

Functorial Models of Differential Linear Logic

Marie Kerjean ✉ 

CNRS, Université Sorbonne Paris Nord, Laboratoire d'Informatique de Paris Nord, LIPN,
F-93430 Villetaneuse, France

Valentin Maestracci ✉ 

Aix-Marseille Université, CNRS, I2M, France

Morgan Rogers ✉  

Université Sorbonne Paris Nord, CNRS, Laboratoire d'Informatique de Paris Nord, LIPN,
F-93430 Villetaneuse, France

Abstract

Differentiation in logic has several sources of inspiration. The most recent is differentiable programming, models of which demand functoriality and good typing properties. More historical is reverse denotational semantics, taking inspiration from models of Linear Logic to differentiate proofs and λ -terms. In this paper, we take advantage of the rich structure of categorical models of Linear Logic to give a new functorial presentation of differentiation. We define differentiation as a functor from a coslice of the category of smooth maps to the category of linear maps. Extending linear–non-linear adjunction models of Linear Logic, this produces models of Differential Linear Logic. We use these functorial presentations to shed new light on integration in differential categories.

2012 ACM Subject Classification Theory of computation \rightarrow Categorical semantics; Theory of computation \rightarrow Denotational semantics; Theory of computation \rightarrow Linear logic

Keywords and phrases Categorical semantics, Differential Programming, Linear Logic

Digital Object Identifier 10.4230/LIPIcs.FSCD.2025.26

Funding *Marie Kerjean*: Work partially funded by the project ANR-24-CE48-1914.

Acknowledgements We are grateful to Jean-Simon Lemay, Zeinab Galal, and Jad Koleilat for enriching discussions on this paper. We also thank the reviewers for the many constructive comments, helping us making this paper a better version of itself

1 Introduction

Differentiation is the bread and butter of mathematical analysis and approximation theory, providing the closest linear approximation of a function. The differential of a function $f : A \rightarrow B$ at a point $a \in A$ is the linear map $D_a f : v \in A \mapsto D_a f v \in B$. For scalar functions $f : \mathbb{R} \rightarrow \mathbb{R}$, this becomes $D_a f : t \mapsto f'(a) \cdot t$, where $_ \cdot _$ denotes multiplication. Models of Linear Logic (LL henceforth) can either be axiomatized as strong monoidal adjunctions between a category whose morphisms are “linear maps” and a category whose morphisms are “non-linear maps”, or more accurately “general maps” (smooth maps in our case), or as *Seely Categories*, a setting in which one considers directly the comonad $!$ that can be deduced from the previous adjunction.

Linear Logic has been extended to Differential Linear Logic (DiLL henceforth) [12], in which non-linear proofs can be turned into linear ones. Categorical presentations of differentiation originate from investigation of denotational models of this system, resulting in a variety of interrelated categorical definitions [1]. Differentiation in such settings might be axiomatized as an external operator acting on morphisms of a category [3], or as a hard-coded natural transformation $\bar{d} : \text{id} \Rightarrow !$ on a Seely category [14].



© Marie Kerjean, Valentin Maestracci, and Morgan Rogers;
licensed under Creative Commons License CC-BY 4.0

10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025).

Editor: Maribel Fernández; Article No. 26; pp. 26:1–26:17

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

26:2 Functorial Models of Differential Linear Logic

The differential of composed functions is famously not equal to the composite of their differentials: the latter may not even typecheck, as the points at which differentials are computed matter. This is expressed by the chain rule¹:

$$D_a(f;g) = D_a f; D_{f(a)}g \neq D_a f; D_a g \quad (1)$$

Functorial differentiation has been brought up to date by differentiable programming: this is a much more recent domain emerging from an effort to express machine learning algorithms in a functional and principled way [5, 13]. In particular, differentiation in differentiable programming is traditionally axiomatized as a transformation of programs *acting on pairs* (Equation 2). This makes D a functor on a suitable category of programs, with axioms enforcing linearity of differentiation.

$$Df : (a, v) \in A \times A \mapsto (f(a), D_a f v) \quad (2)$$

Our work takes a different approach, and exploits the linear–non-linear adjunction at the heart of Linear Logic.

$$(\mathcal{C}, \times, I) \begin{array}{c} \xrightarrow{\mathcal{E}'} \\ \perp \\ \xleftarrow{\mathcal{U}} \end{array} (\mathcal{L}, \otimes, 1)$$

Here \mathcal{U} represents the forgetful functor from linear to general maps; see Definition 8 for the full hypotheses. We consider differentiation as *a functor from the coslice of the category of differentiable maps under the terminal object to the category of linear maps*. The coslice category $(I \downarrow \mathcal{C})$ can be thought of as the category of pointed objects, meaning each object A comes equipped with a point $a : I \rightarrow A$ and morphisms respect these. We re-express models of DiLL in terms of models of LL equipped with a functor

$$\mathcal{D} : (I \downarrow \mathcal{C}) \rightarrow \mathcal{L}$$

with the intuition that \mathcal{D} maps a function $f : (a, A) \rightarrow (f(a), B)$ to $D_a(f)$, its differential at the point a . The only constraint we impose is that \mathcal{D} leaves linear maps unchanged, (11). This simplifies both the pairwise presentation of differentiable programming (equation (2)) and the non-functorial transformation \bar{d} of differential categories. We will show that this definition is equivalent to that coming from differential categories up to a pointed version of the chain rule; the converse is straightforward (Proposition 27).

This approach also yields insights into how related constructions from analysis, most notably integration, can be realised in models of LL. We demonstrate this by using the slice category $(I \downarrow \mathcal{C})$ to present the *fundamental theorem of analysis* as an adjunction.

Related work. Work on differential categories [2], cartesian differential categories [3] and tangent categories [8] have been successful in categorifying differentiation from a mathematical perspective, but place models of DiLL as the strictest in the hierarchy of categorical definitions. Our use of coslice categories in the context of DiLL is novel, to our knowledge, although the dual construction of slice categories appears in various places: in reverse cartesian differential categories, partial differentials are defined on slice objects [9], and slice categories are extensively used in modeling dependent type theory. We hope our work encourages research into the relationship between differentiation and dependent types.

¹ We write composite morphisms within a category in diagrammatic order throughout, so $f;g$ means f followed by g , but we compose functors contravariantly to avoid confusion when writing, say, $\mathcal{U}(!X)$.

Outline and contributions. We provide background material on differential categories in Section 2. The core results of our study are presented in Section 3, where we define \mathcal{D} as a functor on the coslice $(I \downarrow \mathcal{C})$, and demonstrate that with a few assumptions, it induces the structure of a differential category on \mathcal{L} . We formulate categorical models of DiLL with a functorial definition of differentiation (Definition 15), which we prove to be equivalent to differential storage categories under some hypotheses (Proposition 26 and Proposition 27).

In Section 4, we first review the existing presentations of integration and fundamental theorems of analysis in differential categories, then show that the coslice construction allows for a new presentation as an adjunction between $(I \downarrow \mathcal{C})$ and \mathcal{L} (Proposition 34). Finally, in Section 5, we discuss potential future work.

2 Preliminaries

In this section, we give a quick introduction to the syntax and semantics of DiLL. For readers familiar with DiLL, we note that Definition 4 is the only one technically necessary to the rest of the paper; the rest of this section serves as motivation and historical introduction.

Linear Logic [15] (LL) is the result of a decomposition of Intuitionistic Logic via an involutive *linear* negation. Along with multiplicative and additive versions of conjunction and disjunction, it introduces the *exponential* connective $!$ and its dual $?$. Some important intuition for working in LL coming from the Curry-Howard correspondence is that a proof of a sequent of the form $A \vdash B$ corresponds to the construction of a *linear* function from A to B , while a proof of $!A \vdash B$ constructs a *non-linear* function from A to B . When modelling LL in vector spaces, these intuitions are literally valid, in that a proof of $A \vdash B$ is interpreted as a linear map and a proof of $!A \vdash B$ is interpreted as (the transpose across the linear–non-linear adjunction of) a not-necessarily-linear map.

DiLL [12] adds to LL new rules on the connective $!$, expressing the fact that one is able to transform a non-linear proof into a linear one. While LL features a **dereliction** rule (d), DiLL introduces a **codereliction** rule (\bar{d}) which acts like differentiation at 0 into the logic:

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

This creates a duality between linear and non-linear maps in DiLL. We recall also the exponential rules of LL called respectively **weakening** (w), **contraction** (c) and **promotion** (p). DiLL adds to these dual rules of **coweakening** (\bar{w}) and **cocontraction** (\bar{c}).

$$\frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} \text{w} \qquad \frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} \text{c} \qquad \frac{! \Gamma \vdash A}{! \Gamma \vdash !A} \text{p} \qquad \frac{\vdash}{\vdash !A} \bar{w} \qquad \frac{\vdash \Gamma, !A \quad \vdash \Delta, !A}{\vdash \Gamma, \Delta, !A} \bar{c}$$

Since the focus of this paper is on categorical models rather than the proof theory of these logics, we will not give a full presentation of the syntax of DiLL; we omit the cut elimination rules and the corresponding commutative diagrams between their interpretation as these are direct translations of differential calculus rules. We refer the reader to the survey of Ehrhard [11] for a complete introduction. Let us now turn to the categorical models.

