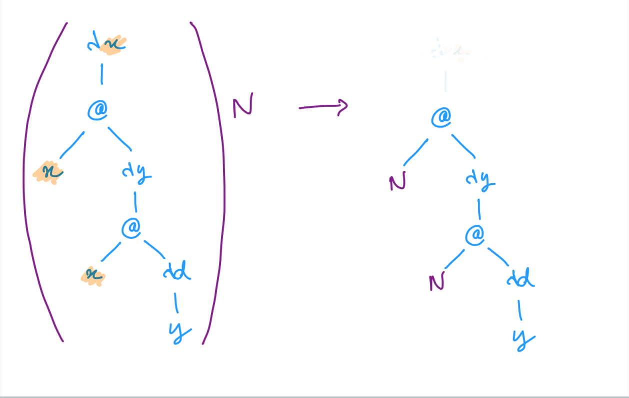
Linear and Non-Linear substitutions in Differential λ-calculus

Substitution

$$
\begin{aligned}
(\lambda x.M)N &\quad\to\quad M[x\backslash N] \\
D\lambda x.M\cdot N &\quad\to\quad \lambda x.\left(\frac{\partial M}{\partial x}\cdot N\right)
\end{aligned}
$$

Linear approximation



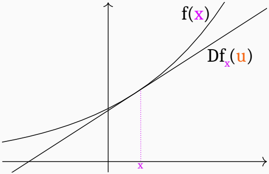


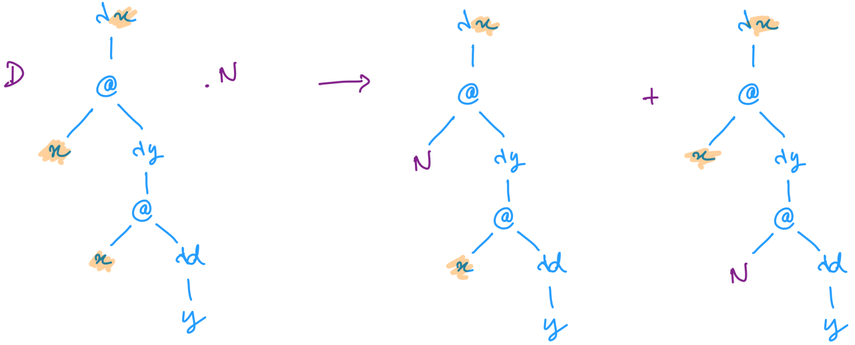
Linear and Non-Linear substitutions in Differential λ-calculus

Substitution

$$\begin{array}{rcl} (\lambda x.M)N & \to & M[x\backslash N] \\ D\lambda x.M \cdot N & \to & \lambda x.\left(\dfrac{\partial M}{\partial x} \cdot N\right) \end{array}$$

Linear approximation





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)

$$\underbrace{x_1 : a_1, \ldots, x_\ell : a_\ell}_{\text{Linear}} \mid \underbrace{\underline{y}_1 : \underline{b}_1, \ldots, \underline{y}_n : \underline{b}_n}_{\text{Non-Linear}} \vdash t : c$$

Linear rules:

$$\frac{}{x : a \mid \Delta \vdash x : a} \qquad \frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^a.t : a \multimap b}$$

$$\frac{\Gamma \mid \Delta \vdash s : a \multimap b \qquad \Gamma' \mid \Delta \vdash t : a}{\Gamma, \Gamma' \mid \Delta \vdash \langle s \rangle t : b}$$

Non-linear rules:

$$\frac{}{\Gamma \mid \Delta, x : \underline{b} \vdash x : \underline{b}} \qquad \frac{\Gamma \mid \Delta, x : \underline{a} \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^{\underline{a}}.t : \underline{a} \to b}$$

$$\frac{\Gamma \mid \Delta \vdash s : a \to b \qquad \cdot \mid \Delta \vdash t : a}{\Gamma \mid \Delta \vdash (s)t : b}$$

Linear-non-linear rule:

$$\frac{\Gamma, \ x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta, \ x : \underline{a} \vdash t : b}$$

A term calculus for Linear-non-linear Logic

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

$$\boxed{\underbrace{x_1 : a_1, \ldots, x_\ell : a_\ell}_{\text{Linear}} \mid \underbrace{\underline{y}_1 : \underline{b}_1, \ldots, \underline{y}_n : \underline{b}_n}_{\text{Non-Linear}} \vdash t : c}$$

Linear rules:
$$\frac{}{x : a \mid \Delta \vdash x : a} \qquad \frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^a.t : a \multimap b}$$

$$\boxed{\frac{\Gamma \mid \Delta \vdash s : a \multimap b \qquad \Gamma' \mid \Delta \vdash t : a}{\Gamma, \Gamma' \mid \Delta \vdash \langle s \rangle t : b}}$$

