On Dialectica and Differentiation, via Categories

Davide Barbarossa[0000-0003-4608-8282]

Department of Computer Science, University of Bath, UK
db2437@bath.ac.uk
https://davidebarbarossa12.github.io/

Abstract. Gödel's Dialectica has been introduced and developed as a logical transformation. Only recently has it been related with the, *a priori* unrelated, notion of differentiation: we can now read it as a differentiable program transformation. Building on that, we formulate the relation between these two notions categorically, in the framework of differential categories. Moreover, we study the relation between differential categories and Dialectica categories. We do this by taking the point of view of fibrations and (dependent) lenses, which allows us to keep a geometrical intuition in mind by considering reverse tangent categories. The viewpoint we propose opens many interesting further developments.

Keywords: Differential Categories · Dialectica Transformation · (Dependent) Lenses · Lambda-Calculus.

1 Introduction

Reverse differentiation and its categorical formulation We define in school the derivative of a function on the reals as a certain limit, and its differential (giving the error in output for a certain error in input) as the product of the derivative at a point times the input error. Using partial derivatives, one generalises this to differentiable maps $f : \mathbf{R}^n \to \mathbf{R}^m$, for which we learned to define its differential as $Df : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}^m$, linear in (say) the second argument, as the directional derivative $Df(a, v) := J_a f \cdot v$ ($J_a f$ being the Jacobian of f at a). Considered as a function of v (thus, linear), we obtain what is called the pushforward $f_* a$ of f at $a \in \mathbb{R}^n$. This situation can be abstracted by Cartesian differential categories [2], where D is an operator on homsets, and the Cartesian *closed* version admits the simply typed differential λ-calculus (ST$\partial\lambda$C for short) [13] as internal language [5,14,21]. It is also natural to consider the case of smooth maps between smooth manifolds [18] as the natural setting for differentiation, as the same constructions can be carried out by working in a local system of coordinates: one constructs the tangent bundle $TA := \sum_{a \in A} T_a A$ of a manifold A and the pushforward of $f : A \to B$ as a linear map on the tangent spaces, which one can think of as a dependent function $f_* : \prod_{a \in A} [T_a A \multimap T_{f(a)} B]$ (we borrow the notation $\multimap$ from Linear Logic for linear functions). The differential is now a map $Tf : (a, v) \in TA \to (fa, f_* av) \in TB$, which gives rise to a functor T. Also this situation can be abstracted categorically, in the so-called tangent categories [6], where T is a given endofunctor. The case of real functions is the one of Euclidean spaces where the tangent spaces are all isomorphic to the base space and thus the tangent bundle trivialises to $TA \simeq A \times A$. All this is can be called *forward* differentiation, as the functoriality of T amounts to the forward mode for computing the chain rule. It is well-known that one can formulate these data in a different way: since f_*a is linear, we can take its dual map $(f_*a)^\perp : T^*_{fa}B \multimap T^*_a A$ (which is defined, in local coordinates, by the transpose of the Jacobian) and consider the *reverse differential* of f as $Rf : (a, v) \in f^*T^*B \to (a, (f_*a)^\perp v) \in T^*A$, where $f^*T^*B := \sum_{a\in A} T^*_{fa}B$ is the pullback of the cotangent bundle $T^*B := \sum_{b\in B} T^*_b B$ along f. Remarking that $T^\perp A := \prod_{a\in A} T^*A$ is precisely the space of diffrential 1-forms, yet another common way of expressing R is to see it as a mapping $T^\perp f : T^\perp B \to T^\perp A$ of differential 1-forms on B to ones on A. In this way $T^\perp$ becomes a *controvariant* functor, so $T^\perp(f \circ g) = T^\perp g \circ T^\perp f$, and this corresponds to the backward mode for computing the chain rule, used e.g. in machine learning algorithms. We will however stick to R instead of $T^\perp$. Reverse differentiation is abstracted in reverse tangent categories [9], where we do not directly have a functor $T^\perp$, but rather we have T together with an involution $(\_)^*$ on fibrations that allows one to construct R. In the Euclidean case where $T^*A \simeq TA \simeq A \times A$, the reverse differential of f is $Rf : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^n$ (note the swap of n and m with respect to Df), linear in the second argument. This situation is abstracted by Cartesian reverse differential categories [4,8], a particular case of reverse tangent ones, where R is a given operator on homsets.

Dialectica, its program-theoretical formulation and its links with differentiation In [15] Gödel defined a transformation $A \mapsto A_D$, known as "Dialectica", from intuitioninstic arithmetic's (say, HA) formulas to System T one. The transformed formula A_D contains two additional kinds of free variables, playing the role of witnesses (w) and counters (c) for A, so $A_D = A_D\{w, c\}$. The theorem is that provability in the source system yields provability in the target, thus obtaining a relative consistency result with respect to the finitistic methods of T, see e.g. [1]: if $\vdash_{\mathrm{HA}} A$ then there are terms $M \in T$ such that $\vdash_{\mathrm{T}} A_D\{M, c\}$ (where c is free). Dialectica has been later broadly applied in the field of proof-mining [17]. Much more recently, on an orthogonal direction, by taking a proof/program-theoretical point of view, i.e. looking at Dialectica as a transformation of proofs (and not just a transformation of formulas preserving provability), [22,23] showed that it is also a genuine program transformation. If we restrict to the simply typed λ-calculus (Λ for short), then one can define a variant, let us call it $\mathbf{P}$, of System T and see Dialectica as a transformation $(\_)^\bullet$, mutually defined with transformations $(\_)_x$ (for each variable x), of (even untyped) programs from Λ to $\mathbf{P}$. Even if this correspondence involves a definitely logically poor source system (and so not suited for, say, proof mining), Λ contains the arrow type, which is the most delicate case in Gödel's Dialectica, and it is definitely interesting from a programming point of view, since it represents high-order computation. The target $\mathbf{P}$ has simple types with products, plus a monadic type constructor $\mathcal{M}[\_]$ (similar to the Diller-Nahm variant of Dialectica [11]), together with two functions W (witness) and C (counter) from simple types to $\mathbf{P}$-types. The relative consistency theorem of Gödel becomes then a soundness theorem for such transformations: if $x : A \vdash_{\Lambda} M : B$ then $x : W(A) \vdash_{\mathbf{P}} M^{\bullet} : W(B)$ and $x : W(A) \vdash_{\mathbf{P}} M_x : C(B) \to \mathcal{M}[C(A)]$ This has several interesting consequences, e.g. understanding Dialectica as a sort of delimited continuation mechanism. But the one of interest for this paper is that it allowed, in [16], to notice that the types of $M^{\bullet}$ and M_x perfectly match the ones of a function and its reverse differential at a point x. This is not just a coincidence, as they show that Dialectica is indeed a differentiable program transformation: given M a simply typed λ-term, $M^{\bullet}$ computes the differential of M, implementing a reverse differentiation algorithm (i.e. by computing the chain rule as the reverse differential), and this is precisely where the $(\_)_x$ appears. The authors show this in several ways, most notably by defining the logical relations at [16, Def. 4.8] between the Dialectica translated programs in $\mathbf{P}$ and the *differential* simply typed λ-calculus, the latter being the syntactical way of handling differentiation in λ-calculus.

This paper

This paper is divided in two different, but related, main sections, both stemming from [16], with the aim of deepening some of its key points. We conclude in Section 4 with many future work directions opened by our point of view.

Section 2:Dialectica and reverse differentiation In [16], the authors explain that Dialectica is a differentiable program transformation by giving, among others, a logical relation $\sim$ (and an auxiliary one, $\bowtie$) between the target language of the transformation and the differential λ-calculus (ST$\partial\lambda$C for short). The central theorem relates then the image of the transformation of a ordinary λ-term with a certain ST$\partial\lambda$C-term. The trained differential λ-calculist can understand why that term indeed encodes the reverse differentiation of the first, even if this is not made explicit in the paper. We propose to explain the connection between Dialectica and reverse differentiation by defining the analogue of the logical relations $\sim$, $\bowtie$ using natural concepts from differential geometry (see for example [18]), namely that of pushforward and, in Section 3, pullbacks. In order to handle λ-calculus constructions we have to move to the world of Cartesian closed differential categories. We achieve this in Theorem 1, Corollary 1 and Proposition 1. The main notions are explained in Subsection 2.1. Moreover, the other reason why we choose a categorical setting is because the ST$\partial\lambda$C encodes *forward* differentiation (the term $\lambda v.\, D[\lambda x.M, v]N$ is the pushforward of M, as a function of x, at N, i.e. its directional derivative at N); but since we know that Dialectica performs a *reverse* differentiation, we would like to express it in a already reverse setting, instead of hiding this reverse operation in the syntactic definition of $\sim$, $\bowtie$. Now, while there is no such a thing as a "reverse differential λ-calculus" in the literature, Cartesian reverse differential (even tangent) categories do exist [4,9]; however, there is no closed version of them yet. But we can still use Cartesian closed differential categories equipped with enough structure in order to express reverse differentiation: that is the framework that we take in Subsection 2.1.

Section 3: Differentiation, from Dialectica categories to Lenses Dialectica categories [10] are a categorical construction which mimics the Dialectica transformation. It is quite natural, by looking at it (see Definition 2), to see a strong analogy with the typing of the reverse differential of a function (an arrow in the base category). Despite this intuition being natural, to the best of our knowledge has only been quickly pointed out in [16], where the authors say "*In our point of view, objects of Dialectica category generalize the relation between a space and its tangent space*". We argue that the correct intuition should involve *co*tangent spaces instead (see Remark 5). Moreover, given a Cartesian closed differential category $\mathcal{L}$, they define a functor from $\mathcal{L}$ to its Dialectica category, and one needs enough structure on $\mathcal{L}$ (see Subsection 2.1) in order to talk about reverse differentials of arrows. They suggest that their construction can be generalised. We will see that the functor does not really use the fact that we start from a Dialectica category, in that it does not use subobjects. In fact, it immediately lifts to (dependent) lenses (Corollary 3, Proposition 6), and we can indeed generalise it to reverse differential/tangent categories (Proposition 7, 8). This allows to keep a geometrical intuition in mind, but we loose the closedness condition and, thus, the possibility of interpreting high-order languages. Importantly, in Proposition 3 and Corollary 2 we show how the syntactical Dialectica transformation of Section 2 can be expressed in the same categorical structure of lenses.

