



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Differential logical relations, Part I: The simply-typed case

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Dal Lago U., Gavazzo F., Yoshimizu A. (2019). Differential logical relations, Part I: The simply-typed case. Schloss Dagstuhl- Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing [10.4230/LIPIcs.ICALP.2019.111].

Availability:

This version is available at: <https://hdl.handle.net/11585/704238> since: 2019-10-31

Published:

DOI: <http://doi.org/10.4230/LIPIcs.ICALP.2019.111>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

Differential Logical Relations

Part I: The Simply-Typed Case

Ugo Dal Lago

University of Bologna, Italy

INRIA Sophia Antipolis, France

ugo.dallago@unibo.it

Francesco Gavazzo

IMDEA Software, Spain

francesco.gavazzo@gmail.com

Akira Yoshimizu

INRIA Sophia Antipolis, France

akiray@bp.iij4u.or.jp

Abstract

We introduce a new form of logical relation which, in the spirit of metric relations, allows us to assign each pair of programs a quantity measuring their distance, rather than a boolean value standing for their being equivalent. The novelty of differential logical relations consists in measuring the distance between terms not (necessarily) by a numerical value, but by a mathematical object which somehow reflects the interactive complexity, i.e. the type, of the compared terms. We exemplify this concept in the simply-typed lambda-calculus, and show a form of soundness theorem. We also see how ordinary logical relations and metric relations can be seen as instances of differential logical relations. Finally, we show that differential logical relations can be organised in a cartesian closed category, contrarily to metric relations, which are well-known *not* to have such a structure, but only that of a monoidal closed category.

2012 ACM Subject Classification Computability → Lambda calculus; Logic → Equational Logic and Rewriting

Keywords and phrases Logical Relations λ -Calculus Program Equivalence Semantics.

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.XXX

Related Version <http://arxiv.org/abs/1904.12137>

Funding The authors are partially supported by the ERC Consolidator Grant DLV-818616 DIAPASoN, as well as by the ANR projects 14CE250005 ELICA and 16CE250011 REPAS.

1 Introduction

Modern software systems tend to be heterogeneous and complex, and this is reflected in the analysis methodologies we use to tame their complexity. Indeed, in many cases the only way to go is to make use of compositional kinds of analysis, in which *parts* of a large system can be analysed in isolation, without having to care about the rest of the system, the *environment*. As an example, one could consider a component A and replace it with another (e.g. more efficient) component B without looking at the context C in which A and B are supposed to operate, see Figure 1. Of course, for this program transformation to be safe, A should be *equivalent* to B or, at least, B should be a *refinement* of A .

Program equivalences and refinements, indeed, are the cruxes of program semantics, and have been investigated in many different programming paradigms. When programs have an interactive behaviour, like in concurrent or higher-order languages, even *defining* a notion of program equivalence is not trivial, while coming out with handy methodologies for *proving* concrete programs to be equivalent can be quite challenging, and has been one of the major



© Ugo Dal Lago and Francesco Gavazzo and Akira Yoshimizu;
licensed under Creative Commons License CC-BY

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).

Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. XXX; pp. XXX:1–XXX:14



Leibniz International Proceedings in Informatics

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



XXX:2 Differential Logical Relations

45 research topics in programming language theory, stimulating the development of techniques
 46 like logical relations [23, 20], applicative bisimilarity [1], and to some extent denotational
 47 semantics [26, 27] itself.

48 Coming back to our example, may we say anything about the case in
 49 which A and B are *not* equivalent, although behaving very similarly? Is
 50 there anything classic program semantics can say about this situation?
 51 Actually, the answer is negative: the program transformation turning
 52 such an A into B cannot be justified, simply because there is no
 53 guarantee about what the possible negative effects that turning A into
 54 B could have on the overall system formed by C and A . There are,
 55 however, many cases in which program transformations like the one
 56 we just described are indeed of interest, and thus desirable. Many
 57 examples can be, for instance, drawn from the field of *approximate*
 58 *computing* [21], in which equivalence-breaking program transformations
 59 are considered as beneficial *provided* the overall behaviour of the
 60 program is not affected too much by the transformation, while its
 61 intensional behaviour, e.g. its performance, is significantly improved.

One partial solution to the problem above consists in considering
 program *metrics* rather than program *equivalences*. This way, any
 pair of programs are dubbed being at a certain numerical distance rather than being
 merely equivalent (or not). This, for example, can be useful in the context of differential
 privacy [24, 6, 32] and has also been studied in the realms of domain theory [13, 5, 14, 16, 4]
 (see also [28] for an introduction to the subject) and coinduction [30, 29, 15, 9]. The common
 denominator among all these approaches is that on the one hand, the notion of a congruence,
 crucial for compositional reasoning, is replaced by the one of a *Lipschitz-continuous* map:
 any context should not amplify (too much) the distance between any pair of terms, when it
 is fed with either the former or the latter:

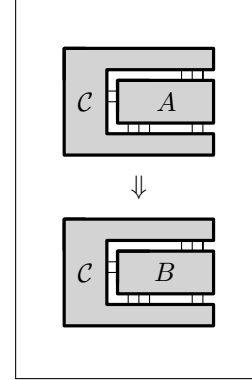
$$\delta(C[M], C[N]) \leq c \cdot \delta(M, N).$$

62 This enforces compositionality, and naturally leads us to consider metric spaces and Lipschitz
 63 functions as the underlying category. As is well known, this is not a cartesian closed category,
 64 and thus does *not* form a model of typed λ -calculi, unless one adopts linear type systems, or
 65 type systems in which the number of uses of each variable is kept track of, like FUZZ [24].
 66 This somehow limits the compositionality of the metric approach [13, 17].

Even if one considers affine calculi, there are program transformations which are intrinsically
 unjustifiable in the metric approach. Consider the following two programs of type
 $REAL \rightarrow REAL$

$$M_{SIN} := \lambda x. \mathbf{sin}(x) \qquad M_{ID} := \lambda x. x.$$

67 The two terms compute two very different functions on the real numbers, namely the sine
 68 trigonometric function and the identity on \mathbb{R} , respectively. The euclidean distance $|\sin x - x|$
 69 is unbounded when x ranges over \mathbb{R} . As a consequence, comparing M_{SIN} and M_{ID} using the
 70 so-called sup metric¹ as it is usually done in metric logical relations [24, 13] and applicative
 71 distances [17, 10], we see that their distance is infinite, and that the program transformation
 72 turning M_{SIN} into M_{ID} cannot be justified this way, for very good reasons. As highlighted



■ **Figure 1** Replacing A with B .

¹ Recall that given (pseudo)metric spaces (X, d_X) , (Y, d_Y) we can give the set Y^X of non-expansive maps between X and Y a (pseudo)metric space structure setting $d_{Y^X}(f, g) = \sup_{x \in X} d_Y(f(x), g(x))$

73 by Westbrook and Chaudhuri [31], this is not the end of the story, at least if the environment
 74 in which M_{SIN} and M_{ID} operate feed either of them *only with* real numbers close to 0. If
 75 this is the case, M_{SIN} can be substituted with M_{ID} without affecting *too much* the overall
 76 behaviour of the system.

