SEMINÁRIO DE LÓGICA MATEMÁTICA (SLM)
March 2022 - year XXXIII

∂ is for Dialectica

Marie Kerjean

CNRS & LIPN, Université Sorbonne Paris Nord

Work in collaboration with Pierre-Marie Pédrot

Differentiable programming

A new area triggered by the advances of deep learning algorithms on neural networks, it tries to attach two very old domains:

- Algorithmic Differentiation.
- λ-calculus.

Goal: Exploring modular way to express (algorithmic) differentiation in functional programming languages:

- Abadi & Plotkin, POPL20. (traces and big-step semantics)
- Brunel & Mazza & Pagani, POPL20, POPL21.
- Elliot, ICFP18, (compositional differentiation)
- Wang and al., ICFP 19, (delimited continuations)
- Interactions with probabilistic programming...

The real inventor of deep learning

Outline of the talk

1. Reverse differentiation and differentiable programming.
2. Dialectica acting on formulas.
3. Dialectica acting on λ-terms.
4. Factorizing Dialectica through differential linear logic.
5. Applications and related work.

Automatic Differentiation

How does one compute the differentiation of an algebraic expression, computed as a sequence of elementary operations ?

$$\text{E.g. : } z = y + cos(x^2) \quad \begin{array}{ll} x_1 = x_0^2 & x_1' = 2x_0x_0' \\ x_2 = cos(x_1) & x_2' = -x_0' sin(x_0) \\ z = y + x_2 & z' = y' + 2x_2x_2' \end{array}$$

The computation of the final results requires the computation of the derivative of all partial computation. But in which order ?

Forward Mode differentiation [Wengert, 1964]
$(x_1, x_1') \rightarrow (x_2, x_2') \rightarrow (z, z')$.
Reverse Mode differentiation: [Speelpenning, Rall, 1980s]
$x_1 \rightarrow x_2 \rightarrow z \rightarrow z' \rightarrow x_2' \rightarrow x_1'$ *while keeping formal the unknown derivative*.

I hate graphs

$$\mathrm{D}_u(f \circ g) = D_{g(u)} f \circ D_u(g)$$

- **Forward Mode differentiation :**
  $g(u) \to D_u g \to f(g(u)) \to D_{g(u)} f \to D_{g(u)} f \circ D_u(g)$.
- **Reverse Mode differentiation:**
  $g(u) \to f(g(u)) \to D_{g(u)} f \to D_u g \to D_{g(u)} f \circ D_u(g)$

The choice of an algorithm is due to complexity considerations:

- Forward mode for $f \circ g : \mathbb{R} \to \mathbb{R}^n$.
- Reverse mode for $f \circ g : \mathbb{R}^n \to \mathbb{R}$

$\rightsquigarrow$ Differentiation is about *linearizing* a function/program. Some people have a very specific idea of what a *linear program* or a *linear type* should be.

Idea: Reverse Differentials are contravariant

- **Forward Mode differentiation :**
$$h : A \Rightarrow B \rightsquigarrow \overrightarrow{D}h : A \Rightarrow A \multimap B.$$

- **Reverse Mode differentiation:**
$$h : A \Rightarrow B \rightsquigarrow \overleftarrow{D}h : A \Rightarrow B^{\perp} \multimap A^{\perp}.$$

AD from a functorial point of view

How to make differentiation functorial ? Make it act on pairs !

$$f : E \Rightarrow F$$

forward:

$$\overrightarrow{D}(f) : \begin{cases} E \times E \to F \times F \\ (a, x) \mapsto (f(a), (\mathrm{D}_a f \cdot x)) \end{cases}$$

backward:

$$\overleftarrow{D}(f) : \begin{cases} E \times F' \to F \times E' \\ (a, \ell) \mapsto (f(a), (\ell \circ \mathrm{D}_a f)) \end{cases}$$

Brunel, Mazza and Pagani [POPL2020]

Key Idea

Reverse derivatives are typed by linear negation.

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$ a function variable.

$$\overleftarrow{D}(f) : \begin{cases} \mathbb{R}^n \times \mathbb{R}^{m\perp} \to \mathbb{R}^m \times \mathbb{R}^{n\perp} \\ \qquad (a, x) \mapsto (f(a), (v \mapsto x \cdot (\mathrm{D}_a f \cdot v)) \end{cases}$$

This leads to a **compositional reverse derivative** transformation over the *linear substitution calculus*, and proven complexity results.

$$A, B, C ::= R \mid A \times B \mid A \to B | R^{\perp_d}$$

$$t, u := x \mid x^! \mid \lambda x.t \mid (t)u \mid t[x^{()!} := u] \mid < t, u > \mid t + u...$$

A Dialectica Transformation

- Gödel Dialectica transformation [1958] : a translation from intuitionistic arithmetic to a finite type extension of primitive recursive arithmetic.

$$A \rightsquigarrow \exists u : \mathbb{W}(A), \forall x : \mathbb{C}(A), A^D[u, x]$$

- De Paiva [1991]: the linearized Dialectica translation operates on Linear Logic (types) and λ-calculus (terms).

