

A Partial Metric Semantics of Higher-Order Types and Approximate Program Transformations

Guillaume Geoffroy

Università di Bologna, Dipartimento Informatica, Scienza e Ingegneria, Italy
guillaume.geoffroy@unibo.it

Paolo Pistone

Università di Bologna, Dipartimento Informatica, Scienza e Ingegneria, Italy
paolo.pistone2@unibo.it

Abstract

Program semantics is traditionally concerned with program equivalence. However, in fields like approximate, incremental and probabilistic computation, it is often useful to describe *to which extent* two programs behave in a similar, although non equivalent way. This has motivated the study of program (pseudo)metrics, which have found widespread applications, *e.g.* in differential privacy. In this paper we show that the standard metric on real numbers can be lifted to higher-order types in a novel way, yielding a metric semantics of the simply typed lambda-calculus in which types are interpreted as quantale-valued partial metric spaces. Using such metrics we define a class of higher-order denotational models, called diameter space models, that provide a quantitative semantics of approximate program transformations. Noticeably, the distances between objects of higher-types are elements of functional, thus non-numerical, quantales. This allows us to model contextual reasoning about arbitrary functions, thus deviating from classic metric semantics.

2012 ACM Subject Classification Theory of computation → Denotational semantics

Keywords and phrases Simply typed λ -calculus, program metrics, approximate program transformations, partial metric spaces

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Guillaume Geoffroy*: ERC CoG 818616 “DIAPASoN”, ANR 16CE250011 “REPAS”

Paolo Pistone: ERC CoG 818616 “DIAPASoN”, ANR 16CE250011 “REPAS”

1 Introduction

In program semantics one is usually interested in capturing notions of behavioral equivalence between programs. However, in several fields like approximate [34], incremental [10, 2] and probabilistic [13] computation, it is often more useful to be able to describe *to which extent* two programs behave in a similar, although non equivalent way, so that one can measure the change in the result produced by replacing one program by the other one.

This idea has motivated much literature on program (pseudo)metrics [4, 41, 5, 19, 6, 13, 11, 14, 21], that is, on semantics in which types are endowed with a notion of distance measuring the differences in their behaviors. This approach has found widespread applications, for example in differential privacy [35, 3, 7], where one is interested in measuring the *sensitivity* of a program, *i.e.* its capacity to amplify changes in its inputs, and in the study of probabilistic processes [16, 43, 11, 42].

Recent literature [44, 32] has highlighted the importance of *contextuality* to reason about program similarity: many common situations require to measure the error produced by a transformation of the form $\mathbb{C}[t] \rightsquigarrow \mathbb{C}[u]$, which replaces a program t by u *within a context* $\mathbb{C}[\]$, as a function of the mismatch between t and u and of the sensitivity of the context $\mathbb{C}[\]$ itself. For instance, the error produced by replacing the program $\lambda x. \sin(x)$ by the identity function $\lambda x. x$ in a given context \mathbb{C} will be highly sensitive to how close to 0 these functions are



© G. Geoffroy and P. Pistone;
licensed under Creative Commons License CC-BY
42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:18



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

45 evaluated in \mathbb{C} . Similar cases of contextual reasoning can be found in many areas of computer
 46 science: for example in techniques from numerical analysis (*e.g.* the Gauss-Newton method),
 47 in which a computationally intensive function is replaced by its Taylor’s expansion around
 48 some given point, or in approximate computing techniques like *loop perforation* [38], in which
 49 a compiler can be asked to skip a certain number of iterations of a loop in a program.

50 **The Problem of Coupling Program Metrics with Higher-Order Types** While several
 51 frameworks for contextual reasoning have been developed in recent years [35, 20, 5, 44, 32],
 52 these approaches suggest that describing program similarity for a fully higher-order language
 53 in terms of program metrics still constitutes a major challenge.

