

A Fibrational Perspective on Differential Linear Logic

Jad Koleilat

koleilat@lipn.univ-paris13.fr

Laboratoire d'Informatique de Paris Nord (LIPN)

23/10/2025

Contents

1. Linear Logic and Differential Linear Logic	2
2. Fibrations and Dependency	22
3. A Simple Fibration for Linear Logic	31
4. Tangent Functor and DiLL	47
5. Dependent Linear Logic and DiLL	62
Bibliography	66

1. Linear Logic and Differential Linear Logic

1.1 Linear Logic

What is Linear Logic [Gir87] ?

- Substructural logic : we are not allowed to duplicate or erase formulas recklessly

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \text{c}$$



$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \text{w}$$

We care about how many hypothesis (resources) are need. In LL being able to derive $A, A \vdash B$ does not imply that the sequent $A \vdash B$ is derivable.

1.1 Linear Logic

1. Linear Logic and Differential Linear Logic

- Modal Logic : For every formula F we can construct the formula $!F$.
This changes the meaning of F .

A formula can only be duplicated or erased if it is of the shape $!F$:

$$\frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{c}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{w}$$

In LL there two kinds of formulas :

- the ones that cannot be duplicated or erased : A .
- the ones that can be : $!A$

There are two additional rules defining the behavior of “!” :

- Dereliction

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{d}$$

- Promotion

$$\frac{!\Gamma \vdash A}{!\Gamma \vdash !A} \mathbf{P}$$

1.2 Differential Linear Logic

1. Linear Logic and Differential Linear Logic

Differential Linear Logic [Ehr18, ER06] by adding three rules to Linear Logic :

$$\frac{\vdash \Gamma, !A \quad \vdash \Delta, !A}{\vdash \Gamma, \Delta, !A} \bar{c}$$

$$\frac{\vdash}{\vdash !A} \bar{w}$$

$$\frac{\vdash \Gamma, !A}{\vdash \Gamma, A} \bar{d}$$

1.2 Differential Linear Logic

1. Linear Logic and Differential Linear Logic

Differential Linear Logic [Ehr18, ER06] by adding three rules to Linear Logic :

$$\frac{\vdash \Gamma, !A \quad \vdash \Delta, !A}{\vdash \Gamma, \Delta, !A} \bar{\mathbf{c}}$$

$$\frac{\vdash}{\vdash !A} \bar{\mathbf{w}}$$

$$\frac{\vdash \Gamma, !A}{\vdash \Gamma, A} \bar{\mathbf{d}}$$

They are the dual rules to

$$\frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{c}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{w}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} \mathbf{d}$$

1.2 Differential Linear Logic

1. Linear Logic and Differential Linear Logic

Why “Differential” ?

A derivation of a sequent $A \vdash B$ can be thought as a “linear” function from $A \rightarrow B$. It is “linear” because A is used only once in the computation.

A derivation of a sequent $!A \vdash B$ can be thought as a non-linear function since the resource A can be used arbitrarily many times.

1.2 Differential Linear Logic

1. Linear Logic and Differential Linear Logic

The codereliction rule

$$\frac{!A \vdash B}{A \vdash B} \bar{d}$$

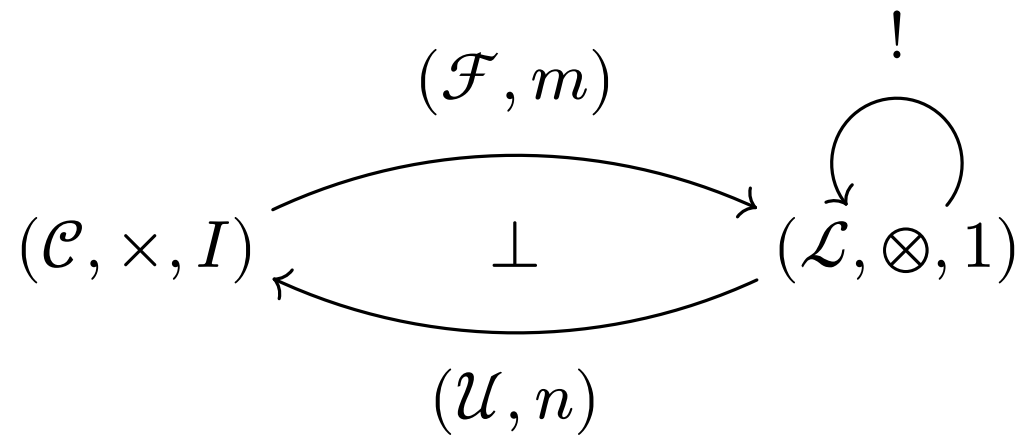
Transforms a non-linear function $f : !A \rightarrow B$ to a linear function $D_0(f) : A \rightarrow B$.

Concretely, in some models, this notion of “syntactic” differentiation corresponds the usual mathematical notion.

Definition (Linear-non-linear Adjunction)

A linear-non-linear (LNL) adjunction is a lax monoidal adjunction between a monoidal category and a cartesian category.

$$\begin{array}{ccc} & (\mathcal{F}, m) & \\ & \curvearrowright & \\ (\mathcal{C}, \times, I) & \perp & (\mathcal{L}, \otimes, 1) \\ & \curvearrowleft & \\ & (\mathcal{U}, n) & \end{array}$$



A formula of Linear Logic $A \otimes !B$ is interpreted in the category \mathcal{L} as

$$\llbracket A \rrbracket \otimes \mathcal{F}\mathcal{U}(\llbracket B \rrbracket) = \llbracket A \rrbracket \otimes !\llbracket B \rrbracket$$

The interpretation of the different rules are given by composition or pre-composition with natural transformations built from the LNL adjunction :

- weakening : $\mathbf{w}_A : !A \rightarrow 1$ (constructed using \mathcal{F})
- contraction : $\mathbf{c}_A : !A \rightarrow !A \otimes !A$ (constructed using \mathcal{F})
- dereliction : $\mathbf{d}_A : !A \rightarrow A$ (counit of the comonad !)
- promotion : $\mathbf{p}_A : !!A \rightarrow !A$ (comultiplication of !)

Instead of asking for an adjunction, we can instead ask for a comonad and recover an adjunction from it.

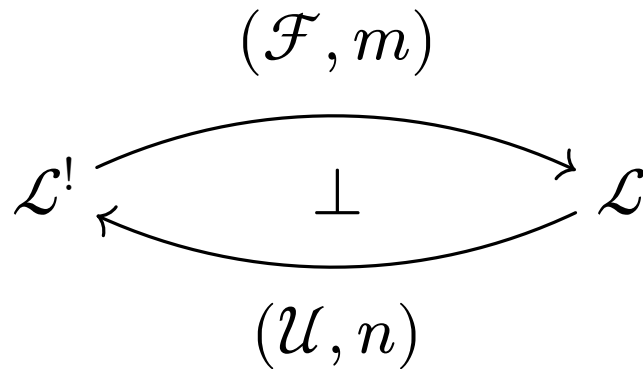
Definition (Linear Category [Mel09])

A Linear Category is a symmetric monoidal category $(\mathcal{L}, \otimes, 1)$ equipped with a symmetric monoidal comonad modality $(!, \mathbf{d}, \mathbf{p}, \mathbf{c}, \mathbf{w})$.

1.3 Semantic of LL

For a symmetric monoidal category $(\mathcal{L}, \otimes, 1)$ equipped with a comonad $(!, \mathbf{d}, \mathbf{p})$, TFAE :

- \mathcal{L} is Linear Category
- \mathcal{L} is involved a linear-non-linear adjunction with its Eilenberg-Moore category $\mathcal{L}!$



Linear Categories are a particular case of linear-non-linear adjunctions

Definition ((Monoidal) Differential Category [Blu+20])

A (monoidal) Differential Category is the data of :

- $((\mathcal{L}, \otimes, 1), !)$ an additive Linear Category
- A natural transformation $\partial_A : !A \otimes A \rightarrow !A$ called the deriving transform that satisfies the following equations :
 - $\partial_A; \mathbf{d}_A = \mathbf{w}_A \otimes \text{id}_A$
 - $\partial_A; \mathbf{w}_A = 0$
 - $\partial_A; \mathbf{c}_A = \mathbf{c}_A \otimes \text{id}_A; (\text{id}_{!A} \otimes \partial_A) + (\text{id}_{!A} \otimes \sigma_{!A,A}; \partial_A \otimes \text{id}_A)$
 - $\partial_A; \mathbf{p}_A = \mathbf{c}_A \otimes \text{id}_A; \mathbf{p}_A \otimes \partial_A; \partial_{!A}$
 - $\text{id}_A \otimes \sigma_{A,A}; \partial_A \otimes \text{id}_A; \partial_A = \partial_A \otimes \text{id}_A; \partial_A$

A closer look at the deriving transform $\partial_A : !A \otimes A \rightarrow !A$

It expects as argument a non-linear function $f : !A \rightarrow B$ and outputs

$$!A \otimes A \xrightarrow{\partial_A} !A \xrightarrow{f} B$$

its differential $D(f) : !A \otimes A \rightarrow B$.