Non-linear rules:
$$\frac{}{\Gamma \mid \Delta, x : \underline{b} \vdash x : \underline{b}} \qquad \frac{\Gamma \mid \Delta, x : \underline{a} \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^{\underline{a}}.t : \underline{a} \to b}$$

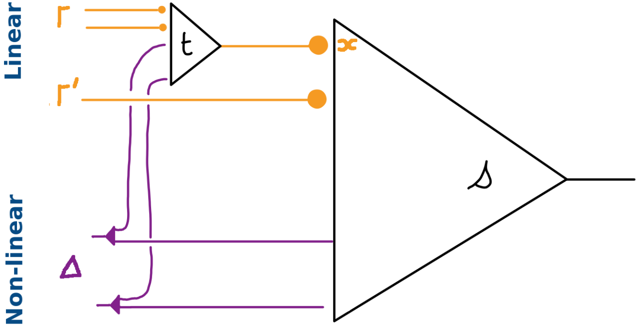
$$\frac{\Gamma \mid \Delta \vdash s : a \to b \qquad \cdot \mid \Delta \vdash t : a}{\Gamma \mid \Delta \vdash (s)t : b}$$

Linear-non-linear rule:
$$\frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta, x : \underline{a} \vdash t : b}$$

A term calculus for Linear-non-linear Logic

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

$$\frac{\Gamma \mid \Delta \vdash t \qquad x, \Gamma' \mid \Delta \vdash s}{\Gamma, \Gamma' \mid \Delta \vdash s[t/x]}$$



A term calculus for Linear-non-linear Logic

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

$$\boxed{\underbrace{x_1 : a_1, \ldots, x_\ell : a_\ell}_{\text{Linear}} \mid \underbrace{\underline{y}_1 : \underline{b}_1, \ldots, \underline{y}_n : \underline{b}_n}_{\text{Non-Linear}} \vdash t : c}$$

Linear rules:

$$\frac{}{x : a \mid \Delta \vdash x : a} \qquad \frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^a.t : a \multimap b}$$

$$\frac{\Gamma \mid \Delta \vdash s : a \multimap b \qquad \Gamma' \mid \Delta \vdash t : a}{\Gamma, \Gamma' \mid \Delta \vdash \langle s \rangle t : b}$$

Non-linear rules:

$$\frac{}{\Gamma \mid \Delta, x : \underline{b} \vdash x : \underline{b}} \qquad \frac{\Gamma \mid \Delta, x : \underline{a} \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^{\underline{a}}.t : \underline{a} \to b}$$

$$\boxed{\frac{\Gamma \mid \Delta \vdash s : a \to b \qquad \cdot \mid \Delta \vdash t : a}{\Gamma \mid \Delta \vdash (s)t : b}}$$

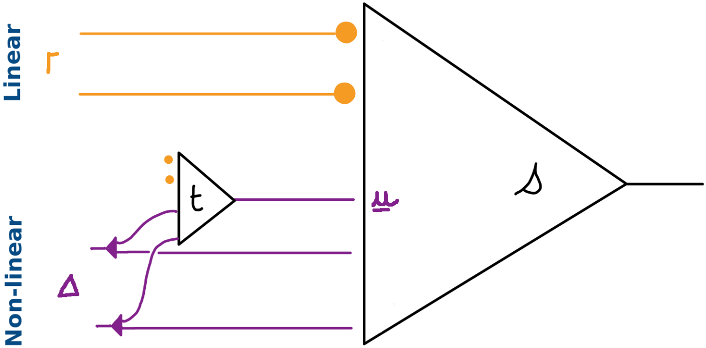
Linear-non-linear rule:

$$\frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta, x : \underline{a} \vdash t : b}$$

A term calculus for Linear-non-linear Logic

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

$$\frac{\cdot \mid \Delta \vdash t \qquad \Gamma \mid \underline{u}, \Delta \vdash s}{\Gamma \mid \Delta \vdash s[t/\underline{u}]}$$



A term calculus for Linear-non-linear Logic

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

$$\boxed{\underbrace{x_1 : a_1, \ldots, x_\ell : a_\ell}_{\text{Linear}} \mid \underbrace{\underline{y}_1 : \underline{b}_1, \ldots, \underline{y}_n : \underline{b}_n}_{\text{Non-Linear}} \vdash t : c}$$

Linear rules:

$$\frac{}{x : a \mid \Delta \vdash x : a} \qquad \frac{\Gamma, x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^a.t : a \multimap b}$$

$$\frac{\Gamma \mid \Delta \vdash s : a \multimap b \qquad \Gamma' \mid \Delta \vdash t : a}{\Gamma, \Gamma' \mid \Delta \vdash \langle s \rangle t : b}$$