The key insight by Westbrook and Chaudhuri is that justifying program transformations like the one above requires taking the difference $\delta(M_{SIN}, M_{ID})$ between M_{SIN} and M_{ID} not merely as a number, but as a more structured object. What they suggest is to take $\delta(M_{SIN}, M_{ID})$ as *yet another program*, which however describes the difference between M_{SIN} and M_{ID} :

$$\delta(M_{SIN}, M_{ID}) := \lambda x. \lambda \varepsilon. |\sin x - x| + \varepsilon.$$

77 This reflects the fact that the distance between M_{SIN} and M_{ID} , namely the discrepancy
 78 between their output, depends not only on the discrepancy on the input, namely on ε , but
 79 also *on the input itself*, namely on x . It both x and ε are close to 0, $\delta(M_{SIN}, M_{ID})$ is itself
 80 close to 0.

81 In this paper, we develop Westbrook and Chaudhuri's ideas, and turn them into a
 82 framework of *differential logical relations*. We will do all this in a simply-typed λ -calculus
 83 with real numbers as the only base type. Starting from such a minimal calculus has at
 84 least two advantages: on the one hand one can talk about meaningful examples like the one
 85 above, and on the other hand the induced metatheory is simple enough to highlight the key
 86 concepts.

87 The contributions of this paper can be summarised as follows:

- 88 ■ After introducing our calculus $ST_{\mathbb{R}}^{\lambda}$, we define differential logical relations inductively
 89 on types, as ternary relations between pairs of programs and *differences*. The latter are
 90 mere set theoretic entities here, and the nature of differences between terms depends on
 91 terms' types.
- 92 ■ We prove a soundness theorem for differential logical relations, which allows us to justify
 93 compositional reasoning about terms' differences. We also prove a *finite difference theorem*,
 94 which stipulates that the distance between two simply-typed λ -terms is finite if mild
 95 conditions hold on the underlying set of function symbols.
- 96 ■ We give embeddings of logical and metric relations into differential logical relations. This
 97 witnesses that the latter are a generalisation of the former two.
- 98 ■ Finally, we show that generalised metric domains, the mathematical structure underlying
 99 differential logical relations, form a cartesian closed category, contrarily to the category
 100 of metric spaces, which is well known not to have the same property.

101 Due to space constraints, many details have to be omitted, but can be found in an Extended
 102 Version of this work [12].

103 **2 A Simply-Typed λ -Calculus with Real Numbers**

104 In this section, we introduce a simply-typed λ -calculus in which the only base type is the
 105 one of real numbers, and constructs for iteration and conditional are natively available.
 106 The choice of this language as the reference calculus in this paper has been made for the
 107 sake of simplicity, allowing us to concentrate on the most crucial aspects, at the same time
 108 guaranteeing a minimal expressive power.

XXX:4 Differential Logical Relations

$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$	$\frac{}{\Gamma \vdash r : REAL}$	$\frac{f_n \in \mathcal{F}_n}{\Gamma \vdash f_n : REAL^n \rightarrow REAL}$	$\frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \rightarrow \rho}$
$\frac{\Gamma \vdash M : \tau \rightarrow \rho \quad \Gamma \vdash N : \tau}{\Gamma \vdash MN : \rho}$	$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \rho}{\Gamma \vdash \langle M, N \rangle : \tau \times \rho}$	$\frac{}{\Gamma \vdash \pi_1 : \tau \times \rho \rightarrow \tau}$	$\frac{}{\Gamma \vdash \pi_2 : \tau \times \rho \rightarrow \rho}$
$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \text{iflz } M \text{ else } N : REAL \rightarrow \tau}$		$\frac{\Gamma \vdash M : \tau \rightarrow \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \text{iter } M \text{ base } N : REAL \rightarrow \tau}$	

■ **Figure 2** Typing rules for $ST_{\mathbb{R}}^{\lambda}$.

109 Terms and Types

$ST_{\mathbb{R}}^{\lambda}$ is a typed λ -calculus, so its definition starts by giving the language of *types*, which is defined as follows:

$$\tau, \rho ::= REAL \mid \tau \rightarrow \rho \mid \tau \times \rho.$$

110 The expression τ^n stands for $\underbrace{\tau \times \dots \times \tau}_{n \text{ times}}$. The set of *terms* is defined as follows:

$$111 \quad 112 \quad M, N ::= x \mid r \mid f_n \mid \lambda x.M \mid MN \mid \langle M, N \rangle \mid \pi_1 \mid \pi_2 \mid \text{iflz } M \text{ else } N \mid \text{iter } M \text{ base } N$$

113 where x ranges over a set \mathbb{V} of variables, r ranges over the set \mathbb{R} of real numbers, n is a natural
 114 number, and f_n ranges over a set \mathcal{F}_n of total real functions of arity n . We do not make any
 115 assumption on $\{\mathcal{F}_n\}_{n \in \mathbb{N}}$, apart from the predecessor $pred_1$ being part of \mathcal{F}_1 . The family, in
 116 particular, could in principle contain non-continuous functions. The expression $\langle M_1, \dots, M_n \rangle$
 117 is simply a shortcut for $\langle \dots \langle \langle M_1, M_2 \rangle, M_3 \rangle \dots, M_n \rangle$. All constructs are self-explanatory,
 118 except for the **iflz** and **iter** operators, which are conditional and iterator combinators,
 119 respectively. An *environment* Γ is a set of assignments of types to variables in \mathbb{V} where
 120 each variable occurs at most once. A *type judgment* has the form $\Gamma \vdash M : \tau$ where Γ is an
 121 environment, M is a term, and τ is a type. Rules for deriving correct typing judgments
 122 are in Figure 2, and are standard. The set of terms M for which $\cdot \vdash M : \tau$ is derivable is
 123 indicated as $CT(\tau)$.

124 Call-by-Value Operational Semantics

A static semantics is of course not enough to give meaning to a paradigmatic programming language, the dynamic aspects being captured only once an *operational* semantics is defined. The latter turns out to be very natural. *Values* are defined as follows:

$$V, W ::= r \mid f_n \mid \lambda x.M \mid \langle M, N \rangle \mid \pi_1 \mid \pi_2 \mid \text{iflz } M \text{ else } N \mid \text{iter } M \text{ base } N$$

125 The set of closed values of type τ is $CV(\tau) \subseteq CT(\tau)$, and the evaluation of $M \in CT(\tau)$
 126 produces a value $V \in CV(\tau)$, as formalised by the rules in Figure 3, through the judgment
 127 $M \Downarrow V$. We write $M \Downarrow$ if $M \Downarrow V$ is derivable *for some* V . The absence of full recursion has
 128 the nice consequence of guaranteeing a form of termination:

129 ► **Theorem 1.** *The calculus $ST_{\mathbb{R}}^{\lambda}$ is terminating: if $\cdot \vdash M : \tau$ then $M \Downarrow$.*