A closer look at the deriving transform $\partial_A : !A \otimes A \rightarrow !A$

It expects as argument a non-linear function $f : !A \rightarrow B$ and outputs

$$!A \otimes A \xrightarrow{\partial_A} !A \xrightarrow{f} B$$

its differential $D(f) : !A \otimes A \rightarrow B$.

From a Differential Geometry point of view, $!A \otimes A$ is the total space of the tangent bundle of A .

- The linear rule : $\partial_A; \mathbf{d}_A = \mathbf{w}_A \otimes \text{id}_A$

A morphism f of type $A \rightarrow B$ is a linear morphism, in other to differentiate it we need to see it as a non-linear morphism.

Pre-composing with $\mathbf{d}_A : !A \rightarrow A$ does exactly this.

$$!A \xrightarrow{\mathbf{d}_A} A \xrightarrow{f} B$$

- The linear rule : $\partial_A; \mathbf{d}_A = \mathbf{w}_A \otimes \text{id}_A$

A morphism f of type $A \rightarrow B$ is a linear morphism, in other to differentiate it we need to see it as a non-linear morphism.

Pre-composing with $\mathbf{d}_A : !A \rightarrow A$ does exactly this.

$$\begin{array}{ccccccc}
 !A \otimes A & \xrightarrow{\partial_A} & !A & \xrightarrow{\mathbf{d}_A} & A & \xrightarrow{f} & B \\
 & \searrow & & & \parallel & & \\
 & & & & 1 \otimes A & & \\
 & \swarrow & & & & & \\
 & & \mathbf{w}_A \otimes \text{id}_A & & & &
 \end{array}$$

- The constant rule : $\partial_A; \mathbf{w}_A = 0$. The differential of a constant function is 0

- The constant rule : $\partial_A; \mathbf{w}_A = 0$. The differential of a constant function is 0
- The Leibniz rule :

$$\partial_A; \mathbf{c}_A = \mathbf{c}_A \otimes \text{id}_A; (\text{id}_{!A} \otimes \partial_A) + (\text{id}_{!A} \otimes \sigma_{!A,A}; \partial_A \otimes \text{id}_A)$$

For a pair of non-linear function $f : !A \rightarrow B$ and $g : !A \rightarrow C$

$$!A \xrightarrow{\mathbf{c}_A} !A \otimes !A \xrightarrow{f \otimes g} B \otimes C$$

is the product of f and g . The Leibniz rule describes the differential of $x \mapsto f(x) \times g(x)$.

- The chain rule : $\partial_A; \mathbf{p}_A = \mathbf{c}_A \otimes \text{id}_A; \mathbf{p}_A \otimes \partial_A; \partial_{!A}$

\mathbf{p}_A allows us to compose non-linear morphisms $f : !A \rightarrow B$ and $g : !B \rightarrow C$

$$!A \xrightarrow{\mathbf{p}_A} !!A \xrightarrow{!f} !B \xrightarrow{g} C$$

This rule describes how to compose differential :

$$D_x(f \circ g)(v) = D_x(g)\left(D_{g(x)}(f)(v)\right)$$

- The chain rule : $\partial_A; \mathbf{p}_A = \mathbf{c}_A \otimes \text{id}_A; \mathbf{p}_A \otimes \partial_A; \partial_{!A}$

\mathbf{p}_A allows us to compose non-linear morphisms $f : !A \rightarrow B$ and $g : !B \rightarrow C$

$$!A \xrightarrow{\mathbf{p}_A} !!A \xrightarrow{!f} !B \xrightarrow{g} C$$

This rule describes how to compose differential :

$$D_x(f \circ g)(v) = D_x(g)\left(D_{g(x)}(f)(v)\right)$$

- The symmetric rules expresses symmetries in second derivatives.

- Semantic defined for Linear Categories.
 - What is a differential linear-non-linear adjunction ?

- Semantic defined for Linear Categories.
 - What is a differential linear-non-linear adjunction ?
- DiLL is related to Differential Geometry, yet Differential Geometry is a dependent theory, DiLL is a propositional logic.
 - Does a dependent version of DiLL exists ?
 - Can we reformulate the semantic of DiLL to match the semantic of dependent type theory ?
 - Does it tell us anything about dependent DiLL ?

What is a differential linear-non-linear adjunction ?

An answer is given by Kerjean, Rogers and Maestracci [KMR25].

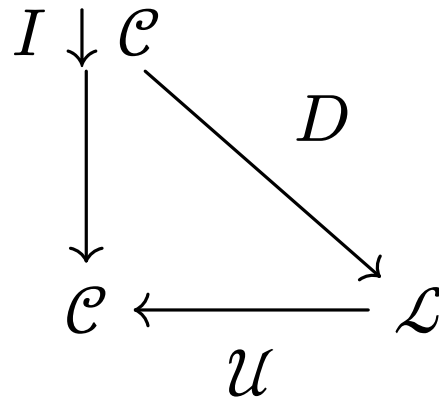
- Requires \mathcal{L} to have products.
- Does not look like models of dependent type theories.

What is a differential linear-non-linear adjunction ?

An answer is given by Kerjean, Rogers and Maestracci [KMR25].

- Requires \mathcal{L} to have products.
- Does not look like models of dependent type theories.

Given an LNL adjunction, they consider a functor from the points of \mathcal{C} to \mathcal{L} .



- We show that the simple category of simply typed theories has a linear equivalent.
- We reformulate models of DiLL in terms of a section (tangent functor) of this fibration.
 - ▶ Generalized to any LNL adjunction
 - ▶ Gives good clues on what the semantics of dependent DiLL should look like

2. Fibrations and Dependency

2.1 Grothendieck Fibrations

Categorically, dependence is modeled using Fibrations.

Definition (Grothendieck Fibration)

A fibration is the data of two categories \mathbb{B} called the base category, \mathbb{E} called the total category and a functor called the fibration $p : \mathbb{E} \rightarrow \mathbb{B}$. Moreover, we ask that each morphisms $f : X \rightarrow Y$ in \mathbb{B} “lifts nicely” to \mathbb{E} .



Consider a dependent type theory, define the category \mathbb{B} as :

- Objects : contexts Γ (example, $n : \mathbb{N}, l : \text{list}(n), b : \text{bool}$)

Consider a dependent type theory, define the category \mathbb{B} as :

- Objects : contexts Γ (example, $n : \mathbb{N}, l : \text{list}(n), b : \text{bool}$)
- Morphisms : let $\Gamma := x_1 : \sigma_1, \dots, x_n : \sigma_n$ and $\Delta := y_1 : \nu_1, \dots, y_m : \nu_m$.
A morphism $M : \Gamma \rightarrow \Delta$ is a m -tuple of terms (M_1, \dots, M_m) such that :

$$\Gamma \vdash M_i : \nu_i[M_1/y_1, \dots, M_{i-1}/y_{i-1}]$$

For \mathbb{E} we take the full subcategory of \mathbb{B}^{\rightarrow} on objects of the shape:

$$\Gamma, x : \sigma \xrightarrow{\pi} \Gamma$$

With π the trivial projection. A morphism in \mathbb{E} is a pair of morphisms in \mathbb{B} such that the following commutes

$$\begin{array}{ccc} \Gamma, x : \sigma & \xrightarrow{M} & \Delta, y : \nu, z : \mu \\ \pi \downarrow & & \downarrow \pi \\ \Gamma & \xrightarrow{N} & \Delta \end{array}$$

2.2 An example from Type Theory

This defines a fibration :

$$\begin{array}{c} \mathbb{E} \\ \downarrow \\ \mathbb{B} \end{array}$$

cod

For each context Γ , $\mathbf{cod}^{-1}(\Gamma)$ is the category of contexts extending Γ . It is the category of objects of \mathbb{B} depending on Γ .

2.3 Fibrations for simple types

We can use this formalism to describe simply typed theories.

2.3 Fibrations for simple types

We can use this formalism to describe simply typed theories.

Definition (Simple Category [Jac99])

Given any cartesian category (\mathcal{C}, \times, I) , define $S(\mathcal{C})$ the simple category of \mathcal{C} as :

- Objects : pairs (X, Y) of objects of \mathcal{C} .
- Morphisms : pairs $(f, u) : (X_1, Y_1) \rightarrow (X_2, Y_2)$ with $f : X_1 \rightarrow Y_1$ and $u : X_1 \times Y_1 \rightarrow Y_2$

2.3 Fibrations for simple types

We can use this formalism to describe simply typed theories.