Remark 1. In all the paper we take, both from in an intuitive way and in a formal way, a point of view based on fibrations and pullbacks. It must be remarked that, while writing this paper, we realised that a similar viewpoint has just very recently been explored in the preprint [3]. However, the direction taken there is orthogonal to ours: we both look at Dialectica categories as a category of lenses, but while we are interested in the links with reverse differentiation and Dialectica as a program transformation, they are interested in lifting the Dialectica construction on formulas (i.e. with non-trivial subobjects) to lenses.

Notations We write $f;g$ for the composition of $f : A \to B$ and $g : B \to C$ in a category. We write 1_A for the identity arrow on A, and we drop A when clear from the context. If $\times$ denotes some product, we write $\pi_i^{A_1,A_2} : A_1 \times A_2 \to A_i$ for the projections (and we drop the "A_1, A_2" if clear from the context). If a category is closed, we denote λ/λ^{-1} its curry/uncurry operators. If the products are symmetric (as it will always be), we confuse them, as well as $\lambda : \mathcal{C}(A \times B, C) \to \mathcal{C}(A, C^B)$ and $\lambda : \mathcal{C}(B \times A, C) \to \mathcal{C}(A, C^B)$ and the same for uncurry.

2 Dialectica and reverse differentiation

2.1 The framework

Dialectica in Pédrot's system **P** The calculus is given in [16, Fig. 1 and Def. 4.1], but for the sake of clarity let us we recall here its the main features.

The first layer of syntax is that of a simply typed λ-calculus with pairs (notation:$\langle M, N\rangle$ for pairs and M^i for projections) and product types (notation:

	α (ground types)	$E \to F$
W	α_W	$(W(E) \to W(F)) \times (W(E) \times C(F) \to \mathcal{M}[C(E)])$
C	α_C	$W(E) \times C(F)$

Fig. 1. Witnesses and Counters of a simple type. α_W and α_C are fixed ground types of **P** associated with α. Remark that in the second componenet of the witness of an arrow, we take a slightly different version, but equivalent, than the original, which would curry our type as $C(F) \to W(E) \to \mathcal{M}[C(E)]$. This is just because we want to highlight the intuition of the reverse differentials: with dependent types in mind, $W(E) \times C(F) = C(E \to F)$ plays the role of $\sum_{e:E} T^*_{fe} F = f^* T^* F$, so we could not give $v : C(F)$ *before* $e : W(E)$ (what would v be cotangent at?).

	x	$\lambda x.M$	PQ
$(\_)^\bullet$	x	$\langle\, \lambda x.M^\bullet \,,\, \lambda\pi.(\lambda x.M_x)\pi^1\pi^2 \,\rangle$	$P^{\bullet 1} Q^\bullet$
$(\_)_y$	$\begin{cases} \lambda\pi.[\pi], & x = y \\ \lambda\pi.0, & x \neq y \end{cases}$	$\lambda\pi.(\lambda x.M_y)\pi^1\pi^2$	$\lambda\pi.(P_y\langle Q^\bullet, \pi\rangle + P^{\bullet 2}\langle Q^\bullet, \pi\rangle \gg\!= Q_y)$

Fig. 2. Untyped Dialectica transformation. Remark that we take a slightly different version, but equivalent, than the original, in order to fit with the modification mentioned in Figure 1. Notice that with a slight abuse of α-equivalence, we have $(\lambda x.M)^\bullet = \langle\, \lambda x.M^\bullet \,,\, (\lambda x.M)_x \,\rangle$, which is already reminiscent of the pair "(f, f_*)".

$A \times B$), quotiened under the usual $\beta\eta$-equality. On the top of that, we have a new monadic type constructor $\mathcal{M}[\_]$, together with its return and bind term constructors (notation: $[M]$ and $M \gg\!= N$), and a commutative monoid structure on it (notation: 0, $M + N$) which is compatible with the monad structure. All these equations (together with $\beta\eta$), constituting its equational semantics, are denoted by $=$. Finally, since we are considering Dialectica as a transformation of proofs, not just of formulas, the transformation is now given by two maps W, C from simple types to **P**-types and two maps $(\_)^\bullet, (\_)_x$ (for x any variable) from λ-terms to **P**-terms, inductively defined in Figure 1 and Figure 2. In [22] it is proved the soundness results mentioned in the introduction, as well as the computational interpretation of Dialectica, together with the proof of the fact that such transformation only depends on the equational semantics classes in **P**.

The ambient setting We fix a model $\mathcal{C}$ of classical Differential Linear Logic as the ambient where the constructions of this section will take place, i.e. $\mathcal{C}$ is left-additive enriched over commutative monoids (we use $0, +$ for the operations on the homsets) endowed with: a symmetric monoidal closed structure $(\otimes, 1)$, whose exponential objects we denote $[A \multimap B]$ and evaluation arrows $\mathrm{ev} : [A \multimap B] \otimes A \multimap B$, finite biproducts $(\&, \top)$, a strong monoidal comonad $(! : \mathcal{C} \to \mathcal{C}, d :! \to \mathrm{id}, p :! \to !!)$ (resp. called *bang*, *dereliction*, *digging*) and natural transformations $c :! \to ! \otimes !$ (*contraction*) and $w :! \to !\top$ (*weakening*) making it a storage modality, isomorphisms $!A \otimes !B \simeq !(A \& B)$ and $1 \simeq !\top$ making $\mathcal{C}$ Seely, a natural transformation $\overline{d} : \mathrm{id} \to !$ (*codereliction*), making $\mathcal{C}$ differential storage, an involutive functor $(\_)^\perp : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$ making $\mathcal{C}$ $\star$-autonomous and a natural bijection $\chi : \mathcal{C}(D \otimes E, F) \simeq \mathcal{C}(D, [F^\perp \multimap E^\perp])$. This means that a series of equations are required, for which we refer to the standard references. We systematically use the notation $A \multimap B$ for arrows in $\mathcal{C}$.

It is well-known that with the above data one can also define natural transformations $\overline{c} : !\otimes! \to !$ (*cocontraction*), $\overline{w} : !\top \to !$ (*coweakening*) and $\partial : \mathrm{id}\otimes! \overset{\overline{d}\otimes 1}{\multimap} !\otimes! \overset{\overline{c}}{\multimap} !$ (*deriving transformation*), and set the *differential of* $f : !A \multimap B$ *in* $\mathcal{C}$ be $\partial f := A\otimes!A \overset{\partial}{\multimap} !A \overset{f}{\multimap} B$. It is well-known that the coKleisli $\mathcal{C}_!$ (same objects as $\mathcal{C}$ and $\mathcal{C}_!(A,B) := \mathcal{C}(!A, B)$, representing non-linear arrows) is a Cartesian closed differential category. We systematically use the notation $A \to B$ for arrows in $\mathcal{C}_!$. Its products are $A \times B := !A\otimes!B \simeq !(A\&B)$, its exponential objects $[A \to B] := [!A \multimap B]$ and the *differential of* $f \in \mathcal{C}_!(A,B)$ *in* $\mathcal{C}_!$ is $Df := d \otimes 1; \partial f \in \mathcal{C}_!(A \times A, B)$. For $f : A \to B$ we have its *promotion* $f^! := !A \overset{p}{\multimap} !!A \overset{!f}{\multimap} !B$. For (finitely many) elements $a_i : \top \to A_i$ of A_i (notation: $a_i : A_i$) and $f : \prod_i A_i \to B$, we define the element $f(\boldsymbol{a}) : B$ as $1 \overset{\simeq}{\multimap} \bigotimes_i 1 \overset{\bigotimes_i a_i^!}{\multimap} \bigotimes_i !A_i \overset{f}{\multimap} B$. For $f : A \to B$ we call its *pushforward* the arrow $f_* : A \overset{\lambda\partial f}{\to} [A \multimap B]$ and $f_* a : A \overset{\lambda^{-1}(f_*(a))}{\multimap} B$ its pushforward at $a : A$. Finally, for $f : A \multimap [E \multimap F]$ and $e : E$, we let $f\big|_e := A \overset{\simeq}{\multimap} A \otimes 1 \overset{1\otimes e}{\multimap} A \otimes E \overset{\lambda^{-1}f}{\multimap} F$.

2.2 Relating Dialectica and Reverse Differentials

We fix now an interpretation of ground simple types in $\mathcal{C}$ extended in the canonical way to all simple types (we still write A for the interpretation of the simple type A in $\mathcal{C}$). Taking inspiration from [16], we define two logical relations $\sim$ and $\bowtie$ relating a closed Dialectica-transformed program of $\mathbf{P}$ (i.e. the image of a proof under Dialectica) with arrows of the suited type in the ambient.

