



Coherent differentiation in models of Linear Logic

Aymeric Walch, aymeric.walch@ens-lyon.fr
ENS DE LYON
M2IF

June, 2022

Under the supervision of Thomas Ehrhard

Proof theory is a field whose objects of study are the proof themselves. A fundamental concept of proof theory is the Curry Howard isomorphism that relates programs with proof in three levels. The types of a programming language relates to formulas. A program of type $A \rightarrow B$ relates to a proof of the formula $A \Rightarrow B$. Finally, program execution (namely the β -reduction) relates to a notion of computation on proof called the *cut elimination*. This observation introduced a deeply influential idea: proof theory is not only about the *provability* of formulas, but also about the *computational content* of their proofs. Two proofs of the same formula can relate deeply between each other, or conversely have nothing to do in common.

The issue is that computation is usually very hard to describe and study, as many operational properties can interact between each other in a very subtle way. Thus designing a proof system (or a calculus) usually comes with a lot of requirements, such as proving subject reduction (computation conserves typing), normalization (computation terminates), confluence (different computational strategy converges to the same result), etc. Denotational semantic is a field that arose in order to simplify those issues. The goal of this field is to describe *what* a program computes while being agnostic on *how* it computes, by giving some object $\mathcal{M}$ called a *model* and an interpretation $\llbracket \cdot \rrbracket : \{\text{proofs/programs}\} \rightarrow \mathcal{M}$ invariant under computation. People then noticed that those models should actually have the structure of categories, hence turning the field of denotational semantic into the field of *categorical semantic*. This field found quite a lot of success, for the reason mentioned above but also in more concrete setting: categorical semantic gives a way to describe the “meaning” of a program, something that is quite crucial when one wants to prove that the program behaves correctly.

Categorical semantic has another crucial application. Sometimes, some properties that are not captured by the syntax might be shared between many different models. It might suggest some new syntactical ideas that should be introduced to account for those. This is what happened when Girard [7] discovered *linear logic*, a refinement of intuitionistic and classical logic in which proofs and morphisms are interpreted through the lenses of linear algebra. This is also what happened when my supervisor Thomas Ehrhard [6] built upon the ideas of linear logic and discovered that many models had a notion of “linear approximation” akin to differentiation, hence leading to the development of *differential linear logic*. This is finally what might be happening again with the very recent introduction of *coherent differentiation* also by Ehrhard [4]. He noticed that many models of linear logic were not models of differential linear logic, yet still admitted a notion of differentiation, suggesting that differential linear logic could be improved.

The goal of this internship was to work in the development of this exciting new concept. Concretely, the internship was split in three part of one month and a half each. The first part consisted in a (somewhat long) appropriation of the basics of categorical semantic, of models of linear logic, of differential logic and of coherent differentiation. The second part consisted in showing that coherent differentiation is indeed a generalization of differential categories, a result that was somewhat expected but required some work. Finally, the third part is too conceptual to explain for now and is hinted in [Section 6](#).

I feel that categorical semantic is a field that is particularly inaccessible and hard to vulgarize. The issue is not that this field is more complex or subtle than any other field. Rather, different notions tends to stack up really quickly. Besides, this field is ultimately based on category theory, a theory that is well known to be quite abstract and quite dividing between its adepts and its detractors. For this reason, there is little hope to explain my work in the scope of a report that is supposed to assume that the reader has no particular background in the field. So this report will mostly consists in an introduction. An introduction to the field of categorical semantic in [Section 1](#). An introduction to how categorical semantic lead to the discovery of linear logic in [Section 2](#). An introduction to how categorical semantic lead to the discovery of differential linear logic in [Section 3](#). And finally, an introduction to coherent differentiation in [Section 4](#). If I do my work correctly, then the reader should be able to understand what I worked on and for what reason I worked on those objects. Then I wrap up the report with [Section 5](#) that gives a taste of the methodology I developed to show that coherent differentiation is a generalization of differentiation and with [Section 6](#) as an appetizer on what I discovered in the last part of the internship.

1 A (brief) introduction to categorical semantic

This introduction to categorical semantic is loosely inspired by this very enlightening paper from Melies [12].

1.1 A small and non exhaustive reminder on proof systems

I will assume that the reader is somewhat familiar with the notion of proof system, and stick to a minimal amount of material.

Definition 1. *Formulas* are inductively defined as the smallest set containing a set of propositional variables V and which is closed under some set of connectives. For the sake of conciseness, there are only two connectives in this report: the conjunction ($\wedge$) and implication ($\Rightarrow$).

$$F, G := v \in V \mid F \wedge G \mid F \Rightarrow G \mid \top$$

A *context* is a list of formulas $\Gamma = A_1, \dots, A_n$. A *sequent* $\Gamma \vdash B$ consists in a context Γ together with a formula B . It should be interpreted as the formula $A_1 \wedge A_2 \wedge \dots \wedge A_n \dots \Rightarrow B$.

Definition 2. A *proof system* consists in a set of *inference rules* in the stylized form

$$\frac{\Gamma_1 \vdash B_1 \quad \dots \quad \Gamma_n \vdash B_n}{\Gamma \vdash B} \quad \text{and, for the case } n=0 \quad \frac{}{\Gamma \vdash B}$$

The $\Gamma_i \vdash B_i$ are called the *premises* of the rule, and $\Gamma \vdash B$ is called the *conclusion*. An *axiom* is a rule with no premises.

Definition 3. A *proof* of a sequent $\Gamma \vdash B$ consists in a labeled tree such that the label of the root is $\Gamma \vdash B$ and such that the label of each node is the conclusion of a rule whose premises are the labels of its children. The labels of the leafs should be the conclusion of an axiom.

Usually, any sensible proof system contains two kinds of rules. First are structural rules, that allows to manipulate sequents. For example, the rules below are structural rules.

$$\frac{}{\Gamma \vdash A}(\text{ax}) \quad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}(\text{weak}) \quad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}(\text{contr}) \quad \frac{\Gamma_1, A_1, A_2, \Gamma_2 \vdash B}{\Gamma_1, A_2, A_1, \Gamma_2 \vdash B}(\text{ex})$$

The rule **ax** is the mandatory axiom rule. The weakening rules **weak** discard hypothesis. The contraction rules **contr** duplicate hypothesis. Finally, the exchange rule **ex** swap the position of formulas. The usage of this last rule is usually kept implicit.

Second, there are connectors rules that manipulate the different connectors. In this report, I stick to rules where the connector is only in the conclusion. The connector rules are classified depending on the position of the connector with respect to the $\vdash$ symbol: right or left.

	Left rules	Right rules	
$\wedge$:	$\frac{\Gamma, A_1 \vdash B \quad \Gamma, A_2 \vdash B}{\Gamma, A_1 \wedge A_2 \vdash B}$	$\frac{\Gamma \vdash B_1 \quad \Gamma \vdash B_2}{\Gamma \vdash B_1 \wedge B_2}$	($\wedge$ -rules)
$\top$	No rule	$\frac{}{\Gamma \vdash \top}$	($\top$ -rules)
$\Rightarrow$:	$\frac{\Gamma_1 \vdash A \quad B, \Gamma_2 \vdash C}{A \Rightarrow B, \Gamma_1, \Gamma_2 \vdash C}$	$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$	($\Rightarrow$ -rules)

Left rules give a way to *use* an hypothesis that contains a connector. Right rules give a way to *build* a conclusion that contains a connector. There is one last structural rule that is crucial for any proof system.

Definition 4. The cut rule is the following structural rule:
$$\frac{\Gamma_1 \vdash B \quad B, \Gamma_2 \vdash C}{\Gamma_1, \Gamma_2 \vdash C}$$

Definition 5. This proof system is a fragment of a bigger proof system called *LJ* (intuitionistic logic). So I will call this system LJ for the rest of the report.

Remark 1. The reader already versed into proof theory and/or linear logic might have noticed that this system has a mix of *additive* and *multiplicative* rules. They have been chosen this way to make the flow of the presentation as minimal and natural as possible.

The cut rule is the cornerstone of any proof systems as it allows to introduce any arbitrary “lemma” B . It can also be seen as a kind of composition of a proof of $A \Rightarrow B$ with a proof of $B \Rightarrow C$. Intuitively, this rule is not necessary: it is always possible to get rid of lemmas by “unfolding their proof on the fly”. The formal pendant of this idea is called the *cut elimination*.

Theorem 1 (Cut elimination). *The cut rule is admissible in LJ. In other word: for any proof using the cut rule, there exists a proof of the same sequent that does not use the cut rule. Besides, the proof of this theorem gives an explicit procedure called the cut elimination procedure that takes as input any proof using the cut rule and outputs a proof of the same sequent without the cut rule.*

This cut elimination procedure is of tremendous importance as it introduces a notion of computation in proof: computing a proof consists in applying the cut elimination procedure, and the result of this computation will be a cut-free proof (a proof with no cut rule). In fact, cut elimination is the proof theoretic equivalent of the normalization in the lambda calculus through the Curry Howard isomorphism. Besides, the the cut free proofs are the proof theoretic equivalent of the β -normal terms.

1.2 Why categories are natural viewpoints on semantic

The cut elimination procedure is all about *how* proofs compute. Denotational semantic is instead more interested in *what* proofs compute. The goal of denotational semantic is to associate to any proof π some interpretation $\llbracket \pi \rrbracket$ that is agnostic to the computational content of π .

Definition 6. The interpretation of a proof should be invariant under cut elimination: if a proof π' is obtained from π by cut elimination, then $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$. Besides, the interpretation should be *modular*: If π is a proof of shape

$$\frac{\frac{\pi_1}{\Gamma_1 \vdash B_1} \quad \dots \quad \frac{\pi_n}{\Gamma_n \vdash B_n}}{\Gamma \vdash B}$$

Then $\llbracket \pi \rrbracket$ should be uniquely determined by the interpretations $\llbracket \pi_1 \rrbracket, \dots, \llbracket \pi_n \rrbracket$ (so even if the proofs π_i change, if their interpretation stays the same, the interpretation of π remains the same).

From those reasonable assumptions, we can infer quite a lot about the structure in which the interpretations $\llbracket \pi \rrbracket$ must live. Take those two proof tree.

$$(\pi): \frac{}{A \vdash A} \quad (\pi'): \frac{\frac{\pi_1}{A \vdash B} \quad \frac{\pi_2}{B \vdash C}}{A \vdash C}$$

The proof π suggest that for all formula A there exists an entity $id_A := \llbracket \pi \rrbracket$. The proof π' and the constraint of modularity suggests that there exists an operation $\circ$ on the entities (tied to some compatibility conditions) defined as $\llbracket \pi_2 \rrbracket \circ \llbracket \pi_1 \rrbracket := \llbracket \pi' \rrbracket$. Besides, the structure of the cut elimination procedure together with the invariance condition implies some algebraic properties on id_A and $\circ$ that are the ones of a *category* (the full detailed process can be found in Mellies’s paper [12]). So the semantic of a proof system *has to be* a category. Denotational semantic can thus be renamed as *categorical* semantic.

Definition 7 (Models). A model of a proof theory consists of a category $\mathcal{L}$ together with

- An interpretation of formulas: there exists a function $\llbracket \cdot \rrbracket : \{\text{formulas}\} \rightarrow \text{Obj}(\mathcal{L})$. This interpretation extends to contexts, by defining $\llbracket A_1, \dots, A_n \rrbracket := \llbracket A_1 \wedge \dots \wedge A_n \rrbracket$.
- An interpretation of proof: for any sequent $\Gamma \vdash B$, there exists a function $\llbracket \cdot \rrbracket_{\Gamma, B} : \{\text{proofs of } \Gamma \vdash B\} \rightarrow \mathcal{L}(\llbracket \Gamma \rrbracket, \llbracket B \rrbracket)$. Usually, Γ and B are kept implicit and I will write $\llbracket \pi \rrbracket$ for $\llbracket \pi \rrbracket_{\Gamma, B}$. This interpretation should be invariant and modular in the sense of [Definition 6](#).

1.3 Exhibiting the structure that $\mathcal{L}$ must have

Assume that $\mathcal{L}$ is a model of LJ. Let us infer the structure that $\mathcal{L}$ must have. First, the interpretation of the proof $\overline{A \vdash \top}$ gives a morphism $\mathbf{t}_{\llbracket A \rrbracket} : \llbracket A \rrbracket \rightarrow \llbracket \top \rrbracket$. Besides, a syntactical property of the cut elimination procedure called η -expansion together with invariance and modularity imply that any proof of $A \vdash \top$ must have for interpretation $\mathbf{t}_{\llbracket A \rrbracket}$ (I will not go into further details). So $\mathcal{L}$ must have what is called a *final object*.

Definition 8. A final object in $\mathcal{L}$ is an object $\top_{\mathcal{L}}$ such that for any object A of $\mathcal{L}$, there exists a unique morphism $\mathbf{t}_A : A \rightarrow \top_{\mathcal{L}}$.