Definition (Simple Category [Jac99])

Given any cartesian category (\mathcal{C}, \times, I) , define $S(\mathcal{C})$ the simple category of \mathcal{C} as :

- Objects : pairs (X, Y) of objects of \mathcal{C} .
- Morphisms : pairs $(f, u) : (X_1, Y_1) \rightarrow (X_2, Y_2)$ with $f : X_1 \rightarrow Y_1$ and $u : X_1 \times Y_1 \rightarrow Y_2$

Intuitively, $X_1 \vdash f : X_2$ and $X_1, Y_1 \vdash u : Y_2$.

2.3 Fibrations for simple types

Composition of $(f, u); (X_1, Y_1) \rightarrow (X_2, Y_2)$ and $(g, v) : (X_2, Y_2) \rightarrow (X_3, Y_3)$ is given by :

$$X_1 \times Y_1 \xrightarrow{\Delta_X \times \text{id}_{Y_1}} X_1 \times X_1 \times Y_1 \xrightarrow{f \times u} X_2 \times Y_2 \xrightarrow{v} Y_3$$

2.3 Fibrations for simple types

Composition of $(f, u); (X_1, Y_1) \rightarrow (X_2, Y_2)$ and $(g, v) : (X_2, Y_2) \rightarrow (X_3, Y_3)$ is given by :

$$X_1 \times Y_1 \xrightarrow{\Delta_X \times \text{id}_{Y_1}} X_1 \times X_1 \times Y_1 \xrightarrow{f \times u} X_2 \times Y_2 \xrightarrow{v} Y_3$$

This is possible because X_1 is a duplicable resource. X_1 is a comonoid.

2.3 Fibrations for simple types

The fibration is simply the projection s sending $(f, u) : (X_1, Y_1) \rightarrow (X_2, Y_2)$ to $f : X_1 \rightarrow X_2$.

$$\begin{array}{c} S(\mathcal{C}) \\ \downarrow s \\ \mathcal{C} \end{array}$$

2.3 Fibrations for simple types

Another construction of this fibration is given by considering the following functors :

- $H : \mathcal{C} \rightarrow \text{CoMonad}(\mathcal{C})$, sending X to the comonad $X \times -$.
- $K : \text{CoMonad}(\mathcal{C})^{\text{op}} \rightarrow \text{Cat}$, sending a comonad T to its Kleisli \mathcal{C}_T

2.3 Fibrations for simple types

Another construction of this fibration is given by considering the following functors :

- $H : \mathcal{C} \rightarrow \text{CoMonad}(\mathcal{C})$, sending X to the comonad $X \times -$.
- $K : \text{CoMonad}(\mathcal{C})^{\text{op}} \rightarrow \text{Cat}$, sending a comonad T to its Kleisli \mathcal{C}_T

Then $KH^{\text{op}} : \mathcal{C}^{\text{op}} \rightarrow \text{Cat}$ is an indexed category.

$$S(\mathcal{C}) = \int KH^{\text{op}}$$

3. A Simple Fibration for Linear Logic

3.1 Linear simple category

We want to construct something that is to propositional LL what $S(\mathcal{C})$ is to simply typed theories.

3.1 Linear simple category

We want to construct something that is to propositional LL what $S(\mathcal{C})$ is to simply typed theories.

Fix a linear-non-linear adjunction.

$$\begin{array}{ccc} & (\mathcal{F}, m) & \\ & \curvearrowright & \\ (\mathcal{C}, \times, I) & \perp & (\mathcal{L}, \otimes, 1) \\ & \curvearrowleft & \\ & (\mathcal{U}, n) & \end{array}$$

Notice that objects in \mathcal{C} are comonoids.

Proposition

In a linear-non-linear adjunction, \mathcal{F} can be equipped with an oplax structure making it a strong monoidal functor.

Corollary

\mathcal{F} sends comonoids in \mathcal{C} to comonoids in \mathcal{L} .

3.1 Linear simple category

Looking back at the simple category $S(\mathcal{C})$, we can adapt the construction by considering pairs (X, A) where X is in \mathcal{C} (a duplicable resource) and A in \mathcal{L} .

Looking back at the simple category $S(\mathcal{C})$, we can adapt the construction by considering pairs (X, A) where X is in \mathcal{C} (a duplicable resource) and A in \mathcal{L} .

Definition 3.1.2 (Linear Simple Category)

Given a linear-non-linear adjunction as above, define $LS(\mathcal{C})$ the linear simple category associated to the adjunction as :

- Objects : pairs (X, A) with $X \in \mathcal{C}$ and $A \in \mathcal{L}$
- Morphisms : pairs $(f, u) : (X, A) \rightarrow (Y, B)$ with $f : X \rightarrow Y$ in \mathcal{C} and $u : \mathcal{F}(X) \otimes A \rightarrow B$ in \mathcal{L} .

3.1 Linear simple category

The same functorial construction can be used to define it :

- $H : \mathcal{C} \rightarrow \text{CoMonad}(\mathcal{L})$, sending X to the comonad $\mathcal{F}(X) \otimes -$.
- $K : \text{CoMonad}(\mathcal{L})^{\text{op}} \rightarrow \text{Cat}$, sending a comonad T to its Kleisli \mathcal{L}_T

Then $KH^{\text{op}} : \mathcal{C}^{\text{op}} \rightarrow \text{Cat}$ is an indexed category.

$$LS(\mathcal{C}) = \int KH^{\text{op}}$$

The same functorial construction can be used to define it :

- $H : \mathcal{C} \rightarrow \text{CoMonad}(\mathcal{L})$, sending X to the comonad $\mathcal{F}(X) \otimes -$.
- $K : \text{CoMonad}(\mathcal{L})^{\text{op}} \rightarrow \text{Cat}$, sending a comonad T to its Kleisli \mathcal{L}_T

Then $KH^{\text{op}} : \mathcal{C}^{\text{op}} \rightarrow \text{Cat}$ is an indexed category.

$$LS(\mathcal{C}) = \int KH^{\text{op}}$$

If we consider the trivial LNL adjunction of (\mathcal{C}, \times, I) with itself we get $LS(\mathcal{C}) = S(\mathcal{C})$.

3.1 Linear simple category

Composition is given in the same way :

$$\mathcal{F}(X) \otimes A \xrightarrow{\mathbf{c}_X \otimes \text{id}_A} \mathcal{F}(X) \otimes \mathcal{F}(X) \otimes A \xrightarrow{f \otimes u} \mathcal{F}(Y) \otimes B \xrightarrow{v} C$$

3.1 Linear simple category

Composition is given in the same way :

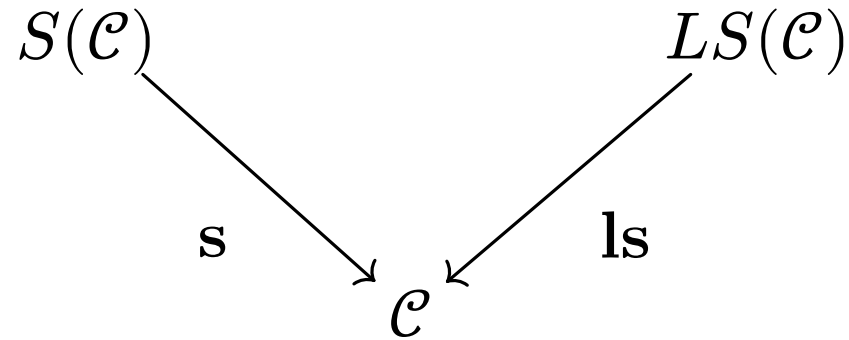
$$\mathcal{F}(X) \otimes A \xrightarrow{\mathbf{c}_X \otimes \text{id}_A} \mathcal{F}(X) \otimes \mathcal{F}(X) \otimes A \xrightarrow{f \otimes u} \mathcal{F}(Y) \otimes B \xrightarrow{v} C$$

The projection ls sending $(f, u) : (X, A) \rightarrow (Y, B)$ to $f : X \rightarrow Y$ is a fibration :

$$\begin{array}{c} LS(\mathcal{C}) \\ \downarrow \text{ls} \\ \mathcal{C} \end{array}$$

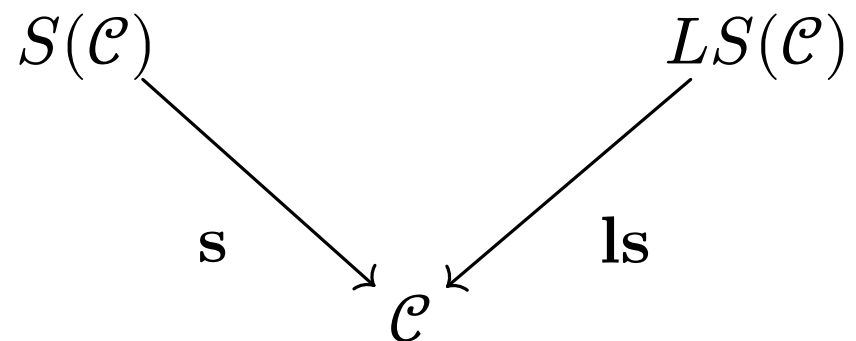
3.2 Fibred LNL adjunction

\mathcal{C} is also the base category of its simple category.