Definition 1. *Given, for any ground type* α *and simple type* A*, two relations* $\sim_\alpha \subseteq \{\vdash_{\mathbf{P}} M : \alpha\} \times \mathcal{C}_!(\top, \alpha)$ *and* $\bowtie^A_\alpha \subseteq \{\vdash_{\mathbf{P}} M : \alpha \to \mathcal{M}_A\} \times \mathcal{C}_!(A, \alpha) \times \mathcal{C}(\alpha^\perp, A^\perp)$*, we lift them at all simple types* B *in order to get relations*

$$\begin{array}{l} \sim_B \subseteq \quad \{\vdash_{\mathbf{P}} M : W(B)\} \quad \times \quad \mathcal{C}_!(\top, B) \\ \bowtie^A_B \subseteq \{\vdash_{\mathbf{P}} M : C(B) \to \mathcal{M}[C(A)]\} \times \mathcal{C}_!(A, B) \times \mathcal{C}(B^\perp, A^\perp) \end{array}$$

by mutual induction on B *as follow (the only case is* $B = E \to F$*):*

- *for* $\vdash_{\mathbf{P}} M : (W(E) \to W(F)) \times (W(E) \times C(F) \to \mathcal{M}[C(E)])$ *and* $f : [E \to F]$*, we set* $M \sim_{E \to F} f$ *iff for all* $H \sim_E e$*, we have*

$$M^1 H \sim_F f\big|_e : F \qquad \lambda\pi.M^2\langle H, \pi\rangle \bowtie^E_F \begin{pmatrix} \lambda^{-1} f : E \to F \\ ((\lambda^{-1} f)_* e)^\perp : F^\perp \multimap E^\perp \end{pmatrix}$$

- *for* $\vdash_{\mathbf{P}} M : W(E) \times C(F) \to \mathcal{M}[W(A)]$*,* $f : A \to [E \to F]$*,* $g : [E \to F] \multimap A$*, we set* $M \bowtie^A_{E \to F} (f, g)$ *iff for all* $H \sim_E e$*, we have*

$$\lambda\pi.M\langle H, \pi\rangle \bowtie^A_F \left(f\big|_e : A \to F \quad , \quad g^\perp\big|_e^\perp : F^\perp \multimap A^\perp \right).$$

$$\frac{M \sim_\alpha f}{N \sim_\alpha f}(\textit{if } M = N) \qquad \frac{M \bowtie^A_\alpha (f,g)}{N \bowtie^A_\alpha (f,g)}(\textit{if } M = N) \qquad \frac{G \bowtie^A_D (h,g) \qquad M \bowtie^D_\alpha (f,s)}{\lambda\pi.(M\pi \ggcurly G) \bowtie^A_\alpha (h^!; f, s; g)}(\ggcurly)$$

$$\frac{}{\lambda\pi.0 \bowtie^A_\alpha \binom{f:A\to\alpha}{0:\alpha^\perp \multimap A^\perp}}(0) \quad \frac{M_1 \bowtie^A_\alpha \binom{f:A\to\alpha}{g_1:\alpha^\perp \multimap A^\perp} \qquad M_2 \bowtie^A_\alpha \binom{f:A\to\alpha}{g_2:\alpha^\perp \multimap A^\perp}}{\lambda\pi.(M_1\pi + M_1\pi) \bowtie^A_\alpha \binom{f:A\to\alpha}{g_1+g_2:\alpha^\perp \multimap A^\perp}}(+) \quad \frac{}{\lambda\pi.[\pi] \bowtie^\alpha_\alpha (d,1)}(d)$$

$$\frac{M_1 \sim_{A_1} a_1 : A_1 \quad \overset{(n\geq 1)}{\cdots} \quad M_n \sim_{A_n} a_n : A_n}{\lambda\pi.[\langle M_1, \ldots, M_n, \pi\rangle] \bowtie^A_\alpha \binom{\mathrm{eval}_{\boldsymbol{a}}:[A_1\to[A_2\to\cdots\to[A_n\to\alpha]]]\to\alpha}{((\mathrm{eval}_{\boldsymbol{a}})_*0)^\perp:\alpha^\perp\multimap[[[\alpha^\perp\multimap A_n^\perp]\multimap A_{n-1}^\perp]\multimap\ldots\multimap A_1^\perp]}}(\mathrm{eval})$$

Fig. 3. In (eval), we let $\mathrm{eval}_{\boldsymbol{a}} : [A_1 \to [A_2 \to \cdots \to [A_n \to B]]] \to B$ be the composition $[1,n] \overset{d}{\to} [1,n] \overset{1_{A_1}\big|_{a_1^!}}{\to} [2,n] \to \cdots \to [n,n] \overset{1_{A_n}\big|_{a_n^!}}{\to} B$, where we put $[i,n] := [A_i \to [A_{i+1} \to \cdots \to [A_n \to B]]]$.

The acquainted differential λ-calculist could relate the arrows in Definition 1 and the terms of [16, Def. 4.8], putting the appropriate duality. In Proposition 1 we show in which sense the two constructions are equivalent. But (one of) the point of our definition is explicit these constructions using the familiar operations of pushforwards and dual maps. Finally, remembering cotangent spaces and their dependently typed nature, the g in $M \bowtie^A_B (f,g)$ should really be understood as $g : T^*_{f(a)}B \multimap T^*_a A$, for an $a \in A$.

Lemma 1. *Suppose $\sim_\alpha, \bowtie^A_\alpha$ are closed w.r.t. the rules of Figure 3. Then the same holds for $\sim_B$ and $\bowtie^A_B$ for all simple type B.*

Proof. Each rule is proved separately by induction on B, except rules (ev) and (d) which are proved by mutual induction on B. The lift of the compatibility with equational equivalence is immediate. The others are all straightforward using the equational semantics of **P** [16, Def. 4.1] and equations which hold in $\mathcal{C}$: (0) uses $0^\perp = 0$. (+) uses $(f+g)^\perp = f^\perp + g^\perp$, $\lambda^{-1}(f+g) = \lambda^{-1}f + \lambda^{-1}g$ and $f;(g+h) = f;g + f;h$. ($\ggcurly$) uses $(h^!;f)\big|_e = h^!;f\big|_e$ and $(s;g)^\perp\Big|_e^\perp = s^\perp\big|_e^\perp;g$. ($d$) uses the inductive hypothesis on (eval) and $\mathrm{eval}_e = d\big|_e$. (eval) uses the fact that $\mathrm{eval}_{\boldsymbol{a}}\big|_e = \mathrm{eval}_{\boldsymbol{a},e}$.

For $f : \prod_i A_i \to B$ and $a_i : A_i$ for $i = 1, \ldots, j-1, j+1, \ldots, n$, we let $f^j_{\boldsymbol{a}} := !A_j \simeq \bigotimes_1^{j-1} 1 \otimes !A_j \otimes \bigotimes_1^{n-j} 1 \overset{\bigotimes_i a_i^! \otimes 1 \otimes \bigotimes_i a_i^!}{\multimap} \bigotimes_i !A_i \overset{f}{\multimap} B$.

Remembering that, with the notations of the following lemma, $c; (p \otimes q^!); \mathrm{ev}$ is the composition of p and q in $\mathcal{C}_!$, the statement expresses the chain rule in its pushforward form[1]. The acquainted differential λ-calculist will notice that it precisely corresponds to the definition of the *linear substitution* of an application.

[1] The usual undergraduate chain rule is obtained when p does not depend on A_j.

Lemma 2. *Let* $p : \prod_i A_i \to [E \to F]$ *and* $q : \prod_i A_i \to E$. *Let us momentarily write* $q\,;_!\,p := \prod_i A_i \overset{c}{\multimap} \bigotimes_i !A_i \otimes \bigotimes_i !A_i \overset{p\otimes q^!}{\multimap} [E \to F] \otimes !E \overset{\mathrm{ev}}{\multimap} F$. *Fix* j *and* $a_i : A_i$. *In* $\mathcal{C}$ *we have:* $((q\,;_!\,p)^j_{\boldsymbol{a}^j})_* a_j = (p^j_{\boldsymbol{a}^j})_* a_j\Big|_{q(\boldsymbol{a})} + ((q^j_{\boldsymbol{a}^j})_* a_j\,;\,(\lambda^{-1}(p(\boldsymbol{a})))_* q(\boldsymbol{a}))$.

From now on, we fix an interpretation $[\![\cdot]\!] : ST\lambda C \to \mathcal{C}_!$ and ground relations $\sim, \bowtie$ satisfying the hypotheses of Lemma 1.

Theorem 1. *Let* $f := [\![x : A_1, \ldots, x : A_n \vdash_\Lambda M : B]\!] : \prod_{i=1}^n A_i \to B$.

For all $\vdash_\Lambda N_i : A_i$ *and* $a_i : A_i$ *s.t.* $N_i \sim_{A_i} a_i$ $(i = 1, \ldots, n)$, *setting* $\boldsymbol{a} := a_1, \ldots, a_n$ *and* $\boldsymbol{a}^j := a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n$, *we have:*

1. $M^\bullet\{\boldsymbol{N}/\boldsymbol{x}\} \sim_B f(\boldsymbol{a})$.
2. *If* $1 \le j \le n \neq 0$, *the following rule is admissible for all simple type* Y*:*

$$\frac{G \bowtie^Y_{A_j} (\, h : !Y \multimap A_j \,,\, g : A_j^\perp \multimap Y^\perp \,)}{\lambda\pi.((M_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\})\pi \ggcurly G) \bowtie^Y_B (\, h^!; f^j_{\boldsymbol{a}^j} : !Y \multimap B \,,\, ((f^j_{\boldsymbol{a}^j})_* a_j)^\perp; g : B^\perp \multimap Y^\perp \,)}$$

Proof. Induction on M. Call (IH1), (IH2) the inductive hypotheses for claim 1,2.

Case $M = x_i$**.** Then $f = \pi_i$.

1). Our goal becomes $N_i \sim_{A_i} a_i$ which is in our hypotheses.

2). If $j = i$, we have $f^j_{\boldsymbol{a}^j} = d_{A_j}$ and one can show that $(f^j_{\boldsymbol{a}^j})_* a_j = d_* a_j = 1$. Our goal then becomes $G \bowtie^Y_{A_j} (h, g)$, which is precisely the premise of our rule. If $j \neq i$, We have $f^j_{\boldsymbol{a}^j} = w^!_{A_j}; a_i$ and one can show that $(f^j_{\boldsymbol{a}^j})_* a_j = (w^!_{A_j}; a_i)_* a_j = 0$. Our goal then becomes $\lambda\pi.0 \bowtie^Y_{A_i} (h^!; f^j_{\boldsymbol{a}^j}, 0)$, which is given by (0).

Case $M = \lambda y.Q$**,** $B = E \to F$. Then there is $\lambda^{-1} f = [\![\boldsymbol{x} : \boldsymbol{A}, y : E \vdash_\Lambda Q : F]\!]$.