► **Definition 1.** A *Seely category* [19] is a monoidal closed category $(\mathcal{L}, \otimes, 1)$ with finite products and a comonad $(!, \mathbf{d}, \nu)$ such that $!$ is a (strong) monoidal functor

$$! : (\mathcal{L}, \times, \top) \longrightarrow (\mathcal{L}, \otimes, 1)$$

where \top denotes a terminal object. Monoidality of $!$ means we have “Seely isomorphisms”:

$$\chi_{A,B} : !A \otimes !B \xrightarrow{\cong} !(A \times B) \quad \text{and} \quad \chi_0 : 1 \xrightarrow{\cong} !\top.$$

26:4 Functorial Models of Differential Linear Logic

Following the LL convention, we will denote linear maps by $\ell : A \multimap B$ and non-linear or smooth maps in the coKleisli category $\mathcal{L}!$ by $f : A \multimap B$ (note the difference in arrow head). To model classical LL, one requires that LL be a $*$ -autonomous category, with an involutive duality $(_)\perp \simeq \mathcal{L}(_, 1)$.

The image of the diagonal map $\Delta_E : E \multimap E \times E$ and terminal map $n_E : E \multimap \top$ for the product in \mathcal{L} under $!$ give each object of the form $!E$ a canonical coalgebra structure:

$$c_E : !E \xrightarrow{!\Delta_E} !(E \times E) \xrightarrow[\cong]{\chi_{E,E}^{-1}} !E \otimes !E \qquad w_E : !E \xrightarrow{!n_E} !\top \xrightarrow[\cong]{\chi_{\top}^{-1}} 1.$$

That is, Seely categories are categorical models of intuitionistic Linear Logic [17], with the above maps interpreting contraction and weakening, while dereliction is interpreted by the counit of the comonad (as suggested by the notation) and promotion is interpreted by coKleisli extension. Moreover, we recover an instance of the linear–non-linear adjunction discussed in the Introduction by constructing the coKleisli category for the comonad $!$.

The usual definition of categorical models of DiLL refines the axiomatization of Seely categories. For instance, to interpret \bar{w} and \bar{c} in this case one only needs to replace products with biproducts. This is a natural choice: to reflect the addition of rules to a logic, one adds more constructors to the categories providing the logic’s semantics.

► **Definition 2.** A *semi-additive category* is a category \mathcal{L} with finite products and coproducts such that the initial object is terminal and the binary product coincides with the binary coproduct. In the present paper, we denote the resulting **biproduct** monoidal structure by \diamond , with unit the **zero object**, denoted 0 . We also give the following names to the canonical morphisms associated to this structure:

$$u_E : 0 \multimap E \qquad n_E : E \multimap 0 \qquad \nabla_E : E \diamond E \multimap E \qquad \Delta_E : E \multimap E \diamond E$$

Note that for any object, (E, u_E, ∇_E) is a commutative monoid and (E, n_E, Δ_E) is a commutative comonoid. The hom-sets of a semi-additive category canonically carry the structure of commutative monoids: given $f, g : A \multimap B$, we may define $f + g$ to be the composite,

$$A \xrightarrow{\Delta_A} A \diamond A \xrightarrow{f \diamond g} B \diamond B \xrightarrow{\nabla_B} B \tag{3}$$

and this operation has the commutativity and associativity properties one would expect.

► **Example 3.** The category of real vector spaces and linear maps is semi-additive, with \diamond the direct sum and unit the zero vector space. The commutative monoid structure on morphisms is just the pointwise sum.

There is a whole hierarchy of axiomatizations of “differential categories” [1], but we will make use of the strongest notion.

► **Definition 4.** A *differential category*, called a “differential storage category” in [2] is a semi-additive Seely category $(\mathcal{L}, \otimes, 1)$ (with biproducts denoted as in Definition 2) in which the comonad $(!, d, \nu)$ is equipped with a natural transformation $\bar{d} : \text{id} \Rightarrow !$ satisfying:

■ invariance of linear maps under differentiation,

$$\bar{d}; d = \text{id} \tag{4}$$

■ and the generalized chain rule, requiring the following diagram to commute:

$$\begin{array}{c}
 \begin{array}{ccccc}
 & \bar{d}_{E \otimes \text{id}_{!E}} & & \bar{c}_E & \\
 & \circ & \circ & \circ & \circ \\
 E \otimes !E & \swarrow & !E \otimes !E & \xrightarrow{\quad} & !E & \xrightarrow{\quad} & !!E \\
 & \searrow & & & & & \uparrow \bar{c}_{!E} \\
 & \bar{d}_{E \otimes c_E} & & \bar{c}_{E \otimes \text{id}_{!E}} & & \bar{d}_{!E \otimes \nu_E} & \\
 & \circ & !E \otimes !E \otimes !E & \xrightarrow{\quad} & !E \otimes !E & \xrightarrow{\quad} & !!E \otimes !!E
 \end{array}
 \end{array} \quad (5)$$

where \bar{c} and \bar{w} are constructed from the algebra structure on each object as follows:

$$\bar{c}_E : !E \otimes !E \xrightarrow{\cong} \chi_{E,E} \circ !(E \diamond E) \xrightarrow{! \nabla_E} !E \qquad \bar{w}_E : 1 \xrightarrow{\cong} \chi^0 \circ !0 \xrightarrow{!u_E} !E.$$

These endow $!E$ with coalgebra structure dualizing the algebra structure constructed earlier.²

The generalized chain rule (5) captures the usual formula for composed functions,

$$D_a(f; g) = D_a(f); D_{f(a)}(g).$$

For the reader unfamiliar with differential categories, we now get a bit more into the categorical transposition of chain rule. We start with the simpler case of differentiation at 0, the *chain rule without a context*, called the “alternative chain rule” [dC.4'] in [1]:

$$\begin{array}{ccc}
 E & \xrightarrow{\bar{d}_E} & !E & \xrightarrow{\nu_E} & !!E \\
 \downarrow j_E; \bar{d}_E \otimes !u_E & & & & \downarrow \bar{c}_{!E} \\
 !E \otimes !E & \xrightarrow{\bar{d}_{!E \otimes \nu_E}} & & & !!E \otimes !!E,
 \end{array} \quad (6)$$

where $j_E := E \rightarrow E \otimes 1$ is the inverse of the right unit map. The relationship becomes more apparent if we post-compose this diagram with $!f; g$, for $f : !E \rightarrow F$ and $g : !F \rightarrow G$. The comultiplication $\nu_E : !E \rightarrow !!E$ expresses composition of transposed non-linear maps: if f and g are respectively the transposes of non-linear maps $f^\#, g^\# \in \mathcal{C}$, then $\nu_F; !f; g : !E \rightarrow G$ transpose to $f^\#; g^\#$. As such, $\bar{d}_E; \nu_E; !f; g : E \rightarrow G$ represents the differential $D_0((f; g)^\#)$. Meanwhile, by naturality of \bar{c} and \bar{d} , the alternative composite can be rearranged to $\bar{d}_E; f; \bar{d}_F; j_E; (!u_E; \nu_E; !f) \otimes \text{id}_F; \bar{c}; g$; the first two terms yield $D_0(f^\#)$, while the rest translates to $D_0(g^\#(f^\#(0) + -))$, which by a change of variables is equal to $D_{f^\#(0)}(g^\#)$. The generalized chain rule (5) adds to this set-up a parameter along which functions are not differentiated, so captures the chain rule for partial differentiation.

► **Proposition 5** ([1, Lemma 6.5]). *The generalized chain rule 5 can be recovered from the alternative chain rule 6 if one asks for the following strength condition on $!$, where $m : !E \otimes !F \rightarrow !(E \times F)$ represents the monoidality of $!$ within $(\mathcal{L}, \otimes, 1)$ and can be reconstructed from the Seelye isomorphisms $\chi_{E,F}$.*

$$\begin{array}{ccc}
 E \otimes !F & \xrightarrow{\bar{d} \otimes \text{id}} & !E \otimes !F & \xrightarrow{m} & !(E \otimes F) \\
 \downarrow \text{id} \otimes \bar{d} & & & & \downarrow \bar{d}_{E \otimes F} \\
 & & E \otimes F & &
 \end{array} \quad (7)$$

In terms of the Seelye isomorphisms, this becomes a more complex condition:

$$\text{id}_{!E} \otimes \bar{d}_E; \chi_{E,E}^{-1}; \bar{d}_{!(E \diamond E)}; !\chi; !(d_E \otimes d_E) = \text{id}_{!E} \otimes \bar{d}_E; \chi^{-1}; \nu_{E \diamond E}; !\chi_{E,E}; !(d_E \otimes d_E)$$

² The definition given in [2] includes more equations; this presentation is equivalent by [1, Corollary 6.3].