3.2 Fibred LNL adjunction

\mathcal{C} is also the base category of its simple category.



The structure of linear-non-linear adjunction lifts to a fibred linear-non-linear adjunction between s and ls .

Proposition

$S(\mathcal{C})$ is a fibred cartesian category and $LS(\mathcal{C})$ is a fibred symmetric monoidal category.

Proposition

$S(\mathcal{C})$ is a fibred cartesian category and $LS(\mathcal{C})$ is a fibred symmetric monoidal category.

We can construct such a structure “pointwise” by defining $(X, Y) \times_X (X, Z)$ as $(X, Y \times Z)$ and $(X, A) \otimes_X (X, B)$ as $(X, A \otimes B)$.

Proposition

$S(\mathcal{C})$ is a fibred cartesian category and $LS(\mathcal{C})$ is a fibred symmetric monoidal category.

We can construct such a structure “pointwise” by defining $(X, Y) \times_X (X, Z)$ as $(X, Y \times Z)$ and $(X, A) \otimes_X (X, B)$ as $(X, A \otimes B)$.

An other way to see it is that $KH^{\text{op}} : \mathcal{C} \rightarrow \text{Cat}$ restricts to a functor $\mathcal{C} \rightarrow \text{SymMonCat}$.

The functor $\mathcal{U} : \mathcal{L} \rightarrow \mathcal{C}$ can be lifted “pointwise” as the fibred functor $\mathcal{U}^s : \mathbf{ls} \rightarrow \mathbf{s}$.

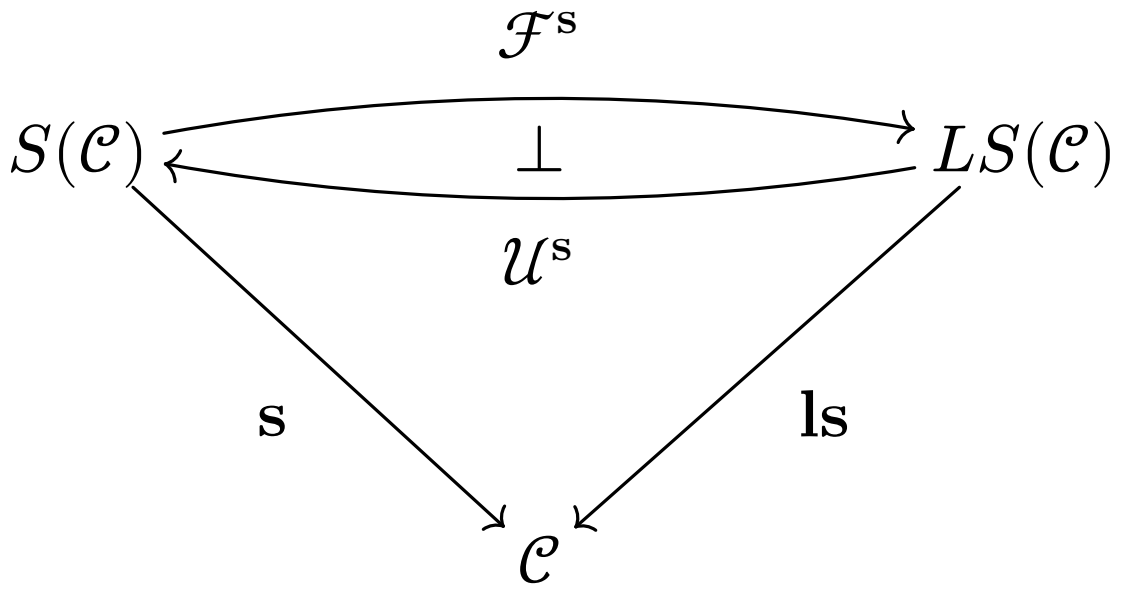
- Objects : $\mathcal{U}^s(X, A) = (X, \mathcal{U}(A)) \in S(\mathcal{C})$
- Morphisms : it sends $u : \mathcal{F}(X) \otimes A \rightarrow B$ to

$$X \times \mathcal{U}(A) \xrightarrow{\eta_X \times \text{id}} \mathcal{U}\mathcal{F}(X) \times \mathcal{U}(A) \xrightarrow{\eta} \mathcal{U}(\mathcal{F}(X) \otimes A) \xrightarrow{u} \mathcal{U}(B)$$

With η the unit of the adjunction $\mathcal{F} \dashv \mathcal{U}$

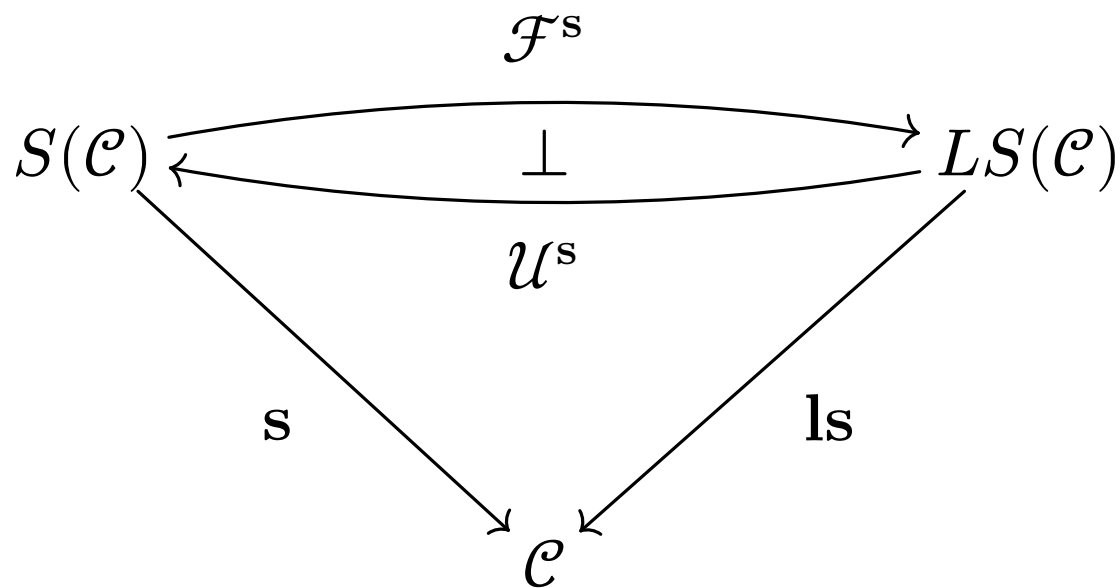
3.2 Fibred LNL adjunction

Similarly, $\mathcal{F} : \mathcal{C} \rightarrow \mathcal{L}$ can be lifted as functor $\mathcal{F}^s : S(\mathcal{C}) \rightarrow LS(\mathcal{C})$.



3.2 Fibred LNL adjunction

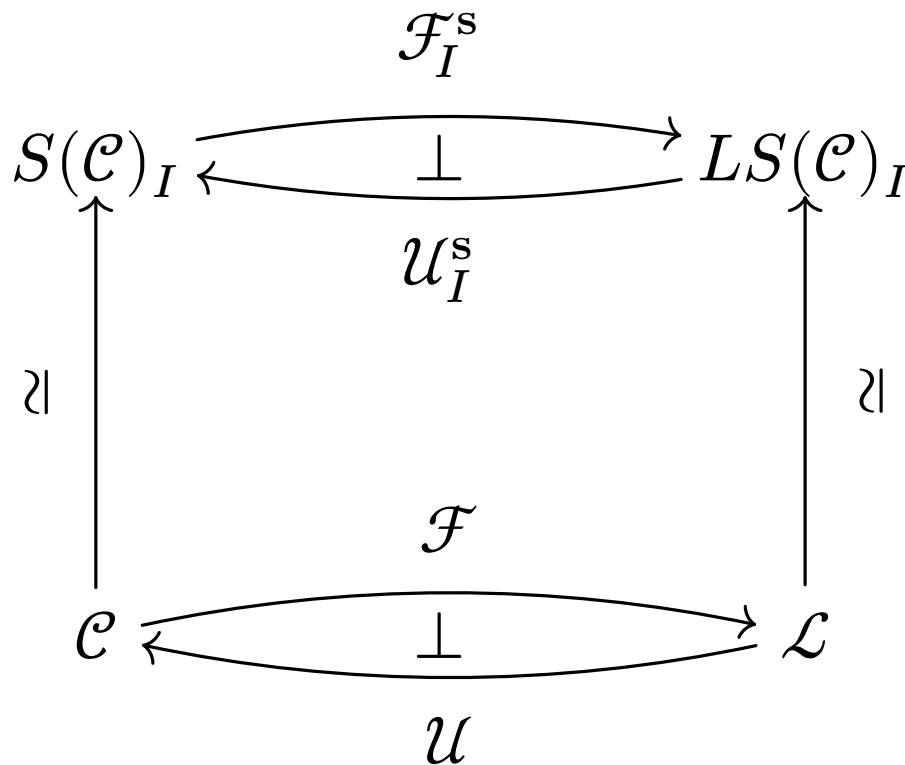
Similarly, $\mathcal{F} : \mathcal{C} \rightarrow \mathcal{L}$ can be lifted as functor $\mathcal{F}^s : S(\mathcal{C}) \rightarrow LS(\mathcal{C})$.