130 Theorem 1 can be proved by way of a standard reducibility argument. Termination implies
 131 the following.

132 ► **Corollary 2.** *If $\cdot \vdash M : REAL$ then there exists a unique $r \in \mathbb{R}$ satisfying $M \Downarrow r$, which
 133 we indicate as $NF(M)$.*

$$\boxed{
\begin{array}{c}
\frac{}{V \Downarrow V} \quad \frac{M \Downarrow f_n \quad N \Downarrow \langle L_1, \dots, L_n \rangle \quad L_i \Downarrow r_i}{MN \Downarrow f(r_1, \dots, r_n)} \quad \frac{M \Downarrow \lambda x.L \quad N \Downarrow V \quad L\{V/x\} \Downarrow W}{MN \Downarrow W} \\
\frac{M \Downarrow \pi_1 \quad N \Downarrow \langle L, P \rangle \quad L \Downarrow V}{MN \Downarrow V} \quad \frac{M \Downarrow \pi_2 \quad N \Downarrow \langle L, P \rangle \quad P \Downarrow V}{MN \Downarrow V} \\
\frac{M \Downarrow \text{iflz } L \text{ else } P \quad N \Downarrow r \quad r < 0 \quad L \Downarrow V}{MN \Downarrow V} \quad \frac{M \Downarrow \text{iflz } L \text{ else } P \quad N \Downarrow r \quad r \geq 0 \quad P \Downarrow V}{MN \Downarrow V} \\
\frac{M \Downarrow \text{iter } L \text{ base } P \quad N \Downarrow r \quad r < 0 \quad P \Downarrow V}{MN \Downarrow V} \\
\frac{M \Downarrow \text{iter } L \text{ base } P \quad N \Downarrow r \quad r \geq 0 \quad L((\text{iter } L \text{ base } P)(\text{pred}_1(r))) \Downarrow V}{MN \Downarrow V}
\end{array}
}$$

■ **Figure 3** Operational semantics for $ST_{\mathbb{R}}^{\lambda}$.

134 Context Equivalence

A *context* C is nothing more than a term containing a single occurrence of a placeholder $[\cdot]$. Given a context C , $C[M]$ indicates the term one obtains by substituting M for the occurrence of $[\cdot]$ in C . Typing rules in Figure 2 can be lifted to contexts by generalising judgments to the form $\Gamma \vdash C[\Delta \vdash \cdot : \tau] : \rho$, by which one captures that whenever $\Delta \vdash M : \tau$, it holds that $\Gamma \vdash C[M] : \rho$. Two terms M and N such that $\Gamma \vdash M, N : \tau$ are said to be *context equivalent* [22] if for every C such that $\emptyset \vdash C[\Gamma \vdash \cdot : \tau] : REAL$ it holds that $NF(C[M]) = NF(C[N])$. Context equivalence is the largest adequate congruence, and is thus considered as the coarsest “reasonable” equivalence between terms. It can also be turned into a pseudometric [11, 10] — called *context distance* — by stipulating that

$$\delta(M, N) = \sup_{\emptyset \vdash C[\Gamma \vdash \cdot : \tau] : REAL} |NF(C[M]) - NF(C[N])|.$$

135 The obtained notion of distance, however, is bound to trivialise [11], given that $ST_{\mathbb{R}}^{\lambda}$ is not
 136 affine. Trivialisation of context distance highlights an important limit of the metric approach
 137 to program difference which, ultimately, can be identified with the fact that program distances
 138 are sensitive to interactions with the environment. Our notion of a differential logical relation
 139 tackles such a problem from a different perspective, namely refining the concept of program
 140 distance which is not just a number, but is now able to take into account interactions with
 141 the environment.

142 Set-Theoretic Semantics

Before introducing differential logical relations, it is useful to remark that we can give $ST_{\mathbb{R}}^{\lambda}$ a standard set-theoretic semantics. To any type τ we associate the set $\llbracket \tau \rrbracket$, the latter being defined by induction on the structure of τ as follows:

$$\llbracket REAL \rrbracket = \mathbb{R}; \quad \llbracket \tau \rightarrow \rho \rrbracket = \llbracket \tau \rrbracket \rightarrow \llbracket \rho \rrbracket; \quad \llbracket \tau \times \rho \rrbracket = \llbracket \tau \rrbracket \times \llbracket \rho \rrbracket.$$

143 This way, any closed term $M \in CT(\tau)$ is interpreted as an element $\llbracket M \rrbracket$ of $\llbracket \tau \rrbracket$ in a natural
 144 way (see, e.g. [20]). Up to now, everything we have said about $ST_{\mathbb{R}}^{\lambda}$ is absolutely standard,
 145 and only serves to set the stage for the next sections.

146 3 Making Logical Relations Differential

147 Logical relations can be seen as one of the *many* ways of defining when two programs are to
 148 be considered equivalent. Their definition is type driven, i.e., they can be seen as a *family*

XXX:6 Differential Logical Relations

149 $\{\delta_\tau\}_\tau$ of binary relations indexed by types such that $\delta_\tau \subseteq CT(\tau) \times CT(\tau)$. This section is
 150 devoted to showing how all this can be made into differential logical relations.

The first thing that needs to be discussed is how to define the space of *differences* between programs. These are just boolean values in logical relations, become real numbers in ordinary metrics, and is type-dependent itself. A function $\langle \cdot \rangle$ that assigns a set to each type is defined as follows:

$$(REAL) = \mathbb{R}_{\geq 0}^\infty; \quad \langle \tau \rightarrow \rho \rangle = \llbracket \tau \rrbracket \times \langle \tau \rangle \rightarrow \langle \rho \rangle; \quad \langle \tau \times \rho \rangle = \langle \tau \rangle \times \langle \rho \rangle;$$

151 where $\mathbb{R}_{\geq 0}^\infty = \mathbb{R}_{\geq 0} \cup \{\infty\}$. The set $\langle \tau \rangle$ is said to be the *difference space* for the type τ and
 152 is meant to model the outcome of comparisons between closed programs of type τ . As an
 153 example, when τ is $REAL \rightarrow REAL$, we have that $\langle \tau \rangle = \mathbb{R} \times \mathbb{R}_{\geq 0}^\infty \rightarrow \mathbb{R}_{\geq 0}^\infty$. This is the type
 154 of the function $\delta(M, N)$ we used to compare the two programs described in the Introduction.

155 Now, which structure could we endow $\langle \tau \rangle$ with? First of all, we can define a partial order
 156 \leq_τ over $\langle \tau \rangle$ for each type τ as follows:

$$\begin{array}{ll} 157 & r \leq_{REAL} s & \text{if } r \leq s \text{ as the usual order over } \mathbb{R}_{\geq 0}^\infty; \\ 158 & f \leq_{\tau \rightarrow \rho} g & \text{if } \forall x \in \llbracket \tau \rrbracket. \forall t \in \langle \tau \rangle. f(x, t) \leq_\rho g(x, t); \\ 159 & (t, u) \leq_{\tau \times \rho} (s, r) & \text{if } t \leq_\tau s \text{ and } u \leq_\rho r. \end{array}$$

161 This order has least upper bounds and greater lower bounds, thanks to the nice structure of
 162 $\mathbb{R}_{\geq 0}^\infty$:

163 ► **Proposition 3.** *For each type τ , $(\langle \tau \rangle, \leq_\tau)$ forms a complete lattice.*