Non-linear rules:

$$\frac{}{\Gamma \mid \Delta, x : \underline{b} \vdash x : \underline{b}} \qquad \frac{\Gamma \mid \Delta,\ x : \underline{a} \vdash t : b}{\Gamma \mid \Delta \vdash \lambda x^{\underline{a}}.t : \underline{a} \to b}$$

$$\frac{\Gamma \mid \Delta \vdash s : a \to b \qquad \cdot \mid \Delta \vdash t : a}{\Gamma \mid \Delta \vdash (s)t : b}$$

Linear-non-linear rule:

$$\frac{\Gamma\ ,\ x : a \mid \Delta \vdash t : b}{\Gamma \mid \Delta\ ,\ x : \underline{a} \vdash t : b}$$

A term calculus for Linear-non-linear Logic

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

$$\underbrace{x_1 : a_1, \dots, x_\ell : a_\ell}_{\text{Linear}} \mid \underbrace{\underline{y}_1 : \underline{b}_1, \dots, \underline{y}_n : \underline{b}_n}_{\text{Non-Linear}} \vdash t : c$$

MLL

$$\frac{}{x : a \vdash x : a} \qquad \frac{\Gamma, x : a \vdash t : b}{\Gamma \vdash \lambda x^a.t : a \multimap b}$$

$$\frac{\Gamma \vdash s : a \multimap b \qquad \Gamma' \vdash t : a}{\Gamma, \Gamma' \vdash \langle s \rangle t : b}$$

λ-calculus

$$\frac{}{\Delta, x : \underline{b} \vdash x : \underline{b}} \qquad \frac{\Delta,\ x : \underline{a} \vdash t : b}{\Delta \vdash \lambda x^{\underline{a}}.t : \underline{a} \to b}$$

$$\frac{\Delta \vdash s : a \to b \qquad \Delta \vdash t : a}{\Delta \vdash (s)t : b}$$

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

$$x_1 : a_1, \ldots, x_\ell : a_\ell \vdash t : c \qquad \text{as} \qquad a_1 \otimes \cdots \otimes a_\ell \multimap c.$$

Axiomatic using Categories

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

Types are interpreted as objects

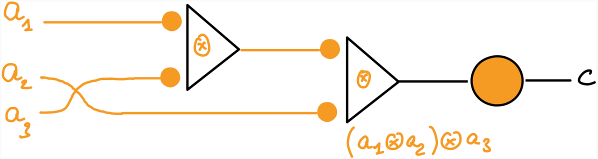
Contexts are interpreted as objects (products/tensors)

Terms are interpreted as morphisms

Substitution is interpreted as composition

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

$$x_1 : a_1, \dots, x_\ell : a_\ell \vdash t : c \qquad \text{as} \qquad a_1 \otimes \cdots \otimes a_\ell \multimap c.$$



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 λ-**calculus,** a term is interpreted as a morphism

$$x_1 : \underline{b}_1, \ldots, x_n : \underline{b}_n \vdash t : c \qquad \text{as} \qquad \underline{b}_1 \times \cdots \times \underline{b}_n \to c\,.$$

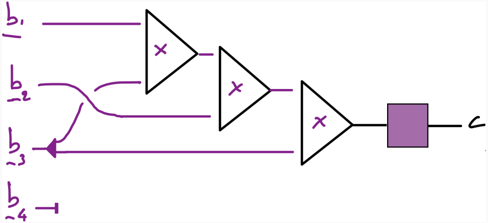
Axiomatic using Categories

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 λ**-calculus,** a term is interpreted as a morphism

$$x_1 : \underline{b}_1, \ldots, x_n : \underline{b}_n \vdash t : c \qquad \text{as} \qquad \underline{b}_1 \times \cdots \times \underline{b}_n \to c\,.$$



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 λ**-calculus,** a term is interpreted as a morphism

$$x_1 : \underline{b}_1, \ldots, x_n : \underline{b}_n \vdash t : c \qquad \text{as} \qquad \underset{!\underline{b}_1 \otimes \cdots \otimes !\underline{b}_n \multimap c}{\underline{b}_1 \times \cdots \times \underline{b}_n \to c}\,.$$

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 λ**-calculus,** a term is interpreted as a morphism

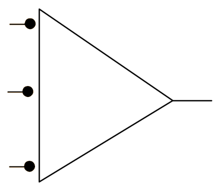
$$x_1 : \underline{b}_1, \ldots, x_n : \underline{b}_n \vdash t : c \qquad \text{as} \qquad \underbrace{\underline{b}_1 \times \cdots \times \underline{b}_n \longrightarrow c}_{!\underline{b}_1 \otimes \cdots \otimes !\underline{b}_n \multimap c} .$$

In **lnl** λ**-calculus,** $x_1 : a_1, \ldots, x_\ell : a_\ell \mid \underline{y}_1 : \underline{b}_1, \ldots, \underline{y}_n : \underline{b}_n \vdash t : c$ as