1). We have to show that, given $H \sim_E e$, we have both $Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}, H/y\} \sim_F f(\boldsymbol{a})\big|_e$ and $Q_y\{\boldsymbol{N}/\boldsymbol{x}, H/y\} \bowtie^E_F (\, \lambda^{-1}(f(\boldsymbol{a}))\,,\, ((\lambda^{-1}(f(\boldsymbol{a})))_* e)^\perp \,)$. The former is given by (IH1), since $f(\boldsymbol{a})\big|_e = (\lambda^{-1} f)(\boldsymbol{a}, e)$. For the latter we have $Q_y\{\boldsymbol{N}/\boldsymbol{x}, H/y\} = \lambda\rho.(Q_y\{\boldsymbol{N}/\boldsymbol{x}, H/y\}\rho \ggcurly \lambda\eta.[\eta])$ so, using rule (d), this is precisely given by (IH2), since $d^!_E; (\lambda^{-1} f)^{n+1}_{\boldsymbol{a}} = \lambda^{-1}(f(\boldsymbol{a}))$.

2). Given $G \bowtie^Y_{A_j} (\, h : !Y \multimap A_j \,,\, g : A_j^\perp \multimap Y^\perp \,)$ and $H \sim_E e$, putting $P := M_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\})\pi \ggcurly G$, our goal is: $\widetilde{P} \bowtie^Y_F (\, (h^!; f^j_{\boldsymbol{a}^j})\big|_e \,,\, \big(((f^j_{\boldsymbol{a}^j})_* a_j)^\perp; g)^\perp\big|_e^{\perp} \,)$, where we put $\widetilde{P} := \lambda\rho.(\lambda\pi.P)\langle H, \rho\rangle$. Since $P = (\lambda y.Q_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\})\pi^1\pi^2 \ggcurly G$, we have $\widetilde{P} = \lambda\rho.(\lambda\pi.P)\langle H, \rho\rangle = \lambda\rho.(Q_{x_j}\{\boldsymbol{N}/\boldsymbol{x}, H/y\}\rho \ggcurly G)$. Now, by (IH2) on Q with (G, h, g), we precisely obtain $\widetilde{P} \bowtie^Y_F (\, h^!; (\lambda^{-1} f)^j_{\boldsymbol{a}^j, e} \,,\, (((\lambda^{-1} f)^j_{\boldsymbol{a}^j, e})_* a_j)^\perp; g \,)$. To conclude, one can see that $(h^!; f^j_{\boldsymbol{a}^j})\big|_e = h^!; f^j_{\boldsymbol{a}^j}\big|_e = h^!; (\lambda^{-1} f)^j_{\boldsymbol{a}^j, e}$ and $((f^j_{\boldsymbol{a}^j})_* a_j)\big|_e = ((\lambda^{-1} f)^j_{\boldsymbol{a}^j, e})_* a_j$ as well as $\big(((f^j_{\boldsymbol{a}^j})_* a_j)^\perp; g)^\perp\big|_e^{\perp} = ((f^j_{\boldsymbol{a}^j})_* a_j)\big|_e^{\perp}; g$.

Case $M = PQ$**.** Then $f = c; (p \otimes q^!); \mathrm{ev}$, where $p = [\![\boldsymbol{x} : \boldsymbol{A} \vdash_\Lambda P : E \to B]\!]$ and $q = [\![\boldsymbol{x} : \boldsymbol{A} \vdash_\Lambda Q : E]\!]$.

1). Since one sees that $f(\boldsymbol{a}) = p(\boldsymbol{a})\big|_{q(\boldsymbol{a})}$, our goal becomes showing that $(P^\bullet\{\boldsymbol{N}/\boldsymbol{x}\})^1(Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\}) \sim_B p(\boldsymbol{a})\big|_{q(\boldsymbol{a})}$. But (IH1) on P gives $(P^\bullet\{\boldsymbol{N}/\boldsymbol{x}\})^1 H \sim_B p(\boldsymbol{a})\big|_e$ for all $H \sim_E e$, and (IH1) on Q gives $Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\} \sim_E q(\boldsymbol{a})$, so we are done.

2). Given $G \bowtie^Y_{A_j} (\, h : !Y \multimap A_j \,,\, g : A_j^\perp \multimap Y^\perp \,)$, let $R := \lambda\eta.((Q_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\}\eta) \ggcurly G)$, $\widetilde{P}_\rho := (P_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\}\langle Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\}, \rho\rangle) \ggcurly G$ and $\widetilde{Q}_\rho := (P^{\bullet 2}\{\boldsymbol{N}/\boldsymbol{x}\}\langle Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\}, \rho\rangle) \ggcurly R$. Now our goal is: $\lambda\pi.((\lambda\rho.\widetilde{P}_\rho)\pi + (\lambda\rho.\widetilde{Q}_\rho)\pi) \bowtie^Y_B (\, h^!; f^j_{\boldsymbol{a}^j} \,,\, ((f^j_{\boldsymbol{a}^j})_* a_j)^\perp; g \,)$. By $(+)$ and Lemma 2, it is enough showing that $\lambda\rho.\widetilde{P}_\rho \bowtie^Y_B (h^!; f^j_{\boldsymbol{a}^j} \,,\, (p^j_{\boldsymbol{a}^j})_* a_j\big|_{q(\boldsymbol{a})}^\perp; g)$ and $\lambda\rho.\widetilde{Q}_\rho \bowtie^Y_B (h^!; f^j_{\boldsymbol{a}^j} \,,\, ((\lambda^{-1}(p(\boldsymbol{a})))_* q(\boldsymbol{a}))^\perp; ((q^j_{\boldsymbol{a}^j})_* a_j)^\perp; g \,)$. For the former, remark that IH1 on Q entails $Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\} \sim_E q(\boldsymbol{a})$. So one can see that IH2 on P precisely gives $\lambda\rho.\widetilde{P}_\rho \bowtie^Y_B (\, (h^!; p^j_{\boldsymbol{a}^j})\big|_{q(\boldsymbol{a})} \,,\, (((p^j_{\boldsymbol{a}^j})_* a_j)^\perp; g)^\perp\big|_{q(\boldsymbol{a})}^\perp \,)$, and it is easy to see that we obtained the desired pair of arrows. For the latter, on the one hand we notice that, by IH2 on Q, we have $R \bowtie^E_B (\, h^!; q^j_{\boldsymbol{a}^j} \,,\, ((q^j_{\boldsymbol{a}^j})_* a_j)^\perp; g \,)$. On the other hand, by IH1 on P, for all $H \sim_E e$ we have $\lambda\pi.P^{\bullet 2}\{\boldsymbol{N}/\boldsymbol{x}\}\langle H, \pi\rangle \bowtie^E_B (\, \lambda^{-1}(p(\boldsymbol{a})) \,,\, ((\lambda^{-1}(p(\boldsymbol{a})))_* e)^\perp \,)$. Now, we already remarked some lines above that $Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\} \sim_E q(\boldsymbol{a})$, thus putting $S := \lambda\pi.P^{\bullet 2}\{\boldsymbol{N}/\boldsymbol{x}\}\langle Q^\bullet\{\boldsymbol{N}/\boldsymbol{x}\}, \pi\rangle$, we have $S \bowtie^E_B (\, \lambda^{-1}(p(\boldsymbol{a})) \,,\, ((\lambda^{-1}(p(\boldsymbol{a})))_* q(\boldsymbol{a}))^\perp \,)$. But by rule $(\ggcurly)$ on R and S, we obtain $\lambda\rho.(S\rho \ggcurly R) \bowtie^Y_B (\, (h^!; q^j_{\boldsymbol{a}^j})^!; \lambda^{-1}(p(\boldsymbol{a})) \,,\, ((\lambda^{-1}(p(\boldsymbol{a})))_* q(\boldsymbol{a}))^\perp; ((q^j_{\boldsymbol{a}^j})_* a_j)^\perp; g \,)$. Now since $S\rho \ggcurly R = \widetilde{Q}_\rho$, one concludes by checking that $(h^!; q^j_{\boldsymbol{a}^j})^!; \lambda^{-1}(p(\boldsymbol{a})) = h^!; f^j_{\boldsymbol{a}^j}$.

Corollary 1. *Under the same hypotheses of Theorem 1(2), we have*

$$M_{x_j}\{\boldsymbol{N}/\boldsymbol{x}\} \bowtie^A_B (\, f^j_{\boldsymbol{a}^j} : A_j \to B \,,\, ((f^j_{\boldsymbol{a}^j})_* a_j)^\perp : B^\perp \multimap A_j^\perp \,).$$

Proof. Immediate using rule (d) as premise of the rule in Theorem 1(2).

Remark 2. For $x : A \vdash_\Lambda M : B$, the results above say that $(\lambda x.M)^\bullet \sim_{A\to B} [\![M]\!]$ and $(\lambda x.M_x)N \bowtie^A_B ([\![M]\!], ([\![M]\!]_* a)^\perp)$ for all $N \sim_A a$. Remembering the reverse differential $R[\![M]\!] : (a, w) \in [\![M]\!]^* T^* B \mapsto (a, ([\![M]\!]_* a)^\perp w) \in T^* A$ of $[\![M]\!]$, we can read it by saying that $\lambda x.M_x$ "represents" $R[\![M]\!]$. Working out a dependently typed framework in which this makes a precise sense is a very interesting goal.

The previous two results and the remark above express the Dialectica as a differentiable program transformation in a categorical way, hopefully clarifying even more the content of [16, Theorem 4.10] and the constructions involved (compare also [16, Fig. 6] with our Definition 1).