- Pedrot [2014] A *computational* Dialectica translation preserving β-equivalence, via the introduction of an "abstract multiset constructor" on types on the target.

Gödel's Dialectica

1. $(\mathrm{F} \wedge \mathrm{G})' = (\exists yv)(zw)[\mathrm{A}(y, z, x) \wedge \mathrm{B}(v, w, u)]$.
2. $(\mathrm{F} \vee \mathrm{G})' = (\exists yvt)(zw)[t=0 \wedge \mathrm{A}(y, z, x) \cdot \vee \cdot t=1 \wedge \mathrm{B}(v, w, u)]$.
3. $[(s)\mathrm{F}]' = (\exists \mathrm{Y})(sz)\mathrm{A}(\mathrm{Y}(s), z, x)$.
4. $[(\exists s)\mathrm{F}]' = (\exists sy)(z)\mathrm{A}(y, z, x)$.
5. $(\mathrm{F} \supset \mathrm{G})' = (\exists \mathrm{VZ})(yw)[\mathrm{A}(y, \mathrm{Z}(yw), x) \supset \mathrm{B}(\mathrm{V}(y), w, u)]$.
6. $(\neg \mathrm{F})' = (\exists \bar{\mathrm{Z}})(y) \neg \mathrm{A}(y, \bar{\mathrm{Z}}(y), x)$.

Kurt Gödel (1958). Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. Dialectica.

Gödel's Dialectica

- Validates semi-classical axioms:
  - Markov's principle : $\neg\neg\exists x A \to \exists x A$ when A is decidable.
  - Independant of premises : $(A \to \exists x B) \to (\exists x.(A \to B))$
- Numerous applications :
  - Soudness results
  - Proof mining

A further distinguishing feature of the D-interpretation is its nice behavior with respect to modus ponens. In contrast to cut-elimination, which entails a global (and computationally infeasible) transformation of proofs, the D-interpretation extracts constructive information through a purely local procedure: when proofs of φ and $\varphi \to \psi$ are combined to yield a proof of ψ, witnessing terms for the antecedents of this last inference are combined to yield a witnessing term for the conclusion. As a result of this modularity, the interpretation of a theorem can be readily obtained from the interpretations of the lemmata used in its proof.

Jeremy Avigad and Solomon Feferman (1999). Gödel's functional ("Dialectica") interpretation

A peek into Dialectica interpretation of functions

$$(A \to B)_D = \exists fg \forall xy (A_D(x, gxy) \to B_D(fx, y))$$

Usual explanation : least unconstructive prenexation.

- Start from $\exists x, \forall u, A_D[x, u] \to \exists y, \forall v, B_D[y, v]$.
- Obvious prenexation : $\forall x\, (\forall u, A_D[x, u] \to \exists y, \forall v, B_D[y, v])$
- Weak form of IP : $\forall x \exists y\, (\forall u, A_D[x, u] \to \forall v, B_D[y, v])$
- Prenexation : $\forall x \exists y, \forall v, \forall \neg\neg \exists u\, (A_D[x, u] \to B_D[y, v])$.
- Markov : $\forall x, \exists y, \forall v, \exists u (A_D[x, u] \to B_D[y, v])$
- Axiom of choice : $\exists f, \exists g, \forall u, \forall v, (A_D(u, guv) \to B_D[fu, v])$.

Dynamic behaviour : agrees to a chain rule.

Mathematical meaning : it's some kind of approximation.

Dialectica verifies the chain rules

$$(A \Rightarrow B)_D[\phi_1; \psi_1, u_1; v_1] := A_D(u_1, \psi_1\, u_1\, v_1) \Rightarrow B_D(\phi_1\, u_1, v_1)$$
$$(B \Rightarrow C)_D[\phi_2; \psi_2, u_2; v_2] := B_D(u_2, \psi_2\, u_2\, v_2) \Rightarrow C_D(\phi_2\, u_2, v_2)$$
$$(A \Rightarrow C)_D[\phi_3; \psi_3, u_3; v_3] := A_D(u_3, \psi_3\, u_3\, v_3) \Rightarrow C_D(\phi_3\, u_3, v_3)$$