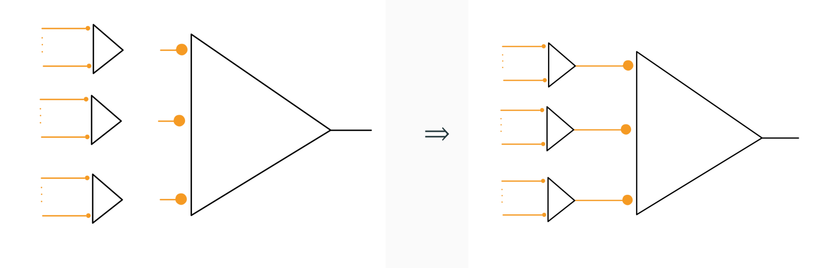
$$a_1 \otimes \cdots \otimes a_\ell \otimes !\underline{b}_1 \otimes \cdots \otimes !\underline{b}_n \multimap c.$$

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 **multi**morphisms

Substitution is interpreted as **multi**composition.

Axiomatic using generalized Multicategories

In a multicategory

Types are interpreted as objects

Terms are interpreted as **multi**morphisms

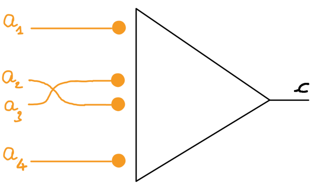
Substitution is interpreted as **multi**composition.

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

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

denoted as

$$a_1, \dots, a_\ell \multimap c.$$



Axiomatic using generalized Multicategories

In a multicategory

Types are interpreted as objects

Terms are interpreted as **multi**morphisms

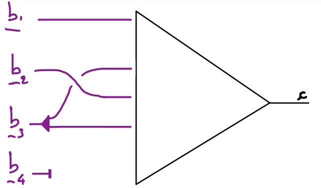
Substitution is interpreted as **multi**composition.

In λ-**calculus,**
a term is interpreted as a multimorphism in a cartesian multicategory:

$$y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c$$

denoted as

$$\underline{b}_1, \ldots, \underline{b}_n \to c.$$



Axiomatic using generalized Multicategories

In a multicategory

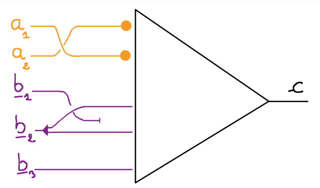
Types are interpreted as objects

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:

$x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c$



denoted as

$$a_1, \ldots, a_\ell, \underline{b}_1, \ldots, \underline{b}_n \to c.$$

Axiomatic using generalized Multicategories

In a multicategory

Types are interpreted as objects

Terms are interpreted as **multi**morphisms

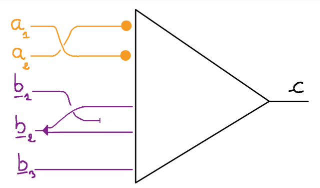
Substitution is interpreted as **multi**composition.

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

$$x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c$$

denoted as

$$a_1, \ldots, a_\ell, \underline{b}_1, \ldots, \underline{b}_n \to c.$$



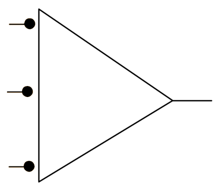
Multicategories can be seen as profunctors combined with a monad.

(Fiore-Plotkin-Turi 1999, Tanaka-Power 2006)

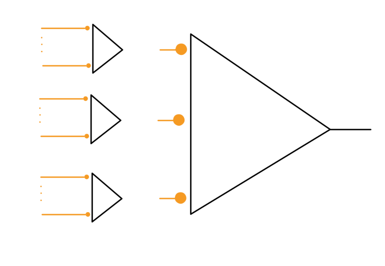
Generalized Multicategories and Context Monads

A **multicategory** is a set of operations: $M : \mathcal{T}X^{\mathrm{op}} \times X \to \mathbf{Set}$

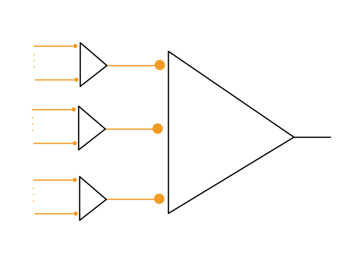
The **context** is represented as a sequence of objects via a **monad** $\mathcal{T}$ on **Cat**.



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



$\Rightarrow$



Axiomatization using Multicategories via Profunctors

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

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 $M : \mathcal{T}X^{op} \times X \to \mathbf{Set}$

Types are interpreted as objects in X

Terms are interpreted as elements of M

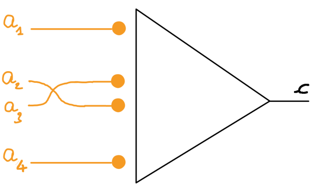
Substitution is interpreted by the monadic structure $M \circ \mathcal{T}M \Rightarrow M$

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

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

Algebras of $\mathcal{L}$ are symmetric strict monoidal categories
$\mathcal{L}X$ is the free one on X

(Fiore-Gambino-Hyland-Winskel 2007)



