The Difference λ-Calculus: A Language for Difference Categories

Mario Alvarez-Picallo
University of Oxford, UK
mario.alvarez-picallo@cs.ox.ac.uk

C.-H. Luke Ong
University of Oxford, UK
luke.ong@cs.ox.ac.uk

Abstract

Cartesian difference categories are a recent generalisation of Cartesian differential categories which introduce a notion of "infinitesimal" arrows satisfying an analogue of the Kock-Lawvere axiom, with the axioms of a Cartesian differential category being satisfied only "up to an infinitesimal perturbation". In this work, we construct a simply-typed calculus in the spirit of the differential λ-calculus equipped with syntactic "infinitesimals" and show how its models correspond to difference λ-categories, a family of Cartesian difference categories equipped with suitably well-behaved exponentials.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Categorical semantics; Theory of computation → Lambda calculus

Keywords and phrases Cartesian difference categories, Cartesian differential categories, Change actions, Differential lambda-calculus, Difference lambda-calculus

Digital Object Identifier 10.4230/LIPIcs.FSCD.2020.32

1 Introduction

A recent series of works introduced the concept of change actions and differential maps between them [5, 4] in order to account for settings equipped with derivative-like operations. Although the motivating example was the eminently practical field of incremental computation, these structures appear in more abstract settings such as the calculus of finite differences and Cartesian differential categories.

Of particular interest are Cartesian difference categories [2], a well-behaved class of change action models [4] that are much closer to the strong axioms of a Cartesian differential category [8] while remaining general enough for interpreting discrete calculus. A Cartesian difference category is a left additive category equipped with an "infinitesimal extension", an operation that sends an arrow f to an arrow $\varepsilon(f)$ which should be understood as f being multiplied by an "infinitesimal" element – infinitesimal in the sense that it verifies the Kock-Lawvere axiom from synthetic differential geometry (see [14] for an introduction to SDG).

The interest of Cartesian differential categories is in part motivated by the fact that they provide models for the differential λ-calculus [13, 11], which extends the λ-calculus with linear combinations of terms and an operator that differentiates arbitrary λ-abstractions. The claim that differentiation in the differential λ-calculus corresponds to the standard, "analytic" notion is then justified by its interpretation in (a well-behaved class of) Cartesian differential categories [9, 16].

It is reasonable to ask, then, whether there is a similar calculus that captures the behavior of derivatives in difference categories – especially since, as it has been shown, these subsume differential categories. The issue is far from trivial, as many of the properties of the differential λ-calculus crucially hinge on derivatives being linear. Through this work we provide an affirmative answer to this question by defining untyped and simply-typed variants for a simple calculus which extends the differential λ-calculus with a notion of derivative more suitable to the Cartesian difference setting.

For brevity most proofs have been omitted. Some of the most important proof sketches and the auxiliary results involved can be found in the appendices. The full details are available as part of the first author's doctoral dissertation [1].

2 Cartesian Difference Categories

The theory of Cartesian difference categories is developed and discussed at length in [2, 3], but we present here the main definitions and results which we will use throughout the paper, referring the reader to [3] for the proofs.

▶ **Definition 1.** A *Cartesian left additive category* ([8, Definition 1.1.1.]) $\mathbf{C}$ is a Cartesian category where every hom-set $\mathbf{C}[A,B]$ is endowed with the structure of a commutative monoid $(\mathbf{C}[A,B], +, 0)$ such that $0 \circ f = 0$, $(f+g)\circ h = (f \circ h) + (g \circ h)$ and $\langle f_1, f_2\rangle + \langle g_1, g_2\rangle = \langle f_1 + g_1, f_2 + g_2\rangle$.

An *infinitesimal extension* ([2, Definition 8]) in a Cartesian left additive category $\mathbf{C}$ is a choice of a monoid homomorphism $\varepsilon : \mathbf{C}[A,B] \to \mathbf{C}[A,B]$ for every hom-set in $\mathbf{C}$. That is, $\varepsilon(f+g) = \varepsilon(f) + \varepsilon(g)$ and $\varepsilon(0) = 0$. Furthermore, we require that ε be compatible with the Cartesian structure, in the sense that $\varepsilon(\langle f, g\rangle) = \langle \varepsilon(f), \varepsilon(g)\rangle$.

▶ **Definition 2.** A *Cartesian difference category* ([2, Definition 9]) is a Cartesian left additive category with an infinitesimal extension ε which is equipped with a *difference combinator* $\partial[-]$ of the form:

$$\frac{f : A \to B}{\partial[f] : A \times A \to B}$$

satisfying the following coherence conditions (writing $\partial^2[f]$ for $\partial[\partial[f]]$):

[C∂.0] $f \circ (x + \varepsilon(u)) = f \circ x + \varepsilon(\partial[f] \circ \langle x, u\rangle)$
[C∂.1] $\partial[f+g] = \partial[f] + \partial[g]$, $\partial[0] = 0$, and $\partial[\varepsilon(f)] = \varepsilon(\partial[f])$
[C∂.2] $\partial[f] \circ \langle x, u+v\rangle = \partial[f] \circ \langle x, u\rangle + \partial[f] \circ \langle x + \varepsilon(u), v\rangle$ and $\partial[f] \circ \langle x, 0\rangle = 0$
[C∂.3] $\partial[\mathbf{id}_A] = \pi_2$ and $\partial[\pi_1] = \pi_1 \circ \pi_2$ and $\partial[\pi_2] = \pi_2 \circ \pi_2$
[C∂.4] $\partial[\langle f, g\rangle] = \langle \partial[f], \partial[g]\rangle$ and $\partial[!_A] = !_{A\times A}$
[C∂.5] $\partial[g \circ f] = \partial[g] \circ \langle f \circ \pi_1, \partial[f]\rangle$
[C∂.6] $\partial^2[f] \circ \langle\langle x, u\rangle, \langle 0, v\rangle\rangle = \partial[f] \circ \langle x + \varepsilon(u), v\rangle$
[C∂.7] $\partial^2[f] \circ \langle\langle x, u\rangle, \langle v, 0\rangle\rangle = \partial^2[f] \circ \langle\langle x, v\rangle, \langle u, 0\rangle\rangle$

As noted in [2], the axioms in a Cartesian differential category (see e.g. **[C∂.1–7]** in [8]) correspond to the analogous axioms of the Cartesian difference operator, modulo certain "infinitesimal" terms, i.e. terms of the form $\varepsilon(f)$. We state here the following two properties, whose proofs can be found in [3].

▶ **Lemma 3.** Given any map $f : A \to B$ in a Cartesian difference category $\mathbf{C}$, its derivative $\partial[f]$ satisfies the following equations:

i. $\partial[f] \circ \langle x, \varepsilon(u)\rangle = \varepsilon(\partial[f]) \circ \langle x, u\rangle$
ii. $\varepsilon(\partial^2[f]) \circ \langle\langle x, u\rangle, \langle v, 0\rangle\rangle = \varepsilon^2(\partial^2[f]) \circ \langle\langle x, u\rangle, \langle v, 0\rangle\rangle$

3 Difference λ-Categories

In order to give a semantics for the differential λ-calculus, it does not suffice to ask for a Cartesian differential category equipped with exponentials – the exponential structure has to be compatible with both the additive and the differential structure, in the sense of [9, Definition 4.4]. For difference categories we will require an identical equation, together with a condition requiring higher-order functions to respect the infinitesimal extension.

▶ **Definition 4.** We remind the reader that a Cartesian left additive category is *Cartesian closed left additive* ([9, Definition 4.2]) whenever it is Cartesian closed and satisfies $\Lambda(f+g) = \Lambda(f) + \Lambda(g), \Lambda(0) = 0$.

A Cartesian difference category $\mathbf{C}$ is a *difference λ-category* if it Cartesian closed left additive and satisfies the following additional axioms:

[$\partial\lambda$.**1**] $\partial[\Lambda(f)] = \Lambda(\partial[f] \circ \langle(\pi_1 \times \mathbf{id}), (\pi_2 \times 0)\rangle)$
[$\partial\lambda$.**2**] $\Lambda(\varepsilon(f)) = \varepsilon(\Lambda(f))$

Equivalently, let **sw** denote the map $\langle\langle\pi_{11}, \pi_2\rangle, \pi_{21}\rangle : (A \times B) \times C \to (A \times C) \times B$. Then the condition [$\partial\lambda$.**1**] can be written in terms of **sw** as:

$$\partial[\Lambda(f)] \coloneqq \Lambda(\partial[f] \circ (\mathbf{id} \times \langle \mathbf{id}, 0\rangle) \circ \mathbf{sw})$$

Axiom [$\partial\lambda$.**1**] is identical to its differential analogue [9, Definition 4.4], and it follows the same broad intuition. Given a map $f : A \times B \to C$, we usually understand the composite $\partial[f] \circ (\mathbf{id}_{A\times B} \times (\mathbf{id}_A \times 0_B)) : (A \times B) \times A \to C$ as a partial derivative of f with respect to its first argument. Hence, just as it was with differential λ-categories, axiom [$\partial\lambda$.**1**] states that the derivative of a curried function is precisely the derivative of the uncurried function with respect to its first argument.

▶ **Example 5.** Let $\mathbf{C}$ be a differential λ-category. Then the trivial Cartesian difference category obtained by setting $\varepsilon(f) = 0$ (as in [2, Proposition 1]) is a difference λ-category. Furthermore, the Kleisli category $\mathbf{C}_{\mathsf{T}}$ induced by its tangent bundle monad (as in [2, Proposition 6]) is also a difference λ-category.

▶ **Example 6.** The category $\overline{\mathbf{Ab}}$ ([2, Section 5.2]), which has Abelian groups as objects and arbitrary functions between their carrier sets as morphisms, is a difference λ-category with infinitesimal extension $\varepsilon(f) = f$ and difference combinator $\partial[f](x, u) = f(x + u) - f(x)$. Given groups G, H, the exponential $G \Rightarrow H$ is the set of (set-theoretic) functions from G into H, endowed with the group structure of H lifted pointwise (that is, $(f+g)(x) = f(x)+g(x)$). Evidently the exponential respects the monoidal structure and the infinitesimal extension. We check that it also verifies axiom [$\partial\lambda$.**1**]:

$$\begin{aligned}\partial[\Lambda(f)](x, u)(y) &= \Lambda(f)(x + u)(y) - \Lambda(f)(x)(y)\\ &= f(x + u, y) - f(x, y)\\ &= \Lambda(\partial[f] \circ (\mathbf{id} \times \langle\mathbf{id}, 0\rangle) \circ \mathbf{sw})(x, u)(y)\end{aligned}$$

A central property of differential λ-categories is a deep correspondence between differentiation and the evaluation map. As one would expect, the partial derivative of the evaluation map gives one a first-class derivative operator (see, for example, [9, Lemma 4.5], which provides an interpretation for the differential substitution operator in the differential λ-calculus). This property still holds in difference categories, although its formulation is somewhat more involved.

▶ **Lemma 7.** For any **C**-morphisms $\Lambda(f) : A \to (B \Rightarrow C), e : A \to B$, the following identities hold:

i. $\partial\left[\mathbf{ev} \circ \langle \Lambda(f), e\rangle\right] = \mathbf{ev} \circ \langle \partial\left[\Lambda(f)\right], e \circ \pi_1\rangle + \partial\left[f\right] \circ \langle\langle \pi_1 + \varepsilon(\pi_2), e \circ \pi_1\rangle, \langle 0, \partial\left[e\right]\rangle\rangle$
ii. $\partial\left[\mathbf{ev} \circ \langle \Lambda(f), e\rangle\right] = \mathbf{ev} \circ \langle \partial\left[\Lambda(f)\right], e \circ \pi_1 + \varepsilon(\partial\left[e\right])\rangle + \partial\left[f\right] \circ \langle\langle \pi_1, e \circ \pi_1\rangle, \langle 0, \partial\left[e\right]\rangle\rangle$

A number of additional auxiliary results from the differential setting also hold for difference λ-categories, possibly with the introduction of some extra "infinitesimal" terms. Some of these will be stated in Appendix C.