The Dialectica interpretation amounts to the following equations:

$$u_3 = u_1 \qquad \psi_3\, u_3\, v_3 = \psi_1\, u_1\, v_1$$
$$v_3 = v_2 \qquad \phi_2\, u_2 = \phi_1\, u_1$$
$$u_2 = \phi_1\, u_1 \qquad v_2 = \phi_1\, u_1\, v_1$$

which can be simplified to:

$$\phi_3\, u_3 = \phi_2\,(\phi_1\, u_3) \textit{ composition of functions}$$
$$\psi_3\, u_3\, v_3 = \psi_2\,(\phi_1\, u_3)\,(\psi_1\, u_3\, v_3) \textit{ composition of their differentials}$$

Types !

$$A \rightsquigarrow \exists \overbrace{x : \mathbb{W}(A)}^{\text{witness}}, \forall \underbrace{u : \mathbb{C}(A)}_{\text{opponent}}, A_D[x, u]$$

Witness and counter types :

$$\mathbb{C}(A \Rightarrow B) = \mathbb{C}(A) \times \mathbb{C}(B)$$

$$\mathbb{W}(A \Rightarrow B) = (\mathbb{W}(A) \Rightarrow \mathbb{W}(B)) \times (\mathbb{W}(A) \Rightarrow \mathbb{C}(B) \Rightarrow \mathbb{C}(A))$$

Types !

$$A \rightsquigarrow \exists \overbrace{x : \mathbb{W}(A)}^{\text{global witness}}, \forall \underbrace{u : \mathbb{C}(A)}_{\text{local opponent}}, A_D[x, u]$$

Witness and counter types :

$$\mathbb{C}(A \Rightarrow B) = \mathbb{C}(A) \times \mathbb{C}(B)$$

$$\mathbb{W}(A \Rightarrow B) = \overbrace{(\mathbb{W}(A) \Rightarrow \mathbb{W}(B))}^{\text{function}} \times \left(\mathbb{W}(A) \Rightarrow \underbrace{\mathbb{C}(B) \Rightarrow \mathbb{C}(A)}_{\text{reverse derivative}} \right)$$

Let's say x, u, f, g are λ-terms.

A reverse Differential λ-calculus

"Behind every successful proof there is a program", *Gödel's wife*

A computational Dialectica

Making Dialectica act on λ-terms instead of formulas:

An abstract multiset $\mathfrak{M}\,(-)$

$$\frac{}{\Gamma \vdash \varnothing : \mathfrak{M}\,A} \qquad \frac{\Gamma \vdash m_1 : \mathfrak{M}\,A \qquad \Gamma \vdash m_2 : \mathfrak{M}\,A}{\Gamma \vdash m_1 \circledast m_2 : \mathfrak{M}\,A}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \{t\} : \mathfrak{M}\,A} \qquad \frac{\Gamma \vdash m : \mathfrak{M}\,A \qquad \Gamma \vdash f : A \Rightarrow \mathfrak{M}\,B}{\Gamma \vdash m \gg\!= f : \mathfrak{M}\,B}$$

$$\begin{aligned}
\mathbb{W}(A \Rightarrow B) \quad &:= \quad (\mathbb{W}(A) \Rightarrow \mathbb{W}(B)) \\
&\qquad \times (\mathbb{C}(B) \Rightarrow \mathbb{W}(A) \Rightarrow \mathfrak{M}\,\mathbb{C}(A)) \\
\mathbb{C}(A \Rightarrow B) \quad &:= \quad \mathbb{W}(A) \times \mathbb{C}(B)
\end{aligned}$$

Pédrot's Dialectica Transformation

Soundness [Ped14]

If $\Gamma \vdash t : A$ in the source then we have in the target

- $\mathbb{W}(\Gamma) \vdash t^\bullet : \mathbb{W}(A)$
- $\mathbb{W}(\Gamma) \vdash t_x : \mathbb{C}(A) \Rightarrow \mathfrak{M}\,\mathbb{C}(X)$ provided $x : X \in \Gamma$.

A global and a local transformation

$$
\begin{array}{lcl@{\qquad}lcl}
x^\bullet & := & x & (\lambda x.\, t)^\bullet & := & (\lambda x.\, t^\bullet, \lambda \pi x.\, t_x\ \pi) \\
x_x & := & \lambda \pi.\, \{\pi\} & (\lambda x.\, t)_y & := & \lambda \pi.\, (\lambda x.\, t_y)\ \pi.1\ \pi.2 \\
x_y & := & \lambda \pi.\, \varnothing \text{ if } x \neq y & (t\ u)^\bullet & := & (t^\bullet.1)\ u^\bullet
\end{array}
$$

$$
(t\ u)_y := \lambda \pi.\, (t_y\, (u^\bullet, \pi)) \circledast ((t^\bullet.2)\, \pi\, u^\bullet \gg\!= u_y)
$$

Flashback: Differential λ-calculus [Ehrhard, Regnier 04]

Inspired by denotational models of Linear Logic in vector spaces of sequences, it introduces a differentiation of λ-terms.