Axiomatization using Multicategories via Profunctors

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

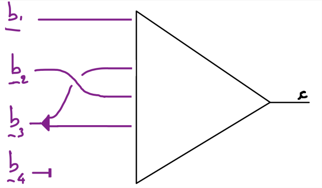
Types	are interpreted as objects in X
Terms	are interpreted as elements of M
Substitution	is interpreted by the monadic structure $M \circ \mathcal{T}M \Rightarrow M$

In **λ-calculus,**
a term is interpreted in a cartesian multicategory $M : \mathcal{M}X^{op} \times X \to \mathbf{Set}$

$$y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c \qquad \text{as} \qquad \underline{b}_1, \ldots, \underline{b}_n \multimap c \quad \text{in} \quad M(\langle \underline{b}_1, \ldots, \underline{b}_n \rangle; c)$$

Algebras of $\mathcal{M}$ are the categories with product
$\mathcal{M}X$ is the free one over X

(Tanaka-Power 2004, Hyland 2017)



Axiomatization using Multicategories via Profunctors

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

Types are interpreted as objects in X

Terms are interpreted as elements of M

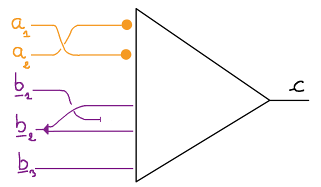
Substitution is interpreted by the monadic structure $M \circ \mathcal{T}M \Rightarrow M$

What is $\mathcal{Q}$ for a **Mixed Linear-Non-Linear calculus**

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

$$x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c \text{ in } M(\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle; c).$$

$\mathcal{Q}X$ is the category whose objects are mixed LNL sequences.



Axiomatization using Multicategories via Profunctors

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

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 $\mathcal{Q}$ for a **Mixed Linear-Non-Linear calculus**

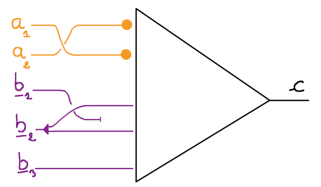
a term is interpreted in a generalized multicategory $M : \mathcal{Q}X^{op} \times X \to \mathbf{Set}$:

$$x_1 : a_1, \ldots, x_\ell : a_\ell \mid y_1 : \underline{b}_1, \ldots, y_n : \underline{b}_n \vdash t : c \text{ in } M(\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle; c).$$

$\mathcal{Q}X$ is the category whose objects are mixed LNL sequences.

Is $\mathcal{Q}$ a monad ?

What are $\mathcal{Q}$-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
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & \\ B \arrow[r, "\ell"'] & C \end{tikzcd}$$

- for any
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "f"] & \\ B \arrow[r, "g"'] & D \end{tikzcd} = \begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & & \\ B \arrow[r, "\ell"'] & C \arrow[r, dotted, "\exists! r"] & D \end{tikzcd}$$

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
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & \\ B \arrow[r, "\ell"'] \arrow[ur, phantom, "\Nearrow\alpha"] & C \end{tikzcd}$$

- for any $$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "f"] & \\ B \arrow[r, "g"'] \arrow[ur, phantom, "\Nearrow\phi"] & D \end{tikzcd} = \begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & & \\ B \arrow[r, "\ell"'] \arrow[ur, phantom, "\Nearrow\alpha"] & C \arrow[r, dotted, "\exists! r"] & D \end{tikzcd}$$

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
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & \\ B \arrow[r, "\ell"'] \arrow[ur, phantom, "\Nearrow\alpha"] & C \end{tikzcd}$$

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

- for any
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "f"] & \\ B \arrow[r, "g"'] \arrow[ur, phantom, "\Nearrow\phi"] & D \end{tikzcd} = \begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "k"] & & \\ B \arrow[r, "\ell"'] \arrow[ur, phantom, "\Nearrow\alpha"] & C \arrow[r, dotted, "\exists! r"] & D \end{tikzcd}$$

- for any
$$\begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left=40, "f'"] \arrow[dr, bend left=10, "f"'] & \\ B \arrow[r, "g"'] \arrow[ur, phantom, "\Nearrow\phi"] & D \end{tikzcd} \ (\Uparrow\rho) = \begin{tikzcd} A \arrow[d, "\lambda"'] \arrow[dr, bend left, "f'"] & \\ B \arrow[r, "g"'] \arrow[r, bend left, "g'"] \arrow[r, phantom, "\sigma\Uparrow"] \arrow[ur, phantom, "\phi'\Nearrow"] & D \end{tikzcd}$$
, $\exists!\ r \overset{\tau}{\Rightarrow} r'$ s.t.