164 The fact that $\langle \tau \rangle$ has a nice order-theoretic structure is not the end of the story. For
 165 every type τ , we define a binary operation $*_\tau$ as follows:

$$\begin{array}{ll} 166 & r *_{REAL} s = r + s \text{ if } r, s \in \mathbb{R}_{\geq 0}; & (f *_{\tau \rightarrow \rho} g)(V, t) = f(V, t) *_\rho g(V, t); \\ 167 & r *_{REAL} s = \infty \text{ if } r = \infty \vee s = \infty; & (t, s) *_{\tau \times \rho} (u, r) = (t *_\tau u, s *_\rho r). \end{array}$$

169 This is precisely what it is needed to turn $\langle \tau \rangle$ into a *quantale*² [25].

170 ► **Proposition 4.** *For each type τ , $\langle \tau \rangle$ forms a commutative unital non-idempotent quantale.*

171 The fact that $\langle \tau \rangle$ is a quantale means that it has, e.g., the right structure to be the
 172 codomain of generalised metrics [19, 18]. Actually, a more general structure is needed for our
 173 purposes, namely the one of a generalised metric domain, which will be thoroughly discussed
 174 in Section 6 below. For the moment, let us concentrate our attention to programs:

175 ► **Definition 5** (Differential Logical Relations). *We define a differential logical relation $\{\delta_\tau \subseteq$
 176 $\Lambda_\tau \times \langle \tau \rangle \times \Lambda_\tau\}_\tau$ as a set of ternary relations indexed by types satisfying*

$$\begin{array}{ll} 177 & \delta_{REAL}(M, r, N) \Leftrightarrow |NF(M) - NF(N)| \leq r; \\ 178 & \delta_{\tau \times \rho}(M, (d_1, d_2), N) \Leftrightarrow \delta_\tau(\pi_1 M, d_1, \pi_1 N) \wedge \delta_\rho(\pi_2 M, d_2, \pi_2 N) \\ 179 & \delta_{\tau \rightarrow \rho}(M, d, N) \Leftrightarrow (\forall V \in CV(\tau). \forall x \in \langle \tau \rangle. \forall W \in CV(\tau). \\ 180 & \delta_\tau(V, x, W) \Rightarrow \delta_\rho(MV, d(\llbracket V \rrbracket), x), NW) \wedge \delta_\rho(MW, d(\llbracket V \rrbracket), x), NV)). \end{array}$$

182 An intuition behind the condition required for $\delta_{\tau \rightarrow \rho}(M, d, N)$ is that $d(\llbracket V \rrbracket), x)$ overapprox-
 183 imates both the “distance” between MV and NW and the one between MW and NV , this
 184 whenever W is within the error x from V .

² Recall that a quantale $\mathbb{Q} = (Q, \leq_Q, 0_Q, *_Q)$ consists of a complete lattice (Q, \leq_Q) and a monoid $(Q, 0_Q, *_Q)$ such that the lattice and monoid structures properly interact (meaning that monoid multiplication distributes over joins). We refer to [25, 18] for details.

3.1 A Fundamental Lemma

Usually, the main result about any system of logical relations is the so-called *Fundamental Lemma*, which states that any typable term is in relation *with itself*. But how would the Fundamental Lemma look like here? Should any term be at *somehow minimal distance* to itself, in the spirit of what happens, e.g. with metrics [24, 13]? Actually, there is no hope to prove anything like that for differential logical relations, as the following example shows.

► **Example 6.** Consider again the term $M_{ID} = \lambda x.x$, which can be given type $\tau = REAL \rightarrow REAL$ in the empty context. Please recall that $(\tau) = \mathbb{R} \times \mathbb{R}_{\geq 0}^{\infty} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$. Could we prove that $\delta_{\tau}(M_{ID}, 0_{\tau}, M_{ID})$, where 0_{τ} is the constant-0 function? The answer is negative: given two real numbers r and s at distance ε , the terms $M_{ID}r$ and $M_{ID}s$ are themselves ε apart, thus at nonnull distance. The best one can say, then, is that $\delta_{\tau}(M_{ID}, f, M_{ID})$, where $f(x, \varepsilon) = \varepsilon$.

As the previous example suggests, a term M being at self-distance d is a witness of M being *sensitive to changes* to the environment according to d . Indeed, the only terms which are at self-distance 0 are the constant functions. This makes the underlying theory more general than the one of logical or metric relations, although the latter can be proved to be captured by differential logical relations, as we will see in the next section.

Coming back to the question with which we opened the section, we can formulate a suitable fundamental lemma for differential logical relations.

► **Theorem 7 (Fundamental Lemma, Version I).** *For every $\cdot \vdash M : \tau$ there is a $d \in (\tau)$ such that $(M, d, M) \in \delta_{\tau}$.*

Proof sketch. The proof proceeds, as usual, by induction on the derivation of $\cdot \vdash M : \tau$. In order to deal with e.g. λ -abstractions we have to strengthen our statement taking into account *open* terms. This turned out to be non-trivial and requires to extend our notion of a differential logical relation to arbitrary terms. First of all, we need to generalise (\cdot) and $[\cdot]$ to environments. For instance, (Γ) is the set of families in the form $\alpha = \{\alpha_x\}_{(x:\rho) \in \Gamma}$, where $\alpha_x \in (\rho)$. Similarly for $[\Gamma]$. This way, a natural space for differences between terms $\Gamma \vdash M, N : \tau$ can be taken as $(\tau)^{[\Gamma] \times (\Gamma)}$, namely the set of maps from $[\Gamma] \times (\Gamma)$ to (τ) . Given an environment Γ , a family $\mathbf{V} = \{V_x\}_{(x:\rho_x) \in \Gamma}$ such that $V_x \in CV(\rho_x)$ is said to be a Γ -family of values. Such a Γ -family of values can naturally be seen as a substitution \mathbf{V} mapping each variable $(x : \rho) \in \Gamma$ to $V_x \in CV(\rho_x)$. As it is customary, for a term $\Gamma \vdash M : \tau$ we write $M\mathbf{V}$ for the closed term of type τ obtained applying the substitution \mathbf{V} to M . We denote by $CV(\Gamma)$ the set of all Γ -family of values. Given a set Z , an environment Γ , and two Γ -indexed families $\alpha = \{\alpha_x\}_{(x:\rho) \in \Gamma}$, $\beta = \{\beta_x\}_{(x:\rho) \in \Gamma}$ over Z (meaning that e.g. $\alpha_x \in Z$, for each $(x : \rho) \in \Gamma$), we introduce the following notational convention. For a Γ -indexed family $B = \{b_x\}_{(x:\rho) \in \Gamma}$ such that $b_x \in \{0, 1\}$, we can construct a ‘choice’ Γ -indexed family B_{β}^{α} as follows:

$$(B_{\beta}^{\alpha})_x = \begin{cases} \alpha_x & \text{if } b_x = 0 \\ \beta_x & \text{if } b_x = 1. \end{cases}$$