Furthermore, take the three proofs below.

$$\pi : \frac{\pi_1}{A \vdash B_1} \quad \frac{\pi_2}{A \vdash B_2} \quad \pi' : \frac{\overline{B_1 \vdash B_1}}{B_1 \wedge B_2 \vdash B_1} \quad \pi'' : \frac{\overline{B_2 \vdash B_2}}{B_1 \wedge B_2 \vdash B_2}$$

By modularity, the interpretation of π is completely characterized by $\llbracket \pi_1 \rrbracket$ and $\llbracket \pi_2 \rrbracket$. So we can define an operator $\langle \cdot, \cdot \rangle$ called the *pairing* as $\langle \llbracket \pi_1 \rrbracket, \llbracket \pi_2 \rrbracket \rangle := \llbracket \pi \rrbracket : \llbracket A \rrbracket \rightarrow \llbracket B_1 \wedge B_2 \rrbracket$. Besides, the interpretations of the proofs π' and π'' provide two morphisms $\mathbf{p}_1 := \llbracket \pi' \rrbracket : \llbracket B_1 \wedge B_2 \rrbracket \rightarrow \llbracket B_1 \rrbracket$ and $\mathbf{p}_2 := \llbracket \pi'' \rrbracket : \llbracket B_1 \wedge B_2 \rrbracket \rightarrow \llbracket B_2 \rrbracket$. Again, invariance implies that $\mathbf{p}_i \circ \langle \llbracket \pi_1 \rrbracket, \llbracket \pi_2 \rrbracket \rangle = \llbracket \pi_i \rrbracket$ and $\mathbf{p}_2 \circ \langle \llbracket \pi_1 \rrbracket, \llbracket \pi_2 \rrbracket \rangle = \llbracket \pi_2 \rrbracket$ and the η -expansion implies that such pairing should be unique. As a result, a model of LJ must be what is called a Cartesian category.

Definition 9. A category $\mathcal{L}$ is said to be Cartesian if it has a final object, and if for any pairs of objects Y_1 and Y_2 , there exists an object $Y_1 \times Y_2$ and two morphisms $\mathbf{p}_1 : Y_1 \times Y_2 \rightarrow Y_1$ and $\mathbf{p}_2 : Y_1 \times Y_2 \rightarrow Y_2$ such that for any object X and morphisms $f_1 : X \rightarrow Y_1$, $f_2 : X \rightarrow Y_2$, there exists a unique morphism $\langle f_1, f_2 \rangle : X \rightarrow Y_1 \times Y_2$ such that the following diagram commutes.

$$\begin{array}{ccc} & X & \\ f_1 \swarrow & \downarrow \langle f_1, f_2 \rangle & \searrow f_2 \\ Y_1 & & Y_2 \\ \mathbf{p}_1 \swarrow & & \searrow \mathbf{p}_2 \\ & Y_1 \times Y_2 & \end{array}$$

The strength of categorical semantic is that it interprets complex syntactical interactions by this kind of simple property that consists in the existence of an object and of a unique morphism following some commutations. Those properties are called *Universal Mapping Property* (UMP). Any UMP generates a functor, such as the one below in the case of the Cartesian product.

Definition 10. This property induces a *bifunctor* (see the [annex](#) for the terminology) $_ \times _ : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ that maps two objects (X_1, X_2) to $X_1 \times X_2$ and two morphism $f_1 : X_1 \rightarrow Y_1$ and $f_2 : X_2 \rightarrow Y_2$ to a morphism $f_1 \times f_2 := \langle f_1 \circ \mathbf{p}_1, f_2 \circ \mathbf{p}_2 \rangle : X_1 \times X_2 \rightarrow Y_1 \times Y_2$ that consists in “applying f_1 to the first coordinate and applying f_2 to the second”.

Remark 2. This functor admits a natural isomorphism $\alpha_{X,Y,Z}^\times := \langle \mathbf{p}_1 \circ \mathbf{p}_1, \langle \mathbf{p}_2 \circ \mathbf{p}_1, \mathbf{p}_2 \rangle \rangle : (X \times Y) \times Z \rightarrow X \times (Y \times Z)$. It allows us to write $X_1 \times \dots \times X_n$ and $f_1 \times \dots \times f_n$ “up to any arbitrary choice of parenthesis” by keeping implicit the use of $\alpha^\times$. $\mathcal{L}$ is said to admit all finite products.

The arrow has a similar interpretation. Because of length consideration, I will not do the whole process again, but directly introduce the universal mapping property required.

Definition 11. Given a bifunctor $_ \times _$, a category $\mathcal{L}$ is said to be closed (with regard to $\times$) if for all pairs of object X and Y , there exists an object $X \Rightarrow_{\mathcal{L}} Y$ and a morphism $\mathbf{ev} : (X \Rightarrow_{\mathcal{L}} Y) \times X \rightarrow Y$ such that for any object Z and morphism $f : X \times Y \rightarrow Z$, there exists a unique morphism $\mathbf{cur}(f) : (X \Rightarrow_{\mathcal{L}} Z)$ such that the diagram below commutes.

$$\begin{array}{ccc}
X \times Y & \xrightarrow{f} & Z \\
\text{cur}(f) \times \text{id}_Y \downarrow & \nearrow \text{ev} & \\
(Y \Rightarrow_{\mathcal{L}} Z) \times Y & &
\end{array}$$

For the reader not used to category theory, this definition might cause some issues. But any computer scientist used to functional programming and currying is in fact already used to this idea. Recall that currying means that any function with two arguments $f : A \times B \rightarrow C$ can be uniquely mapped to a function $\text{cur}(f) : A \rightarrow (B \rightarrow C)$ defined as $a \mapsto (b \mapsto f(a, b))$. Closedness is similar and means that $\mathcal{L}$ contains an object $B \Rightarrow_{\mathcal{L}} C$ that describes the “set of morphism from B to C ” and that any morphism $f : A \times B \rightarrow C$ can be uniquely seen as a morphism $\text{cur}(f) : A \rightarrow (B \Rightarrow_{\mathcal{L}} C)$. In fact, Curryfication is a particular case of closedness where the morphism ev is the function that takes as input a function $h : A \rightarrow B$ and $x \in A$ and returns $h(x)$.

Definition 12. A Cartesian closed category (CCC) is a category with a Cartesian product that is closed with regard to this product.

We saw that the models of LJ have to be Cartesian closed categories. Conversely, any Cartesian closed category $\mathcal{L}$ together with a function $\llbracket \cdot \rrbracket : X \rightarrow \text{Obj}(\mathcal{L})$ defines a model. Indeed, the interpretation of formulas can be defined inductively as $\llbracket x \rrbracket := [x]$, $\llbracket A \wedge B \rrbracket := \llbracket A \rrbracket \times \llbracket B \rrbracket$, $\llbracket \top \rrbracket := \top_{\mathcal{L}}$ and $\llbracket A \Rightarrow B \rrbracket := \llbracket A \rrbracket \Rightarrow_{\mathcal{L}} \llbracket B \rrbracket$. This interpretation extends to context by defining $\llbracket A_1, \dots, A_n \rrbracket = \llbracket A_1 \wedge \dots \wedge A_n \rrbracket = \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket$ (using [Remark 2](#) that ensures parenthesis irrelevance).

The interpretation of a proof is then defined inductively. I will not give the full details, just that each of the rule is interpreted by its semantic equivalent: the rule **(ax)** is interpreted by an identity, the rule **(cut)** by a composition, the rules **($\wedge$ -left)** by the projections, the rule **($\wedge$ -right)** by a pairing, the rule **($\Rightarrow$ -left)** by ev and the **($\Rightarrow$ -right)** rule by a **cur**. It only leaves the rules **(contr)**, **(weak)** and **(ex)**. The rule **(weak)** can directly be interpreted by the first projection $\mathbf{p}_1$. The rule **(contr)** is tied to the existence of a natural transformation $\Delta_X^\times : X \rightarrow X \times X$ that can be defined as $\Delta^\times := \langle \text{id}, \text{id} \rangle$. Finally, the exchange rule is tied to the existence of a natural transformation $\gamma_{X,Y}^\times : X \times Y \rightarrow Y \times X$ than we can be defined as $\gamma := \langle \mathbf{p}_2, \mathbf{p}_1 \rangle$.

2 An introduction to linear logic

Linear logic is a proof theory discovered by Girard [7] while he was studying a model **Coh** (the category of coherent spaces) that he discovered [8]¹. Girard noticed that the object of the closure $A \Rightarrow B$ could be decomposed in two successive constructions $!A \multimap B$ where $! : \text{Obj}(\mathbf{Coh}) \rightarrow \text{Obj}(\mathbf{Coh})$ and $_ \multimap _ : \text{Obj}(\mathbf{Coh})^2 \rightarrow \text{Obj}(\mathbf{Coh})$. Therefore, Girard introduced a category **Lin** whose objects are the objects of **Coh** and whose morphisms are defined as $\mathbf{Lin}(A, B) := A \multimap B$. This category admits a functor $! : \mathbf{Lin} \rightarrow \mathbf{Lin}$ and by design, for any objects A, B , $\mathbf{Coh}(A, B) = \mathbf{Lin}(!A, B)$. Besides, **Lin** has some though inducing properties:

- The object $A \times B$ still gives a Cartesian product in **Lin** and $\top$ is still a final object. So **Lin** is Cartesian.
- However, **Lin** is not closed with regard to this product.
- In fact, **Lin** is closed with regard to some bifunctor $\otimes$. Besides, there exists an object 1 such that $(\mathbf{Lin}, 1, \otimes)$ is what is called a “symetric monoidal category”, see [Definition 13](#) below.

It turns out that there is a strong analogy between these objects and linear algebra. Let **Vect** be the category whose objects are the $\mathbb{R}$ -vector spaces and whose morphisms are the (continuous) linear maps. **Vect** is a Cartesian category: the final object $\top$ is the trivial vector space $\{0\}$ and the Cartesian product is nothing more than the usual product of two vector spaces $E \times F$. Besides, this category seems to be closed: $\mathbf{Vect}(E, F)$ is a vector space itself. However it is not closed with regard to the cartesian product: giving a function $f \in \mathbf{Vect}(E, \mathbf{Vect}(F, G))$ is the same as giving a *bilinear* application $f : E \times F \rightarrow G$, not a linear map.

¹There is a bit of storytelling here. The real story turns out to be even more interesting !

However it is possible to define a vector space $E \otimes F$ called the “tensor product of E and F”. I will only give its definition in finite dimension, as it is more intuitive. If E is a vector space of dimension n and basis (e_i) and if F is a vector space of dimension m and basis (f_j) , then $E \otimes F$ is a vector space of dimension $n.m$ with basis denoted as $(e_i \otimes f_j)$. The idea of this space is to factorize bilinear applications: any bilinear application $\phi : E \times F \rightarrow G$ can be characterized as a (unique) linear application $\bar{\phi} : E \otimes F \rightarrow G$ defined on the basis as $\bar{\phi}(e_i \otimes f_j) := \phi(e_i, f_j)$. So $\mathbf{Vect}(E, \mathbf{Vect}(F, G))$ coincides with the set of bilinear applications $E \times F \rightarrow G$ that coincides with $\mathbf{Vect}(E \otimes F, G)$. This is exactly a closure with regard to $\otimes$.

2.1 Symetric monoidal categories

In the two examples above, the functor $\otimes$ gives to the category the structure of a *symetric monoidal category*. This notion existed prior to linear logic, see [11]. The definition below is a refinement due to Kelly [9].

Definition 13. A *monoidal category* $(\mathcal{L}, \otimes, 1)$ consists in a category $\mathcal{L}$, a bifunctor $\otimes$ on $\mathcal{L}$ and an object 1 of $\mathcal{L}$ called the unit such that there exists three natural isomorphism $\alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$, $\lambda_A : 1 \otimes A \rightarrow A$, $\rho_A : A \otimes 1 \rightarrow A$ that folows the commutations below..

$$\begin{array}{ccccc}
 & & (A \otimes B) \otimes (C \otimes D) & & \\
 & \swarrow \alpha_{A,B,C \otimes D} & & \nwarrow \alpha_{A \otimes B,C,D} & \\
 (A \otimes 1) \otimes B & \xrightarrow{\alpha} & A \otimes (1 \otimes B) & & \\
 \searrow \lambda \otimes id_B & & \downarrow id_A \otimes \rho & & \\
 & A \otimes B & & A \otimes (B \otimes (C \otimes D)) & \\
 & \uparrow id_A \otimes \alpha_{B,C,D} & & \downarrow \alpha_{A,B,C} id_D & \\
 & A \otimes ((B \otimes C) \otimes D) & \xleftarrow{\alpha_{A,B \otimes C,D}} & (A \otimes (B \otimes C)) \otimes D &
 \end{array}$$

A *symetric monoidal category* $\mathcal{L}$ is a monoidal category that is also equiped with a natural transformation $\gamma_{A,B} : A \otimes B \rightarrow B \otimes A$. γ is required to be an involution: $\gamma_{B,A} \circ \gamma_{A,B} = id_{A \otimes B}$. Finally, it should follw those commutations below.

$$\begin{array}{ccccc}
 1 \otimes A & \xrightarrow{\gamma_{1,A}} & A \otimes 1 & & (A \otimes B) \otimes C \xrightarrow{\alpha_{A,B,C}} A \otimes (B \otimes C) \xrightarrow{\gamma_{A,B \otimes C}} (B \otimes C) \otimes A \\
 \searrow \lambda & & \downarrow \rho & \swarrow \gamma_{A,B} id_C & \downarrow \alpha_{B,C,A} \\
 & A & & (B \otimes A) \otimes C & \xrightarrow{\alpha_{B,A,C}} B \otimes (A \otimes C) \xrightarrow{id_B \otimes \gamma_{A,C}} B \otimes (C \otimes A)
 \end{array}$$

The axioms of monoidal categories are the minimal axioms required to ensure that we can rewrite parenthesis, collapse the unit, and swap formulas in any arbitrary order and still end up with the same morphism (it is a generalization of the regular notion of monoid). Associativity in particular allows to define $A_1 \otimes \dots \otimes A_n$ and $f_1 \otimes \dots \otimes f_n$ “up to parenthesis rewriting”, in a similar way as Remark 2. In fact, a Cartesian category is a particular kind of symetric monoidal category.