$D(\lambda x.t)$ is the linearization *of $\lambda x.t$, it substitute x linearly, and then it remains a term t' where x is free.*

Syntax:

$$\Lambda^d : S, T, U, V ::= 0 \mid s \mid s{+}T$$
$$\Lambda^s : s, t, u, v ::= x \mid \lambda x.s \mid sT \mid \mathrm{D}s{\cdot}t$$

Operational Semantics:

$$(\lambda x.s)T \to_\beta s[T/x]$$
$$\mathrm{D}(\lambda x.s) \cdot t \to_{\beta_D} \lambda x.\frac{\partial s}{\partial x} \cdot t$$

where $\frac{\partial s}{\partial x} \cdot t$ is the **linear substitution** of x by t in s.

Linearity in Linear Logic

Linearity is about resources: A proof/program is *linear* iff it uses only once its hypotheses/argument.

Linear	**Non-linear**
$A \vdash A \vee B$	$A \vdash A \wedge A$
$\lambda f \lambda x.fxx$	$\lambda x.\lambda f.fxx$

Usual Implication

A call-by-name translation

$$A \Rightarrow B = \ !A \ \multimap B$$

$$\mathcal{C}^{\infty}(A, B) \simeq \mathcal{L}(!A, B)$$

Linearity in Linear Logic

Linearity is about resources: A proof/program is *linear* iff it uses only once its hypotheses/argument.

Linear	**Non-linear**
$A \vdash A \vee B$	$A \vdash A \wedge A$
$\lambda f \lambda x.fxx$	$\lambda x.\lambda f.fxx$

Usual implication

Linear Implication

A call-by-name translation

$$A \Rightarrow B = \,!\,A \multimap B$$

$$\mathcal{C}^{\infty}(A, B) \simeq \mathcal{L}(!A, B)$$

Linearity in Linear Logic

Linearity is about resources: A proof/program is *linear* iff it uses only once its hypotheses/argument.

Linear	**Non-linear**
$A \vdash A \vee B$	$A \vdash A \wedge A$
$\lambda f \lambda x.fxx$	$\lambda x.\lambda f.fxx$

Usual implication

Linear implication

A call-by-name translation

$$A \Rightarrow B = !\,A \multimap B$$

$$\mathcal{C}^{\infty}(A, B) \simeq \mathcal{L}(!A, B)$$

Exponential

Smooth Semantics

The linear substitution ...

... which is not exactly a substitution

$$\frac{\partial y}{\partial x} \cdot T = \begin{cases} T & \textit{if } x = y \\ 0 & \textit{otherwise} \end{cases} \qquad \frac{\partial}{\partial x}(sU) \cdot T = (\frac{\partial s}{\partial x} \cdot T)U + (\mathrm{D}s \cdot (\frac{\partial U}{\partial x} \cdot T))U$$

$$\frac{\partial}{\partial x}(\lambda y.s) \cdot T = \lambda y.\frac{\partial s}{\partial x} \cdot T \qquad \frac{\partial}{\partial x}(\mathrm{D}s \cdot u) \cdot T = \mathrm{D}(\frac{\partial s}{\partial x} \cdot T) \cdot u + \mathrm{D}s \cdot (\frac{\partial u}{\partial x} \cdot T)$$

$$\frac{\partial 0}{\partial x} \cdot T = 0 \qquad \frac{\partial}{\partial x}(s + U) \cdot T = \frac{\partial s}{\partial x} \cdot T + \frac{\partial U}{\partial x} \cdot T$$

$\frac{\partial s}{\partial x} \cdot t$ represents s where x is linearly (i.e. one time) substituted by t.

Tracking differentiation in Dialectica

7 years ago : "That's Differential λ-calculus"

$$
\begin{array}{lcl}
x_x & := & \lambda\pi.\,\{\pi\} \\
x_y & := & \lambda\pi.\,\varnothing \quad \text{if } x \neq y \\
(\lambda x.\,t)_y & := & \lambda\pi.\,(\lambda x.\,t_y)\ \pi.1\ \pi.2
\end{array}
\qquad
\begin{array}{lcl}
x^\bullet & := & x \\
(\lambda x.\,t)^\bullet & := & (\lambda x.\,t^\bullet, \lambda x\pi.\,t_x\ \pi) \\
(t\ u)^\bullet & := & (t^\bullet.1)\ u^\bullet
\end{array}
$$

$$(t\ u)_y := \lambda\pi.\,(t_y\,(u^\bullet, \pi)) \circledast ((t^\bullet.2)\,u^\bullet\,\pi \gg\!= u_y)$$