4 The Difference λ-Calculus

We now set out to define λ_ε, a calculus in the vein of the differential λ-calculus which adds infinitesimal extensions and relaxes the linearity requirement. We proceed in a manner similar to Vaux [17, 18] in his treatment of the algebraic λ-calculus; that is, we will first define a set of "unrestricted" terms Λ_ε which we will later consider up to an equivalence relation arising from the theory of difference categories.

▶ **Definition 8.** The set Λ_ε of *unrestricted terms* of the λ_ε-calculus is given by the following inductive definition:

$$\text{Terms:} \quad s, t, e \quad := \quad x \mid \lambda x.t \mid (s\ t) \mid \mathrm{D}(s) \cdot t \mid \varepsilon t \mid s + t \mid 0$$

assuming a countably infinite set of variables $x, y, z, \ldots$ is given. In what follows, we will speak of terms only up to α-equivalence, and assume by convention that all bound variables appearing in any term $t \in \Lambda_\varepsilon$ are different from its free variables.

Further to α-equivalence, we introduce here the notion of differential equivalence of terms. The role of this relation is, as in [18], to enforce the elementary algebraic properties of sums and actions. For example, we wish to treat the terms $\lambda x.(0 + \varepsilon(s + t))$ and $(\lambda x.\varepsilon t) + (\lambda x.\varepsilon s)$ as if they were equivalent (as it will be the case in the models). This equivalence relation also has the role of ensuring that the axioms of a Cartesian difference category are satisfied.

▶ **Definition 9.** A binary relation $\sim \subseteq \Lambda_\varepsilon \times \Lambda_\varepsilon$ is *contextual* whenever $t \sim t'$ and $s \sim s'$ implies

$$\lambda x.t \sim \lambda x.t' \quad \varepsilon t \sim \varepsilon t' \quad s\ t \sim s'\ t' \quad \mathrm{D}(s) \cdot t \sim \mathrm{D}(s') \cdot t' \quad s + t \sim s' + t'$$

▶ **Definition 10.** *Differential equivalence* $\sim_\varepsilon \subseteq \Lambda_\varepsilon \times \Lambda_\varepsilon$ is the least equivalence relation which is contextual and contains the relation $\sim^1_\varepsilon$ in Figure 1 below.

The above conditions can be separated in a number of conceptually distinct groups corresponding to their purpose. These are as follows:

- The first block of equations states that $+$ and 0 define a commutative monoid and that ε is a monoid homomorphism.
- The second block of equations amounts to stating that the monoid and infinitesimal extension structure on functions is pointwise.
- The third block of equations implies (and is equivalent to stating) that addition and infinitesimal extension are "linear", in the sense that they coincide with their own derivatives (that is, $\partial\left[+\right] = + \circ \pi_2$ and $\partial\left[\varepsilon\right] = \varepsilon$).

$$
\begin{array}{rcl}
(s+t)+e & \sim^1_\varepsilon & s+(t+e) \\
s+0 & \sim^1_\varepsilon & s \\
s+t & \sim^1_\varepsilon & t+s \\
\varepsilon 0 & \sim^1_\varepsilon & 0 \\
\varepsilon(s+t) & \sim^1_\varepsilon & \varepsilon s + \varepsilon t
\end{array}
\qquad
\begin{array}{rcl}
\lambda x.0 & \sim^1_\varepsilon & 0 \\
\lambda x.(s+t) & \sim^1_\varepsilon & (\lambda x.s) + (\lambda x.t) \\
\lambda x.\varepsilon t & \sim^1_\varepsilon & \varepsilon(\lambda x.t) \\
0\ s & \sim^1_\varepsilon & 0 \\
(s+t)\ e & \sim^1_\varepsilon & (s\ e) + (t\ e) \\
(\varepsilon s)\ t & \sim^1_\varepsilon & \varepsilon(s\ t)
\end{array}
$$

$$
\begin{array}{rcl}
\mathrm{D}(0)\cdot e & \sim^1_\varepsilon & 0 \\
\mathrm{D}(s+t)\cdot e & \sim^1_\varepsilon & (\mathrm{D}(s)\cdot e) + (\mathrm{D}(t)\cdot e) \\
\mathrm{D}(\varepsilon t)\cdot e & \sim^1_\varepsilon & \varepsilon(\mathrm{D}(t)\cdot e)
\end{array}
\qquad
\begin{array}{rcl}
\mathrm{D}(s)\cdot 0 & \sim^1_\varepsilon & 0 \\
\mathrm{D}(s)\cdot(t+e) & \sim^1_\varepsilon & \mathrm{D}(s)\cdot t + \mathrm{D}(s)\cdot e + \varepsilon(\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e) \\
\mathrm{D}(s)\cdot(\varepsilon t) & \sim^1_\varepsilon & \varepsilon(\mathrm{D}(s)\cdot t) \\
\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e & \sim^1_\varepsilon & \mathrm{D}(\mathrm{D}(s)\cdot e)\cdot t \\
\varepsilon^2\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e & \sim^1_\varepsilon & \varepsilon\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e \\
s\ (t+\varepsilon e) & \sim^1_\varepsilon & (s\ t) + \varepsilon((\mathrm{D}(s)\cdot e)\ t)
\end{array}
$$

Figure 1 Differential equivalence on unrestricted Λ_ε-terms.

- The fourth block of equations states structural properties of the derivative, such as the derivative conditions and the commutativity of second derivatives. Similar equations are also present in the differential λ-calculus, where they state instead that the derivative is additive.

Most of these equations correspond directly to properties of Cartesian difference categories, with the only exception being the requirement that $\mathrm{D}(s)\cdot(\varepsilon t) \sim_\varepsilon \varepsilon(\mathrm{D}(s)\cdot t)$ and the "duplication of infinitesimals" in $\varepsilon\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e = \varepsilon^2\mathrm{D}(\mathrm{D}(s)\cdot t)\cdot e$, which should be read as a syntactic formulation of the equations in Lemma 3. It would be possible to give an alternative presentation of the calculus where these equivalences are oriented and understood as reduction rules, and thus part of the operational semantics (as in e.g. Arrighi et al.'s treatment of the linear λ-calculus [7, 6]). While such a formulation would better reflect how a real machine might evaluate these expressions, it would make the study of confluence and termination harder.

▶ **Definition 11.** The set λ_ε of *well-formed terms*, or simply *terms*, of the λ_ε-calculus is defined as the quotient set $\lambda_\varepsilon := \Lambda_\varepsilon/\sim_\varepsilon$. Whenever t is an unrestricted term, we write $\underline{t}$ to refer to the well-formed term represented by t, that is to say, the $\sim_\varepsilon$-equivalence class of t.

The notion of differential equivalence allows us to ensure that our calculus reflects the laws of the underlying models, but has the unintended consequence that our λ_ε-terms are equivalence classes, rather than purely syntactic objects. We will proceed by defining a notion of canonical form of a term and a canonicalization algorithm which explicitly constructs the canonical form of any given term, thus proving that $\sim_\varepsilon$ is decidable.

▶ **Definition 12.** We define the sets $\mathrm{B}_\varepsilon \subset \mathrm{B}^+_\varepsilon \subset \mathrm{B}^*_\varepsilon \subset \mathrm{C}^+_\varepsilon \subset \mathrm{C}_\varepsilon (\subset \Lambda_\varepsilon)$ of *basic*, *positive*, *additive*, *positive canonical* and *canonical* terms according to the following grammar:

$$
\begin{array}{rrcl}
\text{Basic terms:} & s^{\mathbf{b}}, t^{\mathbf{b}}, e^{\mathbf{b}} \in \mathrm{B}_\varepsilon & := & x \mid \lambda x.t^{\mathbf{b}} \mid (s^{\mathbf{b}}\ t^*) \mid \mathrm{D}(s^{\mathbf{b}})\cdot t^{\mathbf{b}} \\
\text{Positive terms:} & s^+, t^+, e^+ \in \mathrm{B}^+_\varepsilon & := & s^{\mathbf{b}} \mid s^{\mathbf{b}} + (t^+) \\
\text{Additive terms:} & s^*, t^*, e^* \in \mathrm{B}^*_\varepsilon & := & 0 \mid s^+ \\
\text{Positive canonical terms:} & S^+, T^+ \in \mathrm{C}_\varepsilon & := & \varepsilon^k s^{\mathbf{b}} \mid \varepsilon^k s^{\mathbf{b}} + (S^+) \\
\text{Canonical terms:} & S, T \in \mathrm{C}_\varepsilon & := & 0 \mid S^+
\end{array}
$$

We will sometimes abuse the notation and write $\underline{t^*}$ or $\underline{t^{\mathbf{b}}}$ to denote well-formed terms whose canonical form is an additive or basic term respectively.

Since every syntactic construct is additive except for application, basic terms may only contain additive terms as the arguments to a function application. As infinitesimal extensions are themselves additive, we also want to disallow terms such as $\varepsilon(s+t)$, instead factoring out the extension into $\varepsilon s + \varepsilon t$. A general canonical term $T \in \mathrm{C}_\varepsilon$ then has the form:

$$T = \varepsilon^{k_1} t_1^{\mathbf{b}} + (\varepsilon^{k_2} t_2^{\mathbf{b}} + (\cdots + \varepsilon^{k_n} t_n^{\mathbf{b}}) \cdots)$$

That is to say, a canonical term is similar to a polynomial with coefficients in the set of basic terms and a variable ε (but note that canonical terms are always written in their "fully distributed" form, that is, we write $\varepsilon s + (\varepsilon t + \varepsilon^2 e)$ rather than $\varepsilon((s+t) + \varepsilon e)$).

We will freely abuse notation and write $\sum_{i=1}^n \varepsilon^{k_i} t_i^{\mathbf{b}}$ to denote a general canonical term, as this form is easier to manipulate in many cases. In particular, the canonical term 0 is precisely the sum of zero terms. We will also write $S + T$ to refer to the obvious canonical term obtained by adding S and T and associating all the additions to the right.