Remark 3. If $\mathcal{L}$ is Cartesian, the functor $_ \times _$ associated to a Cartesian product gives to $\mathcal{L}$ the structure of a symetric monoidal category $(\times, \top, \alpha^\times, \lambda^\times, \rho^\times)$, where $\alpha^\times$ and $\gamma^\times$ are defined in Section 1.3, $\lambda_A^\times = \mathbf{p}_1 : A \times \top \rightarrow A$, $\rho_A^\times = \mathbf{p}_2 : \top \times A \rightarrow A$. Note that $\lambda^\times$ and $\rho^\times$ are isomorphisms because $\top$ is a final object (their inverses are respectively $\langle id_A, \mathbf{t}_A \rangle$ and $\langle \mathbf{t}_A, id_1 \rangle$). So the notion of symetric monoidal category was already the notion we implicitly used when we said that “parenthesis for $\times$ are irrelevant and we can interpret the exchange rule”.

2.2 Deducing the syntax from the semantic

These models suggested the existence of a logic where two notions of conjunction might coexist. So Girard introduced a logical system in which formulas had three binary connectives: one implication $\multimap$ (called the linear arrow), and two conjunction $\otimes$ (called the tensor) and $\&$ (called the with). The semantic of $\otimes$ should be $\otimes$, the semantic of $\multimap$ should be $\multimap$, and the semantic of $\&$ should be $\times$. There are also two constant symbols 1 (associated to $\otimes$) and $\top$ (associated to $\&$) whose semantic should be respectively the objects 1 and $\top$. Girard introduced the following syntax.

	Left rules	Right rules	
$\&$:	$\frac{\Gamma, A_1 \vdash B}{\Gamma, A_1 \& A_2 \vdash B} \quad \frac{\Gamma, A_2 \vdash B}{\Gamma, A_1 \& A_2 \vdash B}$	$\frac{\Gamma \vdash B_1 \quad \Gamma \vdash B_2}{\Gamma \vdash B_1 \& B_2}$	($\&$ -rules)
$\top$	No rule	$\overline{\Gamma \vdash \top}$	($\top$ -rules)
$\multimap$:	$\frac{\Gamma_1 \vdash A \quad \Gamma_2, B \vdash C}{\Gamma_1, \Gamma_2, A \multimap B \vdash C}$	$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}$	($\multimap$ -rules)
$\otimes$	$\frac{\Gamma, A_1, A_2 \vdash B}{\Gamma, A_1 \otimes A_2 \vdash B}$	$\frac{\Gamma_1 \vdash B_1 \quad \Gamma_2 \vdash B_2}{\Gamma_1, \Gamma_2 \vdash B_1 \otimes B_2}$	($\otimes$ -rules)
1	$\frac{\Gamma \vdash B}{\Gamma, 1 \vdash B}$	$\overline{\vdash 1}$	(1 -rules)

Together with the structural rules.

$$\frac{}{A \vdash A}(\text{ax}) \quad \frac{\Gamma_1, A_1, A_2, \Gamma_2 \vdash B}{\Gamma_1, A_2, A_1, \Gamma_2 \vdash B}(\text{ex}) \quad \frac{\Gamma_1 \vdash B \quad \Gamma_2, B \vdash C}{\Gamma_1, \Gamma_2 \vdash C}(\text{cut})$$

Let us motivate those rules with semantic in mind. The rules associated to $\&$, $\top$ and $\multimap$ are the same as the rule that were associated to $\wedge$, $\top$ and $\Rightarrow$, because as we saw the semantic counterpart of those rules is the structure of a Cartesian product, of a final object, and of a closure. Besides, a sequent $A_1, \dots, A_n \vdash B$ should be interpreted as the formula $A_1 \otimes \dots \otimes A_n \multimap B$ (and not $A_1 \& \dots \& A_n \multimap B$) as the linear arrow is closed with regard to $\otimes$ (and not $\times$). In particular, the ($\otimes$ -left) rule is nothing more than some parenthesis rewriting. On the other hand, the ($\otimes$ -right) rule betrays the fact that $\otimes$ is a bifunctor: given two morphisms $f_1 : [\Gamma_1] \rightarrow [B_1]$ and $f_2 : [\Gamma_2] \rightarrow [B_2]$, there exists a morphism $f_1 \otimes f_2 : [\Gamma_1] \otimes [\Gamma_2] \rightarrow [B_1] \otimes [B_2]$. Similarly, the right (1 -rule) does not say anything semantically (the empty context is interpreted as the formula 1), while the left (1 -rule) is the syntactic equivalent of the morphism $\rho_{[\Gamma]} : [\Gamma] \otimes 1 \rightarrow [\Gamma]$. Finally, the exchange rule carries to linear logic thanks to the symmetry $\gamma_{[A],[B]} : [A] \otimes [B] \rightarrow [B] \otimes [A]$.

Definition 14. The proof system introduced in this section is called iMALL (Intuitionistic Multiplicative Additive Linear Logic). A model of iMALL is a symmetric monoidal category $(\mathcal{L}, \otimes, 1, \alpha, \lambda, \rho, \gamma)$ closed with regard to $\otimes$ that also contains a Cartesian product $\times$. From this point onward, I will write the product $\&$ instead of $\times$ in order to match with the logical connector.

Linear logic should be seen as a logic about *resources management*. A formula $A \multimap B$ states that “using the resource A once and only once produces the outcome B”. The formula $A \otimes B$ is interpreted as “having both a resource A and a resource B at the same time”. So $A, A \vdash A \otimes A$ is provable, but not $A \vdash A \otimes A$. The formula $A \& B$ is interpreted as “choosing at will between a resource A and a resource B, but not being able to chose both”. So $A \vdash A \& A$ is provable, but not $A, A \vdash A \& A$.

2.3 About the weakening and contraction rules

The most notable thing about this new system is the absence of weakening and contraction rules. There is an obvious syntactical reason. The presence of those two rules would imply the provability of the sequents $A \otimes B \vdash A \& B$, $A \& B \vdash A \otimes B$, $1 \vdash \top$ and $\top \vdash 1$. It would lead to a collapse between the connector $\otimes$ with the connector $\&$. This is exactly what happens in the system LJ that I introduced in [Section 1.1](#).

There is a similar observation in the world of semantic. Assume that the weakening and contraction rules hold for a given formula A. Take the proofs bellow.

$$\frac{\overline{\vdash 1}}{A \vdash 1} \quad \frac{\frac{A \vdash A \quad A \vdash A}{A, A \vdash A \otimes A}}{A \vdash A \otimes A}$$

The interpretation of the left proof should be a morphism $\epsilon_{\llbracket A \rrbracket} : \llbracket A \rrbracket \rightarrow 1$. The interpretation of the right proof should be a morphism $\Delta_{\llbracket A \rrbracket} : \llbracket A \rrbracket \rightarrow \llbracket A \rrbracket \otimes \llbracket A \rrbracket$. So if the rules hold for every formula, the tensor $\otimes$ would turn into a Cartesian product accordingly to the fact below.

Fact 1. *A symmetric monoidal category is Cartesian if and only if there exists two natural transformations $\Delta_X : X \rightarrow X \otimes X$ and $\epsilon_X : X \rightarrow 1$ that follow some commutations that I will not detail here (see prop.16 in section 6.3 of [12] for the complete statement)*

Sketch of the proof. For the direct implication, take $\epsilon_A := \mathbf{t}_A$ (recall that $\top$ is a final object) and $\Delta_A := \langle id_A, id_A \rangle$ (note that I introduced this morphism in 1.3 for interpreting the contraction rule). For the reverse implication, the pairing and the projections are defined as follow.

$$\begin{aligned} \langle f_1, f_2 \rangle : X &\xrightarrow{\Delta_X} X \otimes X \xrightarrow{f_1 \otimes f_2} Y_1 \otimes Y_2 \\ \mathbf{p}_1 : X_1 \otimes X_2 &\xrightarrow{id_{X_1} \otimes \mathbf{t}_{X_2}} X_1 \otimes 1 \xrightarrow{\lambda} X_1 \quad \text{and} \quad \mathbf{p}_2 : X_1 \otimes X_2 \xrightarrow{\mathbf{t}_{X_1} \otimes id_{X_2}} 1 \otimes X_2 \xrightarrow{\rho} X_2 \end{aligned}$$

□

However, linear logic is still supposed to be a refinement of classical logic (recall that $\mathbf{Lin}(!A, B) = \mathbf{Coh}(A, B)$). So Girard introduced a new unary connector $!$ called *the exponential* (that would be the syntactical equivalent of the functor $!$ in $\mathbf{Lin}$) and constrained the weakening and contraction to formulas of shape $!A$. So there are four new rules, a right rule for $!$ (called promotion), a left rule for $!$ (called dereliction), a weakening and a contraction.

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} (\text{der}) \quad \frac{!A_1, \dots, !A_n \vdash B}{!A_1, \dots, !A_n \vdash !B} (\text{prom}) \quad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} (\text{weak}) \quad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} (\text{contr})$$

One should see a formula $!A$ as a resource that can be “duplicated” and discarded”. The rule **(weak)** states that a resource $!A$ can be discarded. The rule **(contr)** states that a resource $!A$ can be duplicated. The rule **(der)** just “forget” that A was used once and only once. Finally, the rule **(prom)** states that if all of the resources used to produce B are discardable and duplicable, then B can be produced an arbitrary number of time (including zero) by discarding and duplicating the proof.

Besides, a reasoning similar to the one at the beginning of this section shows that the contraction and weakening rules implies the existence of two natural transformations $\mathbf{wk}_X : !X \rightarrow 1$ and $\mathbf{ctr}_X : !X \rightarrow !X \otimes !X$. Conversely, those two morphisms give an interpretation for the proofs that end with a weakening or a contraction rule.

2.4 The Seely isomorphisms

There are still many concurrent notions of models of linear logic, and choosing one is usually a matter of personal taste. I chose the notion of *Seely categories* introduced in [15]. This is not the most general one, but it is quite powerful and it arises in most of the concrete models. The starting point of this axiomatization is the observation made by Girard and deepened by Seely that in syntax as well as in many models, the connective $!$ “transports” the connectives $\&$ to the connectors $\otimes$.

Definition 15. A *Seely category* is a model in which the functor $!$ is a *strong symmetric monoidal functor* from $(\mathcal{L}, \&, \top)$ to $(\mathcal{L}, \otimes, 1)$. It means that there exists an isomorphism $\mathbf{m}^0 : 1 \simeq \top$ and a natural isomorphism $\mathbf{m}_{A,B}^2 : !A \otimes !B \simeq !(A \& B)$ ² called the *Seely isomorphisms* that follow the commutations below.

²Recall that $\&$ is the new name for $\times$, as stated in Definition 14

$$\begin{array}{ccccc}
(!A \otimes !B) \otimes !C & \xrightarrow{m^2 \otimes id} & !(A \& B) \otimes !C & \xrightarrow{m^2} & !((A \& B) \& C) & \quad & !A \otimes !B & \xrightarrow{m^2} & !(A \& B) \\
\alpha \downarrow & & & & \downarrow !\alpha^\& & & \gamma \downarrow & & \uparrow !\gamma^\& \\
!A \otimes (!B \otimes !C) & \xrightarrow{id \otimes m^2} & !A \otimes !(B \& C) & \xrightarrow{m^2} & !(A \& (B \& C)) & \quad & !B \otimes !A & \xrightarrow{m^2} & !(B \& A) \\
& & \uparrow !\rho^\& & & & & & \\
& & !A \otimes 1 & \xrightarrow{\rho} & !A & & 1 \otimes !A & \xrightarrow{\lambda} & !A \\
& & \downarrow id \otimes m^0 & & \uparrow !\rho^\& & & \downarrow m^0 \otimes id & & \uparrow !\lambda^\& \\
& & !A \otimes !\top & \xrightarrow{m^2} & !(A \& \top) & & !\top \otimes !A & \xrightarrow{m^2} & !(\top \& A)
\end{array}$$

Basically, those diagrams states that doing symmetric monoidal reasoning on $!(_ \& _)$ using the symmetric monoidal structure of $\&$ below $!$ is the same as doing symmetric monoidal reasoning on $!_ \otimes !_$ using the symmetric monoidal structure of $\otimes$.