Moreover, given a family B as above, we can construct the *inverse* family \overline{B} as the family $\{1 - b_x\}_{(x:\rho) \in \Gamma}$. We can now talk about *open terms*, and from a differential logical relation $\{\delta_{\tau} \subseteq \Lambda_{\tau} \times (\tau) \times \Lambda_{\tau}\}_{\tau}$ construct a family of relations $\{\delta_{\tau}^{\Gamma} \subseteq \Lambda_{\tau}^{\Gamma} \times (\tau)^{[\Gamma] \times (\Gamma)} \times \Lambda_{\tau, \Gamma}^{\Gamma}\}_{\tau}$ by stipulating that $\delta_{\tau}^{\Gamma}(M, d, N)$ iff

$$\delta_{\Gamma}(\mathbf{V}, Y, \mathbf{W}) \implies \forall B \in \{0, 1\}^{\Gamma}. \delta_{\tau}(MB_{\mathbf{W}}^{\mathbf{V}}, d([\mathbf{V}], Y), N\overline{B}_{\mathbf{W}}^{\mathbf{V}}).$$

We now prove the following strengthening of our main thesis: for any term $\Gamma \vdash M : \tau$, there is a $d \in (\tau)^{[\Gamma] \times (\Gamma)}$ such that $\delta_{\tau}^{\Gamma}(M, d, M)$. At this point the proof is rather standard, and proceeds by induction on the derivation of $\Gamma \vdash M : \tau$. ◀

XXX:8 Differential Logical Relations

208 But what do we gain from Theorem 7? In the classic theory of logical relations, the
209 Fundamental Lemma has, as an easy corollary, that logical relations are compatible: it suffices
210 to invoke the theorem with any context C seen as a term $C[x]$, such that $x : \tau, \Gamma \vdash C[x] : \rho$.
211 Thus, ultimately, logical relations are proved to be a *compositional* methodology for program
212 equivalence, in the following sense: if M and N are equivalent, then $C[M]$ and $C[N]$ are
213 equivalent, too.

214 In the realm of differential logical relations, the Fundamental Lemma plays a similar
215 role, although with a different, *quantitative* flavor: once C has been proved sensitive to
216 changes according to d , and V, W are proved to be at distance e , then, e.g., the impact
217 of substituting V with W in C can be measured by composing d and e (and $\llbracket V \rrbracket$), i.e. by
218 computing $d(\llbracket V \rrbracket, e)$. Notice that the sensitivity analysis on C and the relational analysis on
219 V and W are decoupled. What the Fundamental Lemma tells you is that d can *always* be
220 found.

221 3.2 Our Running Example, Revisited

It is now time to revisit the example we talked about in the Introduction. Consider the following two programs, both closed and of type $REAL \rightarrow REAL$:

$$M_{SIN} = \lambda x. \sin_1(x); \quad M_{ID} = \lambda x. x.$$

222 First of all, let us observe that, as already remarked, comparing M_{SIN} and M_{ID} using the
223 sup metric on $\mathbb{R} \rightarrow \mathbb{R}$, as it is done in metric logical relations and applicative distances,
224 naturally assigns them distance ∞ , the euclidean distance $|x - \sin(x)|$ being unbounded
225 when x ranges over \mathbb{R} .

226 Let us now prove that $(M_{SIN}, f, M_{ID}) \in \delta_{REAL \rightarrow REAL}$, where $f(x, y) = y + |x - \sin x|$.
227 Consider any pair of real numbers $r, s \in \mathbb{R}$ such that $|r - s| \leq \varepsilon$, where $\varepsilon \in \mathbb{R}_{\geq 0}^{\infty}$. We have
228 that:

$$\begin{aligned} 229 \quad |\sin r - s| &= |\sin r - r + r - s| \leq |\sin r - r| + |r - s| \leq |\sin r - r| + \varepsilon = f(r, \varepsilon) \\ 230 \quad |\sin s - r| &= |\sin s - \sin r + \sin r - r| \leq |\sin s - \sin r| + |\sin r - r| \leq |s - r| + |\sin r - r| \\ 231 \quad &\leq \varepsilon + |\sin r - r| = f(r, \varepsilon). \end{aligned}$$

233 The fact that $|\sin s - \sin r| \leq |s - r|$ is a consequence of \sin being 1-Lipschitz continuous
234 (see, e.g., [12] for a simple proof).

Now, consider a context C which makes use of either M_{SIN} or M_{ID} by feeding them with a value close to 0, call it θ . Such a context could be, e.g., $C = (\lambda x. x(x\theta))[\cdot]$. C can be seen as a term having type $\tau = (REAL \rightarrow REAL) \rightarrow REAL$. A self-distance d for C can thus be defined as an element of

$$(\tau) = \llbracket REAL \rightarrow REAL \rrbracket \times (\llbracket REAL \rightarrow REAL \rrbracket) \rightarrow \mathbb{R}_{\geq 0}^{\infty}.$$

235 namely $F = \lambda(g, h). h(g(\theta), h(\theta, 0))$. This allows for compositional reasoning about program
236 distances: the overall impact of replacing M_{SIN} by M_{ID} can be evaluated by computing
237 $F(\llbracket M_{SIN} \rrbracket, f)$. Of course the context C needs to be taken into account, but *once and for all*:
238 the functional F can be built without knowing with which term(s) it will be fed with.

239 any access to either M_{SIN} or M_{ID} .

240 4 Logical and Metric Relations as DLRs

241 The previous section should have convinced the reader about the peculiar characteristics
242 of differential logical relations compared to (standard) metric and logical relations. In

243 this section we show that despite the apparent differences, logical and metric relations can
 244 somehow be retrieved as specific kinds of program differences. This is, however, bound to
 245 be nontrivial. The naïve attempt, namely seeing program equivalence as being captured by
 246 *minimal* distances in logical relations, fails: the distance between a program *and itself* can
 247 be nonnull.

How should we proceed, then? Isolating those distances which witness program equivalence is indeed possible, but requires a bit of an effort. In particular, the sets of those distances can be, again, defined by induction on τ . For every τ , we give $(\tau)^0 \subseteq (\tau)$ by induction on the structure of τ :

$$\begin{aligned} (REAL)^0 &= \{0\}; & (\tau \times \rho)^0 &= (\tau)^0 \times (\rho)^0; \\ (\tau \rightarrow \rho)^0 &= \{f \in (\tau \rightarrow \rho) \mid \forall x \in \llbracket \tau \rrbracket. \forall y \in (\tau)^0. f(x, y) \in (\rho)^0\}. \end{aligned}$$

248 Notice that $(\tau \rightarrow \rho)^0$ is not defined as $\llbracket \tau \rrbracket \times (\tau)^0 \rightarrow (\rho)^0$ (doing so would violate $(\tau \rightarrow \rho)^0 \subseteq$
 249 $(\tau \rightarrow \rho)$). The following requires some effort, and testifies that, indeed, program equivalence
 250 in the sense of logical relations precisely corresponds to being at a distance in $(\tau)^0$:

251 **► Theorem 8.** *Let $\{\mathcal{L}_\tau\}_\tau$ be a logical relation. There exists a differential logical relation*
 252 *$\{\delta_\tau\}_\tau$ satisfying $\mathcal{L}_\tau(M, N) \iff \exists d \in (\tau)^0. \delta_\tau(M, d, N)$.*

What if we want to generalise the argument above to metric relations, as introduced, e.g., by Reed and Pierce [24]? The set $(\tau)^0$ becomes a set of distances parametrised by a single real number:

$$\begin{aligned} (REAL)^r &= \{r\}; & (\tau \times \rho)^r &= (\tau)^r \times (\rho)^r; \\ (\tau \rightarrow \rho)^r &= \{f \in (\tau \rightarrow \rho) \mid \forall x \in \llbracket \tau \rrbracket. \forall y \in (\tau)^s. f(x, y) \in (\rho)^{r+s}\}. \end{aligned}$$

253 A result similar to Theorem 8 is unfortunately outside the scope of this paper, but can be
 254 found in the Extended Version [12]. In particular, metric relations are only available in
 255 calculi, like FUZZ [24], which rely on *linear* type systems, thus more refined than the one we
 256 endow $ST_{\mathbb{R}}^\lambda$ with.