▶ **Definition 13.** Given an unrestricted λ_ε-term $t \in \Lambda_\varepsilon$, we define its *canonical form* $\mathbf{can}(t)$ by structural induction on t as follows:

- $\mathbf{can}(0) \coloneqq 0$
- $\mathbf{can}(x) \coloneqq x$
- $\mathbf{can}(s+t) \coloneqq \mathbf{can}(s) + \mathbf{can}(t)$
- $\mathbf{can}(\varepsilon t) \coloneqq \varepsilon^* \mathbf{can}(t)$, where:

$$\varepsilon^* T \coloneqq \begin{cases} T & \text{if } T = \varepsilon^k \mathrm{D}(\mathrm{D}(e) \cdot u) \cdot v \\ \varepsilon^{k+1} t^{\mathbf{b}} & \text{if } T = \varepsilon^k t^{\mathbf{b}} \neq \varepsilon^k \mathrm{D}(\mathrm{D}(e) \cdot u) \cdot v \\ \sum_{i=1}^n \varepsilon^* T_i & \text{if } T = \sum_{i=1}^n T_i \end{cases}$$

- If $\mathbf{can}(t) = \sum_{i=1}^n \varepsilon^{k_i} t_i^{\mathbf{b}}$ then:

$$\mathbf{can}(\lambda x.t) \coloneqq \sum_{i=1}^n \varepsilon^{k_i} (\lambda x.t_i^{\mathbf{b}})$$

- If $\mathbf{can}(s) = \sum_{i=1}^n \varepsilon^{k_i} s_i^{\mathbf{b}}$ and $\mathbf{can}(t) = T$ then:

$$\mathbf{can}(\mathrm{D}(s) \cdot t) \coloneqq \sum_{i=1}^n ((\varepsilon^*)^{k_i} \mathbf{reg}(s_i^{\mathbf{b}}, T))$$

  where the *regularization* $\mathbf{reg}(s, T)$ is defined by structural induction on T:

$$\begin{aligned} \mathbf{reg}(s, 0) &\coloneqq 0 \\ \mathbf{reg}(s, \varepsilon^k t^{\mathbf{b}} + T') &\coloneqq \left[(\varepsilon^*)^k \, \mathrm{D}(s) \cdot t^{\mathbf{b}}\right] + [\mathbf{reg}(s, T')] + \left[(\varepsilon^*)^{k+1} \, \mathrm{D}^*(\mathbf{reg}(s, T')) \cdot t^{\mathbf{b}}\right] \end{aligned}$$

  and D^* denotes the extension of D by additivity in its first argument, that is to say:

$$\mathrm{D}^*\left(\sum_{i=1}^n \varepsilon^{k_i} s_i^{\mathbf{b}}\right) \cdot t^{\mathbf{b}} \coloneqq \sum_{i=1}^n \varepsilon^{k_i} \left(\mathrm{D}\left(s_i^{\mathbf{b}}\right) \cdot t^{\mathbf{b}}\right)$$

  Observe that, whenever S is canonical and $t^{\mathbf{b}}$ is basic, the term $\mathrm{D}^*(S) \cdot t^{\mathbf{b}}$ is also canonical. Therefore, by induction, the regularization $\mathbf{reg}\left(s^{\mathbf{b}}, T\right)$ is indeed a canonical term, since canonicity is preserved by $\varepsilon^*, +$.

- If $\mathbf{can}\,(s) = \sum_{i=1}^{n} \varepsilon^{k_i} s_i^{\mathbf{b}}$ and $\mathbf{can}\,(t) = T$, then:

$$\mathbf{can}\,(s\ t) := \left[\sum_{i=1}^{n} \varepsilon^{k_i} \left(s_i^{\mathbf{b}}\ \mathbf{pri}(T)\right)\right] + \left[\varepsilon^{*}\left(\sum_{i=1}^{n} \mathbf{ap}(\mathbf{reg}\left(s_i^{\mathbf{b}}, \mathbf{tan}(T)\right), \mathbf{pri}(T))\right)\right]$$

where $\mathbf{ap}$ is the additive extension of application: $\mathbf{ap}\left(\sum_{i=1}^{n} \varepsilon^{k_i} s_i^{\mathbf{b}}, t^{\mathbf{b}}\right) := \sum_{i=1}^{n} \varepsilon^{k_i} (s_i^{\mathbf{b}}\ t^{\mathbf{b}})$ and the primal $\mathbf{pri}$ and tangent $\mathbf{tan}$ components of a canonical term T correspond respectively to the basic terms with zero and non-zero ε coefficients:

$$\begin{array}{rcl@{\qquad}rcl}
\mathbf{pri}\,(0) & := & 0 & \mathbf{tan}\,(0) & := & 0 \\
\mathbf{pri}\left(\varepsilon^{k+1} t^{\mathbf{b}} + T'\right) & := & \mathbf{pri}\,(T') & \mathbf{tan}\left(\varepsilon^{k+1} t^{\mathbf{b}} + T'\right) & := & \varepsilon^{k} t^{\mathbf{b}} + \mathbf{tan}\,(T') \\
\mathbf{pri}\left(\varepsilon^{0} t^{\mathbf{b}} + T'\right) & := & t^{\mathbf{b}} + \mathbf{pri}\,(T') & \mathbf{tan}\left(\varepsilon^{0} t^{\mathbf{b}} + T'\right) & := & \mathbf{tan}\,(T')
\end{array}$$

▶ **Theorem 14.** Every unrestricted Λ_ε-term is differentially equivalent to its canonical form $\mathbf{can}\,(t)$. That is to say, for all $t \in \Lambda_\varepsilon$ we have $t \sim_\varepsilon \mathbf{can}\,(t)$.

This canonicalization procedure is the result of orienting the equivalences in Definition 10. Note, however, that while most of these equivalences have a "natural" orientation to them, two of them are entirely symmetrical: those being commutativity of the sum and the derivative. Barring the imposition of some arbitrary total ordering on terms which would allow us to prefer the term $x + y$ over $y + x$ (or vice versa), we settle for our canonical forms to be unique "up to" these commutativity conditions.

▶ **Definition 15.** *Permutative equivalence* $\sim_+ \subseteq \Lambda_\varepsilon \times \Lambda_\varepsilon$ is the least equivalence relation which is contextual and satisfies the following properties:

$$\begin{array}{rcl}
s + (t + e) & \sim_+ & (s + t) + e \\
s + t & \sim_+ & t + s \\
\mathrm{D}(\mathrm{D}(s) \cdot t) \cdot e & \sim_+ & \mathrm{D}(\mathrm{D}(s) \cdot e) \cdot t
\end{array}$$

▶ **Theorem 16.** Given unrestricted terms $s, t \in \Lambda_\varepsilon$, they are differentially equivalent if and only if their canonical forms are permutatively equivalent. More succinctly, $s \sim_\varepsilon t$ if and only if $\mathbf{can}\,(s) \sim_+ \mathbf{can}\,(t)$

▶ **Corollary 17.** The set λ_ε of well-formed terms corresponds precisely to the set of canonical terms up to permutative equivalence $\mathrm{C}_\varepsilon/\!\sim_+$.

4.1 Substitution

As is usual, our calculus features two different kinds of application: standard function application, represented as $(s\ t)$; and differential application, represented as $\mathrm{D}(s) \cdot t$. These two give rise to two different notions of substitution. The first is, of course, the usual capture-avoiding substitution. The second, differential substitution, is similar to the equivalent notion in the differential λ-calculus, as it arises from the same chain rule that is satisfied in both Cartesian differential categories and change action models.

▶ **Definition 18.** Given terms $s, t \in \Lambda_\varepsilon$ and a variable x, the *capture-avoiding substitution* of s for x in t (which we write as $t\,[s/x]$) is defined by induction on the structure of t:

$$\begin{array}{rcll@{\qquad}rcl}
x\,[s/x] & := & s & & (\mathrm{D}(t) \cdot e)\,[s/x] & := & \mathrm{D}(t\,[s/x]) \cdot (e\,[s/x]) \\
y\,[s/x] & := & y & \text{if } x \neq y & (\varepsilon t)\,[s/x] & := & \varepsilon(t\,[s/x]) \\
(\lambda y.t)\,[s/x] & := & \lambda y.(t\,[s/x]) & \text{if } y \notin \mathrm{FV}(s) & (t + e)\,[s/x] & := & (t\,[s/x]) + (e\,[s/x]) \\
(t\ e)\,[s/x] & := & (t\,[s/x])(e\,[s/x]) & & 0\,[s/x] & := & 0
\end{array}$$

▶ **Proposition 19.** Capture-avoiding substitution respects differential equivalence. That is to say, whenever $s \sim_\varepsilon s'$ and $t \sim_\varepsilon t'$, it is the case that $t\,[s/x] \sim_\varepsilon t'\,[s'/x]$.

▶ **Definition 20.** Given terms $s, t \in \Lambda_\varepsilon$ and a variable x *which is not free in* s the *differential substitution* of s for x in t, which we write as $\frac{\partial t}{\partial x}(s)$, is defined by induction on the structure of t:

$$\begin{aligned}
\frac{\partial x}{\partial x}(s) &:= s \\
\frac{\partial y}{\partial x}(s) &:= 0 && \text{if } x \neq y \\
\frac{\partial(\lambda y.t)}{\partial x}(s) &:= \lambda y.\left(\frac{\partial t}{\partial x}(s)\right) && \text{if } y \notin \mathrm{FV}(s) \\
\frac{\partial(t\ e)}{\partial x}(s) &:= \left[\mathrm{D}(t)\cdot\left(\frac{\partial e}{\partial x}(s)\right)\ e\right] + \left[\frac{\partial t}{\partial x}(s)\ \ (e\,[(x+\varepsilon s)/x])\right] \\
\frac{\partial(\mathrm{D}(t)\cdot e)}{\partial x}(s) &:= \mathrm{D}(t)\cdot\left(\frac{\partial e}{\partial x}(s)\right) + \mathrm{D}\left(\frac{\partial t}{\partial x}(s)\right)\cdot(e\,[(x+\varepsilon s)/x]) \\
&\qquad + \varepsilon\mathrm{D}(\mathrm{D}(t)\cdot e)\cdot\left(\frac{\partial e}{\partial x}(s)\right) \\
\frac{\partial(\varepsilon t)}{\partial x}(s) &:= \varepsilon\left(\frac{\partial t}{\partial x}(s)\right) \\
\frac{\partial(t+e)}{\partial x}(s) &:= \left(\frac{\partial t}{\partial x}(s)\right) + \left(\frac{\partial e}{\partial x}(s)\right) \\
\frac{\partial 0}{\partial x}(s) &:= 0
\end{aligned}$$

We write $\frac{\partial^k t}{\partial(x_1,\ldots,x_k)}(u_1,\ldots,u_k)$ to denote a sequence of nested differential substitutions.

Most of the cases of differential substitution are identical to those in the differential λ-calculus – our definition in fact coincides exactly with the original notion of differential substitution in e.g [12], provided that one assumes the identity $\varepsilon t = 0$ for all terms. This reflects the fact that every Cartesian differential category is in fact a Cartesian difference category with trivial infinitesimal extension.

All the differences in this definition stem from the failure of derivatives to be additive in the setting of Cartesian difference categories. Consider the case for $\frac{\partial \mathrm{D}(t)\cdot s}{\partial x}(e)$, and remember that the "essence" of a derivative in the setting of difference categories lies in **[C∂.0]**, that is to say, if $t(x)$ is a term with a free variable x, we seek our notion of differential substitution to satisfy a condition akin to Taylor's formula:

$$t(x+\varepsilon y) \sim_\varepsilon t(x) + \varepsilon\frac{\partial t}{\partial x}(y)$$

When the term t is a differential application, and assuming the above "Taylor's formula" holds for all of its subterms (which we will show later), this leads us to the following informal argument:

$$\begin{aligned}
&\mathrm{D}(t(x+\varepsilon y))\cdot(s(x+\varepsilon y)) \sim_\varepsilon \mathrm{D}\left(t(x)+\varepsilon\frac{\partial t}{\partial x}(y)\right)\cdot(s(x+\varepsilon y)) \\
&\quad\sim_\varepsilon \mathrm{D}(t(x))\cdot(s(x)) + \varepsilon\mathrm{D}(t(x))\cdot\left(\frac{\partial s}{\partial x}(y)\right) + \varepsilon\mathrm{D}\left(\frac{\partial t}{\partial x}(y)\right)\cdot(s(x+\varepsilon y)) \\
&\qquad\quad + \varepsilon^2\mathrm{D}(\mathrm{D}(t(x))\cdot(s(x)))\cdot\left(\frac{\partial s}{\partial x}(y)\right)
\end{aligned}$$

From this calculation, the differential substitution for this case arises naturally as it results from factoring out the ε and noticing that the resulting expression has precisely the correct shape to be Taylor's formula for the case of differential application. The case for standard application can be derived similarly, although the involved terms are simpler. Differential substitution verifies some useful properties, which we state below (mechanised proofs are available as part of the author's doctoral dissertation [1], although the details are more cumbersome than enlightening).