Remark 4. Here, $!$ is a strong monoidal functor from $(\mathcal{L}, \otimes, 1)$ to $(\mathcal{L}, \&, \top)$ but this notion obviously extend to any functor F that goes from a symmetric monoidal category $(\mathcal{L}, \otimes, 1)$ to another symmetric monoidal category $(\mathcal{L}', \otimes', 1')$. Besides, when the two natural transformations are not isomorphisms, F is called a *lax symmetric monoidal functor*. We will come across this notion latter.

Remark 5. Any Seely category contains the two natural transformation $wk_A : !A \rightarrow 1$ and $ctr_A : !A \rightarrow !A \otimes !A$. They are obtained by “lifting” the two morphisms $\Delta_A^\& := \langle id_A, id_A \rangle : A \rightarrow A \& A$ and $\epsilon_A^\& := t_A : A \rightarrow \top$ that characterize $\&$ as a Cartesian product (recall [Fact 1](#)) through the functor $!$ and carry them to $\otimes$ with the Seely isomorphisms.

$$ctr_A : !A \xrightarrow{!\Delta_A^\&} !(A \& A) \xrightarrow{(m^2)^{-1}} !A \otimes !A \quad wk_A : !A \xrightarrow{!\epsilon_A^\&} !\top \xrightarrow{(m^0)^{-1}} 1$$

So Seely categories does not introduce those two morphisms directly as axioms, but derives them from other objects. This is an usual dynamic of the interaction between syntax and semantic: the axiomatization of the semantic is usually obtained by a direct correspondance with the syntactical rules, but sometimes it is better to divert from it a bit.

2.5 The structure of $!$

Fortunately, the structure that $!$ should have is more consensual. It should be a comonad, commonly called *the exponential comonad*.

Definition 16. A *comonad* on $\mathcal{L}$ consists in $(!, der, dig)$ where $!_ : \mathcal{L} \rightarrow \mathcal{L}$ is a functor together with two natural transformations $der_A : !A \rightarrow A$ and $dig_A : !A \rightarrow !!A$ such that the following diagrams commute.

$$\begin{array}{ccc}
!A & \xrightarrow{dig_A} & !!A \\
dig_A \downarrow & & \downarrow dig_{!A} \\
!!A & \xrightarrow{!dig_A} & !!!A
\end{array}
\quad
\begin{array}{ccc}
& & !A \\
id_{!A} \swarrow & & \searrow id_{!A} \\
!A & \xleftarrow{der_{!A}} & !!A \xrightarrow{!der_A} !A
\end{array}$$

The left diagram is called the Monadic square and the right diagram is called the monadic triangle.

Definition 17 (Dereliction). For any morphism $f : A \rightarrow B$, the morphism $K_1(f) : !A \xrightarrow{der_A} A \xrightarrow{f} B$ is called *the dereliction* of f . It gives an interpretation for the dereliction rule.

Definition 18 (Promotion). For any morphism $f : !A \rightarrow B$ the morphism $f^! : !A \xrightarrow{dig_A} !!A \xrightarrow{!f} !B$ is called the *promotion* of f . It gives an interpretation for the promotion rule when $n = 1$. There is a generalization of this construction for any n that can be derived from the Seely isomorphisms but I will not give the details here.

Putting everything together ends up with the following notion.

Definition 19 (Models of LL). The models of linear logic are *symetric monoidal* categories $(\mathcal{L}, \otimes, A)$, *closed* with regard to $\otimes$, that contains a *Cartesian product* $(\&, \top)$, that contains a *comonad* $(!, \text{der}, \text{dig})$ and two natural isomorphisms $\mathbf{m}^0 : 1 \simeq !\top$ and $\mathbf{m}^2_{A,B} : !A \otimes !B \simeq !(A \& B)$ such that $(!, \mathbf{m}^0, \mathbf{m}^2)$ is a *strong symmetric monoidal functor*.

Besides, a property similar to $\mathbf{Lin}(!A, B) = \mathbf{Coh}(A, B)$ holds in any model of linear logic. Indeed, the category $\mathbf{Coh}$ is a particular instance of the generic notion of *Kleisli category*.

Definition 20. If $(!, \text{der}, \text{dig})$ is a comonad on $\mathcal{L}$, the Kleisli category of $!$ is a category $\mathcal{L}_!$ with objects $\text{Obj}(\mathcal{L}_!) = \text{Obj}(\mathcal{L})$ and morphisms $\mathcal{L}_!(A, B) = \mathcal{L}(!A, B)$. The identity is defined from the dereliction as the morphism $\text{der}_A \in \mathcal{L}(!A, A) = \mathcal{L}_!(A, A)$. The composition is defined from the promotion: for any $f \in \mathcal{L}_!(A, B) = \mathcal{L}(!A, B)$ and $g \in \mathcal{L}_!(B, C) = \mathcal{L}(!B, C)$, $g \circ_! f := g \circ f^!$.

Proof. The proof that it is a category is left for the reader. The neutrality of the identity is a consequence of the triangle equality of the comonad. The associativity is a consequence of the square of the coMonad. $\square$

If the category $\mathcal{L}$ can be seen as the category of the “linear world”, the category $\mathcal{L}_!$ on the other hand can be seen as an extension of $\mathcal{L}$ that introduces a way to produce “non linearity” (by allowing to duplicate the resources). The *dereliction* introduced in [Definition 18](#) allows to “forget” that a given morphism is linear.

Proposition 1. The category $\mathcal{L}_!$ extends $\mathcal{L}$ in the sense that the dereliction $\mathcal{K}_!(f)$ induces a functor $\mathcal{K}_! : \mathcal{L} \rightarrow \mathcal{L}_!$ by defining $\mathcal{K}_!(A) := A$. Besides, this functor is faithful, meaning that if $\mathcal{K}_!(f) = \mathcal{K}_!(g)$, then $f = g$.

Theorem 2. The Kleisli category $\mathcal{L}_!$ is cartesian closed, and thus is a model of LJ.

Proof. I will not detail the proof here, but I will give the important ideas. The final object is still $\top$ and the Cartesian product of A with B is still the object $A \& B$. The projections are defined as $\mathcal{K}_!(\mathbf{p}_1) : !(A \& B) \rightarrow A$ and $\mathcal{K}_!(\mathbf{p}_2) : !(A \& B) \rightarrow B$. The closure then comes from the fact that we can transport the closure in $\mathcal{L}$ with regard to $\otimes$ to a closure in $\mathcal{L}_M$ with regard to $\&$ thanks to the Seely isomorphism $\mathbf{m}^2_{X,Y} : !X \otimes !Y \rightarrow !(X \& Y)$. $\square$

Remark 6. Actually, models of linear logic require one more compatibility condition between $\mathbf{m}^2$ and dig that is necessary in the proof of the closure, but I will not talk about it here.

3 Differentiation in some models of linear logic

As hinted before with **Vect**, the ideas of linear logic have deep ties with linear algebra. However there is no satisfactory way of defining $(!, \text{der}, \text{dig})$ on **Vect**, so interpreting the exponential through the lenses of linear algebra seems at first to be a lost cause. Fortunately, Ehrhard discovered a model of LL called **Fin** (*finiteness spaces*) [5] in which the objects are (topological) vector spaces and the arrows are the (continuous) linear maps. In this model, the exponential can be seen as a power series construction: $\mathbf{Fin}(!X, Y)$ is the set of *analytic* functions from X to Y . For example, a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is analytic if $f(x) = \sum_{n=0}^{\infty} a_n x^n$.

Analytic functions are a particular case of smooth functions. Recall the fundamental idea of differential calculus: for any $x \in X$, the variation a smooth function $f : X \rightarrow Y$ around x can be approximated as a linear variation $f(x+u) \approx f(x) + f'(x).u$, where $f'(x)$ is a (continuous) linear map called the differential of f in x . So there exists a function $f' : X \rightarrow \mathbf{Fin}(X, Y)$. Besides, one can show that when f is analytic f' is also analytic, meaning that $f' \in \mathbf{Fin}(!X, \mathbf{Fin}(X, Y))$. Because of the closure with regard to $\otimes$, it means that the differential of f can be seen as a morphism $f' \in \mathbf{Fin}(!X \otimes X, Y)$. So the fundamental idea of differential calculus seems to corresponds to some categorical properties on **Fin**. Here are those properties, defined in a generic way.

Definition 21. A model of LL $\mathcal{L}$ is said to be *additive* if for all objects X, Y , $\mathcal{L}(X, Y)$ is a monoid. In other word, if there exists a morphism $0 \in \mathcal{L}(X, Y)$ and an associative operator $+$ such that for all $f \in \mathcal{L}(X, Y)$, $0+f = f+0 = f$. Besides, the addition should be *compatible* with the composition and the tensor, meaning:

- $g \circ (f_1 + f_2) \circ h = g \circ f_1 \circ h + g \circ f_2 \circ h$. In other words, the morphisms are “linear” in the sum

- $(f_1 + f_2) \otimes (g_1 \otimes g_2) = f_1 \otimes g_1 + f_2 \otimes g_1 + f_1 \otimes g_2 + f_2 \otimes g_2$. In other word, the tensor is “bilinear” in the sum.

Definition 22. A *differential category* is an additive category $\mathcal{L}$ such that there exists a natural transformation $d : !X \otimes X \rightarrow !X$. This natural transformation allows to define for any $f : !X \rightarrow Y$ a morphism $f' := f \circ d : !X \otimes X \rightarrow Y$. Besides, d is required to follow some commutations that consist in separate interactions between d and $\text{wk}/\text{ctr}/\text{der}/\text{dig}/d$. I will not give the diagrams, but explain to what they correspond in finiteness spaces

- Interaction with der : The derivation of a linear function is the linear function itself
- Interaction with dig : The chain rule holds, that is $(f \circ g)'(x) = f'(g(x)) \circ g'(x)$
- Interaction with wk : The derivative of a constant function is 0
- Interaction with ctr : The Leibniz rule holds, that is if ϕ is a bilinear application, $\phi(f, g)'(x).u = \phi(f'(x).u, g(x)) + \phi(f(x), g'(x).u)$.
- Interaction with d : The Schwarz rule, that is $f''(x).(u, v) = f''(x).(v, u)$

Thus the history of linear logic repeated itself. Ehrhard and Regnier [6] designed a logic called *differential linear logic* whose models should be differential categories. We saw in Section 2.4 that the presentation of the semantic might sometimes deviate a bit from the syntax. Something quite similar happens here. Similarly to how the product is lifted through $!$ in Remark 5, the sum (that induces a structure called a *coproduct*) can be lifted through $!$ to produce two natural transformation $\overline{\text{wk}}_X : 1 \rightarrow !X$ and $\overline{\text{ctr}}_X : !X \otimes !X \rightarrow !X$.

$$\overline{\text{ctr}}_A : !A \otimes !A \xrightarrow{m^2} !(A \& A) \xrightarrow{!(p_1 + p_2)} !A \quad \overline{\text{wk}}_A : 1 \xrightarrow{m^0} !\top \xrightarrow{!0} !X$$

Besides, in most differential categories the natural transformation d can be derived from a natural transformation $\overline{\text{der}} : X \rightarrow !X$ as follows.

$$d_X : !X \otimes X \xrightarrow{id_{!X} \otimes \overline{\text{der}}} !X \otimes !X \xrightarrow{\overline{\text{ctr}}} !X$$

The transformation $\overline{\text{der}}$ can be seen as an operator that computes the derivative in 0, $f \circ \overline{\text{der}} = f'(0)$. The construction above thus corresponds to the idea that the derivative of f in x can be computed as the derivative in zero of the function $y \mapsto f(x + y)$. Again, there are many flavors of differential categories, see [3] for a survey.

Differential logic is a syntactical extension of linear logic that introduces three new rules that account for those morphisms: a co-weakening (that makes an empty proof), a co-contraction (that sums two proofs) and a co-dereliction (that “derives” a proof in 0).

$$(\text{co-weak}) \frac{}{\vdash !A} \quad (\text{co-contr}) \frac{\Gamma_1 \vdash !A \quad \Gamma_2 \vdash !A}{\Gamma_1, \Gamma_2 \vdash !A} \quad (\text{co-der}) \frac{\Gamma \vdash A}{\Gamma \vdash !A}$$

This syntactical notion of derivation has one primary use called the Taylor expansion. Similarly to how an analytic function (hence a morphism in $\mathbf{Fin}(!A, B)$) can be described by its successive derivatives in 0, the operational behaviour of a proof using the promotion rule can be syntactically described by taking its successive co-derelictions. It allows to talk about the *ressource sensitivity* of a proof, that is, how a proof can be “syntactically approximated” by a proof that uses the formulas less than a given amount of time. This notion nicely relates to *Böhm trees*, a notion in denotational semantic that aims at describing the limit behaviour of a non terminating program. See [1] for an enlightening discussion on this topic.