Tracking differentiation in Dialectica

7 years ago : "That's Differential λ-calculus"

$$
\begin{array}{lcl}
x_x & := & \lambda\pi.\,\{\pi\} \\
x_y & := & \lambda\pi.\,\varnothing \quad \text{if } x \neq y \\
(\lambda x.\,t)_y & := & \lambda\pi.\,(\lambda x.\,t_y)\ \pi.1\ \pi.2
\end{array}
\qquad
\begin{array}{lcl}
x^\bullet & := & x \\
(\lambda x.\,t)^\bullet & := & (\lambda x.\,t^\bullet, \lambda x\pi.\,t_x\ \pi) \\
(t\ u)^\bullet & := & (t^\bullet.1)\ u^\bullet
\end{array}
$$

$$(t\ u)_y := \lambda\pi.\,(t_y\,(u^\bullet, \pi)) \circledast ((t^\bullet.2)\,u^\bullet\,\pi \gg\!= u_y)$$

Tracking differentiation in Dialectica

7 years ago : "That's Differential λ-calculus"

$$
\begin{array}{lcl}
x_x & := & \lambda\pi.\, \frac{\partial x}{\partial x} \cdot \pi \\
x_y & := & \lambda\pi.\, \frac{\partial x}{\partial y} \cdot \pi \quad \text{if } x \neq y \\
(\lambda x.\, t)_y & := & \lambda\pi.\, (\lambda x.\, t_y)\ \pi.1\ \pi.2
\end{array}
\qquad
\begin{array}{lcl}
x^\bullet & := & x \\
(\lambda x.\, t)^\bullet & := & (\lambda x.\, t^\bullet, \lambda x\pi.\, t_x\ \pi) \\
(t\ u)^\bullet & := \equiv & (\lambda x.\, (tx)^\bullet)\ u^\bullet
\end{array}
$$

$$(t\ u)_y := \lambda\pi.\, (t_y\, (u^\bullet, \pi)) \circledast ((t^\bullet.2)\, u^\bullet\, \pi \gg\!= u_y)$$

3 years ago : That's reverse differentiation

- $(\_)^\bullet.2$ obeys the chain rule, $(\_)^\bullet$ is the functorial differentiation.
- t_x is contravariant in x.

Tracking differentiation in Dialectica

7 years ago : "That's Differential λ-calculus"

$$
\begin{array}{lcl}
x_x & := & \lambda\pi.\, \frac{\partial x}{\partial x} \cdot \pi \\
x_y & := & \lambda\pi.\, \frac{\partial x}{\partial y} \cdot \pi \quad \text{if } x \neq y \\
(\lambda x.\, t)_y & := & \lambda\pi.\, (\lambda x.\, t_y)\ \pi.1\ \pi.2
\end{array}
\qquad
\begin{array}{lcl}
x^\bullet & := & x \\
(\lambda x.\, t)^\bullet & := & (\lambda x.\, t^\bullet, \lambda x\pi.\, t_x\ \pi) \\
(t\ u)^\bullet & \equiv & (\lambda x.\, (tx)^\bullet)\ u^\bullet
\end{array}
$$

3 years ago : That's reverse differentiation

- $(\_)^\bullet.2$ obeys the chain rule, $(\_)^\bullet$ is the functorial differentiation.
- t_x is contravariant in x.

$$[\![u \gg\!= t_x]\!] \equiv \lambda z.\, \left([\![u]\!]\left(\frac{\partial t}{\partial x} \cdot z\right)\right)$$

up to the linearity of $[\![u]\!]$, IRL we make use of two logical relations

Dialectica is reverse differential λ-calculus

where the linearity of counter terms is not enforced.

Two logical relations : the arrow case

$$
\begin{array}{lcll}
t \sim_{A\to B} T & := & \forall u \sim_A U. & (t.1\ u) \sim_B (T\ U) \\
 & & & \wedge \quad (t.2\ u) \bowtie^A_B (\lambda z.\,(DT \cdot z)\ U) \\
t \bowtie^X_{A\to B} T & := & \forall u \sim_A U. & \lambda\pi.\, t\ (u,\pi) \bowtie^X_B (\lambda z.\, T\ z\ U)
\end{array}
$$

Theorem

If $\Gamma \vdash t : A$ is a simply-typed λ-term, then

- for all $\vec{r} \sim_\Gamma \vec{R}$, $t^\bullet\{\Gamma \leftarrow \vec{r}\} \sim_A t\{\Gamma \leftarrow \vec{R}\}$,
- and for all $\vec{r} \sim_\Gamma \vec{R}$ and $x : X \in \Gamma$,

$$t_x\{\Gamma \leftarrow \vec{r}\} \bowtie^X_A \lambda z. \left(\frac{\partial t}{\partial x} \cdot z\right)\{\Gamma \leftarrow \vec{R}\}.$$