▶ **Proposition 21.** Differential substitution respects differential equivalence. That is to say, whenever $s \sim_\varepsilon s'$ and $t \sim_\varepsilon t'$, it is the case that $\frac{\partial t}{\partial x}(s) \sim_\varepsilon \frac{\partial t'}{\partial x}(s')$.

▶ **Proposition 22.** Whenever x is not free in t, then $\frac{\partial t}{\partial x}(u) \sim_\varepsilon 0$.

▶ **Proposition 23.** Whenever x is not free in u, v, then:

$$\frac{\partial^2 t}{\partial x^2}(u, v) \sim_\varepsilon \frac{\partial^2 t}{\partial x^2}(v, u)$$

As we have previously mentioned, the rationale behind our specific definition of differential substitution is that it should satisfy some sort of "Taylor's formula" (or rather, Kock-Lawvere formula), in the following sense:

▶ **Theorem 24.** For any unrestricted terms s, t, e and any variable x which does not appear free in e, we have

$$s\,[(t + \varepsilon e)/x] \sim_\varepsilon s\,[t/x] + \varepsilon \left(\left(\frac{\partial s}{\partial x}(e) \right) [t/x] \right)$$

We will often refer to the right-hand side of the above equivalence as the *Taylor expansion* of the corresponding term in the left-hand side.

One consequence of this "syntactic Taylor's formula" is that derivatives in the difference λ-calculus can be computed by a sort of quasi-automatic-differentiation algorithm: given an expression of the form $\lambda x.s$, its derivative at point t along u can be computed by reducing the differential application $(\mathrm{D}(\lambda x.s) \cdot (u))\ t$ which, as we shall see later, reduces (by definition) to $\left(\frac{\partial s}{\partial x}(u)\right)[t/x]$. Alternatively, we can simply evaluate $(\lambda x.s)\ (t + \varepsilon(u))$ to compute $s\,[t + \varepsilon(u)/x]$ which, by Theorem 24 is equivalent to $s\,[t/x] + \varepsilon\left(\left(\frac{\partial s}{\partial x}(u)\right)[t/x]\right)$. In an appropriate setting (i.e. one where subtraction of terms is allowed and ε admits an inverse) the derivative can then be extracted from this result by extracting the term under the ε. This process is remarkably similar to forward-mode automatic differentiation, where derivatives are computed by adding "perturbations" to the program input.

▶ **Theorem 25.** Differential substitution is regular, that is, for any unrestricted terms s, u, v where x does not appear free in either u or v, we have:

$$\frac{\partial s}{\partial x}(0) \sim_\varepsilon 0$$

$$\frac{\partial s}{\partial x}(u + v) \sim_\varepsilon \frac{\partial s}{\partial x}(u) + \left(\frac{\partial s}{\partial x}(v)\right)[x + \varepsilon u/x]$$

4.2 The Operational Semantics of λ_ε

With the substitution operations we have introduced so far, we can now proceed to give a small-step operational semantics as a reduction system.

▶ **Definition 26.** The *one-step reduction relation* [1] $\rightsquigarrow\ \subseteq \Lambda_\varepsilon \times \Lambda_\varepsilon$ is the least contextual relation that contains the following reduction rules:

$$\begin{aligned} (\lambda x.t)\ s &\rightsquigarrow_\beta\ t\,[s/x] \\ \mathrm{D}(\lambda x.t) \cdot s &\rightsquigarrow_\partial\ \lambda x.\left(\frac{\partial t}{\partial x}(s)\right) \end{aligned}$$

[1] While the one-step reduction rules for λ_ε may seem identical to those in the differential λ-calculus (compare [12, Section 3]), they are in fact not equivalent, as our notions of differential substitution and term equivalence differ substantially.

We write $\rightsquigarrow^+$ to denote the transitive closure of $\rightsquigarrow$, and $\rightsquigarrow^*$ to denote its transitive, reflexive closure.

The previous one-step reduction is defined as a relation from unrestricted terms to unrestricted terms, but it is not compatible with differential equivalence. That is to say, there may be differentially equivalent terms $t \sim_\varepsilon t'$ such that t' can be reduced but t cannot. For example, consider the term $(\lambda x.x + 0)\ 0$, which contains no β-redexes that can be reduced. This term is, however, equivalent to $(\lambda x.x)\ 0$, which clearly reduces to 0. Fortunately the canonical form of a term t gives us a representative of $\underline{t}$ which is "maximally reducible", that is to say, whenever any representative of $\underline{t}$ can be reduced to a representative of some $\underline{t'}$, then any canonical form $\mathbf{can}\,(t)$ for t can be reduced to a representative of the same $\underline{t'}$, possibly in zero reduction steps.

▶ **Theorem 27.** Reduction is compatible with canonicalization. That is to say, if $s \rightsquigarrow s'$, then $\mathbf{can}\,(s) \rightsquigarrow^* s''$ for some $s'' \sim_\varepsilon s'$.

This result then legitimises our proposed "existential" definition of reduction of well-formed terms, as it shows that, in order to reduce a given term, it suffices to reduce its canonical form. It also gets rid of the "reducing zero" problem, as canonical forms do not contain "spurious" representations of zero.

▶ **Definition 28.** Given well-formed terms $\underline{s}, \underline{s'}$, we say that $\underline{s}$ *reduces to* $\underline{s'}$ *in one step*, and write $\underline{s} \underline{\rightsquigarrow} \underline{t}$, whenever $\mathbf{can}\,(s) \rightsquigarrow s''$ and $s'' \sim_\varepsilon s'$, for some canonical form $\mathbf{can}\,(s)$ of $\underline{s}$.

▶ **Proposition 29.** Whenever $\underline{s} \underline{\rightsquigarrow} \underline{s'}$ then for any term $\underline{t}$ we have $\underline{s + t} \underline{\rightsquigarrow} \underline{s' + t}$.

If $t = t^*$ is an additive term, then additionally $\underline{s\ t^*} \underline{\rightsquigarrow}^+ \underline{s'\ t^*}$.

Furthermore, when $t = t^{\mathbf{b}}$ is a basic term (in particular $t^{\mathbf{b}}$ is not differentially equivalent to zero), we also have $\underline{\mathrm{D}(s) \cdot t} \underline{\rightsquigarrow}^+ \underline{\mathrm{D}(s') \cdot t}$.

Conversely, whenever s is not differentially equivalent to zero and $\underline{t} \underline{\rightsquigarrow} \underline{t'}$, then $\underline{s\ t} \underline{\rightsquigarrow}^+ \underline{s\ t'}$ and $\underline{\mathrm{D}(s) \cdot t} \underline{\rightsquigarrow}^+ \underline{\mathrm{D}(s) \cdot t'}$.

A proof of confluence for λ_ε is sketched in Appendix A This proof follows the standard Tait/Martin-Löf method by introducing a notion of parallel reduction on terms which is shown to have the diamond property although, due to the nature of our setting, the diamond property only holds up to differential equivalence.

▶ **Corollary 30.** The reduction relation $\underline{\rightsquigarrow}$ is confluent.

4.3 Encoding the Differential λ-Calculus

It is immediately clear, from simply inspecting the operational semantics for λ_ε, that it is closely related to the differential λ-calculus – indeed, every Cartesian differential category is a Cartesian difference category, and this connection should also be reflected in the syntax.

As it turns out, there is a clean translation that embeds λ_ε into the differential λ-calculus, which proceeds by deleting every term that contains an ε. The intuition behind this scheme should be apparent: every single differential substitution rule in λ_ε is identical to the corresponding case for the differential λ-calculus, once all the ε terms are cancelled out.

▶ **Definition 31.** Given an unrestricted λ_ε term t, its *ε-erasure* is the differential λ-term $\lceil t \rceil$ defined according to the rules in Figure 2.

▶ **Proposition 32.** The erasure $\lceil t \rceil$ is invariant under equivalence. That is to say, whenever $t \sim_\varepsilon t'$, it is the case that $\lceil t \rceil = \lceil t' \rceil$.

$$
\begin{array}{rcl@{\qquad}rcl}
\lceil x \rceil & := & x & \lceil \varepsilon t \rceil & := & 0 \\
\lceil 0 \rceil & := & 0 & \lceil s\ t \rceil & := & \lceil s \rceil\ \lceil t \rceil \\
\lceil s + t \rceil & := & \lceil s \rceil + \lceil t \rceil & \lceil \mathrm{D}(s) \cdot t \rceil & := & \mathrm{D}(\lceil s \rceil) \cdot \lceil t \rceil
\end{array}
$$

Figure 2 ε-erasure of a term t.

▶ **Proposition 33.** Erasure is compatible with standard and differential substitution. That is to say, for any terms s, t and a variable x, we have $\lceil s\,[t/x] \rceil = \lceil s \rceil\,[\lceil t \rceil/x]$ and $\left\lceil \frac{\partial s}{\partial x}\,(t) \right\rceil = \frac{\partial \lceil s \rceil}{\partial x}\,(\lceil t \rceil)$

▶ **Corollary 34.** Whenever $s \rightsquigarrow s'$, then $\lceil s \rceil \rightsquigarrow^* \lceil s' \rceil$.

These results form the syntactic obverse to the purely semantic correspondence between differential and difference categories [2, Proposition 1]: the former exhibits the differential λ-calculus as an instance of λ_ε where the ε operator is "degenerate", whereas the later shows that every Cartesian differential category can be understood as a "degenerate" Cartesian difference category, in the same sense that the corresponding infinitesimal extension is just the zero map.

5 Simple Types for λ_ε

Much like the differential λ-calculus, λ_ε can be endowed with a system of simple types, built from a set of basic types using the usual function type constructor.

▶ **Definition 35.** The set of *types* and *contexts* of the λ_ε-calculus is given by the following inductive definition:

$$
\begin{array}{rrcl}
\text{Types:} & \sigma, \tau & := & \mathbf{t} \mid \sigma \Rightarrow \tau \\
\text{Contexts:} & \Gamma & := & \emptyset \mid \Gamma, x : \tau
\end{array}
$$

assuming a countably infinite set of basic types $\mathbf{t}, \mathbf{s} \ldots$ is given.

The typing rules for the λ_ε-calculus are given in Figure 3 below, and should not be in the least surprising, as they are identical to the typing rules for the differential λ-calculus, with the addition of a typing rule for the infinitesimal extension of a term. As one would expect, our type system enjoys all the "usual" structural properties and their proofs follow by straightforward induction on the typing derivation. Note, however, that all of these typing rules operate on unrestricted terms, rather than on well-formed terms, for reasons that we will clarify later.

$$
\frac{}{\Gamma, x : \tau \vdash x : \tau}
\qquad
\frac{\Gamma \vdash s : \tau \Rightarrow \sigma \qquad \Gamma \vdash t : \tau}{\Gamma \vdash (s\ t) : \sigma}
\qquad
\frac{\Gamma, x : \tau \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \sigma \Rightarrow \tau}
$$

$$
\frac{}{\Gamma \vdash 0 : \tau}
\qquad
\frac{\Gamma \vdash s : \tau \qquad \Gamma \vdash t : \tau}{\Gamma \vdash s + t : \tau}
\qquad
\frac{\Gamma \vdash t : \tau}{\Gamma \vdash \varepsilon t : \tau}
$$

$$
\frac{\Gamma \vdash s : \tau \Rightarrow \sigma \qquad \Gamma \vdash t : \tau}{\Gamma \vdash \mathrm{D}(s) \cdot t : \tau \Rightarrow \sigma}
$$

Figure 3 Simple types for λ_ε.