The issue of differential logic though is that the notion of sum is closely tied to non determinism. At first glance, the cut elimination equivalent of the Leibniz rule and the constant rule requires to introduce two new rules.

$$(0) \frac{}{\Gamma \vdash B} \quad (\text{sum}) \frac{\Gamma \vdash B \quad \Gamma \vdash B}{\Gamma \vdash B}$$

The first rule introduces a zero proof, the second rule allows to sum two proofs. The first rules makes the logic highly inconsistent as every sequent becomes provable, but it is in fact not that much of an issue because differential logic is more interested in computational properties rather than provability. The second

rule however has a lot of repercussions, as it implies that the proof system allows for a non deterministic branching on two possible paths. For example, assume that some formula A models the boolean, in the sense that the sequent $\vdash A$ only admits two proofs in linear logic, a proof *true* and a proof *false*³. In differential logic, this sequent admits another proof.

$$\frac{\frac{true}{\vdash A} \quad \frac{false}{\vdash A}}{\vdash A}$$

whose semantic would be $\llbracket true \rrbracket + \llbracket false \rrbracket$, a value that should be neither $\llbracket true \rrbracket$ neither $\llbracket false \rrbracket$ but rather a superposition of those two states.

4 Coherent differentiation

In this paper [4], my supervisor Thomas Ehrhard introduced a new notion called *coherent differentiation*. The cornerstone of coherent differentiation is to generalize differentiation to models that are not always additive categories. The motivation behind this idea is twofold. Firstly, Ehrhard noticed that some models such as *probabilistic coherent spaces* were not additive categories yet still admitted a notion of differentiation. Secondly, restraining the sum under some compatibility conditions can hopefully make the system deterministic again. For example, there is no morphism such as $\llbracket true \rrbracket + \llbracket false \rrbracket$ in probabilistic coherent spaces.

4.1 Pre-summability structure

In order to restrict the sum, Ehrhard introduced a notion called a *pre-summability structure*. Let $\mathcal{L}$ be a model of linear logic such that for any objects X, Y , there exists a morphism $0_{X,Y} \in \mathcal{L}(X, Y)$. By abuse of notation, I will usually keep X and Y implicit and write only 0 . Besides, from now on I will start my indices from 0 in order to be consistent with the notations of the article⁴.

Definition 23. A *pre-summability structure* on $\mathcal{L}$ is a tuple $(S, \pi_0, \pi_1, \sigma)$ such that:

- S is an endofunctor $S : \mathcal{L} \rightarrow \mathcal{L}$ that verifies $S(0_{X,Y}) = 0_{SX,SY}$ for all objects X, Y .
- $\pi_0, \pi_1, \sigma : S \Rightarrow Id$ are natural transformations
- π_0, π_1 are jointly monic: if $f, g \in \mathcal{L}(Y, SX)$, $\left\{ \begin{array}{l} \pi_0 \circ f = \pi_1 \circ g \\ \pi_1 \circ f = \pi_1 \circ g \end{array} \right\} \Rightarrow f = g$

The joint monicity of π_0, π_1 means that S is a kind of product: a function $f \in \mathcal{L}(Y, SX)$ is completely characterized by the two morphisms $\pi_0 \circ f$ and $\pi_1 \circ f$. Note however that contrary to Cartesian product, the pairing of two morphisms f_0 and f_1 has no reason to be always defined. When such a pairing exist, we say that f_0 and f_1 are *summable*

Definition 24 (Sumability). (f_0, f_1) are *summable* if there exists a pairing h such that $\left\{ \begin{array}{l} \pi_0 \circ h = f_0 \\ \pi_1 \circ h = f_1 \end{array} \right.$.

As said before when such pairing exists, it is unique. It is written $\langle f_0, f_1 \rangle_S$ and is called the *witness* of the sum. The *sum* is then defined as $f_0 + f_1 := \sigma \langle f_0, f_1 \rangle_S$.

Remark 7. Note that by definition of $\langle f_0, f_1 \rangle_S$, $\pi_i \circ \langle f_0, f_1 \rangle_S = f_i$ and $\langle f_0, f_1 \rangle_S \circ g = \langle f_0 \circ g, f_1 \circ g \rangle_S$. Besides, it is easy to check that π_0, π_1 are summable of witness *id* and sum σ .

The naturality of π_0, π_1 and σ express that the sum is compatible with the composition, in a somewhat similar way to additive categories.

³In fact $A = 1 \oplus 1$ where $\oplus$ is the disjunction associated to the conjunction $\&$. Since I did not talk about disjunction so I will not give further details

⁴Those notations were subject to debate. My supervisor admitted that he took a bad habit by starting indices by 0. On the other hand, these indices make sense in summability structures because the left term is usually though as a “zero order term” while the right term is though as a “one order term”, as we will see in [Sections 4.3 and 4.4](#)

- The naturality of π_0 and π_1 express that $Sf : SX \rightarrow SY$ consists in “applying f coordinate by coordinate”: $Sf \circ \langle g_0, g_1 \rangle_S = \langle f \circ g_0, f \circ g_1 \rangle_S$.
- A consequence of this and [Remark 7](#) is that if f_0, f_1 are summable, $h \circ f_0 \circ g, h \circ f_1 \circ g$ are summable. The witness for this sum is $Sh \circ \langle f_0, f_1 \rangle_S \circ g$.
- The naturality of σ then ensures that $h \circ f_0 \circ g + h \circ f_1 \circ g = h \circ (f_0 + f_1) \circ g$.

Result 1. Let us anticipate a bit over [Section 5](#). This notion of summability generalizes the notion of sum in additive categories. Indeed, let $\mathcal{L}$ be an additive category. Let $S_{\&} : \mathcal{L} \rightarrow \mathcal{L}$ be the functor defined as $S_{\&}X := X \& X$ and $S_{\&}f := f \& f$. This functor can be equipped with the pre-summability structure in which the projections are the projections associated to the cartesian product $\pi_i := \mathbf{p}_i : S_{\&}X \rightarrow X$ and the sum is defined as $\sigma := \mathbf{p}_0 + \mathbf{p}_1 : S_{\&}^2 X \rightarrow S_{\&}X$. The projections are jointly monic by unicity of the pairing in a Cartesian product. Finally, $\forall i \in \{0, 1\}, \mathbf{p}_i \circ S_{\&}(0) = 0 = \mathbf{p}_i \circ 0$ so $S(0) = 0$ by joint monicity of the projections.

The sum induced by this presummability structure coincides with the sum of the additive category. Indeed, any pair of morphisms $f_0, f_1 : X \rightarrow Y$ are summable (in the sense of the pre-summability structure) with witness $\langle f_0, f_1 \rangle$ and sum $\sigma \circ \langle f_0, f_1 \rangle := (\mathbf{p}_0 + \mathbf{p}_1) \circ \langle f_0, f_1 \rangle = \mathbf{p}_0 \circ \langle f_0, f_1 \rangle + \mathbf{p}_1 \circ \langle f_0, f_1 \rangle = f_0 + f_1$.

4.2 Summability structure

The (partial) sum is still quite far from behaving like a sum though, some more axioms are required. The right axiomatic seems to be the notion of *partial commutative monoid*.

Definition 25. The sum is called a *partial commutative monoid* if it follows those properties

- Commutativity: For any morphisms f, g such that (f, g) are summable, (g, f) are summable and $g + f = f + g$
- Neutrality of 0: For any morphism, $(0, f)$ and $(f, 0)$ are summable and $f + 0 = 0 + f = f$
- Associativity: For any morphisms f, g, h , if f, g and $(f + g), h$ are summable, then g, h are summable, $f, (g + h)$ are summable, and $f + (g + h) = (f + g) + h$

A presummability structure whose notion of sum is a partial commutative monoid is called a *summability structure*.

Result 2. The presummability structure $S_{\&}$ defined in [Result 1](#) is clearly a summability structure since partial commutative monoids are without a doubt a generalization of commutative monoids.

Interestingly, commutativity implies the existence of a natural transformation $\gamma = \langle \pi_1, \pi_0 \rangle_S : SX \rightarrow SX$ and the neutrality of 0 implies the existence of two natural transformations $\iota_0 = \langle id, 0 \rangle_S : X \rightarrow SX$ and $\iota_1 = \langle 0, id \rangle_S : X \rightarrow SX$. In fact, Ehrhard shows that those two implications are actually equivalence, but I will not give the proof here. Associativity on the other hand can be reframed as the two following properties.

Definition 26. We call (S-witness) the following property: for any X, Y and $f, g : X \rightarrow SY$, if $\sigma \circ f$ and $\sigma \circ g$ are summable, then f, g are summable.

Definition 27. We call (S-assoc) the following property: assume that $(\langle f_0, f_1 \rangle_S, \langle g_0, g_1 \rangle_S)$ are well defined and summable. Then $\forall i \in \{0, 1\}, (f_i, g_i)$ are summable, $(f_0 + g_0, f_1 + g_1)$ are summable, and $\langle f_0, f_1 \rangle_S + \langle g_0, g_1 \rangle_S = \langle f_0 + g_0, f_1 + g_1 \rangle_S$. In other words: “summing two (summable) pairs consists in summing coordinates by coordinates”.

Proposition 2. If 0 is a neutral element, (S-witness) holds and (S-assoc) holds, the sum is associative.

I spent some amount of time during my internship to try to re frame associativity in a more structural way, in the vein of neutrality of 0 and commutativity. Doing so might gives some insights on the syntactical equivalent of summability structures. However the property (S-witness) is much subtler than it seems and I start to think that it is by nature quite “un-structural”. In fact, I think that partial commutative monoids might be a notion that is a bit too restrictive. Indeed, any major notion of coherent differentiation can be defined in a weaker setting, where (S-witness) is replaced by the existence of three operators. I will not give more details as this observation is still an early development and has no strong semantical nor syntactical backing yet. One result that I noticed in this process though is that (S-assoc) always hold.

Result 3. Any pre-summability structure fulfills (S-*assoc*).

Proof. By compatibility of addition with regard to composition, $(\langle f_0, f_1 \rangle_S, \langle g_0, g_1 \rangle_S)$ summable implies that $(\pi_i \circ \langle f_0, f_1 \rangle_S, \pi_i \circ \langle g_0, g_1 \rangle_S)$ are summable (in other word, (f_i, g_i) are summable). Compatibility also ensures that $\pi_i \circ (\langle f_0, f_1 \rangle_S + \langle g_0, g_1 \rangle_S) = (\pi_i \circ \langle f_0, f_1 \rangle_S) + ((\pi_i \circ \langle g_0, g_1 \rangle_S)) = f_i + g_i$. By definition of summability, it implies that $(f_0 + g_0, f_1 + g_1)$ are summable of witness $\langle f_0, f_1 \rangle_S + \langle g_0, g_1 \rangle_S$. It concludes the proof. $\square$

4.3 The differentiation operator

Erhard introduced coherent differentiation as a natural transformation $\partial_X : !SX \rightarrow S!X$. It allows to define for any $f : !X \rightarrow Y$ (e.g a smooth function) a morphism. $Df : !SX \xrightarrow{\partial_X} S!X \xrightarrow{Sf} SY$. Intuitively, Df is interpreted as the function that takes as input two summable elements (x, u) and return the first order development of f on x for the variation u : $Df(x, u) = (f(x), f'(x).u)$. More formally, Erhard asks ∂ to follow the rule below

$$(\partial\text{-local}) \quad \begin{array}{ccc} !SX & \xrightarrow{\partial_X} & S!X \\ & \searrow \scriptstyle !\pi_0 & \downarrow \scriptstyle \pi_0 \\ & & !X \end{array}$$

This rule basically states that $\partial_X = \langle !\pi_0, \mathbf{d}_X^{\text{Coh}} \rangle_S$ for some natural transformation $\mathbf{d}_X^{\text{Coh}} : !SX \rightarrow !X$. So it provides two informations:

- There is a *derivation* $\mathbf{d}_X^{\text{Coh}} : !SX \rightarrow !X$. Intuitively, $f \circ \mathbf{d}^{\text{Coh}} : !SX \rightarrow Y$ takes as input a summable pair (x, u) and returns $f'(x).u$.
- $Df = Sf \circ \partial_X = \langle f \circ !\pi_0, f \circ \mathbf{d}_X^{\text{Coh}} \rangle_S$, which rewrites intuitively as $Df(x, u) = (f(x), f'(x).u)$. So the existence of ∂ not only ensures that the derivative $f'(x).u$ is defined as soon as (x, u) is summable, but it also ensures that $(f(x), f'(x).u)$ is summable, implying that $f(x) + f'(x).u$ is defined.