The restriction on each fiber $\mathcal{F}_X^s \dashv \mathcal{U}_X^s$ is a linear-non-linear adjunction.

3.2 Fibred LNL adjunction

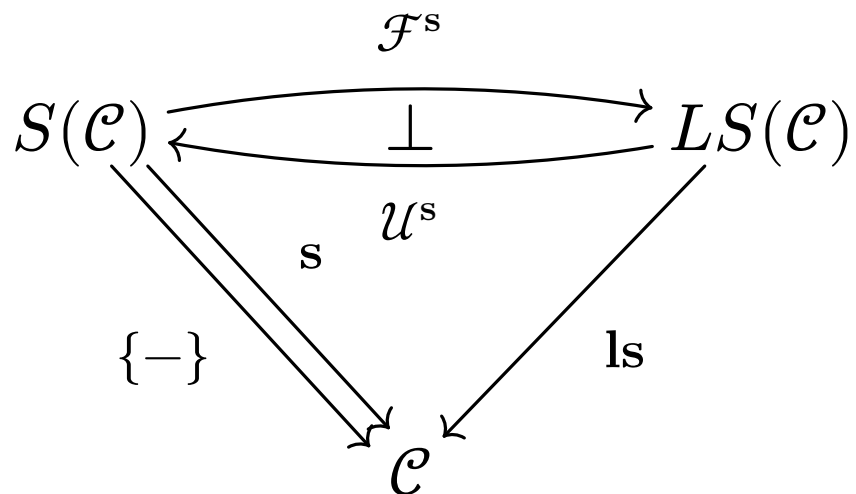
We can recover the starting adjunction by taking the fiber above the terminal object of \mathcal{C} .



3.3 Comprehension

$S(\mathcal{C})$ and $LS(\mathcal{C})$ enjoy additional structure :

- There is a functor $\{-\} : S(\mathcal{C}) \rightarrow \mathcal{C}$ sending an object (X, J) to $X \times J$



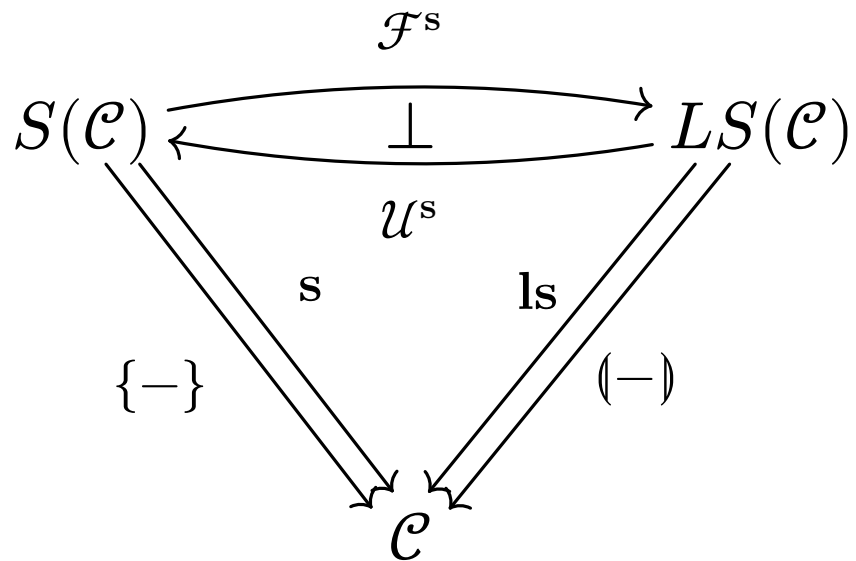
This comprehension structure is what characterizes fibrational models of dependent type theories.

- Logically it's the “comprehension operation”, sending a predicate $P(x)$ on X to the subset $\{P\} \subseteq X$ of elements of X satisfying P .
- Type theoretically it sends a derivation $\Gamma \vdash \sigma : \mathbf{Type}$ to the dependent context $\Gamma, x : \sigma$.
- Geometrically it's the total space functor, sending a fiber bundle above X to its total space.

3.3 Comprehension

By pre-composing $\{-\}$ with \mathcal{U}^S we get a similar functor

$(-)_\perp := \{-\} \circ \mathcal{U}^S : LS(\mathcal{C}) \rightarrow \mathcal{C}$. It sends objects (X, A) to $X \times \mathcal{U}(A)$.



It is the linear equivalent of $\{-\}$.

3.4 A concrete example

$LS(\mathcal{C})$ can be thought as a category of “vector spaces” parametrised by “non-linear spaces”. As an example :

$$\begin{array}{ccc}
 & \text{Free} & \\
 (\text{Set}, \times, 1) & \xrightarrow{\quad} & (\text{Vect}, \otimes, \mathbb{R}) \\
 & \perp & \\
 & \text{Forget} &
 \end{array}$$

3.4 A concrete example

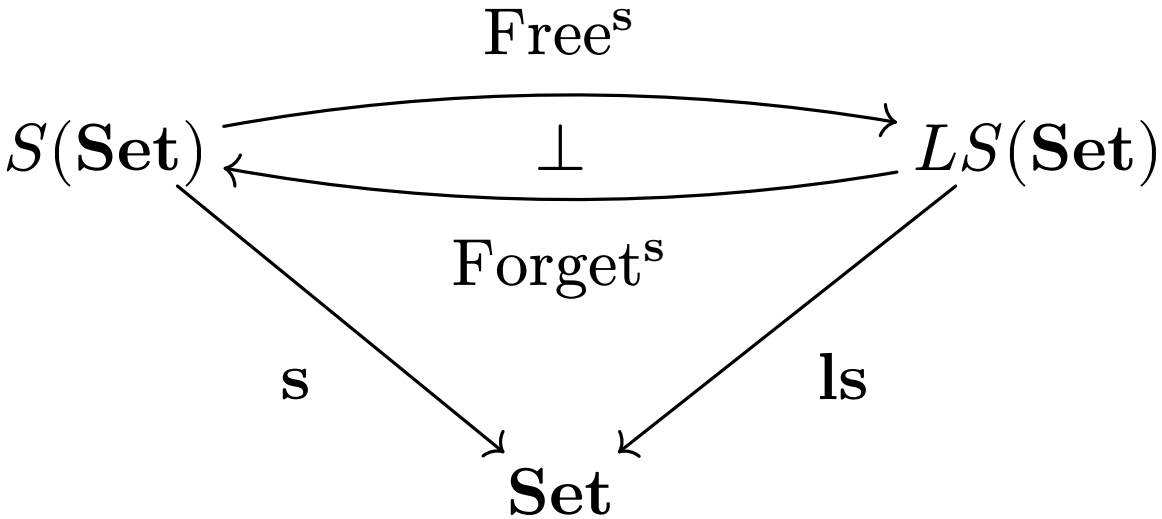
$LS(\mathcal{C})$ can be thought as a category of “vector spaces” parametrised by “non-linear spaces”. As an example :

$$\begin{array}{ccc}
 & \text{Free} & \\
 (\text{Set}, \times, 1) & \begin{array}{c} \xrightarrow{\quad} \\ \perp \\ \xleftarrow{\quad} \end{array} & (\text{Vect}, \otimes, \mathbb{R}) \\
 & \text{Forget} &
 \end{array}$$

$LS(\text{Set})$:

- Objects : pairs of a set and a vector space
- Morphism : a function $f : X \rightarrow Y$ and for each point in X a linear morphisms from $A \rightarrow B$.