Remark 3. $\sim$ can be thought of as a "proof relevant" realisability relation: not only we realise a formula B with **P**-terms (the M's such that $M \sim_B f$, *for some*

$$\frac{M \sim_B [\![S]\!]}{M \overset{\partial\lambda}{\sim}_B S}(1) \qquad \frac{M \bowtie_B^A (f, [\![S]\!]^\perp)}{M \overset{\partial\lambda}{\bowtie}{}_B^A S}(2) \qquad \frac{M \overset{\partial\lambda}{\sim}_B S}{M \sim_B [\![S]\!]}(3) \qquad \frac{M \overset{\partial\lambda}{\bowtie}{}_B^A S \quad [\![S]\!] = f_* a}{M \bowtie_B^A (f, [\![S]\!]^\perp)}(4)$$

Fig. 4. From [16, Def. 4.8] to our Definition 1, and vice versa. We mean by $\overset{\partial\lambda}{\sim}$, $\overset{\partial\lambda}{\bowtie}$ the relations in [16, Figure 6]. The careful reader would notice that, rigorously speaking, one needs to slightly modify the term M when passing from the formulation in [16] and ours, because of the modifications mentioned at Figure 2. We leave it implicit since it is easy to recover this by following the types in Figure 1.

$f : B$), but we also cluster such realisers into classes whose terms realise a certain element $f : B$ (the statement that $M \sim_B f$). Theorem 1 becomes then the usual adequation theorem for realisability: a proof $\vdash_\Lambda M : B$ gives a realiser $M^\bullet$ of B, plus the information that the realiser also realises $[\![M]\!]$ itself. Dialectica is thus the program extraction transformation for such notion of realisability.

As previously mentioned, the following result explains the relation between [16, Def. 4.8] and our Definition 1: the latter appears slightly more general than the former, due to the supplementary hypothesis on $\mathcal{C}$ needed to have an equivalence.

Proposition 1. *Let* $[\![\_]\!]$ *be an interpretation* ST$\partial\lambda$C $\to \mathcal{C}_!$. *Suppose that the rules of Figure 4 hold when B is a ground type.*

If $[\![\_]\!]$ *is full complete (i.e. surjective on all homsets), then the rules lift to all B simple type and, moreover, [16, Theorem 4.10] follows from our Theorem 1 using* (1), (2), *and our Theorem 1 follows from theirs using* (3), (4).

Proof. By straightforward mutual induction on B. We do not give the details because it involves the relation in [16] which we did not report here. The requirement that $[\![\_]\!]$ be full complete is used in the case of (3) and (4).

Remark 4. From Figure 4 one sees that the formulation of the relations using the differential λ-calculus given in [16], explain the reverse differentiation content of Dialectica within a forward differentiation setting. In the absence of a reverse differential λ-calculus or Cartesian closed reverse differential categories (more on that in the final Section 4), our categorical setting allows to explain it within a setting in which one can explicitely talk about reverse differentiation, even though we still start from a forward setting. See also Remark 7.

3 Differentiation, from Dialectica Categories to Lenses

As we have seen in the previous section, Dialectica can be read as a program transformation which mimics the construction of the reverse differential of a morphism. Another way of formulating Dialectica in a categorical way is by means of Dialectica Categories [10]. The departing point of this section is the relation between the latter and the (categorical) notions of (reverse) differentiation.

We fix for all this section a category $\mathcal{L}$ with pullbacks.

Notations We denote[2] with $A \overset{f^*p}{\leftarrow} f^*\beta \overset{\overline{f}}{\to} \beta$ the pullback of a diagram $A \overset{f}{\to} B \overset{p}{\leftarrow} \beta$. The notation is inspired by the canonical example of category with pullbacks that we have in mind, that is the category **SMan** of smooth manifolds and smooth maps. In this case the pullbacks always exist and one thinks of p as a fiber bundle over B, its pullback $f^*\beta$ being the fibre bundle obtained by the disjoint union of the fibres of p with the appropriate topology which makes it a differential manifold (together with its projection map f^*p) and $\overline{f}$ uniquely determining f.

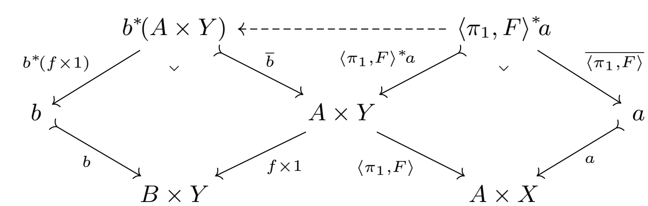
For example, the span $A \overset{p}{\leftarrow} \alpha \overset{1}{\to} \alpha$ is always the pullback of the diagram $A \overset{1}{\to} A \overset{p}{\leftarrow} \alpha$. With our notations, we have thus $(1_A)^*\alpha = \alpha$, $\overline{1_A} = 1_\alpha$ and $(1_A)^*p = p$. Similarly, the span $A \overset{1}{\leftarrow} A \overset{h}{\to} B$ is always the pullback of the diagram $A \overset{h}{\to} B \overset{1}{\leftarrow} B$. With our notations, we have thus $h^*B = A$, $\overline{h} = h$ and $h^*1_A = 1_B$. As another example, in any category with products $\times$, one can always take the pullback of a projection along any arrow: the pullback of $A \overset{f}{\to} B \overset{\pi_1}{\leftarrow} B \times Y$ is given by $f^*(B \times Y) = A \times Y$, $f^*\pi_1^{B,Y} = \pi_1^{A,Y}$ and $\overline{f} = f \times 1_Y$.

Finally, we denote a subobject a of an object A in a category by the abuse of notation $a \overset{a}{\rightarrowtail} A$. We thus mean the equivalence class of the mono a to A.

Definition 2. *The Dialectica Category [10]* Dial$(\mathcal{L})$ *over* $\mathcal{L}$ *is made of:*

Objects are the data of two objects A, X *in* $\mathcal{L}$ *and a subobject* $a \rightarrowtail A \times X$ *in* $\mathcal{L}$ *(in* **Set***,* a *is just a subset of* $A \times X$*, playing the role of a formula with two free variables, e.g. a binary predicate).*

An arrow from (A, X, a) *to* (B, Y, b) *is the data of an* $f : A \to B$ *and a* $F : A \times Y \to X$ *in* $\mathcal{L}$ *such that given the following diagram of pullbacks, there exists exactly one dotted arrow making the triangle commute:*



In **Set***, this reads as:* $(f(a), y) \in b$ *for all* $(a, y) \in A \times Y$ *s.t.* $(a, F(a, y)) \in a$.

The identity on (A, X, a) *is* $(1_A, \pi_2^{A,X})$ *(this is an arrow precisely because* a *is mono) and the composition* $(f, F); (g, G)$ *is* $(f; g, \langle \pi_1, (f \times 1); G \rangle; F)$*. One can immediately see that this is the same arrow given in [10, Proposition 1].*

Remark 5. In a setting where tangent and cotangent spaces are isomorphic, the typing of F in the previous definition is precisely that of the reverse differential of f. It is thus natural to wonder if that analogy can be pushed further and indeed also the composition in such category corresponds to the composition rule

[2] Remark that we make some standard abuse of notation here: the object $f^*\beta$ also depends on the arrow p, not only on its source β and f, and the same for $\overline{f}$.

for reverse differentials. As we will see, this happens because (a very special subcategory of) Dialectica categories can be put in relation with (dependent) lenses (Proposition 5), and the latter is a general framework to talk about constructions such as reverse differentiation (Section 3.3). Contrarily to the typing of F and its composition, the condition involving subobjects in the definition of an arrow of $\mathrm{Dial}(\mathcal{L})$ is not immediately clear in geometric terms. In fact, we will actually get rid of it in the following (e.g. Proposition 6, Corollary 3), as it appears not necessary in order to link Differential categories and Dialectica categories, and this will allow us to generalise it in Section 3.3. It may seem strange that we get rid of such very peculiar aspect of Dialectica; but we can understand this by looking at the computational formulation of Dialectica in $\mathbf{P}$: the subobjects (i.e. the formulas) are not there anymore, because their role (which is that of an orthogonality relation, see [23, Section 8.3.2 and 9.1.4]) is already subsumed and encoded by the witnesses (W) and counters (C), as shown by the soundness of the transformation with respect to them.

Lenses For the purpose of this paper, we take the following:

Definition 3. *The category* $\mathrm{Lens}(\mathcal{L})$ *of lenses over* $\mathcal{L}$ *is defined as follows:*

objects: arrows in $\mathcal{L}$*, which we think as fibre bundles and we write* $p : \binom{\alpha}{A}$ *instead of* $p : \alpha \to A$*;*

arrows from $p : \binom{\alpha}{A}$ *to* $q : \binom{\beta}{B}$ *are the data of both a* $f : A \to B$ *in* $\mathcal{L}$ *and a span* $\alpha \stackrel{F}{\leftarrow} f^*\beta \stackrel{\overline{f}}{\to} \beta$ *in* $\mathcal{L}$*, where* $\overline{f}$ *is part of the following pullback square in* $\mathcal{L}$*, and such that the left triangle commutes:*

$$\begin{array}{ccccc} \alpha & \xleftarrow{F} & f^*\beta & \xrightarrow{\overline{f}} & \beta \\ & \searrow^{p} & \downarrow{\scriptstyle f^*q} & \lrcorner & \downarrow{\scriptstyle q} \\ & & A & \xrightarrow[f]{} & B \end{array}$$

The identity on $p : \binom{\alpha}{A}$ *is given by* 1_A *and the identity span* $\alpha \stackrel{1}{\leftarrow} \alpha \stackrel{1}{\to} \alpha$*. Composition is given by pairwise composition in* $\mathcal{L}$ *and composition in the category* $\mathrm{Span}(\mathcal{L})$ *of spans on* $\mathcal{L}$*. One can check that these data satisfy the conditions for being arrows in* $\mathrm{Lens}(\mathcal{L})$*.*

Proposition 2. $\mathrm{Lens}(\mathcal{L})$ *is indeed a well-defined category.*

Proof. The only non trivial part is that our composition gives indeed an arrow of our claimed category, and that our claimed identity is indeed such. In both cases, the argument is similar to the pasting law for pullbacks. For the composition, one sees that our definition claims to take the composition $p : \binom{\alpha}{A} \stackrel{(f,F)}{\to} q : \binom{\beta}{B} \stackrel{(g,G)}{\to} r : \binom{\gamma}{C}$ to be $(f;g,\ \alpha \xleftarrow{\overline{f}^*G;F} \overline{f}^* g^*\gamma \xrightarrow{\overline{\overline{f}};\overline{g}} \gamma)$. To show that this is indeed an arrow in our claimed category, we consider the left diagram in Figure 5, which is read as follows: given the blue and purple pullbacks, and given the