Result 4. Recall that differential categories are additive categories, and thus admit a summability structure that coincides with the sum, defined in [Results 1](#) and [2](#). Then the operator $\mathbf{d} : !X \otimes X \rightarrow X$ induces a derivation for this summability structure.

$$\mathbf{d}_X^{\text{Coh}} : !S_{\&X} \xrightarrow{(\mathbf{m}^2)^{-1}} !X \otimes !X \xrightarrow{id \otimes \text{der}_X} !X \otimes X \xrightarrow{\mathbf{d}_X} !X$$

It means that a coherent differentiation can be defined as the pairing $\partial_X = \langle !\mathbf{p}_0, \mathbf{d}^{\text{Coh}} \rangle$. In particular, $(\partial\text{-local})$ holds by definition.

4.4 The monad structure on S

Let us start with a practical reasoning that is fundamental in differential calculus. Take two functions f and g with first order development. Those development can be multiplied.

$$f(x + u).g(y + v) = (f(x) + f'(x).u + o(u)).(g(y) + g'(y).v + o(v)) \quad (1)$$

$$= f(x)g(y) + f(x)g'(y).v + g(y)f'(x).u + (f'(x).u)(g'(y).v) + o(u) + o(v) \quad (2)$$

$$\approx f(x)g(y) + v.f(x)g'(y) + u.f'(x)g(y) \quad (3)$$

The term $(f'(x).u)(g'(y).v)$ is of second order so it can be neglected. This computation thus shows that the first order variation of $f.g$ around x is $f(x)g'(y).v + g(y)f'(x).u$: this is the so called *Leibniz rule*. Interestingly, the proof above only uses a kind of algebraic reasoning, not how the derivative itself is defined. It means that this reasoning can carry to the more abstract framework of summability structure: the development of $f \otimes g : !(X_1 \& X_2) \simeq !X_1 \otimes !X_2 \rightarrow Y_1 \otimes Y_2$ can be inferred from the development of f and the development of g , using three categorical constructions:

- A natural transformations $\Phi_{X,Y}^0 := \langle id \otimes \pi_0, id \otimes \pi_1 \rangle_S : X \otimes SY \rightarrow S(X \otimes Y)$ that “distributes on the right pair” and a natural transformation $\Phi_{X,Y}^1 := \langle \pi_0 \otimes id, \pi_1 \otimes id \rangle_S : SX \otimes Y \rightarrow S(X \otimes Y)$ that “distributes on the left pair”. Those two operations are well defined assuming an axiom called ($S \otimes$ -dist) that states the compatibility of the sum with regard to the tensor, in a similar way as in additive categories.
- A natural transformation $\tau_X := \langle \pi_0 \circ \pi_0, \pi_0 \circ \pi_1 + \pi_1 \circ \pi_0 \rangle_S : S^2X \rightarrow SX$ that keeps the constant factor on the left and sum the two order 1 factors on the right. I will not give the details but we can show that it is well defined using associativity and compatibility of addition with regard to composition.

Those constructions allow to define $L : SX \otimes SY \xrightarrow{\Phi^1} S(X \otimes SY) \xrightarrow{S\Phi^0} S^2(X \otimes Y) \xrightarrow{\tau} S(X \otimes Y)$ that “double distributes the two pairs and stash the order two term” (note that Φ^0 could have been applied before Φ^1 for the same outcome). Then the reasoning described above is framed abstractly by the following categorical diagram.

$$\begin{array}{ccc}
!SX_0 \otimes !SX_1 & \xleftarrow{(m^2)^{-1}} & !(SX_0 \& SX_1) \xleftarrow{!(Sp_0, Sp_1)_S} !S(X_0 \& X_1) \\
Df \otimes Dg \downarrow & & \downarrow D(f \otimes g) \\
SY_0 \otimes SY_1 & \xrightarrow{\quad L \quad} & S(Y_0 \otimes Y_1)
\end{array}$$

We can show that $\iota_0 : X \rightarrow SX$ and $\tau_X : S^2X \rightarrow SX$ gives to S the structure of a *Monad*. Monads are the dual notion of comonads and are defined as below.

Definition 28. A monad (M, η, μ) on $\mathcal{L}$ is a functor $M : \mathcal{L} \rightarrow \mathcal{L}$ together with two natural transformations $\eta_A : A \rightarrow MA$ and $\mu_A : M^2A \rightarrow MA$ such that the following diagrams commute.

$$\begin{array}{ccc}
M^3A & \xrightarrow{M\mu_A} & M^2A \\
\mu_{MA} \downarrow & & \downarrow \mu_A \\
M^2A & \xrightarrow{\mu_A} & MA
\end{array}
\quad
\begin{array}{ccc}
MA & \xrightarrow{\eta_{MA}} & M^2A \xleftarrow{M\eta_A} MA \\
& \searrow & \downarrow \mu_A \swarrow \\
& & MA
\end{array}$$

Theorem 3. (S, ι_0, τ) is a monad.

Definition 29. Similarly to comonads, any monad M on $\mathcal{L}$ induces a category $\mathcal{L}_M$ called the *Kleisli category of the monad* where $\mathcal{L}_M(X, Y) = \mathcal{L}(X, MY)$ (same as for comonads but M is on the right). There is also a faithful functor $K_M : \mathcal{L} \rightarrow \mathcal{L}_M$. I will not give the exact constructions as they are quite similar to the constructions of the comonads.

Those categories are extensively studied in semantic. The reason is that they model *effects*. For example, the **Option** type in programming language (an element of **Option** X is either **Some**(x) where $x \in X$ or **None**) is a monad. The Kleisli category of this monad models computations that can either terminate correctly (by outputting **Some**(*result*)) or raise an exception (by outputting **None**). There are also monads that models memory, non determinism, randomness, etc... It suggests that differentiation might be an effect too. As a result, most of the last part of the internship was centered around the study of $\mathcal{L}_S$, see [Section 6](#).

4.5 The axioms of coherent differentiation

Those diagrams turned out to corresponds to the elementary properties that derivation should follows, and their interpretation coincides with the interpretation of the diagram required in DiLL. I will not give the diagram themselves, rather their interpretation.

- (∂ -lin) Two axioms that give the interactions between ∂ and ι_0/τ . They state that the derivative of a function is linear. The first one (∂ -lin-1) ensures that $f'(x).0 = 0$ and the second one (∂ -lin-2) ensures that $f'(x).(u_1 + u_2) = f'(x).u_1 + f'(x).u_2$.
- (∂ -chain) Two axioms that give the interactions between ∂ and **der**/**dig**. The axiom (∂ -chain-1) states that the derivative of a linear function is the function itself. The axiom (∂ -chain-2) states the chain rule.

(∂ -Leibniz) Two axioms that give the interaction between ∂ and wk/ctr . The axiom (∂ -Leibniz-1) states that the derivative of a constant function is null. The axiom (∂ -Leibniz-2) is the Leibniz rule, and is equivalent to diagram motivated in [Section 4.4](#).

(∂ -Schwarz) One axiom that gives the interaction of ∂ with itself. This axiom states the Schwarz rule.

Just as an example, here is the axiom (∂ -lin).

$$\begin{array}{ccc}
 \text{(!}SX \xrightarrow{\partial_X} S!X) & \text{(!}SX \xrightarrow{\partial_X} S!X) & \\
 \text{(\partial-lin)} \quad \text{der}_{SX} \searrow & \text{der}_{SX} \searrow & \text{S}f \circ \text{Sder}_X \searrow \\
 & SX & SX \xrightarrow{\text{S}f} SY
 \end{array}$$

How does this rule says anything about the derivative of a linear function ? Let us plug a morphism of the shape Sf at the end of the diagram. Then by functoriality of S , $Sf \circ Sder = S(f \circ der)$.

Recall that if $f : X \rightarrow Y$, then $f \circ der : !X \rightarrow Y$ can be interpreted as the linear function that maps an element $x \in X$ to $f(x) \in Y$. Thus, thanks to our interpretation of ∂ , the the top path of the diagram $S(f \circ der) \circ \partial_X$ consists in a morphism that maps a pair (x, u) to the first order development $(f(x), f'(x).u)$. On the other hand, the bottom path $Sf \circ der_{SX}$ consists in the linear function that maps a pair (x, u) to the pair $(f(x), f(u))$. So the diagram above interprets as: “if f is a linear function, for any $x, u \in X$ that are summable, $f'(x).u = f.u$ ”.

5 Differentiation in differential linear logic is a particular case of coherent differentiation

The first objective of the internship was to show that the structure of a differential categories induces a structure of coherent differentiation. The goal behind this was threefold. Firstly, it is a nice starting point for an internship, as it allowed me to hone my understanding of both models of differential linear logic and coherent differentiation. Secondly, it ensures that coherent differentiation is a generalization of differential categories, not a divergent notion. Finally, it implies that understanding coherent differentiation might bring some insights to differential categories too.

I already showed that the sum in additive categories induce a summability structure (see [Results 1 and 2](#)) and that any differential operator $d : !X \otimes X \rightarrow X$ induces a candidate for coherent differentiation (see [Result 4](#)). What remains is to check the various axioms of coherent differentiation. Unsurprisingly, each axiom on ∂ ends up being a consequence of its counterpart on d . For example, the axiom (∂ -chain-1) that states that “the derivative of a linear function is the function itself” is a consequence of the interaction between d and der that states the same idea. The only exceptions are the two the axioms (∂ -lin) that have a more structural counterpart expressed in the fact that $\mathbf{d}^{\text{Coh}}$ is defined from d with a dereliction (so, by forgetting the linearity in the second coordinate).

The proof is conceptually not that hard, but many technicalities arose. Consequently, the full redacted proof is 25 pages long. Obviously, I will not write everything down here. I will rather exhibit the proof structure through the example of (∂ -chain-1), as this rule is probably the easiest one that still contains some structural arguments. The first crucial step was to show that this rule was equivalent to a rule about $\mathbf{d}^{\text{Coh}}$.

Result 5. For any summability structure and morphism ∂ that follows (∂ -local), the diagram (∂ -lin) is equivalent to

$$\begin{array}{ccc}
 !SX & \xrightarrow{\mathbf{d}_X^{\text{Coh}}} & !X \\
 !\pi_1 \downarrow & & \downarrow der_X \\
 !X & \xrightarrow{der_X} & X
 \end{array}$$

Proof. By joint monicity of π_0, π_1 , the diagram (∂ -lin) holds if and only if the diagram below on the left holds for any $i \in \{0, 1\}$. This diagram admits the diagram chase on the right.

$$\begin{array}{ccccc}
!SX & \xrightarrow{\partial_X} & S!X & & !SX \xrightarrow{\partial_X} S!X \\
\downarrow \text{der}_{SX} & & \downarrow \text{Sder}_X & & \downarrow \text{der}_{SX} \quad \downarrow \text{Sder}_X \\
SX & \xrightarrow{\pi_i} X \xleftarrow{\pi_i} & SX & & SX \xrightarrow{\pi_i} X \xleftarrow{\pi_i} SX \\
& & & & \downarrow \text{der}_X \quad \downarrow \text{der}_X \\
& & & & X \xleftarrow{\pi_i} X \xleftarrow{\pi_i} X
\end{array}$$

The commutation (a) is the naturality of der and the commutation (b) is the naturality of π_i . This reduced diagram trivially holds for $i = 0$ using that $\pi_0 \circ \partial = !\pi_0$. The induced diagram for $i = 1$ on the other hand is exactly the diagram of the result, using that $\pi_1 \circ \partial = \mathbf{d}^{\text{Coh}}$. $\square$

This first step is crucial, because this diagram is already much closer to its counterpart on $\mathbf{d}$. Here is a side by side comparison.

$$\begin{array}{ccc}
!X \& X & \xrightarrow{\mathbf{d}_X^{\text{Coh}}} & !X \\
\downarrow !p_1 & & \downarrow \text{der}_X & \\
!X & \xrightarrow{\text{der}_X} & X & \\
& & & \\
!X \otimes X & \xrightarrow{\mathbf{d}_X} & !X \\
\downarrow \text{wk}_X \otimes \text{id}_X & & \downarrow \text{der}_X \\
1 \otimes X & \xrightarrow{\lambda} & X
\end{array}$$

The biggest difference is that the “structural work” is done on $\&$ on the left and on $\otimes$ on the right. Fortunately, recall that the Seely isomorphisms relate those two structures.