The transformations \bar{c} , \bar{w} and \bar{d} complement the components of LL interpreted in a general Seely category to produce an interpretation of DiLL. In fact, the differential categories axiomatized above are *denotational* models of intuitionistic DiLL, and of classical DiLL when we impose the extra condition of *-autonomy on \mathcal{L} . The exploration of less strict axiomatizations has been fruitful in incorporating various mathematical examples from functional analysis and differential geometry into this logical framework. However, the definition above crucially does not reflect the symmetry of exponential rules in DiLL, and does not express differentiation using the traditional objects of computer science. Notably, the chain rule is at first sight quite intricate. In this paper, we attempt to simplify this by showing that it is a consequence of functoriality in a new presentation.

Rather than merely introducing new operations such as \bar{d} , our approach requires us to introduce new *categories* on top of the LL setup, expanding Definition 1 to an adjunction involving the coKleisli category of ! mentioned above. Nonetheless, the extension of LL to DiLL described in Definition 4 can be recovered from our construction.

Notation. Given a differentiable function $f : A \rightarrow B$ between two vector spaces, we denote by $D_a f : A \rightarrow B$ the linear map corresponding to the differential of f at a point $a \in A$.

► **Example 6** (The distribution model). The category of convenient spaces [4] provides an illustrative example of a differential category. Formulas are interpreted by real topological vector spaces endowed with a *bornology*, which is a well-behaving collection of bounded sets; we shall not attempt to give detailed definitions of these structures here. The exponential !A is interpreted as the completion of the vector space generated by the Dirac distributions,

$$\llbracket !A \rrbracket := \overline{\langle \delta_x \rangle_{x \in \llbracket A \rrbracket}}.$$

The Dirac distribution δ_x , defined by $f \in \mathcal{C}(E, \mathbb{R}) \mapsto f(x)$, is formally an object of the dual $\mathcal{C}(E, \mathbb{R})'$ of the space of smooth functions, otherwise known as the space of distributions of compact support over E [18]. As the Dirac distributions form a linearly independent family of distributions, to define linear morphisms on !E it is enough to specify their value on Diracs and then extend linearly and continuously to the completion. As such, the components of DiLL are interpreted as follows:

$$\begin{array}{lll} \mathbf{d} : \delta_x \in !E \mapsto x \in E & \mathbf{c} : \delta_x \in !E \mapsto \delta_x \otimes \delta_x \in !E \otimes !E & \mathbf{w} : \delta_x \in !E \mapsto 1 \in \mathbb{R} \\ \bar{\mathbf{d}} : x \in E \mapsto D_0(_)(x) \in !E & \bar{\mathbf{c}} : \delta_x \otimes \delta_y \in !E \otimes !E \mapsto \delta_{x+y} \in !E & \bar{\mathbf{w}} : 1 \in \mathbb{R} \mapsto \delta_0 \in !E \\ & \nu : \delta_x \in !E \mapsto \delta_{\delta_x} \in !!E \end{array}$$

Convenient spaces do *not* form a *-autonomous category, and constructing *-autonomous smooth models of DiLL with a non-trivial duality is tricky, often requiring compromise on the interpretation of the other components of DiLL, such as [10] which features an ad-hoc construction to allow differentiation. An alternative work-around for this issue is to construct polarized models of DiLL [16].

► **Example 7.** We denote by REL the category of sets and relations. The interpretation of LL in REL produces the well-known, historical “relational model”. The tensor product is given by the cartesian product of sets, $X \otimes Y = X \times Y$ and the product (which coincides with the coproduct) is given by the disjoint union of sets $X \sqcup Y$. The exponential ! is interpreted by finite multisets: $!X = \mathcal{M}_f(X)$, the set of all finite multisets of X . The dereliction $\mathbf{d}_X \subseteq !X \times X$ and codereliction $\bar{\mathbf{d}}_X \subseteq X \times !X$ respectively put in relation elements with singleton multisets containing them, and reciprocally. The contraction $\mathbf{c}_X \subseteq !X \times (!X \times !X)$

and cocontraction $\bar{c}_X \subseteq (!X \times !X) \times !X$ relate pairs of finite multisets to their disjoint union, while the weakening $w_X \subseteq !X \times \{*\}$ and coweakening $\bar{w}_X \subseteq \{*\} \times !X$ relate $*$ to the empty multiset. The co-multiplication of the co-monad $!$ is $\nu_X \subseteq !X \times !!X$, relates a finite multiset to each finite multiset of finite multisets of which it is the disjoint union.

3 Differentiation on the coslice

This section represents the heart of this paper: here we will start from a linear–non-linear adjunction and show how defining a functor from a coslice of the non-linear category \mathcal{C} to the linear category \mathcal{L} makes \mathcal{L} into a differential category, up to a well-pointed assumption.

► **Definition 8.** *For the purposes of the present paper, a **linear–non-linear adjunction** is an adjunction:*

$$\begin{array}{ccc}
 & \xrightarrow{\mathcal{E}'} & \\
 (\mathcal{C}, \times, I) & \perp & (\mathcal{L}, \otimes, 1) \\
 & \xleftarrow{\mathcal{U}} &
 \end{array}
 \tag{8}$$

in which:

- \mathcal{L} is both monoidal closed and semi-additive;
- \mathcal{C} has cartesian monoidal structure;
- \mathcal{E}' is a strong monoidal functor, so $\mathcal{E}'(A \times B) \cong \mathcal{E}'(A) \otimes \mathcal{E}'(B)$ and $1 \cong \mathcal{E}'(I)$.

This adds hypothesis to the usual notion of linear–non-linear adjunction, in which the semi-additivity is not required and the adjunction is only lax monoidal [17, Definition 21].

Notation. We denote the endofunctor on \mathcal{L} from the adjunction 8 by $! := \mathcal{E}' \circ \mathcal{U}$, the unit by $\eta : \text{id}_{\mathcal{C}} \Rightarrow \mathcal{U} \circ \mathcal{E}'$ and the counit by $\text{d} : ! \Rightarrow \text{id}_{\mathcal{L}}$. We moreover write $\nu = \mathcal{E}'\eta_{\mathcal{U}}$ for the comultiplication of the induced comonad $(!, \text{d}, \nu)$ on \mathcal{L} . We denote $f \in \mathcal{C}(A, B)$ by $f : A \multimap B$ and $\ell \in \mathcal{L}(E, F)$ by $\ell : E \multimap F$, which is consistent with our notation for Seely categories in the special case where \mathcal{C} is the coKleisli category of $!$. Note that \mathcal{U} preserves products, being a right adjoint. For notational convenience, we work as if \mathcal{U} and \mathcal{E}' are strictly monoidal, in the sense that we suppress the isomorphisms $\mathcal{U}(A \diamond B) \cong \mathcal{U}(A) \times \mathcal{U}(B)$ and $\mathcal{E}'(A \times B) \cong \mathcal{E}'(A) \otimes \mathcal{E}'(B)$. With this convention, the projection map $\pi_1 : A \diamond B \multimap A$ satisfies $\mathcal{U}(\pi_1) = \pi_1 : \mathcal{U}(A) \times \mathcal{U}(B) \multimap \mathcal{U}(A)$ and moreover (considering the case $B = 0$) \mathcal{U} preserves the left and right unit maps for 0 on the nose.

► **Remark 9.** We will not actually use the fact that the monoidal structure \otimes on \mathcal{L} is closed, but we wished to ensure the hypotheses in the various definitions lined up.

► **Example 10.** The distribution model of Example 3 can also be understood from the adjunction point of view instead of the comonadic point of view. Let \mathcal{C} be the category whose objects are convenient vector spaces and whose morphisms are smooth functions and \mathcal{L} the category of convenient vector spaces and linear maps. In adjunction (8), \mathcal{E}' is interpreted via $\mathcal{E}'(E) := \overline{\langle \delta_x \rangle_{x \in [E]}}$, that is the space generated by Dirac distributions on E , and \mathcal{U} is the forgetful functor. \mathcal{C} is a cartesian closed category, \mathcal{L} is monoidal closed and $(\mathcal{E}' \dashv \mathcal{U})$ is a strong monoidal adjunction.

► **Lemma 11** ([17]). *Given a linear–non-linear adjunction as in Definition 8, \mathcal{L} acquires the structure of a Seely category.*

As such, we can interpret the ingredients of Differential Linear Logic, c, w, \bar{c}, \bar{w} and d , as we did in Section 2. In order to interpret differentiation, we need to expose further structure. To begin with, we observe that the semi-additive structure on \mathcal{L} is partially preserved by \mathcal{U} .