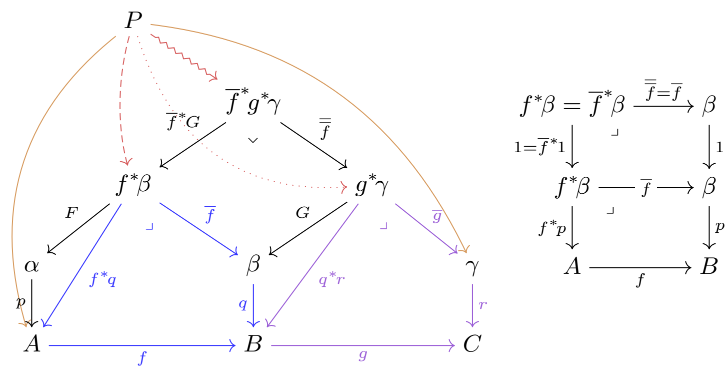


Fig. 5. Lens($\mathcal{L}_!$) is a category. Left: diagram for composition; right: for identities.

composition span (which is defined via the black pullback), in order to show that our composition is well defined, we show both that $A \overset{\overline{f}^*G;f^*q}{\longleftarrow} \overline{f}^*g^*\gamma \overset{\overline{\overline{f}};\overline{g}}{\longrightarrow} \gamma$ is the pullback of $A \xrightarrow{f;g} C \xleftarrow{r} \gamma$ and that $\overline{f}^*G;F;p = \overline{f}^*G;f^*q$. The latter is immediate (because (f,F) is an arrow). For the former, the commutation is immediate (because (g,G) is an arrow), and given the two orange arrows, one obtains the unique squiggly red arrow by first obtaining the unique dotted red arrow (using the purple pullback), then obtaining the unique dotted arrow (using the blue pullback), and finally the desired one (using the black pullback). For identities, one uses the fact that, because Span($\mathcal{L}$) is a category, the upper square of the right diagram of Figure 5 is a pullback as soon as the bottom one is.

Remark 6. The previous definition is basically that of *dependent lenses*. While writing this paper, we found that in the preprint [25, Example 3.7], it has been remarked that those are related with reverse differentiation. In another recent preprint [24, Page 6] one finds our diagram above, but it only appears in the context of polynomial functors. However, no relation with Dialectica nor (reverse) differential/tangent categories has been explicitly made. Summing up, while the point we want to make will probably not surprise the expert "Lens-theorist", we think that it is important to clearly state the links with other recent topics, namely (reverse) Differential/Tangent categories, and older ones like Dialectica categories, which we propose to do in this section.

Definition 4. *Let* $\mathcal{E}$Lens($\mathcal{L}$) *be the full subcategory of* Lens($\mathcal{L}$) *of trivial bundles, i.e. first projections. Concretely:*

- *Objects are first projections* $\pi_1 : \binom{A\times X}{A}$
- *An arrow from* $\pi_1 : \binom{A\times X}{A}$ *to* $\pi_1 : \binom{B\times Y}{B}$ *is given by an* $f : A \to B$ *and a span* $A \times X \xleftarrow{F} A \times Y \xrightarrow{f\times 1} B \times Y$ *such that* $F;\pi_1^{A,X} = \pi_1^{A,Y}$.

The definition above does make sense: the span satisfies the pullback condition of Lens($\mathcal{L}$) so that the arrows above are arrows in Lens($\mathcal{L}$), and the identities and composition are inherited from Lens($\mathcal{L}$). The only non-trivial part

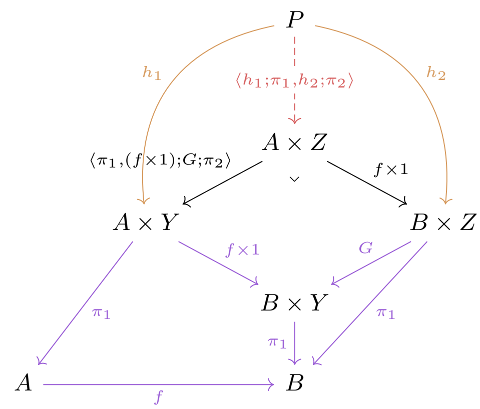


Fig. 6. Figure of Lemma 3.

is to justify that the arrows above are closed w.r.t. composition in Lens($\mathcal{L}$): this follows by the following Lemma 3, which ensures that the composition $\pi_1 : \binom{A\times X}{A} \stackrel{(f,F)}{\to} \pi_1 : \binom{B\times Y}{B} \stackrel{(g,G)}{\to} \pi_1 : \binom{C\times Z}{C}$ in Lens($\mathcal{L}$) of two arrows of $\mathcal{E}$Lens($\mathcal{L}$) is given by the following pair, which is clearly an arrow of $\mathcal{E}$Lens($\mathcal{L}$):

$$(\quad f;g \quad , \quad A\times X \stackrel{\langle \pi_1,(f\times 1);G;\pi_2\rangle;F}{\longleftarrow} A\times Z \stackrel{(f;g)\times 1}{\longrightarrow} C\times Z \quad).$$

We let the "$\mathcal{E}$" in "$\mathcal{E}$Lens($\mathcal{L}$)" stand for "Euclidean".

Lemma 3. *In a category with products, given a commutative square and a triangle as the ones in purple in Figure 6, the pullback of the diagram* $A\times Y \stackrel{f\times 1}{\longrightarrow} B\times Y \stackrel{G}{\longleftarrow} B\times Z$ *is the black one in the same figure.*

3.1 Dialectica and Euclidean-lenses

The structure of Euclidean-lenses is well-suited to shape the Dialectica transformation in $\mathbf{P}$: let Λ_{cat} and $\mathbf{P}_{\text{cat}}$ be the categories induced by the simply-typed λ-calculus and $\mathbf{P}$ as usual, i.e. whose objects are types, an arrow from A to B is the equational semantics class of a term $z : A \vdash M : B$, the identities are variables and composition is substitution. Now, $\mathbf{P}_{\text{cat}}$ does not have pullbacks in general, but we can still define the category $\mathcal{E}$Lens($\mathbf{P}_{\text{cat}}$) over it, exactly as in Definition 4 (ignoring that it is a subcategory of a whole category of lenses). With this, by looking at Figure 7, we can prove:

Proposition 3. *We have a functor* $\Lambda_{\text{cat}} \to \mathcal{E}\text{Lens}(\mathbf{P}_{\text{cat}})$ *defined as follows:*

- *An object* A *is sent to* $(z : W(A)\times\mathcal{M}[C(A)] \vdash_{\mathbf{P}} z^1 : W(A))$*;*
- *An arrow* $(z : A \vdash_{\Lambda} M : B)$ *in* Λ_{cat} *from* A *to* B *is sent to the arrow in* $\mathcal{E}\text{Lens}(\mathbf{P}_{\text{cat}})$ *from* $(z : W(A)\times\mathcal{M}[C(A)] \vdash_{\mathbf{P}} z^1 : W(A))$ *to* $(z : W(B)\times\mathcal{M}[C(B)] \vdash_{\mathbf{P}} z^1 : W(B))$ *given by* $(z : W(A) \vdash_{\mathbf{P}} M^\bullet : W(B))$ *and the span:*

$$W(A)\times\mathcal{M}[C(A)] \stackrel{\langle z^1, z^2 \gg= M_{(z^1)}\rangle}{\longleftarrow} W(A)\times\mathcal{M}[C(B)] \stackrel{\langle M^\bullet, z^2\rangle}{\longrightarrow} W(B)\times\mathcal{M}[C(B)].$$

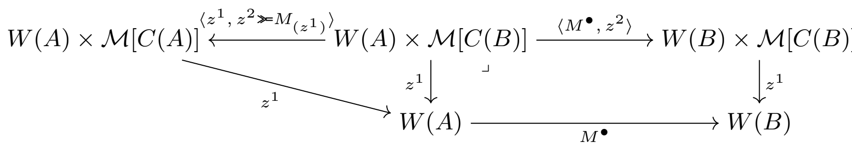


Fig. 7. One can check that, in $\mathbf{P}_{\text{cat}}$, the square is a pullback and the triangle commutes.

Proposition 4. *We have a functor* $G : \mathcal{E}\mathrm{Lens}(\mathcal{L}) \to \mathrm{Dial}(\mathcal{L})$ *defined as follows:*

- *An object* $\pi_1 : \binom{A\times X}{A}$ *is sent to* $(A, X, 1_{A\times X})$ *(here we see that we only use the full subobject, so we actually use a strict subcategoy of* $\mathrm{Dial}(\mathcal{L})$*);*
- *An arrow* $(\quad f : A \to B \quad , \quad A\times X \overset{F}{\longleftarrow} A\times Y \overset{f\times 1}{\longrightarrow} B\times Y \quad)$ *from* $\pi_1 : \binom{A\times X}{A}$ *to* $\pi_1 : \binom{B\times Y}{B}$ *is sent to* $(f,\ F;\pi_2)$ *from* $(A, X, 1_{A\times X})$ *to* $(B, Y, 1_{B\times Y})$*.*

One uses the trivial pullbacks of $B \times X \overset{1}{\to} B \times X \overset{f\times 1}{\leftarrow} A \times X$ and $A \times Y \overset{\langle \pi_1, F;\pi_2\rangle}{\to} A \times A \overset{1}{\leftarrow} A \times A$ in order to see that $G(f, F)$ is an arrow of $\mathrm{Dial}(\mathcal{L})$.