257 5 Strengthening the Fundamental Theorem through Finite Distances

258 Let us now ask ourselves the following question: given any term $M \in CT(\tau)$, what can
 259 we say about its sensitivity, i.e., about the values $d \in (\tau)$ such that $\delta_\tau(M, d, M)$? Two of
 260 the results we have proved about $ST_{\mathbb{R}}^\lambda$ indeed give partial answers to the aforementioned
 261 question. On the one hand, Theorem 7 states that such a d can *always* be found. On the
 262 other hand, Theorem 8 tells us that such a d can be taken in $(\tau)^0$. Both these answers are
 263 not particularly informative, however. The mere existence of such a $d \in (\tau)$, for example, is
 264 trivial since d can always be taken as d_∞ , the maximal element of the underlying quantale.
 265 The fact that such a d can be taken from $(\tau)^0$ tells us that, e.g. when $\tau = \rho \rightarrow \xi$, M returns
 266 equivalent terms when fed with equivalent arguments: there is no quantitative guarantee
 267 about the behaviour of the term when fed with non-equivalent arguments.

Is this the best one can get about the sensitivity of $ST_{\mathbb{R}}^\lambda$ terms? The absence of full recursion suggests that we could hope to prove that infinite distances, although part of the underlying quantale, can in fact be useless. In other words, we are implicitly suggesting that self-distances could be elements of $(\tau)^{<\infty} \subset (\tau)$, defined as follows:

$$\begin{aligned} (REAL)^{<\infty} &= \mathbb{R}_{\geq 0}; & (\tau \times \rho)^{<\infty} &= (\tau)^{<\infty} \times (\rho)^{<\infty}; \\ (\tau \rightarrow \rho)^{<\infty} &= \{f \in (\tau \rightarrow \rho) \mid \forall x \in \llbracket \tau \rrbracket. \forall t \in (\tau)^{<\infty}. f(x, t) \in (\rho)^{<\infty}\}. \end{aligned}$$

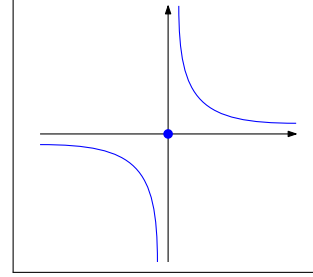
XXX:10 Differential Logical Relations

268 Please observe that $(\tau)^{<\infty}$ is in general a much larger set of differences than $\bigcup_{r \in \mathbb{R}_{\geq 0}^{\infty}} (\tau)^r$:
 269 the former equals the latter only when τ is *REAL*. Already when τ is *REAL* \rightarrow *REAL*, the
 270 former includes, say, functions like $f(r, \varepsilon) = (r + \varepsilon)^2$, while the latter does not.

Unfortunately, there are terms in $ST_{\mathbb{R}}^{\lambda}$ which cannot be proved to be at self-distance in $(\tau)^{<\infty}$, and, surprisingly, this is *not* due to the higher-order features of $ST_{\mathbb{R}}^{\lambda}$, but to $\{\mathcal{F}_n\}_{n \in \mathbb{N}}$ being arbitrary, and containing functions which do not map finite distances to finite distances, like

$$h(r) = \begin{cases} 0 & \text{if } r = 0 \\ \frac{1}{r} & \text{otherwise} \end{cases}$$

271 (see Figure 4). Is this phenomenon *solely* responsible for the
 272 necessity of finite self-distances in $ST_{\mathbb{R}}^{\lambda}$? The answer is positive,
 273 and the rest of this section is devoted precisely to formalising
 274 and proving the aforementioned conjecture.



■ **Figure 4** A total, but highly discontinuous, function.

First of all, we need to appropriately axiomatise the absence of unbounded discontinuities from $\{\mathcal{F}_n\}_{n \in \mathbb{N}}$. A not-so-restrictive but sufficient axiom turns out to be weak boundedness: a function $f_n : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be *weakly bounded* if and only if it maps bounded subsets of \mathbb{R}^n into bounded subsets of \mathbb{R} . As an example, the function h above is *not* weakly bounded, because $h([- \varepsilon, \varepsilon])$ is

$$\left(-\infty, -\frac{1}{\varepsilon}\right] \cup \{0\} \cup \left[\frac{1}{\varepsilon}, \infty\right)$$

275 which is unbounded for any $\varepsilon > 0$. Any term M is said to be weakly bounded iff any
 276 function symbol f_n occurring in M is itself weakly bounded. Actually, this is precisely what
 277 one needs to get the strengthening of the Fundamental Theorem we are looking for.

278 ► **Theorem 9** (Fundamental Theorem, Version II). *For any weakly bounded term $\cdot \vdash M : \tau$,*
 279 *there is $d \in (\tau)^{<\infty}$ such that $(M, d, M) \in \delta_{\tau}$.*

280 The reader may have wondered about how restrictive a condition weak boundedness really
 281 is. In particular, whether it corresponds to some form of continuity. In fact, the introduced
 282 condition only rules out unbounded discontinuities. In other words, weak boundedness can
 283 be equivalently defined by imposing local boundedness *at any point* in the domain \mathbb{R} . This is
 284 weaker than asking for boundedness, which requires the existence of a global bound.

285 6 A Categorical Perspective

286 Up to now, differential logical relations have been treated very concretely, without looking at
 287 them through the lens of category theory. This is in contrast to, e.g., the treatment of metric
 288 relations from [13], in which soundness of metric relations for **FUZZ** is obtained as a byproduct
 289 of a proof of symmetric monoidal closedness for the category **MET** of pseudometric spaces
 290 and Lipschitz functions.

291 But what could take the place of pseudometric spaces in a categorical framework capturing
 292 differential logical relations? The notion of a metric needs to be relaxed along at least two
 293 axes. On the one hand, the codomain of the “metric” δ is not necessarily the set of real
 294 numbers, but a more general structure, namely a quantale. On the other, as we already
 295 noticed, it is not necessarily true that equality implies indistancy, but rather than indistancy
 296 implies inequality. What comes out of these observations is, quite naturally, the notion

297 of a generalized metric domain, itself a generalisation of partial metrics [7]. The rest of
 298 this section is devoted to proving that the category of generalised metric domains is indeed
 299 cartesian closed, thus forming a model of simply typed λ -calculi.