3.4 A concrete example

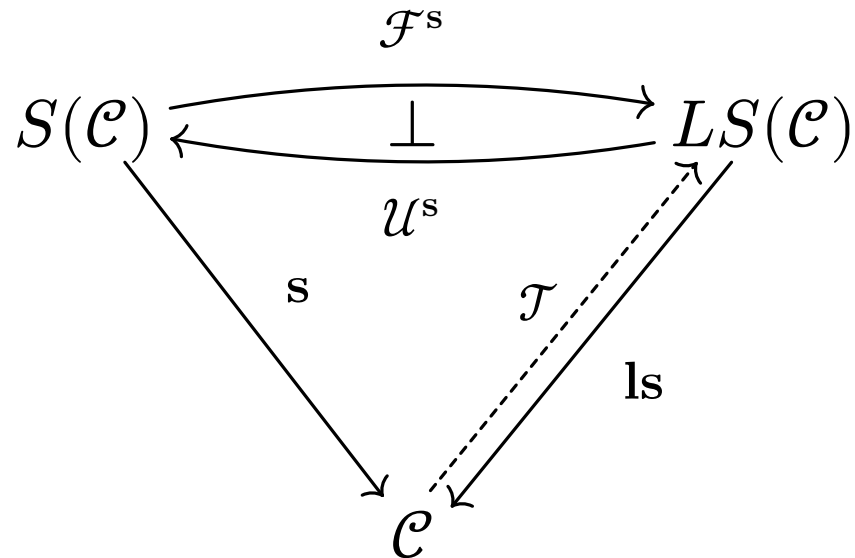


Notice that vectors spaces are exactly the image of $(-)_1 : LS(\mathbf{Set})_1 \rightarrow \mathbf{Set}$

4. Tangent Functor and DiLL

4.1 Intuition

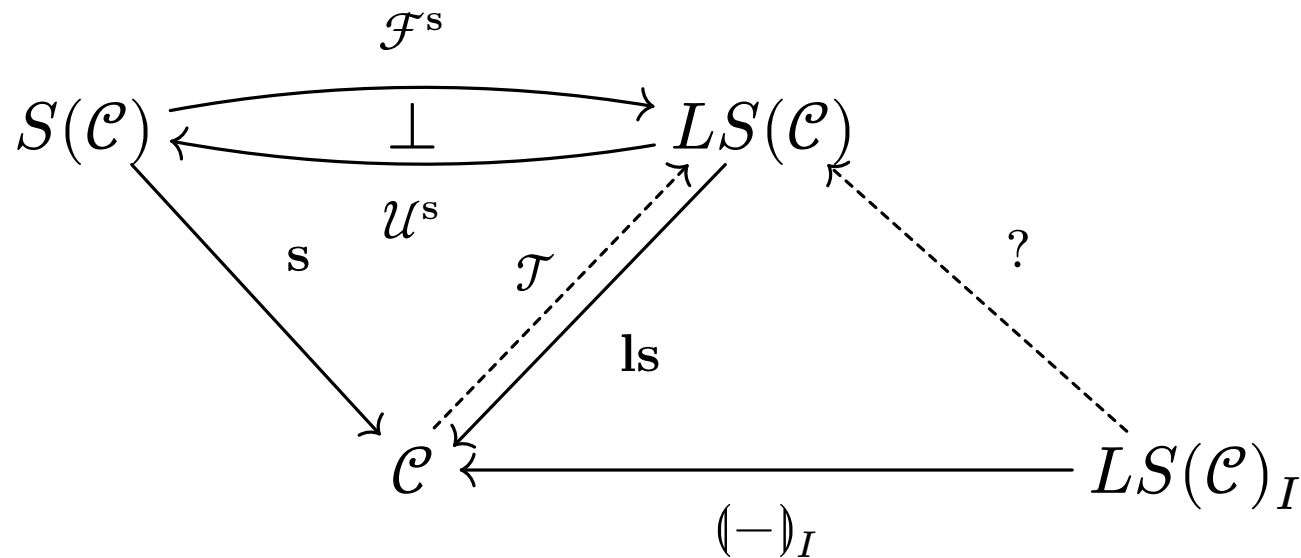
The differential of a (non-linear) function is a family of linear functions indexed by a non-linear parameter. Thus it must live in $LS(\mathcal{C})$.



We ask for the existence of a functor $\mathcal{T} : \mathcal{C} \rightarrow LS(\mathcal{C})$, called a linear tangent functor.

4.1 Intuition

In other categorical setting expressing differentiation [CC14], “vector spaces like objects” are defined based on their interaction with the tangent functor. We already have a notion of “vector spaces” in \mathcal{C} . We must describe its interaction with \mathcal{T}



4.1 Intuition

We know that the tangent space of a vector space A is $A \times A$, where the first A is thought of as a manifold (non-linear type) and the second one as a vector space. This means that the tangent space of A is $!A \otimes A$.

4.1 Intuition

We know that the tangent space of a vector space A is $A \times A$, where the first A is thought of as a manifold (non-linear type) and the second one as a vector space. This means that the tangent space of A is $!A \otimes A$.

Definition

Define the functor $\mathcal{V} : LS(\mathcal{C})_I \rightarrow LS(\mathcal{C})$ as :

- On objects : $A \in LS(\mathcal{C})$ to $\mathcal{P}A^*(A)$ (concretely $((A)_I, A)$)
- On morphisms : $u : \mathcal{F}(I) \otimes A \rightarrow B$ to $\mathcal{P}A^*(u); \bar{u}$

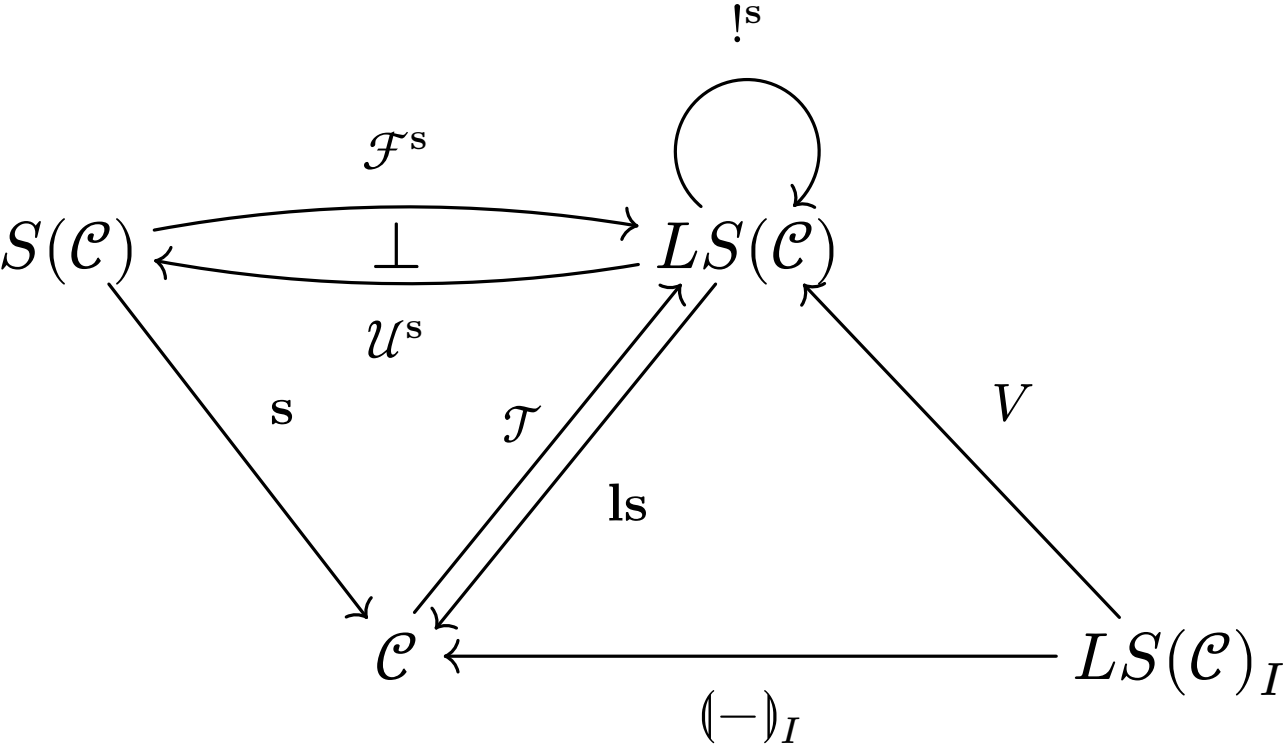
\mathcal{V} sends a non dependent linear type (objects in $LS(\mathcal{C})_I$) to its tangent space in $LS(\mathcal{C})$.