► **Proposition 12.** *Suppose that we are given a linear–non-linear adjunction as in Definition 8. For objects $E \in \mathcal{L}$ and $A \in \mathcal{C}$, $\mathcal{C}(A, \mathcal{U}(E))$ has a distinguished element,*

$$0_{A,E} : A \xrightarrow{n_A} I \xrightarrow{\mathcal{U}(u_E)} \mathcal{U}(E),$$

where n_A is the unique morphism. This is the identity element of a commutative monoid structure on $\mathcal{C}(A, \mathcal{U}(E))$.

Proof. For $f, g : A \rightrightarrows \mathcal{U}(E)$, we define their sum using the codiagonal map:

$$f \boxplus g : A \xrightarrow{\langle f, g \rangle} \mathcal{U}(E \diamond E) \xrightarrow{\mathcal{U}(\nabla_E)} \mathcal{U}(E). \quad (9)$$

To show that $0_{A,E}$ is the identity element, consider the transpose of $f \boxplus 0_{A,E}$:

$$\mathcal{E}'(\langle f, 0_{A,E} \rangle); \mathcal{E}'(\mathcal{U}(\nabla_E)); \mathbf{d}_E = \mathcal{E}'(f) \otimes \bar{w}_E; \bar{c}_E; \mathbf{d}_E$$

by monoidality of \mathcal{E}' . The bialgebra laws over $!E$, as described for example in [1, Section 7] give us $\text{id}_E \otimes \bar{w}_E; \bar{c}_E = \text{id}_E$ and as such $\mathcal{E}'(f) \otimes \bar{w}_E; \bar{c}_E; \mathbf{d}_E = \mathcal{E}'(f); \mathbf{d}_E$, the transpose of f , so $f = f \boxplus 0_{A,E}$ as required. ◀

Next we formalize the category of pointed objects of \mathcal{C} .

► **Definition 13.** *Suppose that \mathcal{L} is a category with a zero object 0, that \mathcal{C} is a category with terminal object I and that we are given a functor $\mathcal{U} : \mathcal{L} \rightarrow \mathcal{C}$ sending 0 to I .*

*Recall that the **coslice category** $(I \downarrow \mathcal{C})$ is defined as follows. The objects of $(I \downarrow \mathcal{C})$ are pairs (A, a) with $A \in \mathcal{C}$ and $a \in \mathcal{C}(I, A)$, and morphisms $(A, a) \rightarrow (B, b)$ are morphisms $g : A \rightarrow B$ such that $a; g = b$.*

We may similarly construct the coslice category $(I \downarrow \mathcal{U})$ where objects are instead pairs (E, a) with $E \in \mathcal{L}$ and $a \in \mathcal{C}(I, \mathcal{U}(E))$, and morphisms $(E, a) \rightarrow (F, b)$ consist of $\ell \in \mathcal{L}(E, F)$ with $a; \mathcal{U}(\ell) = b$.

Notation. In order to more explicitly keep track of the points involved in a morphism of $(I \downarrow \mathcal{C})$, we denote a morphism $g : (A, a) \rightarrow (B, b)$ by $(a]g)$.

The coslice categories $(I \downarrow \mathcal{C})$ and $(I \downarrow \mathcal{U})$ come equipped with projection functors to \mathcal{C} and \mathcal{L} which respectively forget the points; we denote both by Π . Moreover, \mathcal{U} induces a functor $(I \downarrow \mathcal{U}) \rightarrow (I \downarrow \mathcal{C})$ which we denote by $\downarrow \mathcal{U}$, mapping (E, a) to $(\mathcal{U}(E), a)$. Putting these together, we get:

$$\begin{array}{ccc} (I \downarrow \mathcal{C}) & \xleftarrow{\downarrow \mathcal{U}} & (I \downarrow \mathcal{U}) \\ \downarrow \Pi & & \downarrow \Pi \\ \mathcal{C} & \xleftarrow{\mathcal{U}} & \mathcal{L} \end{array} \quad (10)$$

► **Example 14.** Let's see how differentiation can be interpreted in Example 10 using the coslice construction. In this case I is the zero vector space $\{0\}$ and $(I \downarrow \mathcal{C})$ is the category of pairs (A, a) , where a is a vector in A . Differentiation defines a functor $\mathcal{D} : (I \downarrow \mathcal{C}) \rightarrow \mathcal{L}$ as follows: an object $a : I \rightarrow A$ is mapped to A and an arrow $(a]g)$ is sent to the linear map $D_a f : A \rightarrow B$. The chain rule exactly translates to functoriality of \mathcal{D} . Indeed, morphisms $(a]g)$ and $(b]h)$ compose if and only if $b = g(a)$ and then we have

$$\mathcal{D}(a]g; h) = D_a(g; h) = D_a g; D_{g(a)} h = \mathcal{D}(a]g); \mathcal{D}(g(a)]h).$$

Example 14 contains the essential idea of the present section. As we shall shortly see, there will be no need to independently construct a codereliction operator once we equip our adjunction with a differentiation functor, since codereliction can be recovered from \mathcal{D} as differentiation at 0.

► **Definition 15.** A *functorial model of differentiation* consists a linear–non-linear adjunction $(\mathcal{E}' \dashv \mathcal{U})$ between \mathcal{L} and \mathcal{C} equipped with a functor $\mathcal{D} : (I \downarrow \mathcal{C}) \rightarrow \mathcal{L}$ such that

$$\mathcal{D} \circ \downarrow \mathcal{U} = \Pi, \quad (11)$$

which is to say that the following triangle (ignoring the dashed functors) commutes:

$$\begin{array}{ccc} (I \downarrow \mathcal{C}) & \xleftarrow{\downarrow \mathcal{U}} & (I \downarrow \mathcal{U}) \\ \Pi \downarrow & \searrow \mathcal{D} & \downarrow \Pi \\ \mathcal{C} & \xleftarrow{\mathcal{U}} & \mathcal{L} \end{array}$$

► **Remark 16.** The choice of name in Definition 15 is not intended to seem grandiose: as we shall see, these categories do not quite constitute models of DiLL in general. As discussed in the introduction, the idea that the chain rule is compositional on pairs of objects is well-established [13]. Here we explore another presentation of differentiation as a functor compatible with linear–non-linear adjunctions as models of Linear Logic.

More explicitly, (11) says that for any object E of \mathcal{L} and point $a : I \rightarrow \mathcal{U}(E)$ we have $\mathcal{D}(a \downarrow \mathcal{U}(E)) = E$, and that for any arrow $\ell : E \multimap F$ in \mathcal{L} we have $\mathcal{D}(a \downarrow \mathcal{U}(\ell)) = \ell$. This identity also implies that \mathcal{D} preserves products of objects in $\mathcal{U}(\mathcal{L})$, in the following sense:

► **Lemma 17.** The functor \mathcal{D} in Definition 15 preserves limits of objects in the image of $\downarrow \mathcal{U}$. In particular, for $h_1 : A \rightarrow \mathcal{U}(E_1)$ and $h_2 : A \rightarrow \mathcal{U}(E_2)$ in \mathcal{C} and $a : I \rightarrow A$, we have:

$$\mathcal{D}(a \downarrow \langle h_1, h_2 \rangle) = \langle \mathcal{D}(a \downarrow h_1), \mathcal{D}(a \downarrow h_2) \rangle. \quad (12)$$

Proof. A short proof is that Π creates limits and $\downarrow \mathcal{U}$ preserves them (because \mathcal{U} does), whence the identity (11) forces \mathcal{D} to preserve them. ◀

► **Proposition 18.** Given morphisms in \mathcal{C} , $I \xrightarrow{a} A_1 \xrightarrow{g_1} A_2 \xrightarrow{g_2} A_3$, we have:

$$\mathcal{D}(a \downarrow g_1; g_2) = \mathcal{D}(a \downarrow g_1); \mathcal{D}(a; g_1 \downarrow g_2). \quad (13)$$

Similarly, given:

$$I \xrightarrow{a'} A \xrightarrow[h_2]{h_1} \mathcal{U}(E)$$

we have,

$$\mathcal{D}(a' \downarrow h_1 \boxplus h_2) = \mathcal{D}(a' \downarrow h_1) + \mathcal{D}(a' \downarrow h_2). \quad (14)$$

Proof. The first identity is an immediate consequence of functoriality of \mathcal{D} . For the second, the definition of $h_1 \boxplus h_2$ produces the following diagram:

$$\begin{array}{ccc} I & \xrightarrow{a'} & \mathcal{U}(E_1) \\ \parallel & & \downarrow \langle h_1, h_2 \rangle \\ I & \xrightarrow{\langle a'; h_1, a'; h_2 \rangle} & \mathcal{U}(E_2) \times \mathcal{U}(E_2) \\ \parallel & & \downarrow \mathcal{U}(\nabla_{E_2}) \\ I & \xrightarrow{(a'; h_1) \boxplus (a'; h_2)} & \mathcal{U}(E_2) \end{array}$$