Proposition 5. *The functor* $G : \mathcal{E}\mathrm{Lens}(\mathcal{L}) \to \mathrm{Dial}(\mathcal{L})$ *is an isomorphism of categories on its image, which is the following full subcategory* $\mathcal{E}\mathrm{Dial}(\mathcal{L})$ *of* $\mathrm{Dial}(\mathcal{L})$*:*

- *Objects are given by full subobjects* $(A, X, 1_{A\times X})$
- *An arrow from* $(A, X, 1_{A\times X})$ *to* $(B, Y, 1_{B\times Y})$ *is given by arrows* $f : A \to B$ *and* $F : A \times Y \to X$ *(no condition is required here because the Dialectica condition becomes trivially satisfied for full subobjects).*

The inverse $G^{-1} : \mathcal{E}\mathrm{Dial}(\mathcal{L}) \to \mathcal{E}\mathrm{Lens}(\mathcal{L})$ *of* G *is given as follows:*

An object $(A, X, 1_{A\times X})$ *is sent to* $\pi_1^{A,X}$*.*
An arrow $(f : A \to B,\ F : A \times Y \to X)$ *from* $(A, X, 1_{A\times X})$ *to* $(B, Y, 1_{B\times Y})$ *is sent to* $(\quad f \quad , \quad A \times X \overset{\langle \pi_1, F\rangle}{\longleftarrow} A \times Y \overset{f\times 1}{\longrightarrow} B \times Y \quad)$ *from* $\pi_1^{A,X}$ *to* $\pi_1^{B,Y}$*.*

Proof. The fact that $\mathcal{E}\mathrm{Dial}(\mathcal{L})$ is a subcategory of $\mathrm{Dial}(\mathcal{L})$ is immediate to check. In order to show that $G^{-1}((f, F); (g, G)) = G^{-1}(f, F); G^{-1}(g, G)$ one uses the fact that $\langle \pi_1^{A,Z}, (f \times 1_Z); G)\rangle; \langle \pi_1^{A,Y}, F\rangle = \langle \pi_1^{A,Z}, \langle \pi_1^{A,Z}, (f \times 1_Z); G\rangle; F\rangle$, which can be immediately checked. In order to see that $(G; G^{-1})(f, F) = (f, F)$. one uses the fact that $F; \pi_1 = \pi_1$, which is given by the definition of $\mathrm{Lens}(\mathcal{L})$.

Corollary 2. *Composing Propositions 3, 5, one has a functor* $\Lambda_{\text{cat}} \to \mathcal{E}\mathrm{Dial}(\mathbf{P}_{\text{cat}})$ *sending* A *to* $(W(A), C(A), 1)$ *and* $(x : A \vdash_\Lambda M : B)$ *to the pair given by* $x : W(A) \vdash_{\mathbf{P}} M^\bullet : W(B)$ *and* $z : W(A) \times \mathcal{M}[C(B)] \vdash_{\mathbf{P}} z^2 \ggcurly M_{(z^1)} : \mathcal{M}[C(A)]$*. If we let* $z := \langle x, [y]\rangle$ *and remember that* $y \notin M_x$*, then the latter term becomes* $x : W(A) \vdash_{\mathbf{P}} M_x : C(B) \to \mathcal{M}[C(A)]$*.*

The above functor thus *literally* gives the Dialectica transformation in $\mathbf{P}$ together with its soundness Theorem mentioned in the introduction (first lines of Page 3). If Dialectica categories mimic Dialectica, as a *formulas* transformation, in categorical terms, we have here expressed Dialectica, as a *program* transformation, in categorical one. Even if not surprising, we think that it is instructive.

3.2 Forward Differentiation and Lenses

Let us fix in this subsection a Cartesian closed differential category $\mathcal{C}_!$ which is the coKleisli of a category $\mathcal{C}$ as in Section 2. Remember that $\mathcal{C}$ comes with a bijection $\chi : \mathcal{C}(D \otimes E, F) \simeq \mathcal{C}(D, [F^\perp \multimap E^\perp])$. We use it to define the *reverse differential of* $f :!A \to B$ *in* $\mathcal{C}$ as $\rho f :=!A \otimes B^\perp \overset{\lambda^{-1}\chi\partial f}{\multimap} A^\perp$, and the *reverse differential* $Rf \in \mathcal{C}_!(A \times B^\perp, A^\perp)$ *of* f *in* $\mathcal{C}_!$ as $!(A\&B^\perp) \simeq !A \otimes !(B^\perp) \overset{1\otimes d}{\multimap} !A \otimes B^\perp \overset{\rho f}{\multimap} A^\perp$.

Proposition 6. *We have a functor* $D : \mathcal{C}_! \to \mathcal{E}\mathrm{Lens}(\mathcal{C}_!)$ *defined by:*

$$\begin{array}{lcl} A & \mapsto & \pi_1 : \binom{A\times A^\perp}{A} \\ A \xrightarrow{f} B \mapsto (& f \quad , & A \times A^\perp \xleftarrow{\langle \pi_1, Rf\rangle} A \times B^\perp \xrightarrow{f\times 1} B \times B^\perp \quad). \end{array}$$

The statement is analogous to [16, Proposition 5.7], and so is its proof. We will prove a similar result in Proposition 7. We immediately have:

Corollary 3. *The functor in [16, Proposition 5.7] is actually the composition* $\mathcal{C}_! \overset{D}{\to} \mathcal{E}\mathrm{Lens}(\mathcal{C}_!) \overset{G}{\simeq} \mathcal{E}\mathrm{Dial}(\mathcal{C}_!) \hookrightarrow \mathrm{Dial}(\mathcal{C}_!)$.

This also shows, as we anticipated in Remark 5, we do not need all the power of Dialectica categories, i.e. the possibility of taking subobjects, in order to make a link with differentiation: we are only using its Euclidean-lens structure.

Remark 7. Remembering Remark 4 and Figure 4, one sees that the passage from $M \overset{\lambda\partial}{\bowtie} S$ to $M \bowtie S$ contains the same information as the functor D, both building the reverse differential of f, in the same way as the functor in Proposition 3.

3.3 Reverse Differentiation and Lenses

On the same spirit of Proposition 6, one should be able define a functor starting from a reverse tangent category. Let us first immediately see the case with trivial cotangent bundles, i.e. that of reverse differential categories (see [9, Example 28]):

Proposition 7. *Let* $\mathcal{L}$ *a Cartesian reverse differential category ([4, Definition 13]). We have a functor* $\mathcal{T}^* : \mathcal{L} \to \mathcal{E}\mathrm{Lens}(\mathcal{L})$ *defined by:*

$$\begin{array}{lcl} A & \mapsto & \pi_1 : \binom{A\times A}{A} \\ A \xrightarrow{f} B \mapsto (& f \quad , & A \times A \xleftarrow{\langle \pi_1, Rf\rangle} A \times B \xrightarrow{f\times 1} B \times B \quad) \end{array}$$

where $Rf : A \times B \to A$ *in* $\mathcal{L}$ *is the reverse differential of* f *(which is a primitive data in* $\mathcal{L}$*). Therefore, we also have a functor* $\mathcal{T}^*; G : \mathcal{L} \to \mathrm{Dial}(\mathcal{L})$*, factorising the very last lines of [16] as for Corollary 3.*

Proof. By diagram chasings. For the composition one reasons on Figure 8.

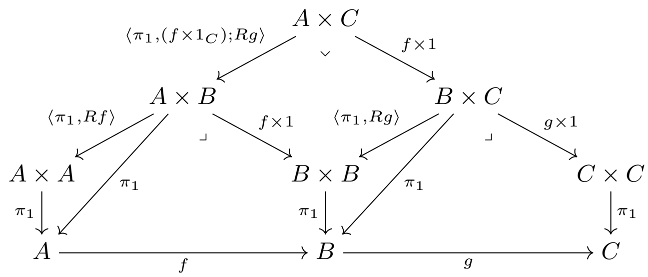


Fig. 8. Diagram for the proof of Proposition 7.

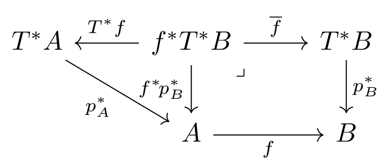
Let us now consider reverse tangent categories [9, Definition 24], whose canonical example is **SMan** ([9, Example 27]). Such a category $\mathcal{L}$ is, broadly speaking, a tangent category $\mathcal{L}$, i.e. a differential category with non-trivial tangent bundles $p_A : \binom{TA}{A}$ [9, Definition 1], equipped with: a full subcategory $\mathrm{DBun}_D(\mathcal{L})$ of $\mathcal{L}$ of differential bundles which behave like cotangent bundles ([9, Definition 16]); its canonical fibration $\mathrm{DBun}_D(\mathcal{L}) \to \mathcal{L}$ and dual fibration $\mathrm{DBun}^{\circ}_D(\mathcal{L}) \to \mathcal{L}$ ([9, Propositions 17, 21]); an involutive fibration morphism $\mathrm{DBun}_D(\mathcal{L}) \to \mathrm{DBun}^{\circ}_D(\mathcal{L})$ giving the dual bundle $p^* : \binom{\alpha^*}{A}$ of a differential bundle $p : \binom{\alpha}{A}$ [9, Definition 23].

Proposition 8. *Let $\mathcal{L}$ be a reverse tangent category. We have a functor $\mathcal{T}^*$: $\mathcal{L} \to \mathrm{Lens}(\mathcal{L})$ defined by:*

$$A \quad \mapsto \quad p^*_A : \binom{T^*A}{A}$$

$$A \xrightarrow{f} B \mapsto \left(\quad f \quad , \quad T^*A \xleftarrow{T^*f} f^*T^*B \xrightarrow{\overline{f}} T^*B \quad \right)$$

where p^*_A is the dual of the tangent bundle on A, $\overline{f}$ is part of the pullback square in $\mathcal{L}$ below and (f, T^*f) is the image of (f, Tf) under the involution of $\mathcal{L}$ ([9, Definition 23]), where (f, Tf) is defined in [7, Example 2.4(ii)].



Remark that the left triangle above commutes because because of [9, Proposition 21.(ii).(3), left diagram], so the functor $\mathcal{T}^*$ is well defined in $\mathrm{Lens}(\mathcal{L})$.