4.2 Linear Tangent Functor

Definition (First Order Differential LNL adjunction)

A first order differential linear-non-linear adjunction is the data of a additive linear-non-linear adjunction and a functor $\mathcal{T} : \mathcal{C} \rightarrow LS(\mathcal{C})$, called a linear tangent functor, such that :

1. \mathcal{T} is a section of ls .
2. $\mathcal{T}(-)_I = \mathcal{V}$



4.2 Linear Tangent Functor

Definition ((Monoidal) Differential Category [Blu+20])

A (monoidal) Differential Category is the data of :

- $((\mathcal{L}, \otimes, 1), !)$ an additive Linear Category
- A natural transformation $\partial_A : !A \otimes A \rightarrow !A$ called the deriving transform that satisfies the following equations :
 - $\partial_A ; \mathbf{d}_A = \mathbf{w}_A \otimes \text{id}_A$
 - $\partial_A ; \mathbf{w}_A = 0$
 - $\partial_A ; \mathbf{c}_A = \mathbf{c}_A \otimes \text{id}_A ; (\text{id}_{!A} \otimes \partial_A) + (\text{id}_{!A} \otimes \sigma_{!A,A} ; \partial_A \otimes \text{id}_A)$
 - $\partial_A ; \mathbf{p}_A = \mathbf{c}_A \otimes \text{id}_A ; \mathbf{p}_A \otimes \partial_A ; \partial_{!A}$
 - $\text{id}_A \otimes \sigma_{A,A} ; \partial_A \otimes \text{id}_A ; \partial_A = \partial_A \otimes \text{id}_A ; \partial_A$

4.2 Linear Tangent Functor

How do we recover the deriving transform $\partial_A : !A \otimes A \rightarrow A$?

¹idea from [KMR25]

4.2 Linear Tangent Functor

How do we recover the deriving transform $\partial_A : !A \otimes A \rightarrow A$?

By considering the tangent $\mathcal{T} \{ \eta_{\mathcal{U}^s(A)} \}^1$ for $A \in LS(\mathcal{C})_I$. We can build :

$$\mathcal{D}_A : \mathcal{V}(A) \rightarrow !^s \mathcal{V}(A)$$

More concretely :

$$\mathcal{D}_A : ((A), A) \rightarrow ((A), !A)$$

Similar to : $\bar{\mathbf{d}}_A : A \rightarrow !A$

¹idea from [KMR25]

4.2 Linear Tangent Functor

We can prove the rules :

- Linear rule : $\partial_A; \mathbf{d}_A = \mathbf{w}_A \otimes \text{id}_A$
- Constant rule : $\partial_A; \mathbf{w}_A = 0$
- Chain rule : $\partial_A; \mathbf{p}_A = \mathbf{c}_A \otimes \text{id}_A; \mathbf{p}_A \otimes \partial_A; \partial_{!A}$

In a first order differential linear-non-linear adjunction.

4.2 Linear Tangent Functor

Definition (Differential LNL adjunction)

A differential linear-non-linear adjunction, is the data of an additive linear-non-linear adjunction and a functor $\mathcal{T} : \mathcal{C} \rightarrow LS(\mathcal{C})$ called a linear tangent functor such that :

1. \mathcal{T} is a section of ls .
2. $\mathcal{T}(-)_I = \mathcal{V}$
3. $D((\mathbf{c}_A)) = (\mathcal{D}_A \otimes \nu) + (\nu \otimes \mathcal{D}_A)$

4.2 Linear Tangent Functor

Theorem

For every Differential LNL adjunction, the category $LS(\mathcal{C})_I$ is a monoidal differential category.

4.2 Linear Tangent Functor

Theorem

For every Differential LNL adjunction, the category $LS(\mathcal{C})_I$ is a monoidal differential category.

Is the converse true ? Can we build a differential LNL adjunction from a monoidal differential category ?

4.2 Linear Tangent Functor

Theorem

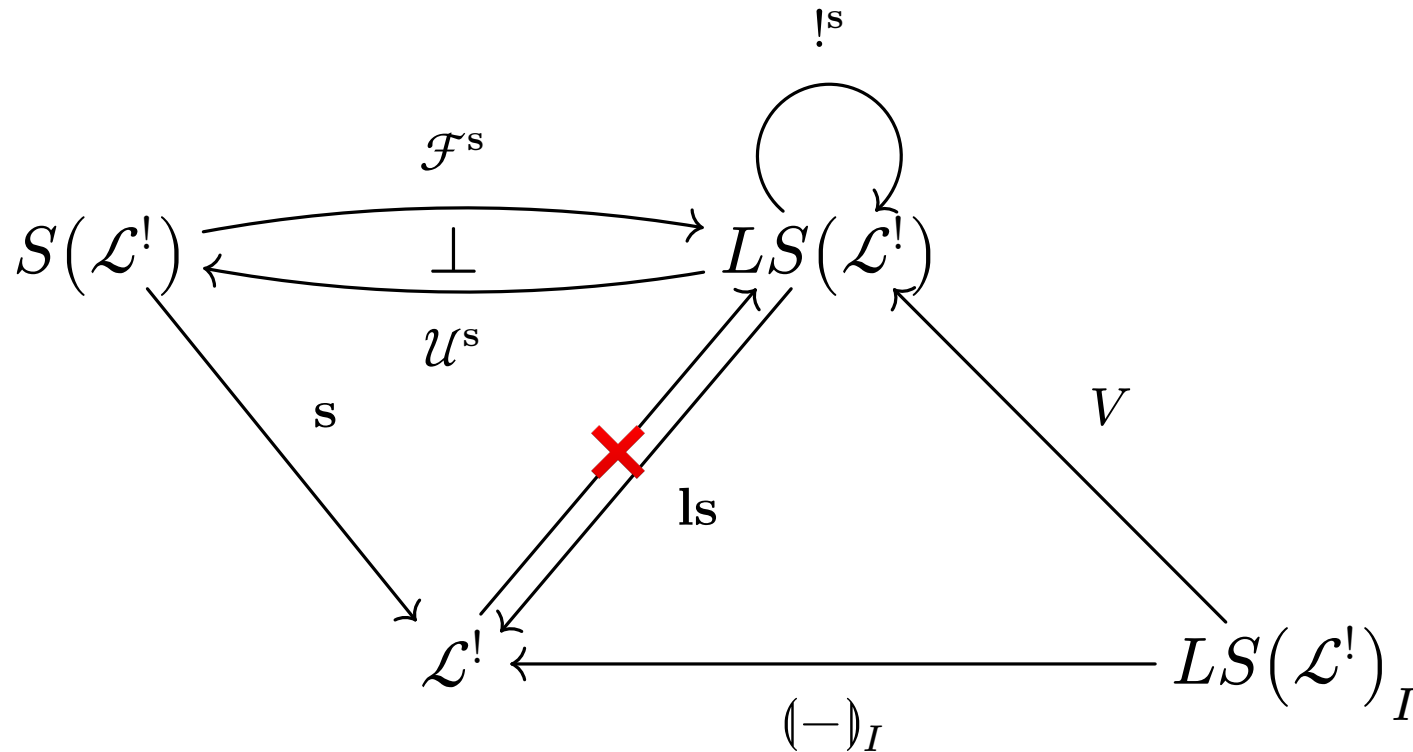
For every Differential LNL adjunction, the category $LS(\mathcal{C})_I$ is a monoidal differential category.

Is the converse true ? Can we build a differential LNL adjunction from a monoidal differential category ?

NO

4.2 Linear Tangent Functor

Let $(\mathcal{L}, \otimes, 1)$ be a monoidal differential category. It is involved in a LNL with its Eilenberg-Moore category $\mathcal{L}^!$.



4.2 Linear Tangent Functor

Theorem

Every monoidal differential category with biproducts defines a differential LNL with its Kleisli category.

4.3 Shortcoming

Axioms 2 and 3 are probably not optimal.

- Axiom 2 : $\mathcal{T}(-)_I = \mathcal{V}$. Describes the tangent space of vectors. Yet, we should be able to describe to some extent the tangent space of all vector bundle.

Proposition

For every smooth function $f : M \times V \rightarrow W$ with M a manifold, V, W vector spaces and for every $x \in M$, $f(x, -) : V \rightarrow W$ is linear then :

$$D_{(x,a)}(f)(0, v) = f(x, v)$$

4.3 Shortcoming

- Axiom 3 : $D(\mathbf{c}_A) = (\mathcal{D}_A \otimes \nu) + (\nu \otimes \mathcal{D}_A)$. It would be more elegant to replace it by a more “functorial” condition.

4.3 Shortcoming

- Axiom 3 : $D(\mathbf{c}_A) = (\mathcal{D}_A \otimes \nu) + (\nu \otimes \mathcal{D}_A)$. It would be more elegant to replace it by a more “functorial” condition.

The category $LS(\mathcal{C})_I$ is a Differential Category. What about the other slices $LS(\mathcal{C})_X$? They are not. Consequence of axiom 2 being not general enough.

4.3 Shortcoming

- Axiom 3 : $D(\mathbf{c}_A) = (\mathcal{D}_A \otimes \nu) + (\nu \otimes \mathcal{D}_A)$. It would be more elegant to replace it by a more “functorial” condition.