26:10 Functorial Models of Differential Linear Logic

Applying \mathcal{D} , the upper square is mapped to $\langle \mathcal{D}(a']h_1), \mathcal{D}(a']h_2) \rangle$ by Lemma 17 and the lower square is mapped to ∇_{E_2} by (11), yielding the desired sum, as required. \blacktriangleleft

As such, \mathcal{D} satisfies the chain rule and is linear with respect to the additive structure on morphisms. It follows that \mathcal{D} interacts well with products of morphisms too:

► **Corollary 19.** *Let $g_1 : \mathcal{U}(E'_1) \rightarrow \mathcal{U}(E)$ and $g_2 : \mathcal{U}(E'_2) \rightarrow \mathcal{U}(E)$ and $a = \langle a_1, a_2 \rangle : I \rightarrow \mathcal{U}(E'_1) \times \mathcal{U}(E'_2)$. Then we have:*

$$\mathcal{D}(a]g_1 \times g_2) = \mathcal{D}(a_1]g_1) \diamond \mathcal{D}(a_2]g_2). \quad (15)$$

Proof. By definition, $g_1 \times g_2 = \langle \mathcal{U}(\pi_1); g_1, \mathcal{U}(\pi_2); g_2 \rangle$. Thus by Lemma 17,

$$\mathcal{D}(a]g_1 \times g_2) = \langle \mathcal{D}(a]\mathcal{U}(\pi_1); g_1, \mathcal{D}(a]\mathcal{U}(\pi_2); g_2 \rangle.$$

We have: $\mathcal{D}(a]\mathcal{U}(\pi_1); g_1) = \mathcal{D}(a]\mathcal{U}(\pi_1)); \mathcal{D}(a;\mathcal{U}(\pi_1)]g_1) = \pi_1; \mathcal{D}(a_1]g_1)$, by (11) and similarly for $\mathcal{D}(a]\mathcal{U}(\pi_2); g_2)$, whence we recover the claimed product of morphisms. \blacktriangleleft

Comparing with differential categories, the only thing missing from \mathcal{L} to make it a differential category is a codereliction $\bar{d} : \text{id} \Rightarrow !$ satisfying the required identities.

► **Definition 20.** *For E an object of \mathcal{L} , we define: $\bar{d}_E := \mathcal{D}(\mathcal{U}(u_E)]\eta_{\mathcal{U}(E)}) : E \multimap !E$.*

► **Lemma 21.** *The transformation $\bar{d} : \text{id}_{\mathcal{L}} \Rightarrow !$ is natural.*

Proof. We have a functor $R : \mathcal{L} \rightarrow (I \downarrow \mathcal{C})$ sending E to $\mathcal{U}(u_E) : I \rightarrow \mathcal{U}(E)$ and $\ell : E \multimap F$ to $(\mathcal{U}(u_E)]\mathcal{U}(\ell))$; this is a section of \mathcal{D} . By inspection, $\eta_{\mathcal{U}(-)}$ lifts to a natural transformation $R \Rightarrow R \circ !$, and \bar{d} is the result of applying \mathcal{D} to this lifting, so is natural as required. \blacktriangleleft

We denote by $\ell^\# : \mathcal{U}(E) \rightarrow \mathcal{U}(F)$ the transpose of a morphism $\ell : !E \multimap F$ in \mathcal{L} across the adjunction $(\mathcal{E}' \dashv \mathcal{U})$. Since $\ell^\# = \eta_{\mathcal{U}(E)}; \mathcal{U}(\ell)$, using Proposition 18, the definition of \bar{d}_E and (11) we have:

$$\mathcal{D}(\mathcal{U}(u_E)]\ell^\#) = \bar{d}; \ell, \quad (16)$$

In particular, the linear rule $\bar{d}_E; \mathbf{d}_E = \text{id}_E$ holds for any object E of \mathcal{L} , since the counit morphism \mathbf{d}_E transposes to the identity in (16). Following [1, Proposition 6.2], by making use of the semi-additive structure we can deduce that the product rule is satisfied:

$$\bar{d}; \mathbf{c} = \bar{d} \otimes \mathbf{w} + \mathbf{w} \otimes \bar{d}. \quad (17)$$

The following uses functoriality to capture differentiation of translated functions: intuitively, $D_a g = D_0(x \mapsto g(a+x))$.

► **Lemma 22.** *The functor \mathcal{D} internalizes translations, meaning for any point $a : I \rightarrow \mathcal{U}(E)$ and morphism $(a]g)$ in $(I \downarrow \mathcal{C})$, we have:*

$$\mathcal{D}(a]g) = \mathcal{D}(\mathcal{U}(u_E)]\mathcal{U}(i_E); (\text{id}_{\mathcal{U}(E)} \times a); \mathcal{U}(\nabla_E); g),$$

where $i_E : E \multimap E \diamond 0$ is the inverse of the right unit morphism (or equivalently, the coproduct injection morphism).

Proof. By compositionality, we may without loss of generality consider the case $g = \text{id}_{\mathcal{U}(E)}$. By Proposition 12, we have $a = a \boxplus \mathcal{U}(u_E)$, so we can construct the following sequence of arrows in $(I \downarrow \mathcal{C})$:

$$\begin{array}{ccc}
 I & \xrightarrow{\mathcal{U}(u_E)} & \mathcal{U}(E) \\
 \parallel & & \downarrow \mathcal{U}(i_E) \\
 I & \xrightarrow{\langle \mathcal{U}(u_E), \text{id}_I \rangle} & \mathcal{U}(E) \times I \\
 \parallel & & \downarrow \mathcal{U}(\text{id}_E) \times a \\
 I & \xrightarrow{\langle \mathcal{U}(u_E), a \rangle} & \mathcal{U}(E) \times \mathcal{U}(E) \\
 \parallel & & \downarrow \mathcal{U}(\nabla_A) \\
 I & \xrightarrow{a} & \mathcal{U}(E)
 \end{array}$$

Observing $\mathcal{D}(\text{id}_I]a) = u_E$ (since its domain is 0), the image under \mathcal{D} of the composite is:

$$E \xrightarrow{i_E} \circ E \diamond 0 \xrightarrow{\text{id}_E \times u_E} \circ E \diamond E \xrightarrow{\nabla_E} \circ E,$$

where (11) gives left- and right-hand morphisms, while Corollary 19 gives the middle morphism. By the left unit law for i and ∇ , the composite is the identity, as required. \blacktriangleleft

We can use Lemma 22 to strengthen (16): we now consider the differential of a function at any point a and express it with respect to \bar{d} .

► **Lemma 23.** *For any $a : I \rightarrow \mathcal{U}(E)$ and $\ell : !E \multimap F$, we have:*

$$\mathcal{D}(a] \ell^\#) = j_E; \bar{d}_E \otimes \mathcal{E}'(a); \bar{c}_E; \ell \quad (18)$$

where again $j_E := E \multimap E \otimes 1$ is the inverse of the right unit map.

Proof. Consider first the case $\ell = \text{id}_{!E}$, so $\ell^\# = \eta_{\mathcal{U}(E)}$. By Lemma 22,

$$\mathcal{D}(f] \eta_{\mathcal{U}(E)}) = \mathcal{D}(\mathcal{U}(u_E)] \mathcal{U}(i_E); (\text{id}_{\mathcal{U}(E)} \times f); \mathcal{U}(\nabla); \eta_{\mathcal{U}(E)});$$

By naturality of η and monoidality of \mathcal{E}' , the morphism part of the right-hand side is equal to

$$(!i_E; \text{id}_{!E} \otimes \mathcal{E}'(a); !\nabla_E)^\#,$$

whence we can apply (16) to conclude,

$$\begin{aligned}
 \mathcal{D}(a] \eta_{\mathcal{U}(E)}) &= \bar{d}_E; !i_E; \text{id}_{!E} \otimes \mathcal{E}'(a); \bar{c}_E \\
 &= j_E; \bar{d}_E \otimes \mathcal{E}'(a); \bar{c}_E
 \end{aligned} \quad (19)$$

by naturality of j_E in E , observing that $!i_E = j_{!E}$. The general case follows by the argument used to derive (16). \blacktriangleleft

We now begin working toward the chain rule, beginning with the context-free case, (6). This turns out to be a consequence of naturality of η .