$$\begin{tikzcd} A \arrow[r, bend left, "f'"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "\rho\Uparrow"] & D \end{tikzcd} = \begin{tikzcd} A \arrow[r, "k"] & C \arrow[r, bend left, "r'"] \arrow[r, bend right, "r"'] \arrow[r, phantom, "\tau\Uparrow"] & D \end{tikzcd} \qquad \begin{tikzcd} B \arrow[r, bend left, "g'"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\sigma\Uparrow"] & D \end{tikzcd} = \begin{tikzcd} B \arrow[r, "\ell"] & C \arrow[r, bend left, "r'"] \arrow[r, bend right, "r"'] \arrow[r, phantom, "\tau\Uparrow"] & D \end{tikzcd}$$

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$
- $\mathcal{Q}X$ contains Linear objects

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

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$
- $\mathcal{Q}X$ 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

$$\ell_X : \langle \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{M}X \mapsto \langle \cdot \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{Q}X$$

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$
- $\mathcal{Q}X$ 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

$$\ell_X : \langle \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{M}X \mapsto \langle \cdot \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{Q}X$$

Solution: Colax Colimit over λ in the 2-category of SymStMonCat

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

$$\alpha_{X,\langle a_1,\ldots,a_\ell \rangle} \in \mathcal{Q}X(\langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell \rangle\ ,\ \langle a_1, \ldots, a_\ell \mid \cdot \rangle)$$

Mixing Linear and Non-Linear contexts via a colimit

Remark:

- Every category with products is a symmetric strict monoidal category

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$
- $\mathcal{Q}X$ 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

$$\ell_X : \langle \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{M}X \mapsto \langle \cdot \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{Q}X$$

Solution: Colax Colimit over λ in the 2-category of SymStMonCat

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[drr, bend left, "k_X"] & & \\ \mathcal{M}X \arrow[rr, "\ell_X"'] \arrow[urr, phantom, "\Nearrow \alpha_X"] & & \mathcal{Q}X \end{tikzcd}$$

$$\alpha_{X, \langle a_1, \ldots, a_\ell \rangle} \in \mathcal{Q}X(\langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell \rangle \,,\ \langle a_1, \ldots, a_\ell \mid \cdot \rangle)$$

$$\boxed{\dfrac{x_1 : a_1, \ldots, x_\ell : a_\ell \mid \cdot \vdash t : b}{\cdot \mid x_1 : \underline{a}_1, \ldots, x_\ell : \underline{a}_\ell \vdash t : b}}$$

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

$$\lambda_X : \langle a_1, \ldots, a_\ell \rangle \in \mathcal{L}X \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell \rangle \in \mathcal{M}X$$

Wanted

- $\mathcal{Q}X$ is in SymStMonCat and objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$
- $\mathcal{Q}X$ contains Linear objects $k : \mathcal{L} \to \mathcal{Q}$ is a map of 2-monad

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

- $\mathcal{Q}X$ contains Non-Linear ones $\ell : \mathcal{M} \to \mathcal{Q}$ is almost a map of 2-monad

$$\ell_X : \langle \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{M}X \mapsto \langle \cdot \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \in \mathcal{Q}X$$

Solution: Colax Colimit over λ in the 2-category of SymStMonCat

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

$$\alpha_{X,\langle a_1,\ldots,a_\ell \rangle} \in \mathcal{Q}X(\langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell \rangle\ ,\ \langle a_1, \ldots, a_\ell \mid \cdot \rangle)$$

$$\boxed{\dfrac{x_1 : a_1, \ldots, x_\ell : a_\ell \mid \cdot \ \vdash t : b}{\cdot \mid x_1 : \underline{a}_1, \ldots, x_\ell : \underline{a}_\ell \vdash t : b}}$$

A Colimit construction

$\mathcal{Q}$ is a 2-monad on **Cat** and $\mathcal{Q}$-algebras

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd}
\mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\
\mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & \mathcal{Q}X
\end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra

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

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & QX \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- QX objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- QX is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra

$$\begin{tikzcd} Z \arrow[r, "\eta^{\mathcal{L}}"] \arrow[dr, equal] & \mathcal{L}Z \arrow[d, "w"] \\ & Z \end{tikzcd} \qquad \begin{tikzcd} \mathcal{L}^2 Z \arrow[r, "\mathcal{L}w"] \arrow[d, "\mu^{\mathcal{L}}"'] & \mathcal{L}Z \arrow[d, "w"] \\ \mathcal{L}Z \arrow[r, "w"] & Z \end{tikzcd}$$

and coherences

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$
\begin{tikzcd}
\mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\
\mathcal{M}X \arrow[r, "\ell_X"'] \arrow[r, phantom, shift left=3, "\Nearrow \alpha_X"] & \mathcal{Q}X
\end{tikzcd}
$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f : \quad \mathcal{Q}X \quad \xrightarrow{h_X} \quad \mathcal{M}X \quad \xrightarrow{k_X} \quad \mathcal{Q}X$$