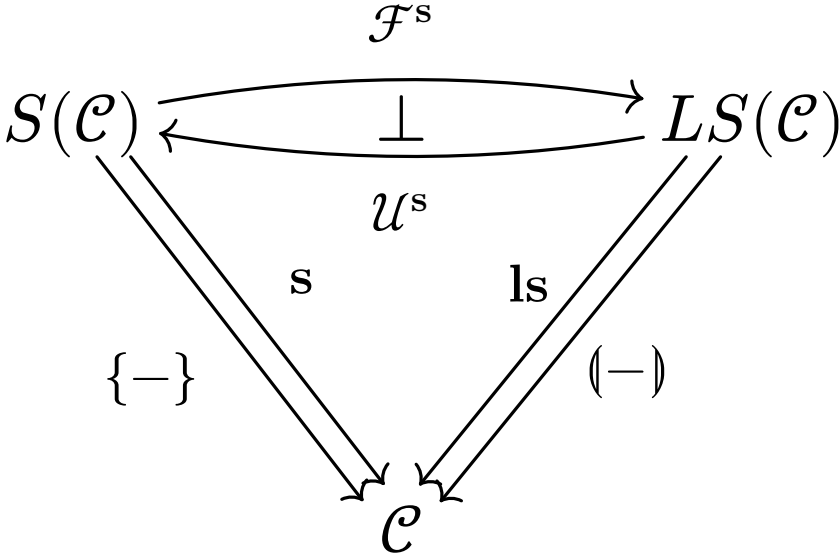
The category $LS(\mathcal{C})_I$ is a Differential Category. What about the other slices $LS(\mathcal{C})_X$? They are not. Consequence of axiom 2 being not general enough.

We have studied a very minimal fragment of LL, only allowing the multiplicative conjunction. It raises the questions of the interaction between \mathcal{T} , \oplus and the linear implication. Perhaps a solution to axioms 2 and 3 ?

5. Dependent Linear Logic and DiLL

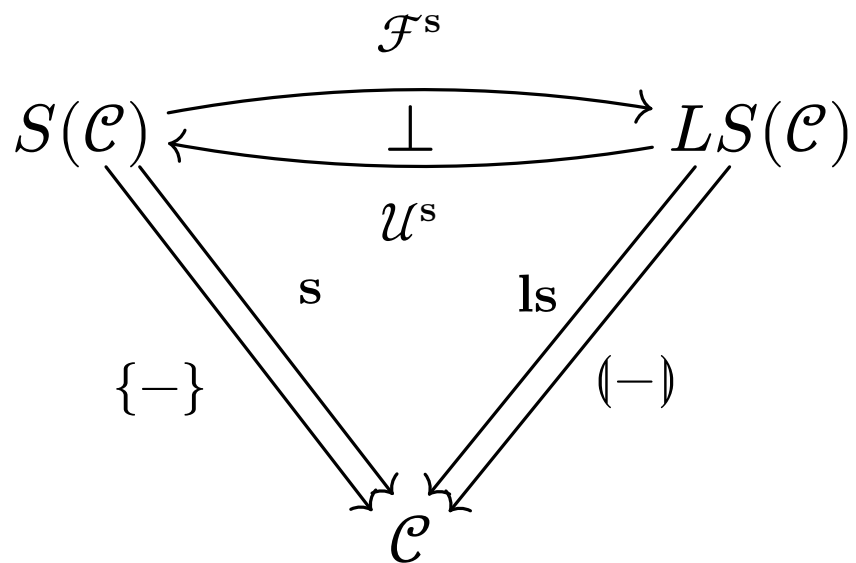
5.1 Linear Simple Category

This construction is a model of Dependent Linear Logic (as in [Lun, Vák15])



5.1 Linear Simple Category

This construction is a model of Dependent Linear Logic (as in [Lun, Vák15])



The fibration $LS(\mathcal{C})$ admits linear sigma types.

- None of the constructions in Differential linear-non-linear adjunctions rely crucially on the exact structure of $LS(\mathcal{C})$.

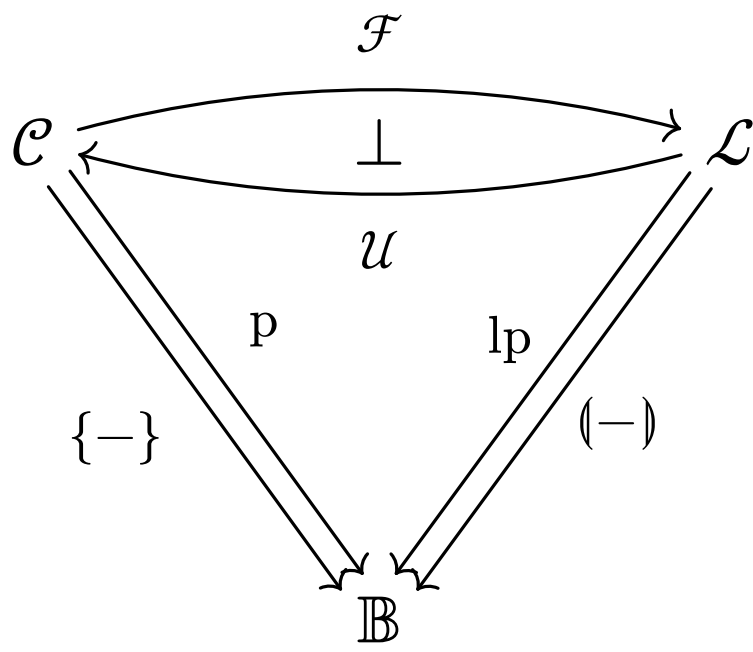
5.2 Generalisation

- None of the constructions in Differential linear-non-linear adjunctions rely crucially on the exact structure of $LS(\mathcal{C})$.
- They require linear sigma types and the “usual”/“expected” coherence conditions imposed on those sigma types.
 - ▶ Beck-Chevalley
 - ▶ Frobenius reversed
 - ▶ Coherence with !

- None of the constructions in Differential linear-non-linear adjunctions rely crucially on the exact structure of $LS(\mathcal{C})$.
- They require linear sigma types and the “usual”/“expected” coherence conditions imposed on those sigma types.
 - Beck-Chevalley
 - Frobenius reversed
 - Coherence with !
- A very good sign for generalisation to any model of Dependent Linear Logic.

5.2 Generalisation

A model of Dependent Differential Linear Logic should look like a model of DLL + a linear tangent functor.



Bibliography

- [Gir87] J.-Y. Girard, “Linear logic,” *Theoretical Computer Science*, vol. 50, no. 1, pp. 1–101, Jan. 1987, doi: 10.1016/0304-3975(87)90045-4.
- [ER06] T. Ehrhard and L. Regnier, “Differential interaction nets,” *Theoretical Computer Science*, vol. 364, no. 2, pp. 166–195, Nov. 2006, doi: 10.1016/j.tcs.2006.08.003.
- [Ehr18] T. Ehrhard, “An introduction to differential linear logic: proof-nets, models and antiderivatives,” *Mathematical Structures in Computer Science*, vol. 28, no. 7, pp. 995–1060, Aug. 2018, doi: 10.1017/S0960129516000372.
- [Mel09] P.-A. Melliès, “Categorical Semantic of Linear Logic,” *Interactive models of computation and program behavior*, no. numéro27. in Panoramas et synthèses. Société mathématique de France, Paris, pp. 1–196, 2009.
- [Blu+20] R. F. Blute, J. R. B. Cockett, J.-S. P. Lemay, and R. A. G. Seely, “Differential Categories Revisited,” *Applied Categorical Structures*, vol. 28, no. 2, pp. 171–235, Apr. 2020, doi: 10.1007/s10485-019-09572-y.
- [KMR25] M. Kerjean, V. Maestracci, and M. Rogers, “Functorial Models of Differential Linear Logic,” *LIPICs, Volume 337, FSCD 2025*, vol. 337, pp. 1–17, 2025, doi: 10.4230/LIPICS.FSCD.2025.26.
- [Jac99] B. Jacobs, *Categorical logic and type theory*, 1st ed., no. v.141. in Studies in logic and the foundations of mathematics. Amsterdam ; New York: Elsevier Science, 1999.
- [CC14] J. R. B. Cockett and G. S. H. Cruttwell, “Differential Structure, Tangent Structure, and SDG,” *Applied Categorical Structures*, vol. 22, no. 2, pp. 331–417, Apr. 2014, doi: 10.1007/s10485-013-9312-0.
- [Lun] M. Lundfall, “Models of linear dependent type theory.”
- [Vák15] M. Vákár, “A Categorical Semantics for Linear Logical Frameworks,” *Foundations of Software Science and Computation Structures*, vol. 9034. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 102–116, 2015. doi: 10.1007/978-3-662-46678-0_7.