300 Formally, given a quantale $\mathbb{Q} = (Q, \leq_Q, 0_Q, *_Q)$ ³, a *generalised metric domain* on \mathbb{Q} is a
 301 pair (A, δ_A) , where A is a set and δ_A is a subset of $A \times \mathbb{Q} \times A$ satisfying some axioms akin
 302 to those of a metric domain:

$$303 \quad \delta_A(x, 0_Q, y) \Rightarrow x = y; \quad (\text{Indistancy Implies Equality})$$

$$304 \quad \delta_A(x, d, y) \Rightarrow \delta_A(y, d, x); \quad (\text{Symmetry})$$

$$305 \quad \delta_A(x, d, y) \wedge \delta_A(y, e, z) \wedge \delta_A(y, f, z) \Rightarrow \delta_A(x, d * e * f, z). \quad (\text{Triangularity})$$

307 Please observe that δ_A is a *relation* rather than a function. Moreover, the first axiom is dual
 308 to the one typically found in, say, pseudometrics. The third axiom, instead, resembles the
 309 usual triangle inequality for pseudometrics, but with the crucial difference that since objects
 310 can have non-null self-distance, such a distance has to be taken into account. Requiring
 311 equality to imply indistancy (and thus $\delta_A(x, 0_Q, y) \Leftrightarrow x = y$), we see that (Triangularity)
 312 gives exactly the usual triangle inequality (properly generalised to quantale and relations
 313 [18, 19]).

314 In this section we show that generalised metric domains form a cartesian closed category,
 315 unlike that of metric spaces (which is known to be non-cartesian closed). As a consequence,
 316 we obtain a firm categorical basis of differential logical relations. The category of generalised
 317 metric domain, denoted by **GMD**.

318 ► **Definition 10.** *The category **GMD** has the following data.*

319 ■ *An object \mathcal{A} is a triple (A, \mathbb{Q}, δ) where \mathbb{Q} is a quantale and (A, δ) is a generalised metric
 320 domain on \mathbb{Q} .*

321 ■ *An arrow $(A, \mathbb{Q}, \delta) \rightarrow (B, \mathbb{S}, \rho)$ is a pair (f, ζ) consisting of a function $f: A \rightarrow B$ and an-
 322 other function $\zeta: Q \times A \rightarrow S$ satisfying $\forall a, a' \in A. \forall q \in Q. \delta(a, q, a') \Rightarrow \rho(f(a), \zeta(q, a), f(a'))$
 323 and $\rho(f(a), \zeta(q, a'), f(a'))$.*

324 We can indeed give **GMD** the structure of a category. In fact, the identity on the object
 325 $\mathcal{A} = (A, \mathbb{Q}, \delta)$ in **GMD** is given by $(\text{id}_{\mathcal{A}}, \text{id}'_{\mathcal{A}})$ where $\text{id}_{\mathcal{A}}: A \rightarrow A$ is the set-theoretic
 326 identity on A and $\text{id}'_{\mathcal{A}}: Q \times A \rightarrow Q$ is defined by $\text{id}'_{\mathcal{A}}(q, a) = q$. The composition of two
 327 arrows $(f, \zeta): (A, \mathbb{Q}, \delta) \rightarrow (B, \mathbb{S}, \rho)$ and $(g, \eta): (B, \mathbb{S}, \rho) \rightarrow (C, \mathbb{T}, \nu)$ is the pair (h, θ) where
 328 $h: A \rightarrow C$ is given by the function composition $g \circ f: A \rightarrow C$ and $h: Q \times A \rightarrow T$ is given by
 329 $\theta(q, a) = \eta(\zeta(q, a), f(a))$. Straightforward calculations show that composition is associative,
 330 and that the identity arrow behaves as its neutral element.

331 Most importantly, we can give **GMD** a cartesian closed structure, as shown by the
 332 following result⁴.

333 ► **Theorem 11.** ***GMD** is a cartesian closed category.*

334 **Proof sketch.** Before entering details, it is useful to remark that the cartesian product of two
 335 quantales is itself a quantale (with lattice and monoid structure defined pointwise). Similarly,
 336 for any quantale \mathbb{Q} and set X , the function space \mathbb{Q}^X inherits a quantale structure from \mathbb{Q}
 337 pointwise. Let us now show that **GMD** is cartesian closed. We begin showing that **GMD**
 338 has a terminal object and binary products. The former is defined as $(\{*\}, \mathbb{O}, \delta_0)$, where \mathbb{O}

³ When unambiguous, we will omit subscripts in \leq_Q , 0_Q , and $*_Q$.

⁴ See [12] for a detailed proof.

XXX:12 Differential Logical Relations

339 is the one-element quantale $\{0\}$, and $\delta_0 = \{(*, 0, *)\}$ (notice that $(\{*\}, \delta_0)$ is a generalized
340 metric domain on \mathbb{O}), whereas the binary product $\mathcal{A} \times \mathcal{B}$ of two objects \mathcal{A} and \mathcal{B} in **GMD**
341 is given by a triple $(A \times B, \mathbb{Q} \times \mathbb{S}, \delta \times \rho)$. Finally, we define exponentials in **GMD**. Given
342 \mathcal{C}, \mathcal{B} in **GMD**, their exponential $\mathcal{C}^{\mathcal{B}}$ is the triple $(C^{\mathcal{B}}, \mathbb{T}^{\mathbb{S} \times \mathcal{B}}, \nu^\rho)$, where $C^{\mathcal{B}}$ is the function
343 space $\{f \mid f: B \rightarrow C\}$, $\mathbb{T}^{\mathbb{S} \times \mathcal{B}}$ is the exponential quantale, and ν^ρ is a ternary relation over
344 $C^{\mathcal{B}} \times T^{\mathbb{S} \times \mathcal{B}} \times C^{\mathcal{B}}$ defined by: if $\rho(b, s, b')$ then $\nu(f(b), d(s, b), f'(b'))$ and $\nu(f(b), d(s, b'), \zeta(b'))$.
345 Please notice that the relation ν^ρ is indeed a differential logical relation. ◀

346 Interestingly, the constructions of product and exponential objects in the proof of The-
347 orem 11 closely match the definition of a differential logical relation. In other words,
348 differential logical relations as given in Definition 5 can be seen as providing a denota-
349 tional model of $ST_{\mathbb{R}}^\lambda$ in which base types are interpreted by the generalised metric domain
350 corresponding to the Euclidean distance.

7 Conclusion

352 In this paper, we introduced differential logical relations as a novel methodology to evaluate
353 the “distance” between programs of higher-order calculi akin to the λ -calculus. We have been
354 strongly inspired by some unpublished work by Westbrook and Chaudhuri [31], who were the
355 first to realise that evaluating differences between interactive programs requires going beyond
356 mere real numbers. We indeed borrowed our running examples from the aforementioned
357 work.

358 This paper’s contribution, then consists in giving a simple definition of differential logical
359 relations, together with some results about their underlying metatheory: two formulations of
360 the Fundamental Lemma, a result relating differential logical relations and ordinary logical
361 relations, and a proof that generalised metric domains — the metric structure corresponding
362 to differential logical relations — form a cartesian closed category. Such results give evidence
363 that, besides being *more expressive* than metric relations, differential logical relations are
364 somehow *more canonical*, naturally forming a model of simply-typed λ -calculi.

365 As the title of this paper suggests, we see the contributions above just as a very first step
366 towards understanding the nature of differences in a logical environment. In particular, at
367 least two directions deserve to be further explored.