One property that fails to hold is uniqueness of typings: indeed the term 0 admits any type, as do terms such as $0+0$ or $(\lambda x.0)\ y$. Typing judgements are nonetheless invertible. The following "standard" properties also hold, and can be proven by straightforward induction on the relevant typing derivation.

▶ **Proposition 36** (Weakening)**.** Whenever $\Gamma \vdash t : \tau$, then for any context Σ which is disjoint with Γ it is also the case that $\Gamma, \Sigma \vdash t : \tau$.

▶ **Proposition 37** (Substitution)**.** Whenever $\Gamma, x : \tau \vdash s : \sigma$ and $\Gamma \vdash t : \tau$, we have:

(i) $\Gamma \vdash s\,[t/x] : \sigma$
(ii) $\Gamma, x : \tau \vdash \frac{\partial s}{\partial x}\,(t) : \sigma$

▶ **Theorem 38** (Subject reduction)**.** Whenever $\Gamma \vdash t : \tau$ and $t \rightsquigarrow t'$ then $\Gamma \vdash t' : \tau$.

Since we have defined well-formed terms as equivalence classes of unrestricted terms, we might ask if typing is compatible with this equivalence relation. The answer is unfortunately no, that is to say, there are ill-typed terms that are differentially equivalent to well-typed terms. In particular, the term $(0\ t)$ is differentially equivalent to the term 0, but while the later is trivially well-typed, the former will not be typable for many choices of t (for example, whenever $t = (x\ x)$). A weaker version of this property does hold, however, that makes use of canonicity.

▶ **Proposition 39.** Whenever $\Gamma \vdash t : \tau$, then $\Gamma \vdash \mathbf{can}\,(t) : \tau$, and furthermore whenever $\Gamma \vdash \mathbf{can}\,(t) : \tau$ then every canonical form of t admits the same type.

Before stating a progress theorem for λ_ε, we must point out one small subtlety, as the definition of reduction of unrestricted terms depends on the particular representation chosen for the term. For example, the terms $((\lambda x.x) + 0)\ 0$ and $(\lambda x.x)\ 0$ are equivalent, but the first one contains no β-redexes, whereas the second one reduces to 0 in one step. We can prove that progress holds for canonical terms, however, as those are "maximally reducible".

▶ **Definition 40.** A canonical term T is a *canonical value* whenever it is of the form

$$T = \sum_{i=1}^{i} \varepsilon^{k_i}(\lambda x_i.t_i)$$

▶ **Theorem 41** (Progress)**.** Whenever a canonical term T admits a typing derivation $\vdash T : \tau$, then either T is a canonical value or there is some term t' with $T \rightsquigarrow t'$.

▶ **Definition 42.** We extend typing judgements to well-formed terms by setting $\Gamma \,\underline{\vdash}\, \underline{t} : \tau$ whenever $\Gamma \vdash \mathbf{can}\,(t) : \tau$.

▶ **Corollary 43** (Subject reduction for well-formed terms)**.** Whenever $\Gamma \,\underline{\vdash}\, \underline{t} : \tau$ and $\underline{t} \,\underline{\rightsquigarrow}\, \underline{t'}$, then $\Gamma \,\underline{\vdash}\, \underline{t'} : \tau$.

▶ **Corollary 44** (Progress for well-formed terms)**.** Whenever $\Gamma \,\underline{\vdash}\, \underline{t} : \tau$ then either $\underline{t} \,\underline{\rightsquigarrow}\, \underline{t'}$ or every canonical form $\mathbf{can}\,(\underline{t})$ is a canonical value.

Finally, strong normalisation can be shown by a proof similar to Ehrhard and Regnier's [12] and Vaux's [17], which themselves proceed by an argument similar to the standard reducibility candidates method. We defer the details to Appendix B.

▶ **Theorem 45** (Strong normalisation)**.** Whenever a closed well-formed term is typable with type $\underline{\vdash}\, \underline{t} : \tau$, it is strongly normalising.

6 Semantics

It is a well-known result that the simply-typed differential λ-calculus can be soundly interpreted in any differential λ-category, that is to say, any Cartesian differential category where differentiation "commutes with" abstraction (in the sense of [9, Definition 4.4]).

The exact same result holds for the difference λ-calculus and difference λ-categories. In what follows we will consider a fixed difference λ-category $\mathbf{C}$, and proceed to define interpretations for the types, contexts and terms of the simply-typed λ_ε-calculus.

▶ **Definition 46.** Given a $\mathbf{t}$-indexed family of objects $O_{\mathbf{t}}$, we define the interpretation $[\![\tau]\!]$ of a type τ by induction on its structure by setting $[\![\mathbf{t}]\!] := O_{\mathbf{t}}$, $[\![\sigma \Rightarrow \tau]\!] := [\![\sigma]\!] \Rightarrow [\![\tau]\!]$. We lift the interpretation of types to contexts in the usual way. Or, more formally, we have: $[\![\cdot]\!] := \mathbf{1}$, $[\![\Gamma, x : \tau]\!] := [\![\Gamma]\!] \times [\![\tau]\!]$.

As is the case in differential λ-categories, we can define a "differential substitution" operator on the semantic side. This operator is akin to post-composition with a partial derivative, and can be defined as follows.

▶ **Definition 47.** Given morphisms $s : A \times B \to C, u : A \to B$, we define their *differential composition* $s \star u : A \times B \to C$ by $s \star u := \partial [s] \circ \langle \mathbf{id}_{A\times B}, \langle 0_A, u \circ \pi_1 \rangle \rangle$

▶ **Definition 48.** Given a well-typed unrestricted λ_ε-term $\Gamma \vdash t : \tau$, we define its interpretation $[\![t]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$ inductively as in Figure 4 below. When Γ and τ are irrelevant or can be inferred from the context, we will simply write $[\![t]\!]$.

$$\begin{array}{rcll}
[\![(x_i : \tau_i)_{i=1}^n \vdash x_k : \tau_k]\!] & := & \pi_2 \circ \pi_1^{n-k} & : \prod_{i=1}^n [\![\tau_i]\!] \to [\![\tau_k]\!] \\
[\![\Gamma \vdash 0 : \tau]\!] & := & 0 & : [\![\Gamma]\!] \to [\![\tau]\!] \\
[\![\Gamma \vdash s + t : \tau]\!] & := & [\![s]\!] + [\![t]\!] & : [\![\Gamma]\!] \to [\![\tau]\!] \\
[\![\Gamma \vdash \varepsilon t : \tau]\!] & := & \varepsilon\, [\![t]\!] & : [\![\Gamma]\!] \to [\![\tau]\!] \\
[\![\Gamma \vdash \lambda x.t : \sigma \Rightarrow \tau]\!] & := & \Lambda\, [\![t]\!] & : [\![\Gamma]\!] \to [\![\sigma]\!] \Rightarrow [\![\tau]\!] \\
[\![\Gamma \vdash (s\ t) : \tau]\!] & := & \mathbf{ev} \circ \langle [\![s]\!], [\![t]\!] \rangle & : [\![\Gamma]\!] \to [\![\tau]\!] \\
[\![\Gamma \vdash \mathrm{D}(s) \cdot t : \sigma \Rightarrow \tau]\!] & := & \Lambda\, (\Lambda^-([\![s]\!]) \star [\![t]\!]) & : [\![\Gamma]\!] \to [\![\tau]\!]
\end{array}$$

Figure 4 Interpreting λ_ε in $\mathbf{C}$.

▶ **Theorem 49.** Whenever $s \sim_\varepsilon t$ are equivalent unrestricted terms that admit typing derivations $\Gamma \vdash s : \tau$ and $\Gamma \vdash t : \tau$, then their interpretations are identical, that is to say:

$$[\![\Gamma \vdash s : \tau]\!] = [\![\Gamma \vdash t : \tau]\!]$$

▶ **Definition 50.** Given well-formed terms $\underline{s}, \underline{s'}$, we define the equivalence relation $\sim_{\beta\partial}$ as the least contextual equivalence relation that contains the one-step reduction relation $\underset{\sim}{\rightsquigarrow}$.

▶ **Theorem 51.** The interpretation $[\![\cdot]\!]$ is *sound*, that is to say, whenever $\underline{s} \sim_{\beta\partial} \underline{s'}$ then $[\![s]\!] = [\![s']\!]$, independently of the choice of representatives s, s'.

Proof. Straightforward consequence of the results in Appendix C. ◀

▶ **Definition 52.** Recall that a simply-typed theory is a collection of equational judgements of the form $\Gamma \vdash s = t : \sigma$ where $\Gamma \vdash s : \sigma$ and $\Gamma \vdash t : \sigma$ are derivable. We say that a simply-typed theory is a *difference λ-theory* if it is closed under all rules in the system $\lambda_\varepsilon^\times\beta\eta\partial$ (comprising the contextual rules for all the constructs of the λ_ε-calculus augmented by products, $\sim_\varepsilon$ equivalence, and the surjective pairing, β, η and ∂ laws, and last being the equational version of $\rightsquigarrow_\partial$).

Given an interpretation $[\![\cdot]\!]_\mathcal{M}$ of λ_ε in $\mathbf{C}$, we say that $\mathcal{M} = [\![\cdot]\!]_\mathcal{M}$ is a *model* of a difference λ-theory $\mathscr{T}$ if for every typed equational judgement $\Gamma \vdash s = t : \sigma$ in $\mathscr{T}$, we have that $[\![\Gamma \vdash s : \sigma]\!]_\mathcal{M}$ and $[\![\Gamma \vdash t : \sigma]\!]_\mathcal{M}$ are the same morphism.

A *model homomorphism* $h : \mathcal{M} \to \mathcal{N}$ is given by isomorphisms $h_\mathbf{t} : [\![\mathbf{t}]\!]_\mathcal{M} \to [\![\mathbf{t}]\!]_\mathcal{N}$ for each basic type $\mathbf{t}$, and $h_{\sigma\times\tau} := h_\sigma \times h_\tau$, and

$$h_{\sigma\Rightarrow\tau} := h_\sigma^{-1} \Rightarrow h_\tau := \Lambda(h_\tau \circ \mathbf{ev} \circ (\mathbf{id} \times h_\sigma^{-1})).$$

We write $\mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{C})$ for the category whose objects are all models of difference λ-theory $\mathscr{T}$ in a difference λ-category $\mathbf{C}$, and whose morphisms are model homomorphisms.

▶ **Definition 53.** Let $\mathbf{C}$ and $\mathbf{D}$ be difference λ-categories. We say that a functor $F : \mathbf{C} \to \mathbf{D}$ is a *difference λ-functor* if F preserves the following:

- additive structure: $F(f+g) = F(f) + F(g)$, and $F(0) = 0$
- infinitesimal extension: $F(\varepsilon(f)) = \varepsilon(F(f))$
- products via the isomorphism $\Phi := \langle F(\pi_1), F(\pi_2)\rangle$
- exponentials via the isomorphism $\Psi := \Lambda(F(\mathbf{ev}) \circ \Phi)$
- difference combinator: $F(\partial[f]) = \partial[F(f)] \circ \Phi$.

We write $\mathsf{Dif}\lambda\text{-}\mathsf{Func}(\mathbf{C}, \mathbf{D})$ for the category of difference λ-functors $\mathbf{C} \to \mathbf{D}$ and natural isomorphisms.