► **Lemma 24.** *For any morphism $a : I \rightarrow \mathcal{U}(E)$, we have:*

$$\mathcal{D}(a] \nu_E^\#) = \mathcal{D}(a] \eta_{\mathcal{U}(E)}); \mathcal{D}(a; \eta_{\mathcal{U}(E)}] \eta_{\mathcal{U}(!E)}) \quad (20)$$

26:12 Functorial Models of Differential Linear Logic

Proof. Recalling that $\nu_E = \mathcal{E}'\eta_{\mathcal{U}(E)}$, we have a naturality square for η that makes the arrows involved equal in $(\mathcal{U} \downarrow \mathcal{C})$, illustrated in the following diagram, to which we apply \mathcal{D} :

$$\begin{array}{ccccc}
 I & \xrightarrow{a} & \mathcal{U}(E) & & \\
 \parallel & & \swarrow \eta_{\mathcal{U}(E)} & \searrow \eta_{\mathcal{U}(E)} & \\
 I & \xrightarrow{a; \eta_{\mathcal{U}(E)}} & \mathcal{U}(!E) & \xrightarrow{\nu^\#} & \mathcal{U}(!E) \\
 \parallel & & \swarrow \eta_{\mathcal{U}(!E)} & \searrow \eta_{\mathcal{U}(!E)} & \\
 I & \xrightarrow{\quad} & \mathcal{U}(!E) & \xrightarrow{\mathcal{U}\nu_E} & \mathcal{U}(!E)
 \end{array}$$

► **Corollary 25.** *The codereliction \bar{d}_A satisfies the chain rule without a context, (6):*

$$\bar{d}_E; \nu_E = j_E; (\bar{d}_E \otimes !u_E); (\bar{d}_{!E} \otimes \nu_E); \bar{c}_{!E} \quad (21)$$

Proof. Let $a = \mathcal{U}(u_E)$ in (20). We can expand the left-hand side with (16); the first term on the right-hand side is the definition of \bar{d}_E and the second term can be expanded using (19). Observing that $\mathcal{E}'(\mathcal{U}(u_E); \eta_{\mathcal{U}(E)}) = !u_A; \nu_A$, the resulting identity is:

$$\bar{d}_E; \nu_E = \bar{d}_E; j_{!E}; (\bar{d}_{!E} \otimes (!u_E; \nu_E)); \bar{c}_{!E}.$$

The right-hand side can be manipulated into the expression in (21) by exploiting naturality of j and rearranging parallel tensored terms. ◀

However, to fully interpret DiLL, one also needs to interpret the chain rule *within a context* (5). This is the translation of a cut rule with a co-dereliction on the context of a promotion in DiLL with other elements not being cut.

$$\frac{\frac{!A, !B \vdash C}{!A, !B \vdash !C} \text{p} \quad \frac{\vdash A}{\vdash !A} \bar{d}}{!B \vdash !C} \text{cut}$$

Interpreting this rule in the context of Example 14, the proof takes a two-variable function $f : A \times B \rightarrow C$ (with transpose $f^\# : !A \otimes !B \multimap C$) and promotes it:

$$\begin{aligned}
 \text{p}(f) : !A \otimes !B \multimap !C; \\
 \delta_a \otimes \delta_b \mapsto ((g : C \multimap \mathbb{R}) \mapsto g(f(a, b))).
 \end{aligned}$$

Cutting with the codereliction rule means computing, for each vector $v \in A$, the function $\delta_b \mapsto (g \mapsto D_{(0,b)}(g \circ f)(v, 0))$, which expands to, $\delta_b \mapsto (g \mapsto D_{f(0,b)}g(D_{(0,b)}f)(v, 0))$ via the chain rule. Abstracting this reasoning leads us to a “pointed” version of the generalized chain rule.

► **Proposition 26.** *The codereliction \bar{d}_E satisfies a pointed version of the generalized chain rule (13). Namely, for each $a : I \rightarrow \mathcal{U}(!E)$, we have:*

$$\text{id}_E \otimes \mathcal{E}'(a); \bar{d}_E \otimes \text{id}_{!E}; \bar{c}_E; \nu_E = \text{id}_E \otimes \mathcal{E}'(a); \bar{d}_E \otimes c_E; \bar{c}_E \otimes \text{id}_{!E}; \bar{d}_{!E} \otimes \nu_E; \bar{c}_{!E}.$$

Proof. We deduce the equation from Lemma 24 as follows. By Lemma 23, the left-hand side of (20) expands as:

$$\begin{aligned}
 \mathcal{D}(a \upharpoonright \nu_E^\#) &= j_E; \bar{d}_E \otimes \mathcal{E}'(a); \bar{c}_E; \nu_E. \\
 &= j_E; \text{id}_E \otimes \mathcal{E}'(a); \bar{d}_E \otimes \text{id}_{!E}; \bar{c}_E; \nu_E.
 \end{aligned}$$

Meanwhile, the right-hand side expands via two applications of (19) to give:

$$\begin{aligned} \mathcal{D}(a \upharpoonright \nu^\#) &= j_E; \bar{d}_E \otimes \mathcal{E}'(a); \bar{c}_E; j_{!E}; \bar{d}_{!E} \otimes (\mathcal{E}'(a); \nu_E); \bar{c}_{!E} \\ &= j_E; \text{id}_E \otimes c_1; \bar{d}_E \otimes \mathcal{E}'(a) \otimes \mathcal{E}'(a); \bar{c}_E \otimes \text{id}_E; \bar{d}_{!E} \otimes \nu_E; \bar{c}_{!E} \\ &= j_E; \text{id}_E \otimes \mathcal{E}'(a); \bar{d}_E \otimes c_E; \bar{c}_E \otimes \text{id}_E; \bar{d}_{!E} \otimes \nu_E; \bar{c}_{!E} \end{aligned}$$

Here we pass from the first to the second line by naturality of j and the fact that $j_E; j_{E \otimes 1} = j_E; \text{id}_E \otimes c_0$, and from the second to the third line by naturality of c . Since j_E is an isomorphism, the desired identity follows. ◀

These developments would be of questionable value if they diverged too far from the established formalism of differential categories, so the following result is important.

► **Proposition 27.** *A differential category is a functorial model of differentiation (Def.15).*

Proof. Let \mathcal{L} be a differential category, with notation as in Definition 4. Let us denote by $\mathcal{C} = \mathcal{L}_!$ the coKleisli category and by \times the product on $\mathcal{L}_!$ inherited from \diamond . Then define a functor $\mathcal{D} : (0 \downarrow \mathcal{C}) \rightarrow \mathcal{L}$ by:

$$\mathcal{D} \begin{cases} (a : !0 \multimap E) \mapsto E \\ (a \upharpoonright g : !E \multimap E') \mapsto j_E; (\nu_0; !a) \otimes \bar{d}_E; \bar{c}_E; g \end{cases}$$

The effect of \mathcal{D} on morphisms is the exact categorical translation of the effect of $D_a g$ in Example 14. To show that \mathcal{D} is functorial, consider:

$$0 \xrightarrow{a} E \xrightarrow{g} E' \xrightarrow{g'} E''.$$

Expanding the definition, we first apply naturality of j and ν , then the argument in the proof of Proposition 26 and naturality of \bar{c} , then naturality of c , and finally the chain rule of Definition 4:

$$\begin{aligned} \mathcal{D}(a \upharpoonright g); \mathcal{D}(\nu_0; !a; g \upharpoonright g') &= j_E; (\nu_0; !a) \otimes \bar{d}_E; \bar{c}_E; g; j_{E'}; (\nu_0; !\nu_0; !!a; !g) \otimes \bar{d}_{E'}; \bar{c}_{E'}; g' \\ &= j_E; \text{id}_{!0} \otimes (j_E; (\nu_0; !a) \otimes \bar{d}_E; \bar{c}_E); (\nu_0; \nu_0; !!a; !g) \otimes (\bar{d}_E; !g); \bar{c}_{!E}; g' \\ &= j_E; c_0 \otimes \text{id}_E; \text{id}_{!0} \otimes ((\nu_0; !a) \otimes \bar{d}_E; \bar{c}_E); (\nu_0; !a; \nu_E) \otimes \bar{d}_E; \bar{c}_{!E}; !g; g' \\ &= j_E; (\nu_0; !a) \otimes \text{id}_E; c_E \otimes \bar{d}_E; \text{id}_{!E} \otimes \bar{c}_E; \nu_E \otimes \bar{d}_E; \bar{c}_{!E}; !g; g' \\ &= j_E; (\nu_0; !a) \otimes \text{id}_E; \text{id}_{!E} \otimes \bar{d}_E; \bar{c}_E; \nu_E; !g; g' \\ &= \mathcal{D}(a \upharpoonright \nu_E; !g; g'), \end{aligned}$$

as required. The fact that $\mathcal{D}(a \upharpoonright d; \ell) = \ell$ follows directly from $\bar{d}; d = \text{id}$, as

$$\mathcal{D}(a \upharpoonright d; \ell) = !a \otimes \bar{d}; \bar{c}; d; \ell = (!a; d) \otimes \text{id}_A; \bar{c}; \ell = !u_A \otimes \text{id}_A; \bar{c}; \ell. \quad \blacktriangleleft$$

► **Remark 28.** The separation in axiomatic strength between differential categories and our model thus comes down to the separation between the pointed chain rule of Proposition 26 and the generalized version (5). If morphisms of the form $\text{id}_E \otimes \mathcal{E}'(a)$ are jointly epimorphic (a sort of *well-pointedness* condition), then the latter follows directly from the former. However, this condition is not satisfied by the category of sets and relations of Example 7 (see [6, section 4]) so this condition seems undesirably strong.