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f : \quad \mathcal{Q}X \quad \xrightarrow{h_X} \quad \mathcal{M}X \quad \xrightarrow{k_X} \quad \mathcal{Q}X$$

$$\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell, \underline{b}_1, \ldots \underline{b}_n \rangle \mapsto \langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell, \underline{b}_1, \ldots, \underline{b}_n \rangle$$

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[rr, bend left, "k_X"] & & \\ \mathcal{M}X \arrow[rr, "\ell_X"'] \arrow[r, phantom, "\Nearrow \alpha_X"] & & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f : \quad \mathcal{Q}X \quad \xrightarrow{h_X} \quad \mathcal{M}X \quad \xrightarrow{k_X} \quad \mathcal{Q}X$$

$$\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle \mapsto \langle \underline{a}_1, \ldots, \underline{a}_\ell, \underline{b}_1, \ldots \underline{b}_n \rangle \mapsto \langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell, \underline{b}_1, \ldots, \underline{b}_n \rangle$$

$$f \overset{\beta}{\Rightarrow} \mathrm{id}_{\mathcal{Q}X}$$

$$\boxed{\frac{x_1 : a_1, \ldots, x_\ell : a_\ell \mid \Delta \vdash t : b}{\cdot \mid x_1 : \underline{a}_1, \ldots, x_\ell : \underline{a}_\ell, \Delta \vdash t : b}}$$

$$\beta_{\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle} : \langle \cdot \mid \underline{a}_1, \ldots, \underline{a}_\ell, \underline{b}_1, \ldots, \underline{b}_n \rangle \rightarrow \langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$$

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Uparrow\alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f: \quad \mathcal{Q}X \xrightarrow{h_X} \mathcal{M}X \xrightarrow{k_X} \mathcal{Q}X$$

- $\mathcal{Q}X$ is almost with products, i.e. a left-semi $\mathcal{M}$-algebra:

$$\mathcal{M}\mathcal{Q}X \xrightarrow{z} \mathcal{Q}X$$

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[drr, bend left, "k_X"] & & \\ \mathcal{M}X \arrow[rr, "\ell_X"'] \arrow[urr, phantom, "\Nearrow \alpha_X"] & & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f : \quad \mathcal{Q}X \xrightarrow{h_X} \mathcal{M}X \xrightarrow{k_X} \mathcal{Q}X$$

- $\mathcal{Q}X$ is almost with products, i.e. a left-semi $\mathcal{M}$-algebra:

$$\mathcal{M}\mathcal{Q}X \xrightarrow{z} \mathcal{Q}X$$

$$\begin{tikzcd} Z \arrow[r, "\eta^{\mathcal{M}}"] \arrow[dr, equal] & \mathcal{M}Z \arrow[d, "z"] \arrow[dl, phantom, "\Swarrow \epsilon"] \\ & Z \end{tikzcd} \qquad \begin{tikzcd} \mathcal{M}^2 Z \arrow[r, "\mathcal{M}z"] \arrow[d, "\mu^{\mathcal{M}}"'] & \mathcal{M}Z \arrow[d, "z"] \\ \mathcal{M}Z \arrow[r, "z"] & Z \end{tikzcd}$$

and coherences

Properties of the $\mathcal{Q}X$ from universality for 1-cell and 2-cell

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Uparrow \alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

- $\mathcal{Q}X$ is a symmetric strict monoidal category, i.e. an $\mathcal{L}$-algebra
- $\mathcal{Q}X$ is equipped with a strictly idempotent comonad

$$f : \quad \mathcal{Q}X \xrightarrow{h_X} \mathcal{M}X \xrightarrow{k_X} \mathcal{Q}X$$

- $\mathcal{Q}X$ is almost with products, i.e. a left-semi $\mathcal{M}$-algebra:

$$\mathcal{M}\mathcal{Q}X \xrightarrow{z} \mathcal{Q}X$$

- The induced left-semi $\mathcal{L}$-algebras are equal:

$$\mathcal{L}\mathcal{Q}X \xrightarrow{\lambda_{\mathcal{Q}X}} \mathcal{M}\mathcal{Q}X \xrightarrow{z} \mathcal{Q}X \qquad \mathcal{L}\mathcal{Q}X \xrightarrow{w} \mathcal{Q}X \xrightarrow{f} \mathcal{Q}X$$

Structure category of $\mathcal{Q}$

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[rr, bend left, "k_X"] & & \\ \mathcal{M}X \arrow[rr, "\ell_X"'] \arrow[r, phantom, "\Uparrow \alpha_X"] & & \mathcal{Q}X \end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

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

$$f : \quad Z \xrightarrow{\eta_z} \mathcal{M}Z \xrightarrow{z} Z$$

- The induced left-semi $\mathcal{L}$-algebras are equal:

$$\mathcal{L}Z \xrightarrow{\lambda_Z} \mathcal{M}Z \xrightarrow{z} Z \qquad \qquad \mathcal{L}Z \xrightarrow{w} Z \xrightarrow{f} Z$$

Structure category of $\mathcal{Q}$

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

$$\begin{tikzcd}
\mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[rd, bend left, "k_X"] & \\
\mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ru, phantom, "\Uparrow \alpha_X"] & \mathcal{Q}X
\end{tikzcd}$$

- $\mathcal{L}X$ the free symmetric st. monoidal category
- $\mathcal{M}X$ the free category with products
- $\mathcal{Q}X$ objects are $\langle a_1, \ldots, a_\ell \mid \underline{b}_1, \ldots, \underline{b}_n \rangle$

Structure Category $\mathcal{Q}X$ has this structure !