A Linear Logic Refinement

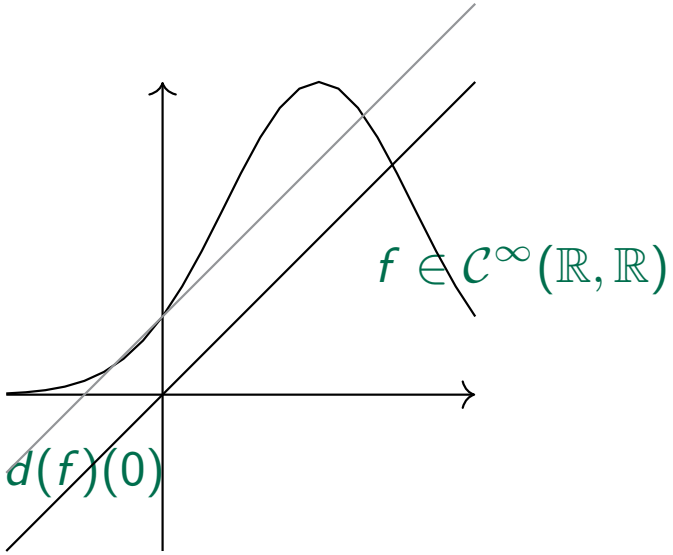
Differential Linear Logic

$$\frac{\vdash \ell : A \multimap B}{\vdash \ell : !A \multimap B}\ d$$

A linear proof
is in particular non-linear.

$$\frac{\vdash f : !A \multimap B}{\vdash D_0 f : A \multimap B}\ \bar{d}$$

From a non-linear proof
we can extract a linear proof



Differential interaction nets, Ehrhard and Regnier, TCS (2006)

Exponential rules of Differential Linear Logic

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\, w \qquad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\, c \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\, d$$

$$\frac{}{\vdash !A}\, \bar{w} \qquad \frac{\Gamma \vdash !A \qquad \Delta \vdash !A}{\Gamma, \Delta, \vdash !A}\, \bar{c} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash !A}\, \bar{d}$$

$$\frac{?\Gamma \vdash A}{?\Gamma \vdash !A}\, p$$

Dialectica factorizes through Linear Logic

$$
\begin{array}{lcl}
\mathbb{W}(A^{\perp}) & := & \mathbb{C}(A) \\
\mathbb{W}(A \oplus B) & := & \mathbb{W}(A) + \mathbb{W}(B) \\
\mathbb{W}(!A) & := & \mathbb{W}(A)
\end{array}
\qquad
\begin{array}{lcl}
\mathbb{C}(A^{\perp}) & := & \mathbb{W}(A) \\
\mathbb{C}(A \oplus B) & := & \mathbb{C}(A) \times \mathbb{C}(B) \\
\mathbb{C}(!A) & := & \mathbb{W}(A) \Rightarrow \mathbb{C}(A)
\end{array}
$$

$$
\begin{array}{lcl}
\mathbb{W}(A \otimes B) & := & \mathbb{W}(A) \times \mathbb{W}(B) \\
\mathbb{C}(A \otimes B) & := & (\mathbb{W}(A) \Rightarrow \mathbb{C}(B)) \times (\mathbb{W}(B) \Rightarrow \mathbb{C}(A))
\end{array}
$$

$$
\begin{tikzcd}
 & \mathrm{LL} \arrow[dr, "\mathbb{W}", "\mathbb{C}"'] & \\
\lambda^{+,\times} \arrow[ur, "[\![-]\!]_e"] \arrow[rr, "\mathbb{W}", "\mathbb{C}"'] & & \lambda^{+,\times}
\end{tikzcd}
$$

Valeria de Paiva, 1989, A dialectica-like model of linear logic.

Dialectica factorizes through Differential Linear Logic

$$
\begin{array}{lcl}
\mathbb{W}(!A) & := & !\mathbb{W}(A) \qquad \mathbb{C}(!A) := !\mathbb{W}(A) \multimap \mathbb{C}(A) \\
\mathbb{W}(A \otimes B) & := & \mathbb{W}(A) \otimes \mathbb{W}(B) \\
\mathbb{C}(A \otimes B) & := & (\mathbb{W}(A) \multimap \mathbb{C}(B)) \oplus (\mathbb{W}(B) \multimap \mathbb{C}(A)) \\
\mathbb{W}(A \multimap B) & := & (\mathbb{W}(A) \multimap \mathbb{W}(B)) \,\&\, (\mathbb{C}(B) \multimap \mathbb{C}(A)) \\
\mathbb{C}(A \multimap B) & := & \mathbb{W}(A) \otimes \mathbb{C}(B)
\end{array}
$$

If $\Gamma \vdash A$ in LL, then $\mathbb{W}(\Gamma) \vdash \mathbb{W}(A)$ in classical DiLL.