4 Integration from a functorial point of view

We now explain how presenting differentiation as a functor on the coslice allows for a new understanding of integration. We begin with a review of integration in Seely categories, based on work by Lemay, Cockett and Ehrhard [7, 11].

4.1 Integration and antiderivatives

The fundamental theorems of calculus relate differentiation and integration for functions of a single variable. The first (FTC1) states that differentiating the integral of a function returns the original function. For a function $f : \mathbb{R} \mapsto \mathbb{R}$ and $a \in \mathbb{R}$, one has:

$$f(a) = D_x \left(x \mapsto \int_0^x f(t) dt \right) (a).$$

The second (FTC2) states that integrating a function's derivative returns the function itself:

$$f = x \mapsto f(a) + \int_a^x f'(t) dt.$$

These are valid for real-valued functions only. For higher-order functions, one replaces expressions $f'(t) dt$ by $D_a f(dt)$. This means that integration is an operation on functions with a linear and a non-linear argument:

$$\int_0^a : (F : !A \otimes A \multimap B) \mapsto \left(\left(\int_0^a F \right) : B \right)$$

In Section 3, we considered differentiation at 0, as encoded by \bar{d} . More generally, we can present differentiation at any point in terms of a so-called deriving transformation.

► **Definition 29.** *In a differential category \mathcal{L} , we denote by $\bar{\partial} : ! \otimes \text{id} \Rightarrow !$ the natural transformation:*

$$\bar{\partial}_E := \text{id}_{!E} \otimes \bar{d}_E; \bar{c}_E.$$

In the context of Example 14, this represents the differential combinator $\delta_a \otimes v \mapsto (f \mapsto D_a f(v))$. Dually, as introduced by Ehrhard [11], we write $\partial : ! \Rightarrow ! \otimes \text{id}$ for the natural transformation: $\partial_E = c_E; \text{id}_{!E} \otimes d_E$ corresponding to the map $\delta_v \mapsto \delta_v \otimes v$.

► **Definition 30** ([11]). *A differential category is said to **have antiderivatives** if the following morphism is invertible on every object E :*

$$J_E = \text{id}_{!E} + (\partial_E; \bar{\partial}_E) : !E \multimap !E$$

In the distribution model J_E would be interpreted as $\delta_v \mapsto (f \mapsto f(v) + D_v(f)v)$. We refer to [11, section 3.2] for detailed computations using this intuition, which involves integration by parts of the function $x \mapsto \int_0^1 f(tx) dt$.

Lemay and Cockett [7] gave a more general axiomatization of integration independent from differentiation. Integration is represented by a natural transformation $s : !A \multimap !A \otimes A$. The basic intuition is that post-composing a function F by s results in the integral from zero, $\int_0^- F(t, dt)$ where dt stands for any linear argument and not specifically for a measure. Cockett and Lemay then formulate (FTC2) categorically as follows:

$$s; \bar{\partial} + !0 = \text{id}, \tag{22}$$

where $!0 = w; \bar{w}$. Theorem (FTC1) involves an extra commutativity of variables condition (Poincaré's lemma), see [11, Prop. 13] and [7, Sec. 5.2]. We leave its study in our context for future work.

4.2 Derivatives and Anti-Derivatives in an adjunction

Consider a model of Differential Linear Logic, notated as in Definition 4. For the sake of simplicity, we consider \mathcal{C} as the co-Kleisli category over \mathcal{L} for the co-monad $!$. We denote $f^\# : A \rightarrow B$ the transpose in \mathcal{C} of the map $f : !A \multimap B$. We still write $\mathcal{U}(\ell)$ for the embedding of a linear map in \mathcal{C} , meaning $\mathcal{U}(\ell) = (\mathbf{d}; \ell)^\#$.

Note that all statements and proofs, suitably modified, work in the more general context of a linear–non-linear adjunction $\mathcal{E}' \dashv \mathcal{U}$, with a functor $\mathcal{D} : (I \downarrow \mathcal{C}) \rightarrow \mathcal{L}$, provided that the generalized chain rule (5) holds: this is used in Proposition 32.

We would like to study a bijection of the following form:

$$\Theta : (I \downarrow \mathcal{C})((A, a), (B, \mathcal{U}(u_B))) \rightleftharpoons \mathcal{L}(!A \otimes A, B) : \Theta^{-1} \quad (23)$$

The core intuition of our development is that Θ maps a function $f^\# : A \rightarrow B$ to its derivative $\bar{\partial}; f$. Meanwhile, Θ^{-1} should map a function $F : !A \otimes A \multimap B$ to its integral $s; F := x \in A \mapsto \int_a^x F(t, dt)$. Then (FTC1) will translate to $\Theta^{-1}; \Theta = \text{id}$. We now develop that statement categorically.

► **Definition 31.** We define the functor $\mathfrak{D} : (I \downarrow \mathcal{C}) \rightarrow \mathcal{L}$ and the mapping Θ such that:

$$\mathfrak{D} : \begin{cases} (a : I \rightarrow A) \mapsto !A \otimes A \\ (_ \rfloor f^\#) \mapsto \mathbf{c}_A \otimes \text{id}_A; (\nu; !f) \otimes (\bar{\partial}_A; f) \end{cases}$$

$$\Theta : \begin{cases} (I \downarrow \mathcal{C})((A, a), (B, \mathcal{U}(u_B))) \mapsto \mathcal{L}(\mathfrak{D}(A, a), B) \\ (_ \rfloor f^\#) \mapsto \bar{\partial}_A; f \end{cases}$$

With the notation of Section 3 and of Example 6, one has in particular $\mathfrak{D}(a \rfloor f^\#) (\delta_{x'} \otimes v) = (\delta_{f^\#(x')} \otimes \mathcal{D}(x' \rfloor f^\#))$, and Θ mapping $(a \rfloor f^\#)$ to $(\delta_{x'} \otimes v) \mapsto \mathcal{D}(x' \rfloor f^\#)(v)$.

► **Proposition 32.** Θ is natural in A and in B . If Θ is moreover an isomorphism, then it defines an adjunction between \mathfrak{D} and the functor $R : B \mapsto (B, u_B)$ described as a section of \mathcal{D} in the proof of Lemma 21.

Proof. Consider morphisms $f^\# : (A', a') \rightarrow (A, a)$ in $(I \downarrow \mathcal{C})$ and $\ell : B \multimap B' \in \mathcal{L}$. To show naturality, one has to prove that $(I \downarrow \mathcal{C})(f^\#, \mathcal{U}(\ell)); \Theta = \Theta; \mathcal{L}(\mathfrak{D}(f), \ell)$. Consider then $g^\# \in (I \downarrow \mathcal{C})((A, a), (B, \mathcal{U}(u_B)))$. By definition of Θ , \mathfrak{D} and functoriality of \mathcal{D} one has:

$$\begin{aligned} \Theta(a' \rfloor f^\#; g^\#; \mathcal{U}(\ell)) &= \bar{\partial}; \nu; !\nu; !!f; !g; \mathbf{d}; \ell = \text{id} \otimes \bar{\mathbf{d}}; \nu; !\nu; !!f; !g; \mathbf{d}; \ell \\ &= \text{id} \otimes \bar{\mathbf{d}}; \nu; !f; g; \ell \text{ by naturality and comonad laws} \\ &= \mathbf{c}_{A'} \otimes \bar{\mathbf{d}}_{A'}; \text{id} \otimes \bar{\mathbf{c}}_{A'}; \nu_{A'} \otimes \bar{\mathbf{d}}_{!A'}; \bar{\mathbf{c}}_{!A'}; !f; g; \ell \text{ from Equation (5)} \\ (\Theta; \mathcal{L}(\mathfrak{D}(f^\#), \ell))(g^\#) &= \mathfrak{D}(f^\#); \Theta(a \rfloor g^\#); \ell = \mathbf{c}_{A'} \otimes \text{id}_{A'}; (\nu; !f) \otimes (\bar{\partial}_A; f); \text{id} \otimes \bar{\mathbf{d}}; \bar{\mathbf{c}}_A; g; \ell \\ &= \mathbf{c}_{A'} \otimes \text{id}_{A'}; (\nu; !f) \otimes (\text{id} \otimes \bar{\mathbf{d}}_{A'}; \bar{\mathbf{c}}_{A'}; f); \text{id} \otimes \bar{\mathbf{d}}; \bar{\mathbf{c}}_A; g; \ell \\ &= \mathbf{c}_{A'} \otimes \bar{\mathbf{d}}_{A'}; (\nu; !f) \otimes (\bar{\mathbf{c}}_{A'}; f); \text{id} \otimes \bar{\mathbf{d}}; \bar{\mathbf{c}}_A; g; \ell \\ &= \mathbf{c}_{A'} \otimes \bar{\mathbf{d}}_{A'}; \nu \otimes \bar{\mathbf{c}}_{A'}; \text{id} \otimes \bar{\mathbf{d}}; \bar{\mathbf{c}}_{!A}; !f; g; \ell \text{ by naturality of } \bar{\mathbf{d}} \text{ and } \bar{\mathbf{c}} \quad \blacktriangleleft \end{aligned}$$

Note that Equation 23 describe (FTC1) only for functions which are null at some point a . Hence, we will need an abelian structure on homsets. Following Proposition 12, $\mathcal{C}(A, \text{git}E)$ has an additive structure \boxplus for any $A \in \mathcal{C}$ and $E \in \mathcal{L}$.