- 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

$$f : \quad Z \xrightarrow{\eta_z} \mathcal{M}Z \xrightarrow{z} Z$$

- The induced left-semi $\mathcal{L}$-algebras are equal:

$$\mathcal{L}Z \xrightarrow{\lambda_Z} \mathcal{M}Z \xrightarrow{z} Z \qquad \mathcal{L}Z \xrightarrow{w} Z \xrightarrow{f} Z$$

A general colax colimit construction on 2-monads

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

$$\begin{tikzcd} \mathcal{L}X \arrow[d, "\lambda_X"'] \arrow[dr, bend left, "k_X"] & \\ \mathcal{M}X \arrow[r, "\ell_X"'] \arrow[ur, phantom, "\Nearrow \alpha_X"] & \mathcal{Q}X \end{tikzcd}$$

Theorem

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

The proof uses universality of the colimit.

It is not the end of the story

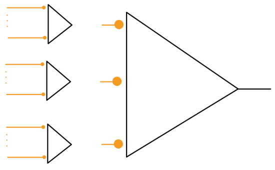
Does $\mathcal{Q}$ lifts from Cat to Prof ?

Multicategories as Profunctors with Context Monad

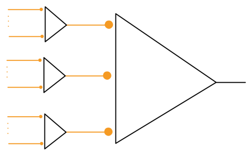
A **multicategory** can be seen as a **profunctor** in the Kleisli bicat of $\mathcal{T}$:

$$M : X \nrightarrow \mathcal{T}X \qquad M : \mathcal{T}X^{\mathrm{op}} \times X \to \mathbf{Set}$$

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



$\Rightarrow$



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

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

(Tanaka 2005, Fiore-Gambino-Hyland-Winskel 2016)

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

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

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

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

Conjecture: Pseudo $\mathcal{Q}$-algebras are pseudo Structure Categories.

Does Q extends from Cat to Prof ?

Wanted: The presheaf pseudomonad **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 $\mathcal{Q}$ extends from Cat to Prof ?

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

Conjecture: Pseudo $\mathcal{Q}$-algebras are pseudo Structure Categories.

Problem: Pseudo $\mathcal{Q}$-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 bulid a $\mathcal{Q}$-multicategory from a LNL adjunction ?

Linear-non-linear adjunction (Benton 1994)

A monoidal adjunction $\begin{tikzcd} X \arrow[r, bend left, "s"] \arrow[r, phantom, "\top"] & Y \arrow[l, bend left, "r"] \end{tikzcd}$

with X symmetric strict monoidal and Y with products

Can we bulid a $\mathcal{Q}$-multicategory from a LNL adjunction ?

Linear-non-linear adjunction (Benton 1994)

A monoidal adjunction
$$\begin{tikzcd} X \arrow[r, bend left, "s"] \arrow[r, phantom, "\top"] & Y \arrow[l, bend left, "r"] \end{tikzcd}$$
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 bulid a $\mathcal{Q}$-multicategory from a LNL adjunction ?

Linear-non-linear adjunction (Benton 1994)

A monoidal adjunction $\begin{tikzcd} X \arrow[r, bend left, "s"] \arrow[r, phantom, "\top"] & Y \arrow[l, bend left, "r"] \end{tikzcd}$

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

Work in progress: recover a structure category by going through multicategories.

It is not the end of the story

Differential λ-calculus axiomatization

Towards a multicategorical model of differential λ-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 λ-calculus

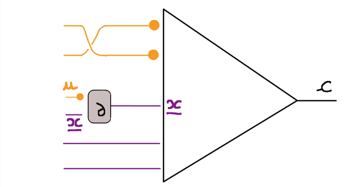
Substitution: linear-non-linear multicategories

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 λ-calculus

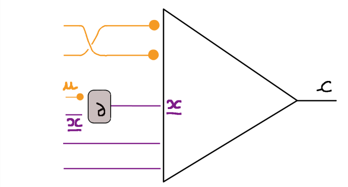
Substitution: linear-non-linear multicategories

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.

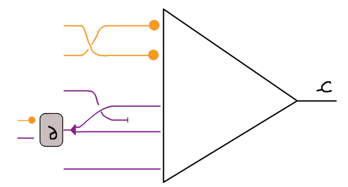
Derivation $u, x \mapsto D_x f(u)$

- 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)