54 In particular, when considering higher-order languages with a type Real for real numbers,
 55 it is not clear how to *lift* the standard metric on Real to higher-order types, *e.g.* to $\text{Real} \rightarrow \text{Real}$,
 56 so that the distances between higher-order programs are measured in a contextual way.

57 A standard solution is to take the sup-distance, that is, to let, for $f, g : \text{Real} \rightarrow \text{Real}$,
 58 $d(f, g) = \sup\{d(f(r), g(r)) \mid r \in \text{Real}\}$. This solution works well in models in which programs
 59 are interpreted as *non-expansive* or *Lipschitz-continuous* maps [25, 5]. However such models
 60 are not *cartesian-closed*¹, so they do not account for the simply-typed lambda-calculus
 61 in its full generality, but only for linear or sub-exponential variations of it (such as Fuzz
 62 [35, 20, 5]). Also, it has been shown [13] that in a probabilistic setting the non-linearity of
 63 higher-order programs has the effect of *trivialising* metrics, that is, of forcing distances to be
 64 either 0 or 1, hence collapsing program distances onto usual notions of program equivalence.
 65 Most importantly, even if one restricts to a sub-exponential language, the sup-distance is
 66 inadequate to account for contextual transformations as the replacement of $\lambda x. \sin(x)$ by
 67 $\lambda x. x$ around 0, as the sup-distance between these two programs is infinite (see Fig. 3).

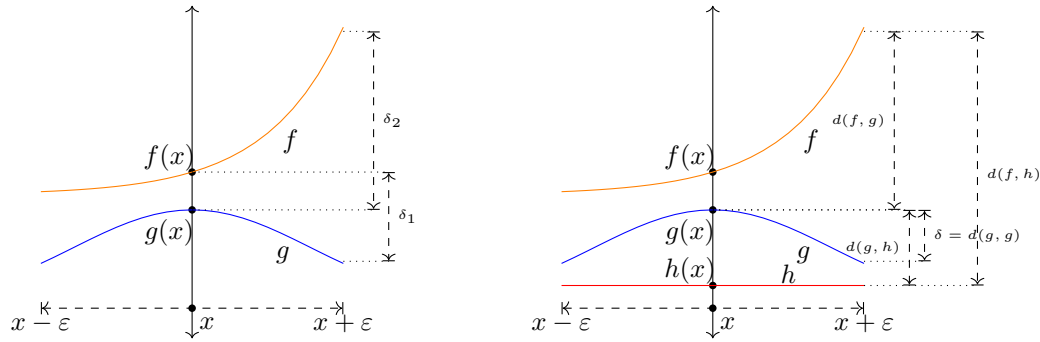
68 On the other side of the coin, other approaches like [44, 32] are fully contextual and
 69 higher-order, but provide, at best, only weak approximations of a standard notion of metric.
 70 Nonetheless, these approaches introduce the idea, which we retain here, that program
 71 differences must be taken as being themselves some kind of programs, relating errors in input
 72 with errors in output, and that accordingly, programs should be split in two different classes:
 73 *exact* programs, computing mappings from well-defined inputs to well-defined outputs, and
 74 *approximate* programs, mapping errors in the input to errors in the output.

75 **Diameter Spaces** In this paper we introduce a new semantic framework to reason about
 76 program similarity and approximate program transformations based on a class of higher-order
 77 denotational models that we call *diameter space models*. Compared to existing higher-order
 78 frameworks, the main novelty of these models is that program similarities are measured by
 79 associating each simple type with a *generalized partial metric space*, yielding a lifting of the
 80 standard metric on Real to higher-order types.

81 Generalized partial metric spaces are a well-investigated class of metric spaces that has
 82 been widely applied in program semantics [8, 9, 33, 37, 36, 26, 23]. Such spaces generalize
 83 standard metric spaces in that distances need not be real numbers, but can be functions or
 84 any other type of object that lives in a suitable *quantale* [25], and *self-distances* $d(x, x)$ need
 85 not be 0 (which leads to a stronger triangular inequality: $d(x, y) + d(z, z) \leq d(x, z) + d(z, y)$).