▶ **Definition 54.** Given a difference λ-theory $\mathscr{T}$, we say that a category, denoted $\mathbf{Cl}(\mathscr{T})$, is *classifying* if there is a model of the theory in $\mathbf{Cl}(\mathscr{T})$, and this model is "generic", meaning that for every differential λ-category $\mathbf{D}$, there is a natural equivalence

$$\mathsf{Dif}\lambda\text{-}\mathsf{Func}(\mathbf{Cl}(\mathscr{T}), \mathbf{D}) \simeq \mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{D}). \tag{1}$$

The classifying category (unique up to isomomrphism) is the "smallest" in the sense that given a model of the theory $[\![\cdot]\!]_\mathbf{D}$ in a difference λ-category $\mathbf{D}$, there is a difference λ-functor $F : \mathbf{Cl}(\mathscr{T}) \to \mathbf{D}$ such that the interpretation $[\![\cdot]\!]_\mathbf{D}$ can be factored through the canonical interpretation in the classifying category, i.e., $[\![\cdot]\!]_\mathbf{D} = F \circ [\![\cdot]\!]_{\mathbf{Cl}(\mathscr{T})}$.

▶ **Conjecture 6.1** (Completeness). Every difference λ-theory $\mathscr{T}$ has a classifying difference λ-category $\mathbf{Cl}(\mathscr{T})$.

7 Future Work

We have defined here the difference λ-calculus, which generalises the differential λ-calculus in exactly the same manner as Cartesian difference categories generalise their differential counterpart. While this calculus is of theoretical interest, it lacks most practical features, such as iteration or conditionals, and it is not immediately obvious how to extend it with these. It is not clear, for example, precisely when iteration combinators are differentiable in the difference category sense.

The problem of iteration is closely related to integration, which is itself the focus of current work on the differential side [10, 15]. Indeed, consider a hypothetical extension of the difference λ-calculus equipped with a type of natural numbers (with the identity as its corresponding infinitesimal extension, that is to say, $\varepsilon_{\mathbb{N}} = \mathbf{id}_{\mathbb{N}}$). How should an iteration operator **iter** be defined? The straightforward option would be to give it the usual behavior, that is to say:

$$\begin{aligned} \mathbf{iter}\ \mathbf{Z}\ z\ s &\rightsquigarrow z \\ \mathbf{iter}\ (\mathbf{S}\ n)\ z\ s &\rightsquigarrow s\ (\mathbf{iter}\ n\ z\ s) \end{aligned}$$

These reduction rules entail that every object involved must be complete, that is to say, for every $s, t : A$, there is some $u : A$ with $s + \varepsilon(u) = t$ – such an element is given by the term $((\mathrm{D}(\lambda n.\mathbf{iter}\ n\ s\ (\lambda x.t)) \cdot (\mathbf{S}\ \mathbf{Z}))\ \mathbf{Z})$.

This would rule out a number of interesting models and so it seems unsatisfactory. An alternative is to define the iteration operator by:

$$\begin{aligned} \mathbf{iter}\ \mathbf{Z}\ z\ s &\rightsquigarrow z \\ \mathbf{iter}\ (\mathbf{S}\ n)\ z\ s &\rightsquigarrow (\mathbf{iter}\ n\ z\ s) + \varepsilon(s\ (\mathbf{iter}\ n\ z\ s)) \end{aligned}$$

Fixed z, s, and defining the map $\mu(n) := \mathbf{iter}\ n\ z\ s$, its derivative $\mathrm{D}[\mu](n, \mathbf{S}\ \mathbf{Z})$ is precisely $s(\mu(n))$. Or, in other words, the function $\mu : \mathbb{N} \to A$ is a "curve" which starts at z and whose derivative at a given point n is $s(\mu(n))$ – this boils down to stating that the curve μ is an integral curve for the vector field s satisfying the initial condition $\mu(\mathbf{Z}) = z$! Hence it may be possible to understand iteration as a discrete counterpart of the Picard-Lindelöf theorem, which states that such integral curves always exist (locally).

It would be of great interest to extend λ_ε with an interation operator and give its semantics in terms of differential (or difference) equations. Studying recurrence equations using the language of differential equations is a very useful tool in discrete analysis; for example, one can treat the recursive definition of the Fibonacci sequence as a discrete ODE and use differential equation methods to find a closed-form solution. We believe that in a language which frames iteration in such terms may be amenable to optimisation by similar analytic methods.

References

1 Mario Alvarez-Picallo. Change actions: from incremental computation to discrete derivatives, 2020. `arXiv:2002.05256`.

2 Mario Alvarez-Picallo and Jean-Simon P. Lemay. Cartesian difference categories. In *International Conference on Foundations of Software Science and Computation Structures*, page to appear. Springer, 2020.

3 Mario Alvarez-Picallo and Jean-Simon Pacaud Lemay. Cartesian difference categories: Extended report, 2020. `arXiv:2002.01091`.

4 Mario Alvarez-Picallo and C.-H. Luke Ong. Change actions: models of generalised differentiation. In *International Conference on Foundations of Software Science and Computation Structures*, pages 45–61. Springer, 2019.

5 Mario Alvarez-Picallo, Michael Peyton-Jones, Alexander Eyers-Taylor, and C.-H. Luke Ong. Fixing incremental computation. In *European Symposium on Programming*. Springer, 2019. in press.

6 P Arrighi and G Dowek. Linear-algebraic lambda-calculus: higher-order, encodings and confluence in A. Voronkov, Rewriting techniques and applications. *Lecture Notes in Computer Science, Springer-Verlag*, 2008.

7 Pablo Arrighi and Gilles Dowek. A computational definition of the notion of vectorial space. *Electronic Notes in Theoretical Computer Science*, 117:249–261, 2005.

8 Richard F. Blute, J. Robin B. Cockett, and Robert A. G. Seely. Cartesian differential categories. *Theory and Applications of Categories*, 22(23):622–672, 2009.

9 Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Categorical models for simply typed resource calculi. *Electronic Notes in Theoretical Computer Science*, 265:213–230, 2010.

10 J Robin B Cockett and J-S Lemay. Integral categories and calculus categories. *Mathematical Structures in Computer Science*, 29(2):243–308, 2019.

11 Thomas Ehrhard. An introduction to differential linear logic: proof-nets, models and antiderivatives. *Mathematical Structures in Computer Science*, 28(7):995–1060, 2018.

12 Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1-3):1–41, 2003. `doi:10.1016/S0304-3975(03)00392-X`.

13 Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 403(2):347–372, 2008.

14 René Lavendhomme. *Basic concepts of synthetic differential geometry*, volume 13. Springer Science & Business Media, 2013.

15 Jean-Simon Pacaud Lemay. Exponential functions in cartesian differential categories. *arXiv preprint arXiv:1911.04790*, 2019.

16 Giulio Manzonetto. What is a Categorical Model of the Differential and the Resource λ-Calculi? *Mathematical Structures in Computer Science*, 22(03):451–520, 2012.

17 Lionel Vaux. λ-calculus in an algebraic setting. *unpublished note*, 2006.

18 Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(05):1029–1059, 2009.

A Confluence

The following lemmas relate differential substitution and standard substitution, and will be of much use later.

▶ **Lemma 55.** Whenever x, y are (distinct) variables then for any unrestricted terms t, u, v where x is not free in v we have:

$$\left(\frac{\partial t}{\partial x}(u)\right)[v/y] = \frac{\partial t\,[v/y]}{\partial x}(u\,[v/y])$$

▶ **Lemma 56.** Whenever x, y are (distinct) variables, with y not free in either u, v, we have:

$$\frac{\partial t\,[v/y]}{\partial x}(u) \sim_\varepsilon \left(\frac{\partial t}{\partial x}(u)\right)[(v\,[x+\varepsilon u/x])/y] + \left(\frac{\partial t}{\partial y}\left(\frac{\partial v}{\partial x}(u)\right)\right)[v/y]$$

▶ **Definition 57.** The *parallel reduction* relation between (unrestricted) terms is defined according to the deduction rules in Figure 5.

$$(\rightrightarrows_x)\ \frac{}{x \rightrightarrows x} \qquad (\rightrightarrows_0)\ \frac{}{0 \rightrightarrows 0} \qquad (\rightrightarrows_\lambda)\ \frac{t \rightrightarrows t'}{\lambda x.t \rightrightarrows \lambda x.t'}$$

$$(\rightrightarrows_\varepsilon)\ \frac{t \rightrightarrows t'}{\varepsilon t \rightrightarrows \varepsilon t'} \qquad (\rightrightarrows_+)\ \frac{s \rightrightarrows s' \qquad t \rightrightarrows t'}{s + t \rightrightarrows s' + t'}$$

$$(\rightrightarrows_{\mathbf{ap}})\ \frac{s \rightrightarrows s' \qquad t \rightrightarrows t'}{s\ t \rightrightarrows s'\ t'} \qquad (\rightrightarrows_{\mathrm{D}})\ \frac{s \rightrightarrows s' \qquad t \rightrightarrows t'}{\mathrm{D}(s)\cdot t \rightrightarrows \mathrm{D}(s')\cdot t'}$$

$$(\rightrightarrows_\beta)\ \frac{s \rightrightarrows \lambda x.s' \qquad t \rightrightarrows t'}{s\ t \rightrightarrows s'\,[t'/x]} \qquad (\rightrightarrows_\partial)\ \frac{s \rightrightarrows \lambda x.s' \qquad t \rightrightarrows t'}{\mathrm{D}(s)\cdot t \rightrightarrows \lambda x.\frac{\partial s'}{\partial x}(t')}$$

Figure 5 Parallel reduction rules for λ_ε.

The parallel reduction relation can be extended to well-formed terms by setting $\underline{t} \,\underline{\rightrightarrows}\, \underline{t'}$ whenever $\mathbf{can}\,(t) \rightrightarrows t''$ with $t'' \sim_\varepsilon t'$ for some canonical form of $\underline{t}$.

▶ Remark 58. Our definition of parallel reduction differs slightly from the usual in the rule $(\rightrightarrows_\beta)$, which allows reducing a newly-formed λ-abstraction. This is necessary because our calculus contains terms of the shape $(\mathrm{D}(\lambda x.s) \cdot u)\ t$, which we need to parallel reduce in a single step to $\left(\frac{\partial s}{\partial x}(u)\right)[t/x]$. The original presentation of the differential λ-calculus opted instead for adding an extra parallel reduction rule to allow for the case of reducing an abstraction under a differential application. Similarly, our rule $(\rightrightarrows_\partial)$ allows reducing terms of the form $\mathrm{D}(\mathrm{D}(\lambda x.s) \cdot u) \cdot v$ in a single step.

One convenient property of the parallel reduction relation lies in its relation to canonical forms. As we saw in Theorem 27, canonical forms are "maximally reducible", but don't respect the number of reduction steps. This is no longer the case for parallel reduction: the process of canonicalization only duplicates regexes "in parallel" (that is, by copying them onto multiple separate summands) or in a "parallelizable series" (i.e. a differential application may be regularized into a term of the form $\mathrm{D}(\mathrm{D}(\ldots) \cdot u) \cdot v$, which can be entirely reduced in a single parallel reduction step).

▶ **Theorem 59.** Whenever $s \rightrightarrows s'$, then $\mathbf{can}\,(s) \rightrightarrows s''$ for some $s'' \sim_\varepsilon s'$.

We also state the following standard properties of parallel reduction, all of which can be proven by straightforward induction on the term.

▶ **Lemma 60.** Parallel reduction sits between one-step and many-step reduction. That is to say: $\rightsquigarrow\ \subseteq\ \rightrightarrows\ \subseteq\ \rightsquigarrow^*$, and furthermore $\underline{\rightsquigarrow}\ \subseteq\ \underline{\rightrightarrows}\ \subseteq\ \underline{\rightsquigarrow}^*$.