Result 6 (Rewriting the projections). The following diagrams commute

$$\begin{array}{ccc}
!(X \& Y) & \xrightarrow{!p_0} & !X \\
(m^2)^{-1} \downarrow & & \uparrow \rho \\
!X \& !Y & \xrightarrow{\text{id}_{!X} \otimes \text{wk}_Y} & !X \otimes 1
\end{array}
\quad
\begin{array}{ccc}
!(X \& Y) & \xrightarrow{!p_1} & !Y \\
(m^2)^{-1} \downarrow & & \uparrow \lambda \\
!X \otimes !Y & \xrightarrow{\text{wk}_X \otimes \text{id}_{!Y}} & 1 \otimes !Y
\end{array}$$

Proof. I do the proof for the second diagram. Unfolding the definition of the weakening (recall [Remark 5](#)) leads up to the following diagram chase.

$$\begin{array}{ccccc}
!(X \& Y) & \xrightarrow{p_1} & !Y & & \\
\downarrow (m^2)^{-1} & \searrow !(\epsilon_X \& \text{id}_{!Y}) & \nearrow !p_1 & & \\
& & !(\top \& Y) & & \\
& & \downarrow (m^2)^{-1} & & \\
!X \otimes !Y & \xrightarrow{! \epsilon_X \otimes \text{id}_{!Y}} & !\top \otimes !Y & \xrightarrow{m^0 \circ 1} & 1 \otimes !Y
\end{array}$$

The commutation (a) is the naturality of m^2 , the commutation (b) is one of the diagrams that ensures that $(!, m^0, m^2)$ is a symmetric monoidal functor. The triangle at the top can be directly computed. $\square$

We now have all of the ingredients necessary to prove that $(\partial\text{-chain-1})$ holds.

Result 7. The diagram of [Result 5](#) commutes, so $(\partial\text{-chain-1})$ holds.

Proof. We perform the following diagram chase.

$$\begin{array}{ccccccc}
!(X \& X) & \xrightarrow{(m^2)^{-1}} & !X \otimes !X & \xrightarrow{!X \otimes \text{der}_X} & !X \otimes X & \xrightarrow{\mathbf{d}_X} & !X \\
\downarrow !p_1 & & \downarrow \text{wk}_X \otimes \text{id}_{!X} & & \downarrow \text{wk}_X \otimes \text{id}_X & & \downarrow \text{der}_X \\
!X & \xrightarrow{\lambda^{-1}} & 1 \otimes !X & \xrightarrow{\text{id}_1 \otimes \text{der}_X} & 1 \otimes X & \xrightarrow{\lambda} & X
\end{array}$$

The commutation (a) comes from [Result 6](#), the commutation (b) is the functoriality of $\otimes$, and the commutation (c) is the linear rule in models of differential linear logic. $\square$

To sum up, I proved that the different axioms of coherent differentiation hold by using the following proof method.

- I proved that all of the axioms of coherent differentiation are equivalent to axioms on the derivation operator $\mathbf{d}^{\text{Coh}}$ that is introduced by $(\partial\text{-local})$, as in [Result 5](#)
- Those axioms are always similar to their counterpart in differential categories, except that the structural work is done on $\&$ rather than on $\otimes$. Fortunately I showed that the properties of the Seely isomorphisms stated in [Definition 15](#) close the bridge between those two worlds, as in [Result 6](#).
- I wrapped everything up using basic properties such as naturality and functoriality, as in [Result 7](#).

I cannot resist to give one last structural commutation as this one is particularly insightful on the fact that the Seely isomorphisms lift the structure of the Cartesian product $\&$ under $!$ to give to the tensor $\otimes$ a structure similar to a Cartesian product on the objects of shape $!X$, accordingly to [Fact 1](#) and [Remark 5](#).

$$\begin{array}{ccc} !X & \xrightarrow{\text{ctr}_X} & !X \otimes !X \\ \downarrow \langle f_0, f_1 \rangle & & \downarrow f_0 \otimes f_1 \\ !(Y_0 \& Y_1) & \xrightarrow{(\mathbf{m}^2)^{-1}} & !Y_1 \otimes !Y_1 \end{array}$$

6 Distributive laws, distributive laws everywhere

In his paper, Ehrhard noticed that all of the concrete instances of summability structure he found in the various models consisted in a functor $\mathbf{S} = I \multimap _$ where $I = 1 \& 1$. The monads of this shape are said to be “right adjoints” to a comonad of shape $\bar{\mathbf{S}} = _ \otimes I$, a relation written $\bar{\mathbf{S}} \dashv \mathbf{S}$ that I will not detail here. Ehrhard then showed that this relation relates coherent differentiation on $\mathbf{S}$ to what is called a coalgebra on I , that is a morphism $\delta : I \rightarrow !I$ with some properties. It is somewhat folklore in the field that under those conditions, the Kleisli category of $\bar{\mathbf{S}}$ are still models of linear logic, as it was already answered by Girard in a particular example. The initial goal of this last part of the internship was to make a definitive proof of this statement.

In parallel, Ehrhard noticed that the rules $(\partial\text{-lin})$ and $(\partial\text{-chain})$ made ∂ what is called a *distributive law* from the monad $\mathbf{S}$ to the comonad $!$, a notion introduced a while ago by Beck [\[2\]](#) and quite studied in some modern fields of semantic, see [\[14\]](#) for example. I then noticed that a similar notion of distributive law existed, this time from a comonad to another comonad. This notion was surprisingly very close to the commutations required on δ . This is when I noticed that Ehrhard implicitly showed in his paper that the adjunction $\bar{\mathbf{S}} \dashv \mathbf{S}$ carries distributive laws ∂ from the monad $\bar{\mathbf{S}}$ to the comonad $!$ to distributive laws from the comonad $\bar{\mathbf{S}}$ to the comonad $!$, and that the existence of such distributive law is equivalent to the existence of a coalgebra δ .

The fact that the kleisli category $\mathcal{L}_{\bar{\mathbf{S}}}$ is a model of linear logic thus seemed to hint that $\mathcal{L}_{\mathbf{S}}$ should be a model of a linear logic as well (and that a proof of the first statement should carry to the second through an adjunction). So I decided to devise some generic conditions under which the Kleisli category of a monad (or a comonad) is a model of linear logic to make everything clearer.

Recall that a model of linear logic $\mathcal{L}$ is characterized by nothing more than the existence of some functors together with the existence of some natural transformations that follow some commutations: a functor $\otimes$ together with four natural transformations $\alpha, \lambda, \rho, \gamma$, a functor $!$ together with two natural transformations der, dig , and two natural transformations $\mathbf{m}^0 : \top \rightarrow !1$ and $\mathbf{m}^2_{X,Y} : !X \otimes !Y \rightarrow !(X \& Y)$. Finally, there are two universal mapping properties (the Cartesian product $\&$ and the closure with regard to $\otimes$). I will not talk about those here, but they can also be looked upon this point of view using the fact that giving an universal mapping property is the same as giving two functors and two natural transformations (thanks to the notion of *adjunction*).

Now let us assume that we can extend those functors to $\mathcal{L}_{\mathbf{S}}$ in the sense of [Definition 30](#) below and that we can extend those natural transformations in the sense of [Definition 31](#).

Definition 30. Given a monad $\mathbf{H}$ on a category $\mathcal{C}$ and a monad $\mathbf{K}$ on a category $\mathcal{D}$, a functor $\widehat{F} : \mathcal{C}_{\mathbf{H}} \rightarrow \mathcal{D}_{\mathbf{K}}$ *lifts* the functor $F : \mathcal{C} \rightarrow \mathcal{D}$ to the Kleisli categories if for any object X of $\mathcal{C}$, $\widehat{F}X = FX$, and if for any morphism $f \in \mathcal{L}(X, Y)$, $\widehat{F}(\mathcal{K}_{\mathbf{H}}(f)) = \mathcal{K}_{\mathbf{K}}(Ff)$. In other word, “ $\widehat{F}$ coincides with F on $\mathcal{C}$ ”.

Definition 31. Take two functors $F, G : \mathcal{C} \rightarrow \mathcal{D}$ with some extensions $\widehat{F}, \widehat{G} : \mathcal{C}_H \rightarrow \mathcal{D}_K$. Given a natural transformation $\alpha : F \Rightarrow G$, we can define a family of morphisms

$$\mathcal{K}_K(\alpha) := (\mathcal{K}_K(\alpha_X))_X \in \mathcal{D}_K(\widehat{F}X, \widehat{G}X)$$

We say that α extends to $\widehat{F}$ and $\widehat{G}$ if $\mathcal{K}_K(\alpha)$ is a natural transformation $\widehat{F} \Rightarrow \widehat{G}$.

Then all the commutations on the natural transformations $\mathcal{K}_S(\cdot)$ would be obtained for free using the functoriality of $\mathcal{K}_S$. So $\mathcal{L}_S$ would “inherit” from $\mathcal{L}$ the necessary structure to be a model of linear logic. The litterature is already on point on the necessary and sufficient conditions that allows to extend a functor, see [13] for example.

Definition 32. A *distributive law* from a functor F to two monads (H, η^H, μ^H) and (K, η^K, μ^K) is a natural transformation $\lambda : FH \Rightarrow KF$ such that the two diagrams commute.

$$\begin{array}{ccc} \text{(Lift-Unit)} & FX \xrightarrow{F\eta^H} FHX & \\ & \searrow \eta^K \quad \downarrow \lambda & \\ & KFX & \end{array} \quad \begin{array}{ccc} \text{(Lift-Sum)} & FH^2X \xrightarrow{\lambda_{HX}} KFHX \xrightarrow{K\lambda_X} K^2FX & \\ \mu^H \downarrow & & \downarrow \mu^K \\ FHX & \xrightarrow{\lambda} & KFX \end{array}$$

We write $(F, \lambda) : (\mathcal{C}, H) \rightarrow (\mathcal{D}, K)$ if λ is a distributive law of F on H and K

This notion is also called distributive law, because a distributive law from a monad M to a comonad $!$ is in fact a distributive law from the functor $F = !$ to the monads $H = K = M$ that fulfills some additional commutations. This intersection of terminology was somewhat troublesome because it made my bibliographic work much harder.

Theorem 4. *There is a bijection between the set of functors $\widehat{F} : \mathcal{C}_H \rightarrow \mathcal{D}_K$ that extends F and the set of distributive laws $\lambda : FH \Rightarrow KF$*

However, I did not see anything in the litterature (at first) that characterizes when a natural transformation extends to the Kleisli category. Surprisingly, I discovered during my internship this very straightforward characterization.

Definition 33. Take $(F, \lambda_F), (G, \lambda_G) : (\mathcal{C}, H) \rightarrow (\mathcal{D}, K)$ two distributive laws. A morphism of distributive laws $\alpha : (F, \lambda_F) \rightarrow (G, \lambda_G)$ is a natural transformations $\alpha : F \Rightarrow G$ that verifies the following diagram

$$\begin{array}{ccc} FH & \xrightarrow{\alpha^H} & GH \\ \lambda^F \downarrow & & \downarrow \lambda^G \\ KF & \xrightarrow{K\alpha} & KG \end{array}$$

Result 8. Take $(F, \lambda_F), (G, \lambda_G) : (\mathcal{C}, H) \rightarrow (\mathcal{D}, K)$ two distributive laws with associated extension $\widehat{F}, \widehat{G}$. Take $\alpha : F \Rightarrow G$ a natural transformation. The the following are equivalent:

- (1) $\alpha : (F, \lambda_F) \rightarrow (G, \lambda_G)$ is a morphism of distributive laws
- (2) α extends to the kleisli category, in other words $\mathcal{K}_K(\alpha) : \widehat{F} \Rightarrow \widehat{G}$ is a natural transformation

The proof is quite simple (just unfold the definitions and use naturality), yet this result provides a powerful toolbox to extend natural transformations. So it *had* to be an already existing notion. It turns out to be the case: the notion of morphism of distributive law is introduced in a more abstract setting in [16]. Besides, an article by Power and Watabane [14] brought this notion to the semantic community. A recent discussion with Paul André Meliès suggested that this fact is somewhat folklore in some communities, but also that this notion seems to be somewhat underutilized in concrete settings such as mine. So the conclusions I draw below seem to be new and are quite enlightening.

- Applying the method to the ressource comonad gives back the notion of distributive law between a monad and a comonad. For example, in the setting of coherent differentiation, $(\partial\text{-lin})$ are the rules required for ∂ to be a distributive law between the monad S and the functor $!$, and $(\partial\text{-chain})$ are the rules required to extend der and dig .

- The functor $\otimes$ extends to $\mathcal{L}_M$ if and only if there exists a distributive law $\lambda_{X,Y}^\otimes : MX \otimes MY \rightarrow M(X \otimes Y)$. Interestingly, the commutations required to extend the natural transformations $\alpha, \lambda, \rho, \gamma$ are exactly the commutations that states that $(M, \eta, \lambda^\otimes)$ is a lax symmetric monoidal functor from $(\mathcal{L}, \otimes, 1)$ to itself. Adding the commutations required for $\lambda^\otimes$ to be a distributive laws exactly gives the conditions required to makes M what is called a *symetric monoidal monad*. This notion has been developped independently precisely to show when the Kleisli category of a monad inherits the structure of a symmetric monoidal category, see [10]. Besides, S precisely fulfills those conditions for $\lambda^\otimes = L$ (defined in Section 4.4). Thus $\mathcal{L}_S$ inherits from $\mathcal{L}$ the structure of a symmetric monoidal category.
- Finally, it turns out that in the setting of coherent differentiation, the conditions to extend m^0 and m^2 are exactly the two (∂ -Leibniz) axioms.