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax}}{\vdash A, !A^\perp}\,\bar{d} \qquad \dfrac{}{\vdash ?A, !A^\perp}\,\mathsf{ax}}{\vdash ?A, A, !A^\perp}\,\bar{c} \qquad \dfrac{\pi}{\Gamma \vdash ?A}}{\Gamma \vdash ?A, A}\,\mathsf{cut}
$$

Dialectica factorizes through Differential Linear Logic

The economical translation

$$
\begin{aligned}
[\![A \Rightarrow B]\!]_e &:= !A \multimap B \\
[\![A \times B]\!]_e &:= A \,\&\, B \\
[\![A + B]\!]_e &:= A \oplus B
\end{aligned}
$$

$$
\begin{tikzcd}
\mathrm{ILL} \arrow[r, "\mathbb{W}", "\mathbb{C}"'] & \mathrm{IDiLL} \arrow[d, "\cdots"] \\
\lambda^{+,\times} \arrow[u, "{[\![-]\!]_e}"] \arrow[r, "\mathbb{W}", "\mathbb{C}"'] & \lambda^{+,\times}
\end{tikzcd}
$$

IDILL : Intuitionnistic Differential Linear Logic ? Oh no ...

Dialectica categories through Differential Categories

Categories representing specific relations

Consider a category $\mathcal{C}$. **Dial**$(\mathcal{C})$ is constructed as follows:

- Objects : relations $\alpha \subseteq U \times X$, $\beta \subseteq V \times Y$.
- Maps from α to β :

$$(f : U \to V, F : U \times Y \to X)$$

- Composition : the chain rule !

Consider

$$\begin{array}{rrcl} & (f, F) : \quad \alpha \subseteq (A, X) & \to & \beta \subseteq (B, Y) \\ \text{and} & (g, G) : \quad \beta \subseteq (B, Y) & \to & \gamma \subseteq (C, Z) \end{array}$$

two arrows of the Dialectica category. Then their composition is defined as

$$(g, G) \circ (f, F) := (g \circ f, (a, z) \mapsto G(f(a), F(a, z))).$$

Dialectica categories through Differential Categories

In a $*$-autonomous differential category :

$$\partial : Id \otimes ! \to !$$

$$\mathcal{L}(B \otimes A, C^{\perp}) \simeq \mathcal{L}(A, (B \otimes C)^{\perp})$$

From $f : !A \to B$ one constructs :

$$\overleftarrow{D}(f) \in \mathcal{L}(!A \otimes B^{\perp}, A^{\perp}).$$

Dialectica categories factorize through differential categories

If $\mathcal{L}$ is a model of DiLL such that $\mathcal{L}_!$ has finite limits:

$$\begin{cases} \mathcal{L}_! & \to & \mathscr{D}(\mathcal{L}_!) \\ A & \mapsto & A \times A^{\perp} \\ f & \mapsto & (f, \overleftarrow{D}(f)) \end{cases}$$

To be declined in reverse/cartesian differential categories...

Conclusion and applications

Take home message:

Dialectica is functorial reverse differentiation,
extracting ~~intensional~~ local content from proofs.

Related work and applications:

- Semantics : Ehrhard's differentiation without sums.
- Markov's principle and delimited continuations on positive formulas.
- Proof mining and backpropagation.

Ehrhard's differentiation without sums

Content. We base our approach on a concept of summable pair that we axiomatize as a general categorical notion in Section 2: a *summable category* is a category $\mathcal{L}$ with 0-morphisms[1] together with a functor $\mathbf{S} : \mathcal{L} \to \mathcal{L}$ equipped with three natural transformations from $\mathbf{S}X$ to X: two projections and a sum operation. The first projection also exists in the "tangent bundle" functor of a tangent category but the two other morphisms do not. Such a summability structure induces a monad structure on $\mathbf{S}$ (a similar phenomenon occurs in tangent categories). In Section 3 we consider the case where the category is a cartesian SMC equipped with a resource comonad $!\_$ in the sense of LL where we present differentiation as a distributive law between the monad $\mathbf{S}$ and the comonad $!\_$. This allows to extend $\mathbf{S}$ to a strong monad D on the Kleisli category $\mathcal{L}_!$ which implements differentiation of non-linear maps. In Section 4 we study the case where the functor $\mathbf{S}$ can be defined using a more basic structure of $\mathcal{L}$ based on the object $1 \,\&\, 1$ where $\&$ is the cartesian product and 1 is the unit of $\otimes$: this is actually what happens in

Thomas Ehrhard. Coherent differentiation. 2021

Dialectica is differentiation ...