- 368 ■ The first one concerns *language features*: admittedly, the calculus $ST_{\mathbb{R}}^\lambda$ we consider here
369 is very poor in terms of its expressive power, lacking full higher-order recursion and
370 thus not being universal. Moreover, $ST_{\mathbb{R}}^\lambda$ does not feature any form of effect, including
371 probabilistic choices, in which evaluating differences between programs would be very
372 helpful. Addressing such issues seems to require to impose a domain structure on
373 generalised metric domain, on one hand, and to look at monads on **GMD**, on the other
374 hand (for the latter, the literature on monadic lifting for quantale-valued relations might
375 serve as a guide [18]).
- 376 ■ The second one is about *abstract differences*: defining differences as functions with *the*
377 *same rank* as that of the compared programs implies that reasoning about them is complex.
378 Abstracting differences so as to facilitate differential reasoning could be the way out,
379 given that deep connections exist between logical relations and abstract interpretation [2].
380 Another way to understand program difference better is to investigate whether differential
381 logical relations can be related to abstract structures for differentiation, as in [3]. Indeed,
382 Example 6 suggests that an interesting distance between a program and itself can be
383 taken as its derivative, the latter being defined as in [8].

384 — References —

- 385 1 S. Abramsky. The lazy lambda calculus. In D. Turner, editor, *Research Topics in Functional*
 386 *Programming*, pages 65–117. Addison Wesley, 1990.
- 387 2 Samson Abramsky. Abstract interpretation, logical relations and Kan extensions. *J. Log.*
 388 *Comput.*, 1(1):5–40, 1990.
- 389 3 Mario Alvarez-Picallo and C.-H. Luke Ong. Change actions: Models of generalised differenti-
 390 ation. In *Proc. of FOSSACS 2019*, pages 45–61, 2019.
- 391 4 A. Arnold and M. Nivat. Metric interpretations of infinite trees and semantics of non
 392 deterministic recursive programs. *Theor. Comput. Sci.*, 11:181–205, 1980.
- 393 5 C. Baier and M.E. Majster-Cederbaum. Denotational semantics in the CPO and metric
 394 approach. *Theor. Comput. Sci.*, 135(2):171–220, 1994.
- 395 6 Gilles Barthe, Marco Gaboardi, Justin Hsu, and Benjamin C. Pierce. Programming language
 396 techniques for differential privacy. *SIGLOG News*, 3(1):34–53, 2016.
- 397 7 Michael A. Bukatin, Ralph Kopperman, Steve Matthews, and Homeira Pajoohesh. Partial
 398 metric spaces. *The American Mathematical Monthly*, 116(8):708–718, 2009.
- 399 8 Yufei Cai, Paolo G. Giarrusso, Tillmann Rendel, and Klaus Ostermann. A theory of changes
 400 for higher-order languages: incrementalizing λ -calculi by static differentiation. In *Proc. of*
 401 *PLDI*, pages 145–155, 2014.
- 402 9 Konstantinos Chatzikokolakis, Daniel Gebler, Catuscia Palamidessi, and Lili Xu. Generalized
 403 bisimulation metrics. In *CONCUR 2014 - Concurrency Theory - 25th International Conference,*
 404 *CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, pages 32–46, 2014.
- 405 10 Raphaëlle Crubillé and Ugo Dal Lago. Metric reasoning about λ -terms: The affine case. In
 406 *Proc. of LICS 2015*, pages 633–644, 2015.
- 407 11 Raphaëlle Crubillé and Ugo Dal Lago. Metric reasoning about λ -terms: The general case. In
 408 *Proc. of ESOP 2017*, pages 341–367, 2017.
- 409 12 Ugo Dal Lago, Francesco Gavazzo, and Akira Yoshimizu. Differential logical relations, Part I:
 410 The simply-typed case (extended version). 2018. URL: <https://arxiv.org/abs/1904.12137>.
- 411 13 A.A. de Amorim, M. Gaboardi, J. Hsu, S. Katsumata, and I. Cherigui. A semantic account of
 412 metric preservation. In *Proc. of POPL 2017*, pages 545–556, 2017.
- 413 14 J.W. de Bakker and J.I. Zucker. Denotational semantics of concurrency. In *STOC*, pages
 414 153–158, 1982.
- 415 15 Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for
 416 labelled markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004.
- 417 16 M.H. Escardo. A metric model of pcf. In *Workshop on Realizability Semantics and Applications*,
 418 1999.
- 419 17 Francesco Gavazzo. Quantitative behavioural reasoning for higher-order effectful programs:
 420 Applicative distances. In *Proc. of LICS 2018*, pages 452–461, 2018.
- 421 18 D. Hofmann, G.J. Seal, and W. Tholen, editors. *Monoidal Topology. A Categorical Approach to*
 422 *Order, Metric, and Topology*. Number 153 in Encyclopedia of Mathematics and its Applications.
 423 Cambridge University Press, 2014.
- 424 19 F.W. Lawvere. Metric spaces, generalized logic, and closed categories. *Rend. Sem. Mat. Fis.*
 425 *Milano*, 43:135–166, 1973.
- 426 20 John C. Mitchell. *Foundations for Programming Languages*. MIT Press, 1996.
- 427 21 Sparsh Mittal. A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4),
 428 2016.
- 429 22 J. Morris. *Lambda Calculus Models of Programming Languages*. PhD thesis, MIT, 1969.
- 430 23 Gordon D. Plotkin. Lambda-definability and logical relations. Memorandum SAI-RM-4,
 431 University of Edinburgh, 1973.
- 432 24 J. Reed and B.C. Pierce. Distance makes the types grow stronger: a calculus for differential
 433 privacy. In *Proc. of ICFP 2010*, pages 157–168, 2010.
- 434 25 K.I. Rosenthal. *Quantales and their applications*. Pitman research notes in mathematics series.
 435 Longman Scientific & Technical, 1990.

XXX:14 Differential Logical Relations

- 436 **26** Dana Scott. Outline of a mathematical theory of computation. Technical Report PRG02,
437 OUCL, November 1970.
- 438 **27** Dana Scott and Christopher Strachey. Toward a mathematical semantics for computer
439 languages. Technical Report PRG06, OUCL, August 1971.
- 440 **28** F. Van Breugel. An introduction to metric semantics: operational and denotational models
441 for programming and specification languages. *Theor. Comput. Sci.*, 258(1-2):1–98, 2001.
- 442 **29** F. Van Breugel and J. Worrell. A behavioural pseudometric for probabilistic transition systems.
443 *Theor. Comput. Sci.*, 331(1):115–142, 2005.
- 444 **30** Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic
445 transition systems. In *Proc. of ICALP 2001*, pages 421–432, 2001.
- 446 **31** Edwin M. Westbrook and Swarat Chaudhuri. A semantics for approximate program trans-
447 formations. *CoRR*, abs/1304.5531, 2013. URL: <http://arxiv.org/abs/1304.5531>.
- 448 **32** Lili Xu, Konstantinos Chatzikokolakis, and Huimin Lin. Metrics for differential privacy in
449 concurrent systems. In *Proc. of FORTE 2014*, pages 199–215, 2014.