▶ **Lemma 61.** The parallel reduction relation is contextual. In particular, every term parallel-reduces to itself.

▶ **Lemma 62.** Parallel reduction cannot introduce free variables. That is to say: whenever $t \rightrightarrows t'$, we have $\mathrm{FV}(t') \subseteq \mathrm{FV}(t)$.

▶ **Lemma 63.** Whenever $\lambda x.t \rightrightarrows u$, it must be the case that $u = \lambda x.t'$ and $t \rightrightarrows t'$.

▶ **Lemma 64.** Whenever $s \rightrightarrows s'$ and $t \rightrightarrows t'$ then $s\,[t/x] \rightrightarrows s'\,[t'/x]$, and furthermore there is some w with $\frac{\partial s}{\partial x}(t) \rightrightarrows w \sim_\varepsilon \frac{\partial s'}{\partial x}(t')$.

We first prove that parallel reduction has the diamond property when applied to canonical terms, taking care that it holds up to differential equivalence (note that, much like one-step reduction, the result of parallel-reducing a canonical term need not be canonical). For this, we introduce the usual notion of a full parallel reduct of a term.

▶ **Definition 65.** Given an unrestricted term t, its *full parallel reduct* $t_\downarrow$ is defined inductively by:

$$\begin{array}{rcl@{\qquad}rcl}
x_\downarrow & := & x & (\lambda x.t)_\downarrow & := & \lambda x.(t_\downarrow) \\
(\varepsilon t)_\downarrow & := & \varepsilon(t_\downarrow) & (s\ t)_\downarrow & := & \begin{cases} e\,[t_\downarrow/x] & \text{if } s_\downarrow = \lambda x.e \\ (s_\downarrow)\ (t_\downarrow) & \text{otherwise} \end{cases} \\
(s+t)_\downarrow & := & (s_\downarrow) + (t_\downarrow) & & & \\
0_\downarrow & := & 0 & (\mathrm{D}(s) \cdot t)_\downarrow & := & \begin{cases} \lambda x.\frac{\partial e}{\partial x}(t_\downarrow) & \text{if } s_\downarrow = \lambda x.e \\ D(s_\downarrow) \cdot (t_\downarrow) & \text{otherwise} \end{cases}
\end{array}$$

▶ **Lemma 66.** Whenever $s \rightrightarrows \lambda x.v$, then $s_\downarrow$ is of the form $\lambda x.w$, for some term w.

▶ **Theorem 67.** For any unrestricted terms s, s' such that $s \rightsquigarrow s'$, there is an unrestricted term w such that $s' \rightsquigarrow w$ and $w \sim_\varepsilon s_\downarrow$.

▶ **Corollary 68.** Parallel reduction has the diamond property up to differential equivalence. That is to say, for any unrestricted term t and terms t_1, t_2 such that $t \rightsquigarrow t_1$ and $t \rightsquigarrow t_2$, there are terms u, v making the following diagram commute:

$$\begin{tikzcd}
t_1 \arrow[d, squiggly] & T \arrow[l, squiggly] \arrow[r, squiggly] & t_2 \arrow[d, squiggly] \\
u \arrow[rr, phantom, "\sim_\varepsilon"] & & v
\end{tikzcd}$$

▶ **Lemma 69.** Given unrestricted terms $s \sim_+ s'$ which are permutatively equivalent, that is, which differ only up to a reordering of their additions and differential applications, their full parallel reducts are differentially equivalent.

▶ **Theorem 70.** The reduction relation $\underline{\rightsquigarrow}$ has the diamond property. That is, whenever $\underline{s} \underline{\rightsquigarrow} \underline{u}$ and $\underline{s} \underline{\rightsquigarrow} \underline{v}$ there is a term $\underline{c}$ such that $\underline{u} \underline{\rightsquigarrow} \underline{c}$ and $\underline{v} \underline{\rightsquigarrow} \underline{c}$.

Proof. Consider a well-formed term $\underline{s}$, and suppose that $\underline{s} \underline{\rightsquigarrow} \underline{u}$ and $\underline{s} \underline{\rightsquigarrow} \underline{v}$. In particular, this means there are two canonical forms $\mathbf{can}(s)_1, \mathbf{can}(s)_2$ of s such that $\mathbf{can}(s)_1 \rightsquigarrow u$ and $\mathbf{can}(s)_2 \rightsquigarrow v$. These canonical forms $\mathbf{can}(s)_1, \mathbf{can}(s)_2$ are equivalent up to permutative equivalence, and so their full parallel reducts are differentially equivalent as per Lemma 69. Denote their $\sim_\varepsilon$-equivalence class by $\underline{c}$. Therefore since $\mathbf{can}(s)_1 \rightsquigarrow \mathbf{can}(s)_{1\downarrow} \sim_\varepsilon c$ and $\mathbf{can}(s)_2 \rightsquigarrow \mathbf{can}(s)_{2\downarrow} \sim_\varepsilon c$ it follows that $\underline{u} \underline{\rightsquigarrow} \underline{c}$ and $\underline{v} \underline{\rightsquigarrow} \underline{c}$. ◀

B Strong Normalisation

With our typing rules in place, we set out to show that λ_ε is strongly normalising. Our proof follows the structure of Ehrhard and Regnier's [12] and Vaux's[17], which use an adaptation of the well-known argument by reducibility candidates. Our proof will be somewhat simpler, however, due to two main reasons: first, we are not concerning ourselves with terms with coefficients on some general rig; and second, we have defined unrestricted and canonical terms as inductive types, and so we can freely use induction on the syntax of our terms. We will need some auxiliary results, which we prove now.

▶ **Lemma 71.** A term $\underline{s + t}$ is strongly normalising if and only if $\underline{s}, \underline{t}$ are strongly normalising. A term $\underline{\varepsilon s}$ is strongly normalising if and only if $\underline{s}$ is.

▶ **Definition 72.** For every type τ we define inductively a set $\mathcal{R}_\tau$ of well-formed terms of type τ.

- Whenever $\tau = \mathbf{t}$ is a primitive type, $\underline{s} \in \mathcal{R}_\mathbf{t}$ if and only if $\underline{s}$ is strongly normalising.
- Whenever $\tau = \sigma_1 \Rightarrow \sigma_2$, $\underline{s} \in \mathcal{R}_{\sigma_1 \Rightarrow \sigma_2}$ if and only if for any additive term $\underline{t^*} \in \mathcal{R}_{\sigma_1}$ and for any sequence $\underline{v_1^\mathbf{b}}, \ldots, \underline{v_n^\mathbf{b}}$ of basic terms $\underline{v_i^\mathbf{b}} \in \mathcal{R}_{\sigma_1}$ of length $n \geq 0$ we have $\underline{\left(\mathrm{D}^n(s) \cdot (v_1^\mathbf{b}, \ldots, v_i^\mathbf{b})\right)\ t^*} \in \mathcal{R}_{\sigma_2}$

  If $\underline{t} \in \mathcal{R}_\tau$ we will often just say that $\underline{t}$ is *reducible* if the choice of τ is clear from the context.

▶ **Lemma 73.** Whenever $\underline{t} \in \mathcal{R}_\tau$, then for any two distinct variables x, y the renaming $\underline{t\,[y/x]}$ is also in $\mathcal{R}_\tau$.

▶ **Lemma 74.** Whenever $\underline{t} \in \mathcal{R}_\tau$, then $\underline{t}$ is strongly normalising.

▶ **Lemma 75.** Whenever $\underline{s}, \underline{t} \in \mathcal{R}_\tau$, then both $\underline{s+t}, \underline{\varepsilon s}$ are in $\mathcal{R}_\tau$. Conversely, whenever $\underline{s+t}$ is in $\mathcal{R}_\tau$ then so are $\underline{s}, \underline{t}$.

▶ **Lemma 76.** Whenever $\underline{s} \in \mathcal{R}_{\sigma \Rightarrow \tau}$ and $\underline{t} \in \mathcal{R}_\sigma$ then $\underline{\mathrm{D}(s) \cdot t} \in \mathcal{R}_{\sigma \Rightarrow \tau}$.

▶ **Corollary 77.** A well-formed term $\underline{t}$ is in $\mathcal{R}_\tau$ if and only if some canonical form $T = \mathbf{can}\,(\underline{t})$ is of the form $\sum_{i=1}^n \varepsilon^{k_i} t_i^{\mathbf{b}}$ with $\underline{t_i^{\mathbf{b}}} \in \mathcal{R}_\tau$ for each $1 \leq i \leq n$.

▶ **Lemma 78.** Whenever $\underline{t} \in \mathcal{R}_\tau, \underline{t} \rightsquigarrow^+ \underline{t'}$, then $\underline{t'} \in \mathcal{R}_\tau$.

▶ **Definition 79.** A basic term $t^{\mathbf{b}}$ is *neutral* whenever it is not a λ-abstraction. In other words, a basic term is neutral whenever it is of the form $x, (s\ t)$ or $\mathrm{D}(s) \cdot u$. A canonical term T is neutral whenever it is of the form $\sum_{i=1}^n \varepsilon^{k_i} s_i^{\mathbf{b}}$, where each of the $s_i^{\mathbf{b}}$ are neutral. In particular, 0 is a neutral term. A well-formed term $\underline{t}$ is neutral whenever some (equivalently, all) canonical form is neutral.

▶ **Lemma 80.** Whenever $\underline{t}$ is neutral and every $\underline{t'}$ such that $\underline{t} \rightsquigarrow^+ \underline{t'}$ is in $\mathcal{R}_\tau$, then so is $\underline{t}$.

▶ **Lemma 81.** If, for all $\underline{t^*} \in \mathcal{R}_{\sigma_1}$ where x does not appear free, the term $\underline{s\,[t^*/x]}$ is in $\mathcal{R}_{\sigma_2}$ and, for all $\underline{u^{\mathbf{b}}}$ where x does not appear free, the term $\underline{\left(\frac{\partial s}{\partial x}\left(u^{\mathbf{b}}\right)\right)[t^*/x]}$ is in $\mathcal{R}_{\sigma_2}$, then the term $\underline{\lambda x.s}$ is in $\mathcal{R}_{\sigma_1 \Rightarrow \sigma_2}$.

▶ **Theorem 82.** Consider a well-formed term $\underline{t}$ which admits a typing of the form $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash \underline{t} : \tau$ and assume given the following data:

- A sequence of basic terms $\underline{d_1^{\mathbf{b}}} \in \mathcal{R}_{\sigma_1}, \ldots, \underline{d_n^{\mathbf{b}}} \in \mathcal{R}_{\sigma_n}$.
- An arbitrary sequence of indices $i_1, \ldots, i_k \in \{1, \ldots, n\}$ (possibly with repetitions).
- A sequence of additive terms $\underline{s_1^*} \in \mathcal{R}_{\sigma_{i_1}}, \ldots, \underline{s_k^*} \in \mathcal{R}_{\sigma_{i_k}}$.

such that none of the variables $x_1, \ldots, x_i$ appear free in the $d_i^{\mathbf{b}}, s_i^*$. Then the term

$$\underline{t'} = \underline{\left(\frac{\partial^k t}{\partial(x_{i_1}, \ldots, x_{i_k})}(d_1^{\mathbf{b}}, \ldots, d_k^{\mathbf{b}})\right)[s_1^*, \ldots, s_n^*/x_1, \ldots, x_n]}$$

is in $\mathcal{R}_\tau$.

C Soundness

The following result corresponds to well-known properties of differential λ-categories (see e.g. [9, Lemma 4.8]), the proof being identical to the differential case (unlike some other lemmas in that work which hinge on derivatives being additive).