Remark 8. Actually, the point of a reverse tangent category, is that in it one can always define a reverse tangent bundle functor $\mathcal{T}^* : \mathcal{L} \to \mathrm{DBun}^{\circ}_D(\mathcal{L})$ as in [9, Definition 25]. Now, one can think of $\mathrm{DBun}^{\circ}_D(\mathcal{L})$ as a subcategory of $\mathrm{Lens}(\mathcal{L})$ (just consider the left diagram of [9, Proposition 21.ii(3)], i.e. ignore the differential bundle part): composing with the inclusion we immediately get the functor above. In conclusion if, after [9], Proposition 8 is not a surprise, the point we are making here is that this is the direct generalisation of the functor in [16, Proposition 5.7], something which was not remarked there.

4 Conclusions and future work

We took a categorical point of view on the relation between Dialectica and (reverse) differentiation; we did this by defining two logical relations (Definition 1) between the target language of the Dialectica transformation of simply-typed λ-terms, and arrows in an opportune class of differential categories. We showed that, on the image of the transformation, such relations perform reverse differentiation (Theorem 1, Corollary 1). We then investigated reverse differentiation in terms of lenses; we did this by considering reverse differential and tangent categories (Proposition 6, 7 8); we have also factored the functor in [16, Proposition 5.7] by remarking that it only uses a subcategory of the Dialectica category not involving non-trivial subobjects (Corollary 3), which can be seen as a subcategory of lenses (Proposition 5). Finally, we have seen how the same categorical structure shapes Dialectica as a program transformation (Proposition 3, Corollary 2). This point of view opens many natural and interesting future work:

1. As already pointed out in Remark 4, the first natural question is to express Definition 1 in a *reverse* differential category. This should be definitely possible, but the category should be Cartesian closed in order to interpret λ-calculus, and this has not been explored in the literature yet.
2. Related with the previous point is that of defining a "*reverse* differential λ-calculus. One could then formulate Definition 1 in order to relate it to Dialectica in a syntactic way. In a sense, such language already appears in the preprint [20], where the authors introduce a language precisely defined with the model of pullbacks of differential 1-forms in mind. A natural goal is therefore to express Dialectica within such calculus.
3. Remembering Remark 7, an interesting question is whether one can lift the relations in Definition 1 to dependently typed languages (or different logical systems). The natural starting point would be [22,23], where it is shown how to formulate Dialectica for dependent types. This would require both linear and dependent types and, on the categorical side, Cartesian *closed* reverse *tangent* categories which, again, have not been explored in the literature yet.
4. In our work there is a clear distinction between linear and non-linear arrows, handled by the use of models of differential linear logic on the lines of [12]. However, in the Cartesian closed case, this does not include geometrical models like **SMan**. The recent setting of *linearly closed reverse differential categories* of [19] allows to keep geometric examples while still being able to (un)curry linear maps. Can Sections 2 and 3 be phrased within it?
5. In Subsection 3.3 we considered the lens structure from a differential one. Can one conversely build a differential structure from a lens/Dialectica one?
6. In Section 3, the main point is really to have a functor $\mathcal{L} \to \mathrm{Lens}(\mathcal{L})$. Are reverse tangent categories the only ones admitting such functors?

Acknowledgments. Thanks to Thomas Powell for many instructive discussions about Dialectica. This work has been funded by the Engineering and Physical Sciences Research Council (EPSRC) project "Imperative programs from proofs" (EP/W035847/1).

References

1. Avigad, J., Feferman, S.: *Chapter V - Gödel's Functional ("Dialectica") Interpretation*. In: Buss, S.R. (ed.) Handbook of Proof Theory, Studies in Logic and the Foundations of Mathematics, vol. 137, pp. 337–405. Elsevier (1998). https://doi.org/https://doi.org/10.1016/S0049-237X(98)80020-7, https://www.sciencedirect.com/science/article/pii/S0049237X98800207
2. Blute, R., Cockett, J.R.B., Lemay, J.P., Seely, R.A.G.: *Differential Categories Revisited*. Appl. Categorical Struct. **28**(2), 171–235 (2020). https://doi.org/10.1007/S10485-019-09572-Y, https://doi.org/10.1007/s10485-019-09572-y
3. Capucci, M., Gavranovic, B., Malik, A., Rios, F., Weinberger, J.: *On a fibrational construction for optics, lenses, and Dialectica categories*. CoRR **abs/2403.16388** (2024), https://doi.org/10.48550/arXiv.2403.16388
4. Cockett, J.R.B., Cruttwell, G.S.H., Gallagher, J., Lemay, J.P., MacAdam, B., Plotkin, G.D., Pronk, D.: *Reverse Derivative Categories*. In: Fernández, M., Muscholl, A. (eds.) 28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain. LIPIcs, vol. 152, pp. 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPICS.CSL.2020.18, https://doi.org/10.4230/LIPIcs.CSL.2020.18
5. Cockett, J.R.B., Gallagher, J.: *Categorical Models of the Differential λ-Calculus Revisited*. In: Birkedal, L. (ed.) The Thirty-second Conference on the Mathematical Foundations of Programming Semantics, MFPS 2016, Carnegie Mellon University, Pittsburgh, PA, USA, May 23-26, 2016. Electronic Notes in Theoretical Computer Science, vol. 325, pp. 63–83. Elsevier (2016). https://doi.org/10.1016/J.ENTCS.2016.09.032, https://doi.org/10.1016/j.entcs.2016.09.032
6. Cockett, J., Cruttwell, G.: *Differential Structure, Tangent Structure*. SDG. Appl Categor Struct 22, 331–417 (2014). https://doi.org/https://doi.org/10.1007/s10485-013-9312-0
7. Cockett, R., Cruttwell, G.: *Differential bundles and fibrations for tangent categories*. Cahiers de Topologie et Géométrie Différentielle Catégoriques **59**(1), 10–92 (2018), https://arxiv.org/pdf/1606.08379
8. Cruttwell, G.S.H., Gallagher, J., Lemay, J.P., Pronk, D.: *Monoidal reverse differential categories*. Math. Struct. Comput. Sci. **32**(10), 1313–1363 (2022). https://doi.org/10.1017/S096012952200038X, https://doi.org/10.1017/S096012952200038X
9. Cruttwell, G.S.H., Lemay, J.P.: *Reverse Tangent Categories*. In: Murano, A., Silva, A. (eds.) 32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy. LIPIcs, vol. 288, pp. 21:1–21:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2024), https://doi.org/10.4230/LIPIcs.CSL.2024.21
10. De Paiva, V.C.V.: *The Dialectica categories*. PhD thesis (1991), https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-213.pdf
11. Diller, J.: *Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen*. Arch math Logik 16, 49–66 (1974). https://doi.org/https://doi.org/10.1007/BF02025118
12. Ehrhard, T.: *An introduction to differential linear logic: proof-nets, models and antiderivatives*. Math. Struct. Comput. Sci. **28**(7), 995–1060 (2018). https://doi.org/10.1017/S0960129516000372, https://arxiv.org/abs/1606.01642
13. Ehrhard, T., Regnier, L.: *The differential lambda-calculus*. Theor. Comput. Sci. **309**(1-3), 1–41 (2003). https://doi.org/10.1016/S0304-3975(03)00392-X, https://doi.org/10.1016/S0304-3975(03)00392-X

14. Gallagher, J.: *The differential lambda-calculus: syntax and semantics for differential geometry*. Ph.D. thesis, Doctoral thesis, University of Calgary, Calgary, Canada (2018), https://pages.cpsc.ucalgary.ca/~robin/Theses/GallagherPhD.pdf
15. Gödel, V.K.: *Über Eine Bisher Noch Nicht Benützte Erweiterung des Finiten Standpunktes*. Dialectica **12**(3-4), 280–287 (1958). https://doi.org/10.1111/j.1746-8361.1958.tb01464.x
16. Kerjean, M., Pédrot, P.: *∂ is for Dialectica*. In: Sobocinski, P., Lago, U.D., Esparza, J. (eds.) Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024. pp. 48:1–48:13. ACM (2024), https://doi.org/10.1145/3661814.3662106
17. Kohlenbach, U.: *Applied Proof Theory: Proof Interpretations and Their Use in Mathematics*. Springer monographs in mathematics (2008) (01 2008). https://doi.org/10.1007/978-3-540-77533-1
18. Lee, J.M.: *Introduction to Smooth Manifolds*. Springer New York, NY (2012). https://doi.org/https://doi.org/10.1007/978-1-4419-9982-5
19. Lemay, J.P.: *Jacobians and Gradients for Cartesian Differential Categories*. In: Kishida, K. (ed.) Proceedings of the Fourth International Conference on Applied Category Theory, ACT 2021, Cambridge, United Kingdom, 12-16th July 2021. EPTCS, vol. 372, pp. 29–42 (2021). https://doi.org/10.4204/EPTCS.372.3, https://doi.org/10.4204/EPTCS.372.3
20. Mak, C., Ong, L.: *A Differential-form Pullback Programming Language for Higher-order Reverse-mode Automatic Differentiation*. CoRR **abs/2002.08241** (2020), https://arxiv.org/abs/2002.08241
21. Manzonetto, G.: *What is a categorical model of the differential and the resource λ-calculi?* Math. Struct. Comput. Sci. **22**(3), 451–520 (2012). https://doi.org/10.1017/S0960129511000594, http://arxiv.org/abs/1011.2307
22. Pédrot, P.: *A functional functional interpretation*. In: Henzinger, T.A., Miller, D. (eds.) XXIX Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), Vienna, Austria, July 14 - 18, 2014. pp. 77:1–77:10. ACM (2014), https://doi.org/10.1145/2603088.2603094
23. Pédrot, P.M.: *A Materialist Dialectica*. Theses, Paris Diderot (Sep 2015), https://hal.science/tel-01247085
24. Spivak, D.I.: *Poly: An abundant categorical setting for mode-dependent dynamics*. CoRR (2020), https://arxiv.org/pdf/2005.01894
25. Spivak, D.I.: *Generalized Lens Categories via functors $\mathcal{C}^{\mathrm{op}} \to \mathsf{Cat}$* (2022), https://arxiv.org/abs/1908.02202