To conclude, I built some generic tools that exhibit that the axioms of coherent differentiation (excluding the Schwarz rule) turn out to be the necessary and sufficient conditions to ensures that $\mathcal{L}_S$ inherits from $\mathcal{L}$ the structure of a model of linear logic. So the abstract principles of first order differential calculus turns out to be *closely* tied to the very generic notion of structure extension applied to the specific monad (S, ι_0, τ) .

There is dually a notion of distributive law from a functor to two *comonads*, as well as similar notion of morphism between such distributive laws. Applying this notion to the comonad $_ \otimes I$ should solve the initial goal of showing that the Kleisli category of this comonad is a model of linear logic, but I was a bit short on time so I could not write it down yet. Besides, I think that the adjunction $\bar{S} \dashv S$ should canonically relate distributive laws to monads and their morphisms with distributive laws to comonads and their morphisms. Most of this work is already implicitly done in Ehrhard’s paper.

7 Conclusion

To sum up most of my internship, I showed in Section 5 that coherent differentiation is a generalization of differential categories and that its axioms except (∂ -lin) are in a one to one correspondence to the axioms on differential categories. I built in Section 6 some generic tools that, when applied to coherent differentiation, exhibit that those axioms (excluding the Schwarz rule) turn out to be the necessary and sufficient conditions to ensures that $\mathcal{L}_S$ inherits from $\mathcal{L}$ the structure of a model of linear logic.

Plan for future work Recall that all of the concrete instances of summability structure Ehrhard found in the various models actually consists in the functor $S = 1 \& 1 \multimap _$. A natural question is whether or not a summability structure is always of this shape. A step in the right direction would be to find instances of Monads M that are not of shape $1 \& 1 \multimap _$ such that M follows the properties discussed in Section 6, even if they does not correspond to a notion of differentiation. But those considerations require to learn a bunch of different models, this is why the question was not studied during the internship.

Finally, we expect that coherent differentiation might relate to the growing field of automated differentiation, used in machine learning. The difference is that automated differentiation only differentiates with regard to a type of real numbers, while coherent differentiation differentiates with regard to everything (which is admittedly a weird thing). Further work has to be done in order to relate those two fields, as it could provide an exciting application

References

- [1] Davide Barbarossa and Giulio Manzonetto. “Taylor Subsumes Scott, Berry, Kahn and Plotkin”. In: *Proc. ACM Program. Lang.* 4.POPL (Dec. 2019). DOI: [10.1145/3371069](https://doi.org/10.1145/3371069). URL: <https://doi.org/10.1145/3371069>.
- [2] Jon Beck. “Distributive laws”. In: *Seminar on Triples and Categorical Homology Theory*. Ed. by B. Eckmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1969, pp. 119–140. ISBN: 978-3-540-36091-9.
- [3] R. F. Blute et al. *Differential Categories Revisited*. 2018. DOI: [10.48550/ARXIV.1806.04804](https://arxiv.org/abs/1806.04804). URL: <https://arxiv.org/abs/1806.04804>.

- [4] Thomas Ehrhard. *Coherent differentiation*. 2021. DOI: [10.48550/ARXIV.2107.05261](https://doi.org/10.48550/ARXIV.2107.05261). URL: <https://arxiv.org/abs/2107.05261>.
- [5] Thomas Ehrhard. “Finiteness spaces”. In: *Mathematical Structures in Computer Science* 15.4 (July 2005). 32 pages, pp. 615–646. DOI: [10.1017/S0960129504004645](https://doi.org/10.1017/S0960129504004645). URL: <https://hal.archives-ouvertes.fr/hal-00150276>.
- [6] Thomas Ehrhard and Laurent Regnier. “The differential lambda-calculus”. In: *Theoretical Computer Science* 309.1-3 (Dec. 2003). 41 pages, pp. 1–41. DOI: [10.1016/S0304-3975\(03\)00392-X](https://doi.org/10.1016/S0304-3975(03)00392-X). URL: <https://hal.archives-ouvertes.fr/hal-00150572>.
- [7] Jean-Yves Girard. “Linear logic”. In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4). URL: <https://www.sciencedirect.com/science/article/pii/0304397587900454>.
- [8] Jean-Yves Girard. “The system F of variable types, fifteen years later”. In: *Theoretical Computer Science* 45 (1986), pp. 159–192. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/0304-3975\(86\)90044-7](https://doi.org/10.1016/0304-3975(86)90044-7). URL: <https://www.sciencedirect.com/science/article/pii/0304397586900447>.
- [9] G.M Kelly. “On MacLane’s conditions for coherence of natural associativities, commutativities, etc.” In: *Journal of Algebra* 1.4 (1964), pp. 397–402. ISSN: 0021-8693. DOI: [https://doi.org/10.1016/0021-8693\(64\)90018-3](https://doi.org/10.1016/0021-8693(64)90018-3). URL: <https://www.sciencedirect.com/science/article/pii/0021869364900183>.
- [10] Anders Kock. “Monads on symmetric monoidal closed categories”. In: *Archiv der Mathematik* 21 (1970), pp. 1–10.
- [11] Saunders Mac Lane. “Natural associativity and commutativity”. In: *Rice University Studies* (1963).
- [12] Paul-André Melliès. “CATEGORICAL SEMANTICS OF LINEAR LOGIC”. In: 2009.
- [13] Philip S. Mulry. “Lifting theorems for Kleisli categories”. In: *Mathematical Foundations of Programming Semantics*. Ed. by Stephen Brookes et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 304–319. ISBN: 978-3-540-48419-6.
- [14] John Power and Hiroshi Watanabe. “Combining a monad and a comonad”. In: *Theoretical Computer Science* 280.1 (2002). Coalgebraic Methods in Computer Science, pp. 137–162. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/S0304-3975\(01\)00024-X](https://doi.org/10.1016/S0304-3975(01)00024-X). URL: <https://www.sciencedirect.com/science/article/pii/S030439750100024X>.
- [15] R. A. G. Seely. “Linear Logic, *-Autonomous Categories and Cofree Coalgebras”. In: *Categories in Computer Science and Logic*. Vol. 92. Contemporary Mathematics. Boulder, Colorado: American Mathematical Society, 1989, pp. 371–382.
- [16] Ross Street. “The formal theory of monads”. In: *Journal of Pure and Applied Algebra* 2.2 (1972), pp. 149–168. ISSN: 0022-4049. DOI: [https://doi.org/10.1016/0022-4049\(72\)90019-9](https://doi.org/10.1016/0022-4049(72)90019-9). URL: <https://www.sciencedirect.com/science/article/pii/0022404972900199>.

Appendix: basic category theory definition

I assumed in this report that the reader knows the very basic definitions of a category (the . This appendix is here just in case to give a refresher on those notions.

Definition 34 (Category). A category $\mathcal{L}$ consists in

- In a set $Obj(\mathcal{L})$ called the set of objects of $\mathcal{L}$
- For any pair of objects $A, B \in Obj(\mathcal{L})$, a set $\mathcal{L}(A, B)$ called the set of morphisms from A to B .

Such that :

- For any object A , there exists a morphism $id_A^{\mathcal{L}} \in \mathcal{L}(A, A)$
- For any morphisms $f \in \mathcal{L}(A, B)$ and $g \in \mathcal{L}(B, C)$, there exists a morphism $g \circ_{\mathcal{L}} f \in \mathcal{L}(A, C)$
- For any $f \in \mathcal{L}(A, B)$, $f \circ_{\mathcal{L}} id_A^{\mathcal{L}} = id_B^{\mathcal{L}} \circ_{\mathcal{L}} f = f$
- For any $f \in \mathcal{L}(A, B)$, $g \in \mathcal{L}(B, C)$ and $h \in \mathcal{L}(C, D)$, $h \circ_{\mathcal{L}} (g \circ_{\mathcal{L}} f) = (h \circ_{\mathcal{L}} g) \circ_{\mathcal{L}} f$

Remark 8. I directly introduce $Obj(\mathcal{L})$ and $\mathcal{L}(X, Y)$ as sets rather than classes because I do not want to deal with foundational issues here (the set of all set is not defined, the category of all category is not defined, etc).

Notations. When there is no doubt about the category $\mathcal{L}$ considered, I will write $\circ$ for $\circ_{\mathcal{L}}$ and id for $id^{\mathcal{L}}$. Besides, I will often write $f : A \rightarrow B$ for $f \in \mathcal{L}(A, B)$ (taking inspiration from functional notations).

Definition 35 (Isomorphism). A morphism $f : A \rightarrow B$ is an *isomorphism* if there exists a morphism $g : B \rightarrow A$ such that $f \circ g = id_B$ and $g \circ f = id_A$.

Definition 36 (Functor). Given two categories $\mathcal{C}$ and $\mathcal{D}$, a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ consists in:

- A function $F^{Obj} : Obj(\mathcal{C}) \rightarrow Obj(\mathcal{D})$
- For any objects $A, B \in Obj(\mathcal{C})$, a function $F_{A,B} : \mathcal{C}(A, B) \rightarrow \mathcal{D}(F^{Obj}(A), F^{Obj}(B))$ such that $F_{A,A}(id_A^{\mathcal{C}}) = id_{F^{Obj}(A)}^{\mathcal{D}}$ and $F_{A,C}(f \circ_{\mathcal{C}} g) = F_{A,B}(f) \circ_{\mathcal{D}} F_{B,C}(g)$.

We will often write F for both F^{Obj} and $F_{A,B}$ (depending on what it is applied, we can infer if we actually use F^{Obj} or $F_{A,B}$).

Definition 37 (Product category). Given two categories $\mathcal{C}$ and $\mathcal{D}$, we can define a category $\mathcal{C} \& \mathcal{D}$ where:

- $Obj(\mathcal{C} \& \mathcal{D}) := \{(A_1, A_2) \mid A_1 \in Obj(\mathcal{C}) \text{ and } A_2 \in Obj(\mathcal{D})\}$
- $(\mathcal{C} \& \mathcal{D})((A_1, A_2), (B_1, B_2)) := \{(f, g) \mid f \in \mathcal{C}(A_1, B_1) \text{ and } g \in \mathcal{D}(A_2, B_2)\}$
- The identity is defined as $id_{(A_1, A_2)} := (id_{A_1}^{\mathcal{C}}, id_{A_2}^{\mathcal{D}})$ and the composition is defined as $(g_1, g_2) \circ (f_1, f_2) := (g_1 \circ_{\mathcal{C}} f_1, g_2 \circ_{\mathcal{D}} f_2)$. We can check that this is indeed a category.

Definition 38 (Endofunctor, bifunctor). We call an *endofunctor* any functor of shape $F : \mathcal{L} \rightarrow \mathcal{L}$. We call a *bifunctor* any functor of shape $F : \mathcal{C} \& \mathcal{D} \rightarrow \mathcal{L}$.

A very powerful and intuitive way of showing equalities between morphisms in categories is to use what are called *commutative diagrams*. For example, the equality $h = g \circ f$ can be represented as the diagram.

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ & \searrow h & \downarrow g \\ & & Z \end{array}$$

More formally, a diagram is an oriented graph where vertices are indexed by objects and every edges from X to Y are indexed by a morphism of $\mathcal{L}(X, Y)$. A diagram states that for any objects X, Y and any path from X to Y , the morphisms obtained by the successive compositions are equal.

Diagrams can be combined, doing what is called a *diagram chase*. The diagram below is an example of a diagram chase.

$$\begin{array}{ccccc}
X_1 & \xrightarrow{f_1} & Y_1 & \xrightarrow{g_1} & Z_1 \\
h_X \downarrow & & \downarrow h_Y & & \downarrow h_Z \\
X_2 & \xrightarrow{f_2} & Y_2 & \xrightarrow{g_2} & Z_2
\end{array}$$

This chase states that $g_2 \circ f_2 \circ h_X = h_Z \circ g_1 \circ f_1$ using the fact that $f_2 \circ h_X = h_Y \circ f_1$ and $g_2 \circ h_Y = h_Z \circ g_1$.

Definition 39 (Natural transformation). A natural transformation α between two functors $F, G : \mathcal{C} \rightarrow \mathcal{D}$, written $\alpha : F \Rightarrow G$, is a family of morphisms $(\alpha_X)_{X \in \text{Obj}(\mathcal{C})}$ where $\alpha_X \in \mathcal{D}(FX, GX)$ such that for any objects X, Y of $\mathcal{C}$ and $f \in \mathcal{C}(X, Y)$, the following diagram commutes.

$$\begin{array}{ccc}
FX & \xrightarrow{F(f)} & FY \\
\alpha_X \downarrow & & \downarrow \alpha_Y \\
GX & \xrightarrow{G(f)} & GY
\end{array}$$

When the morphisms are all isomorphisms, α is called a natural isomorphism.