► **Lemma 33.** For morphisms $\ell, \ell' : !F \multimap E$ in \mathcal{L} , one has $\mathcal{E}'(\ell^\# \boxplus \ell'^\#); \mathbf{d} = \ell + \ell'$. In particular, when $\mathcal{C}(A, E)$ is an abelian group, transposition across the adjunction preserves subtraction.

26:16 Functorial Models of Differential Linear Logic

Assuming enrichment in abelian groups, we may now apply Θ to the translation of $\eta_A = \text{id}_{!A}^\#$ which equals 0 at 0, namely $\eta_A - \mathbf{w}_A; \bar{\mathbf{w}}_A; \eta_A$. Since $\bar{\mathbf{d}}; \mathbf{w} = 0$, we have:

$$\bar{\partial} = \Theta(\mathcal{U}(u_A)]\eta_A - \mathbf{w}_A; \bar{\mathbf{w}}_A; \eta_A) \quad (24)$$

Define $S_A := \Theta^{-1}(\text{id}_{!A \otimes A}) : (A, \mathcal{U}(u_A)) \longrightarrow (!A \otimes A, \mathcal{U}(u_{!A \otimes A}))$. Intuitively, S_A maps a function $F : !A \otimes A \multimap A$ to $x \mapsto \int_0^x F(t, dt)$. Projecting S_A in \mathcal{C} one gets $\Pi_2(S_A) : A \longrightarrow !A \otimes A$ and as such a morphism of \mathcal{L} :

$$s := !A \multimap !A \otimes A$$

► **Proposition 34.** *Consider a functorial model of differentiation, enriched over abelian groups, such that \mathcal{D} and R form adjunction via (23). Then (FTC2) as formalised in (22) holds.*

Proof. The proof amounts to a straightforward computation:

$$\begin{aligned} s; \bar{\partial} + !0 &= \Theta^{-1}(\text{id}_{!A \otimes A}); \bar{\partial} + \mathbf{w}_A; \bar{\mathbf{w}}_A = \Theta^{-1}(\text{id}_{!A \otimes A}; \bar{\partial}) + \mathbf{w}_A; \bar{\mathbf{w}}_A \text{ by naturality of } \Theta^{-1} \\ &= \Theta^{-1}(\Theta(\mathcal{U}(u_A)]\eta_A - \mathbf{w}_A; \mathbf{w}_A; \eta_A)) + \mathbf{w}_A; \bar{\mathbf{w}}_A \text{ by (24).} \\ &= \text{id}_{!A} - \mathbf{w}_A; \bar{\mathbf{w}}_A + \mathbf{w}_A; \bar{\mathbf{w}}_A = \text{id}_{!A} \text{ by Lemma 33} \quad \blacktriangleleft \end{aligned}$$

5 Conclusion

In this paper, we showed that differentiation as defined in Differential Linear Logic can be described as an extension of the formalism of linear–non-linear adjunction by a functor from a co-slice of the non-linear category to the linear category, $\mathcal{D} : (I \downarrow \mathcal{C}) \longrightarrow \mathcal{L}$, and that this recovers differential categories up to a well-pointedness condition. We then extended \mathcal{D} to an adjoint functor in order to describe antiderivatives. We believe that our presentation provides a more accessible axiomatization of differentiation that offers simpler intuitions.

The following avenues present natural next steps in this research.

Polarity and Differentiation. Once this work stabilizes, we would like to investigate what polarized models of DiLL should be, and refine diagram (8) in the case where \mathcal{L} is $*$ -autonomous, or more generally decomposed into a polarized chirality. One could also use chiralities to relax the coincidence of products and coproducts we relied on in the present paper (constraining us to semi-additive categories) to the mere existence of these structures.

Integration. Section 4 studies integration and (FTC2) only as an inverse to differentiation. A further step in this line of work would be to integrate the formulation of (FTC1) into this presentation. To complement our work on differentiation, we would like to understand how the axioms of integration can be lifted to a functorial presentation on their own to reach a more complete picture of the interactions between integration and differentiation. An interesting question would be whether any adjunction of the form $\mathcal{D} \dashv R$ results in antiderivatives, in the sense that the natural family of bijections is of the form proposed for Θ and its inverse.

Well-pointedness. We would like to find an extension of our set-up witnessing the generalized chain rule without recourse to the compromises of Remark 28. An avenue suggested by Zeinab Galal is to extend the domain of definition of \mathcal{D} to a comma category such as $(\mathcal{U} \downarrow \mathcal{C})$

to provide “generalized points” at which to differentiate, possibly abstracting the pairwise functorial differentiation of differentiable programming, $\mathcal{D}(f) : \delta_x \otimes v \mapsto \delta_{f(x)} \otimes D_x f(v)$. This was the origin of the functor \mathfrak{D} used to describe antiderivatives in Section 4.

Dependent types. Finally, this setting gives a dependent flavor to differentiation, and it feels natural to investigate a possible link with dependent types.

References

- 1 R. F. Blute, J. R. B. Cockett, J.-S. P. Lemay, and R. A. G. Seely. Differential categories revisited. *Applied Categorical Structures*, 2020. doi:10.1007/s10485-019-09572-y.
- 2 R. F. Blute, J. R. B. Cockett, and R. A. G. Seely. Differential categories. *Mathematical Structures in Computer Science*, 2006. doi:10.1017/S0960129506005676.
- 3 R. F. Blute, J. R. B. Cockett, and R. A. G. Seely. Cartesian differential categories. *Theory and Applications of Categories*, 2009.
- 4 Richard Blute, Thomas Ehrhard, and Christine Tasson. A convenient differential category. *Cahiers de topologie et géométrie différentielle catégoriques*, 53(3):211–232, 2012. arXiv:1006.3140.
- 5 A. Brunel, D. Mazza, and M. Pagani. Backpropagation in the Simply Typed Lambda-calculus with Linear Negation. *POPL*, 2020. doi:10.1145/3371132.
- 6 Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In *Computer Science Logic*, 2007. doi:10.1007/978-3-540-74915-8_24.
- 7 J. R. B. Cockett and J.-S. Lemay. Integral categories and calculus categories. *Mathematical Structures in Computer Science*, 29:243–308, 2019. doi:10.1017/S0960129518000014.
- 8 J. Robin B. Cockett and Geoff S. H. Cruttwell. Differential Structure, Tangent Structure, and SDG. *Appl. Categorical Struct.*, 22(2):331–417, 2014. doi:10.1007/s10485-013-9312-0.
- 9 J. Robin B. Cockett, Geoff S. H. Cruttwell, Jonathan Gallagher, Jean-Simon Pacaud Lemay, Benjamin MacAdam, Gordon D. Plotkin, and Dorette Pronk. Reverse derivative categories. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPICs*, pages 18:1–18:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CSL.2020.18.
- 10 Y. Dabrowski and M. Kerjean. Models of Linear Logic based on the Schwartz ϵ product. *Theory and Applications of Categories*, 2020. doi:10.48550/arXiv.1712.07344.
- 11 T. Ehrhard. An introduction to differential linear logic: proof-nets, models and antiderivatives. *Math. Struct. Comput. Sci.*, 2017. doi:10.1017/S0960129516000372.
- 12 T. Ehrhard and L. Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2), 2006. doi:10.1016/j.tcs.2006.08.003.
- 13 Conal Elliott. The simple essence of automatic differentiation. In *Proceedings of the ACM on Programming Languages (ICFP)*, 2018. doi:10.1145/3236765.
- 14 M. P. Fiore. Differential structure in models of multiplicative biadditive intuitionistic linear logic. In *TLCA*, 2007. doi:10.1007/978-3-540-73228-0_13.
- 15 J.-Y. Girard. Linear logic. *Theoret. Comput. Sci.*, 50(1), 1987. doi:10.1016/0304-3975(87)90045-4.
- 16 Marie Kerjean. A logical account for linear partial differential equations. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 589–598. ACM, 2018. doi:10.1145/3209108.3209192.
- 17 Paul-André Mellies. Categorical semantics of linear logic. *Société Mathématique de France*, 2008.
- 18 L. Schwartz. *Théorie des distributions*. Hermann, Paris, 1966.
- 19 R. A. G. Seely. Linear Logic, *-Autonomous Categories and Cofree Coalgebras. In *In Categories in Computer Science and Logic*. American Mathematical Society, 1989.