... We knew it already !

The codereliction of differential proof nets: In terms of polarity in linear logic [23], the $\forall$-$\rightarrow$-free constraint characterizes the formulas of intuitionistic logic that can be built only from positive connectives ($\oplus$, $\otimes$, 0, 1, !) and the why-not connective ("?"). In this framework, Markov's principle expresses that from such a $\forall$-$\rightarrow$-free formula A (e.g. $?\oplus_x (?A(x) \otimes ?B(x))$) where the presence of "?" indicates that the proof possibly used weakening (`efq` or `throw`) or contraction (`catch`), a linear proof of A purged from the occurrences of its "?" connective can be extracted (meaning for the example above a proof of $\oplus_x(A(x) \otimes B(x))$). Interestingly, the removal of the "?", i.e. the steps from $?P$ to P, correspond to applying the codereliction rule of differential proof nets [24].

Differentiation : $(?P = (P \multimap \bot) \Rightarrow \bot) \rightarrow ((P \multimap \bot) \multimap \bot) \equiv P)$

Hugo Herbelin, "An intuitionistic logic that proves Markov's principle", LICS '10 .

Differentiation and delimited continuations

Herbelin Lics'10

Markov's principle is proved by allowing `catch` and `throw` operations on hereditary positive formulas.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a : \neg\neg T \vdash_{\alpha:T} a : \neg\neg T}\,\text{AXIOM} \qquad \dfrac{\dfrac{\dfrac{}{b : T \vdash_{\alpha:T} b : T}\,\text{AXIOM}}{b : T \vdash_{\alpha:T} \mathtt{throw}_\alpha\, \mathtt{b} : \bot}\,\text{THROW}}{\vdash_{\alpha:T} \lambda b.\mathtt{throw}_\alpha\, \mathtt{b} : \neg \mathtt{T}}\,\to_I}{a : \neg\neg T \vdash_{\alpha:T} a\,(\lambda b.\mathtt{throw}_\alpha\, \mathtt{b}) : \bot}\,\to_E}{a : \neg\neg T \vdash_{\alpha:T} \mathtt{efq}\ \mathtt{a}\,(\lambda \mathtt{b}.\mathtt{throw}_\alpha\, \mathtt{b}) : \mathtt{T}}\,\bot_E}{a : \neg\neg T \vdash \mathtt{catch}_\alpha\ \mathtt{efq}\ \mathtt{a}\,(\lambda \mathtt{b}.\mathtt{throw}_\alpha\, \mathtt{b}) : \mathtt{T}}\,\text{CATCH}}{\vdash \lambda a.\mathtt{catch}_\alpha\ \mathtt{efq}\ \mathtt{a}\,(\lambda \mathtt{b}.\mathtt{throw}_\alpha\, \mathtt{b}) : \neg\neg \mathtt{T} \to \mathtt{T}}\,\to_I
$$

Figure 3. Proof of MP

Proof Mining

Extracting quantitative information from proofs.

Effective moduli from ineffective uniqueness proofs. An unwinding of de La Vallée Poussin's proof for Chebycheff approximation*

Ulrich Kohlenbach

Fachbereich Mathematik, J.W. Goethe Universität

Robert Mayer Str. 6 10, 6000 Frankfurt am Main, FRG

Abstract

We consider uniqueness theorems in classical analysis having the form

$$(+)\ \forall u \in U, v_1, v_2 \in V_u \big(G(u, v_1) = 0 = G(u, v_2) \to v_1 = v_2\big),$$

where U, V are complete separable metric spaces, V_u is compact in V and $G : U \times V \to \mathbb{R}$ is a constructive function.
If $(+)$ is proved by arithmetical means from analytical assumptions

$$(++)\ \forall x \in X \exists y \in Y_x \forall z \in Z \big(F(x, y, z) = 0\big)$$

only (where X, Y, Z are complete separable metric spaces, $Y_x \subset Y$ is compact and $F : X \times Y \times Z \to \mathbb{R}$ constructive), then we can extract from the proof of $(++) \to (+)$ an effective modulus of uniqueness, i.e.

$$(+++)\ \forall u \in U, v_1, v_2 \in V_u, k \in \mathbb{N} \big(|G(u, v_1)|, |G(u, v_2)| \leq 2^{-\Phi u k} \to d_V(v_1, v_2) \leq 2^{-k}\big).$$

Differentiate the function $(\epsilon \to \eta)$ in :

$$\forall u, v_1 v_2, \forall \epsilon > 0, \exists \eta > 0, \| G(u, v_1) - G(u, v_2)| < \eta \to d_V(v_1, v_2) < \epsilon \|$$