▶ **Lemma 83.** Let $f : A \to B, g : A \to C, h : (A \times B) \times E \to F$ be arbitrary $\mathbf{C}$-morphisms. Then the following properties hold:

i. $\partial\,[\mathbf{sw}] = \mathbf{sw} \circ \pi_2$
ii. $(g \circ \pi_1) \star f = 0$
iii. $\Lambda(h) \star f = \Lambda(((h \circ \mathbf{sw}) \star (f \circ \pi_1)) \circ \mathbf{sw})$
iv. $\Lambda^-\,(\Lambda(h) \star f) = ((h \circ \mathbf{sw}) \star (f \circ \pi_1)) \circ \mathbf{sw}$

The results below also have rough analogues in the theory of Cartesian differential categories, but the correspondence starts growing a bit more distant as any result that hinges on derivatives being additive will, in general, only hold up to some second-order term in the theory of difference categories.

▶ **Lemma 84.** Let $f : A \times B \times C \to D, g : A \to B, g' : A \times B \to B, e : A \times B \to C$ be arbitrary $\mathbf{C}$-morphisms. Then the following identities hold:

i. $(\mathbf{ev} \circ \langle \Lambda(f), e\rangle) \star g = \mathbf{ev} \circ \langle \Lambda(f \star (e \star g)), e\rangle + \mathbf{ev} \circ \langle \Lambda(f) \star g, e \circ \langle \pi_1, \pi_2 + \varepsilon(g) \circ \pi_1\rangle\rangle$
ii. $\Lambda(f \star e) \star g = \Lambda\Big[\Lambda^-(\Lambda(f) \star g) \star (e \circ (\mathbf{id} + \langle 0, \varepsilon(g)\rangle))) + \varepsilon(f \star e) \star (e \star g) + (f \star (e \star g))\Big]$
iii. $\Lambda(f \star e) \circ \langle \pi_1, g'\rangle = \Lambda(\Lambda^-(\Lambda(f) \circ \langle \pi_1, g'\rangle) \star (e \circ \langle \pi_1, g'\rangle))$

▶ **Lemma 85.** Let t be some unrestricted λ_ε-term. The following properties hold:

i. If $\Gamma \vdash t : \tau$ and x does not appear in Γ then $[\![\Gamma, x : \sigma \vdash t : \tau]\!] = [\![\Gamma \vdash t : \tau]\!] \circ \pi_1$
ii. If $\Gamma, x : \sigma_1, y : \sigma_2 \vdash t : \tau$ then $[\![\Gamma, y : \sigma_2, x : \sigma_1 \vdash t : \tau]\!] = [\![\Gamma, x : \sigma_1, y : \sigma_2 \vdash t : \tau]\!] \circ \mathbf{sw}$

The morphism $\mathbf{sw}$ above is the obvious isomorphism between $(A \times B) \times C$ and $(A \times C) \times B$, which we can define explicitly by $\mathbf{sw} \coloneqq \langle\langle \pi_{11}, \pi_2\rangle, \pi_{21}\rangle : (A \times B) \times C \to (A \times C) \times B$

▶ **Lemma 86.** Let $\Gamma, x : \tau \vdash s : \sigma$, with s some unrestricted λ_ε-term. Then:

i. Whenever $\Gamma, x : \tau \vdash t : \tau$, then $[\![s\,[t/x]]\!]_\Gamma = [\![s]\!]_{\Gamma, x:\tau} \circ \left\langle \pi_1, [\![t]\!]_{\Gamma, x:\tau} \right\rangle$
ii. Whenever $\Gamma \vdash t : \tau$, then $\left[\!\!\left[\frac{\partial s}{\partial x}(t)\right]\!\!\right]_{\Gamma, x:\tau} = \partial\left[[\![s]\!]_{\Gamma, x:\tau}\right] \circ \langle \mathbf{id}, \langle 0, [\![t]\!]_\Gamma \circ \pi_1\rangle\rangle$. Or, using the notation in Definition 47, $\left[\!\!\left[\frac{\partial s}{\partial x}(t)\right]\!\!\right] = [\![s]\!] \star [\![t]\!]$.

D Completeness

We set up the equivalence (1) in the forward direction via modelling functors. Using the preceding notations, let $\mathcal{M} = [\![\cdot]\!]_\mathcal{M}$ be a model of a difference λ-theory $\mathscr{T}$ in $\mathbf{C}$, we define a family of *modelling functors* $\mathsf{mod}_\mathcal{M} : \mathsf{Dif}\lambda\text{-}\mathsf{Func}(\mathbf{C}, \mathbf{D}) \to \mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{D})$ by $[\![\mathbf{t}]\!]_{\mathsf{mod}_\mathcal{M} F} \coloneqq F([\![\mathbf{t}]\!]_\mathcal{M})$ where $F : \mathbf{C} \to \mathbf{D}$ is a difference λ-functor; and for any natural isomorphism $\phi : F \to G$, $\mathsf{mod}_\mathcal{M}\,\phi : \mathsf{mod}_\mathcal{M} F \to \mathsf{mod}_\mathcal{M} G$ is a model homomorphism where $(\mathsf{mod}_\mathcal{M}\,\phi)_\mathbf{t} \coloneqq \phi_{[\![\mathbf{t}]\!]_\mathcal{M}}$.

We first build a syntactic difference λ-category using the difference λ-theory $\mathscr{T}$, and then prove that it is classifying by presenting the "inverse" for the functor

$$\mathsf{mod}_\mathcal{G} : \mathsf{Dif}\lambda\text{-}\mathsf{Func}(\mathbf{Cl}(\mathscr{T}), \mathbf{D}) \to \mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{D}),$$

writing $\mathcal{G} = [\![\cdot]\!]_\mathcal{G}$ for the canonical "generic" interpretation in $\mathbf{Cl}(\mathscr{T})$.

We build the difference λ-category, $\mathbf{Cl}(\mathscr{T})$, using a standard construction. Objects are types of $\mathscr{T}$, and morphisms $\boldsymbol{f} : \sigma \to \tau$ are equivalence classes of term-in-context judgements $[x : \sigma \vdash M : \tau]$, where two terms are equivalent if they are provably equal in $\mathscr{T}$. It remains to show that $\mathbf{Cl}(\mathscr{T})$ is classifying by exhibiting the "inverse" of the modelling functors.

It is straightforward to see that the canonical interpretation $\mathcal{G}$, that sets $[\![\mathbf{t}]\!]_\mathcal{G} \coloneqq \mathbf{t}$, is a model of $\mathscr{T}$ in $\mathbf{Cl}(\mathscr{T})$. Take a difference λ-category $\mathbf{D}$. We define

$$\mathsf{mod}_\mathcal{G}^{-1} : \mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{D}) \to \mathsf{Dif}\lambda\text{-}\mathsf{Func}(\mathbf{Cl}(\mathscr{T}), \mathbf{D})$$

as follows. Given a model $\mathcal{M}$ of $\mathscr{T}$ in $\mathbf{D}$, the functor $\mathsf{mod}_\mathcal{G}^{-1}\,\mathcal{M} : \mathbf{Cl}(\mathscr{T}) \to \mathbf{D}$ is defined by

$$\begin{aligned} \sigma &\mapsto [\![\sigma]\!]_\mathcal{M} \\ [x : \sigma \vdash M : \tau] &\mapsto [\![x : \sigma \vdash M : \tau]\!]_\mathcal{M} : [\![\sigma]\!]_\mathcal{M} \to [\![\tau]\!]_\mathcal{M} \end{aligned}$$

Soundness of the interpretations tells us that $\mathsf{mod}_\mathcal{G}^{-1}\,\mathcal{M}$ is functorial. Note that Φ and Ψ are both identity functors, and so, $\mathsf{mod}_\mathcal{G}^{-1}\,\mathcal{M}$ preserves products and exponentials. It remains to check that $\mathsf{mod}_\mathcal{G}^{-1}\,\mathcal{M}$ is a difference λ-functor.

Given a model homomorphism $h : \mathcal{M} \to \mathcal{N}$ in $\mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T}, \mathbf{D})$, we define the natural isomorphism $\mathsf{mod}_{\mathcal{G}}^{-1}\, h : \mathsf{mod}_{\mathcal{G}}^{-1}\, \mathcal{M} \to \mathsf{mod}_{\mathcal{G}}^{-1}\, \mathcal{N}$ by setting

$$(\mathsf{mod}_{\mathcal{G}}^{-1}\, h)_\sigma \coloneqq h_{\mathbf{t}} : [\![\sigma]\!]_{\mathcal{M}} \to [\![\sigma]\!]_{\mathcal{N}} .$$

We can easily check that $(\mathsf{mod}_{\mathcal{G}}^{-1}\, h)_\sigma$ is a natural transformation by induction on the length of the derivation of the $\lambda_\varepsilon^\times$-term $\Gamma \vdash M : \tau$. Since h_σ is an isomorphism for any type σ, $\mathsf{mod}_{\mathcal{G}}^{-1}$ is a natural isomorphism.

We check that $(\mathsf{mod}_{\mathcal{G}}^{-1}\, h)_\sigma$ is a natural transformation by induction on the length of the derivation of the $\lambda_\varepsilon^\times$-term-in-context, $\Gamma \vdash M : \tau$, that the following diagram commutes:

$$[\![\Gamma \vdash M : \sigma]\!]_{\mathcal{N}} \circ h_\Gamma = h_\tau \circ [\![\Gamma \vdash M : \sigma]\!]_{\mathcal{M}} .$$

Lastly we check that $\mathsf{mod}_{\mathcal{G}}$ and $\mathsf{mod}_{\mathcal{G}}^{-1}$ define an equivalence via the natural isomorphisms

$$\mu : \mathsf{mod}_{\mathcal{G}} \circ \mathsf{mod}_{\mathcal{G}}^{-1} \simeq \mathbf{id}_{\mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T},\mathbf{D})}$$
$$\nu : \mathbf{id}_{\mathsf{Mod}_{\mathsf{Dif}\lambda}(\mathscr{T},\mathbf{D})} \simeq \mathsf{mod}_{\mathcal{G}}^{-1} \circ \mathsf{mod}_{\mathcal{G}}$$

defined as follows. For any model $\mathcal{M}$ of $\mathscr{T}$ in $\mathbf{D}$, $\mu_{\mathcal{M}} : \mathsf{mod}_{\mathcal{G}}(\mathsf{mod}_{\mathcal{G}}^{-1}\, \mathcal{M}) \to \mathcal{M}$ is defined by

$$(\mu_{\mathcal{M}})_{\mathbf{t}} \coloneqq \mathbf{id}_{[\![\mathbf{t}]\!]_{\mathcal{M}}} : [\![\mathbf{t}]\!]_{\mathsf{mod}_{\mathcal{G}}(\mathsf{mod}_{\mathcal{G}}^{-1}\, \mathcal{M})} = [\![\mathbf{t}]\!]_{\mathcal{M}} \to [\![\mathbf{t}]\!]_{\mathcal{M}}$$

and for any difference λ-functor $F : \mathbf{Cl}(\mathscr{T}) \to \mathbf{D}$, we define

$$(\nu_F)_\sigma \coloneqq \mathbf{id}_{F\sigma} : F\,\sigma \to (\mathsf{mod}_{\mathcal{G}}^{-1}(\mathsf{mod}_{\mathcal{G}}\, F))\sigma = F([\![\sigma]\!]_{\mathcal{G}}) = F\,\sigma.$$

Obviously μ and ν are natural isomorphisms. Thus $\mathbf{Cl}(\mathscr{T})$ is a classifying category with the model $\mathcal{G}$.