86 In our models a higher-order type A is interpreted as a 4-tuple $(|A|, \llbracket A \rrbracket, \langle A \rangle, \delta_A)$ called
 87 a *diameter space*, where $|A|$ is a set of *exact* values, $\llbracket A \rrbracket \subset \mathcal{P}(|A|)$ is a complete lattice of

¹ In fact, cartesian closed categories of metric spaces and non-expansive functions *do* exist [19, 12], but, to our knowledge, none of these categories contains the real numbers with the standard metric.



(a) In differential logical relations the distance between two functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, computed at (x, ε) is the maximum between $\delta_1 = \max\{d(f(x), g(y)); y \in [x-\varepsilon, x+\varepsilon]\}$ and $\delta_2 = \max\{d(g(x), f(y)); y \in [x-\varepsilon, x+\varepsilon]\}$.

(b) The distance arising from differential logical relations is not a partial metric: the example above shows that $d(f, h) > d(f, g) + d(g, h) - d(g, g)$ (with all distances computed at (x, ε)).

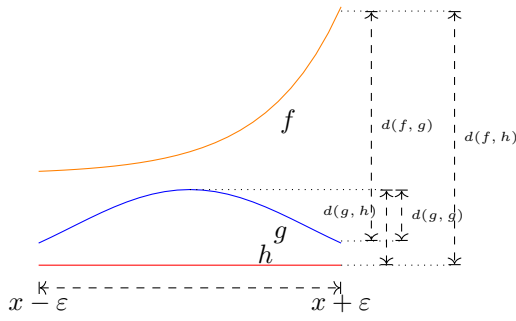
Figure 1 Differential logical relations do not yield partial metrics.

approximate values, $\langle A \rangle$ is a quantale, and $\delta_A : \llbracket A \rrbracket \rightarrow \langle A \rangle$ is a function, called *diameter*, which provides a quantitative measure of approximate values. The map δ_A generalizes some properties of the diameter function of the standard metric on real numbers. In particular, just like the distance between two real numbers can be described as the diameter of the smallest interval containing them, the map δ_A yields a generalized partial metric $d_A : |A| \times |A| \rightarrow \langle A \rangle$ in which the distance between two exact values of A is measured as the diameter of the smallest approximate value containing them, i.e. $d_A(x, y) = \delta_A(x \vee y)$.

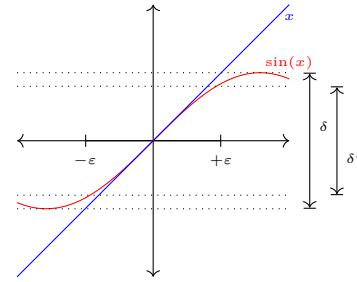
Measuring Distances between Programs of Functional Type A primary source of inspiration for our approach is the recent work by Dal Lago, Gavazzo and Yoshimizu on *differential logical relations* [32]. This is a semantical framework for higher-order languages in which a type is interpreted as a set X endowed with a kind of metric structure expressed by a ternary relation $\rho \subseteq X \times Q \times X$, where Q is an arbitrary quantale. To our knowledge, this is the first place where the idea of varying the quantales in which distances are measured is introduced as a key ingredient to obtain a cartesian closed category.

However, although such a relation ρ induces a distance function $d_\rho(x, y) = \sup\{\varepsilon \mid \rho(x, \varepsilon, y)\}$, this function is not a (partial) metric. We can show this fact with a simple example: in this model the distance between two programs $f, g : \mathbb{Real} \rightarrow \mathbb{Real}$ is taken in the quantale of functions from $\mathbb{R} \times \mathbb{R}_+^\infty$ to \mathbb{R}_+^∞ : intuitively, $d(f, g)$ associates a closed interval $[x - \varepsilon, x + \varepsilon]$ (corresponding to the pair (x, ε)) with the smallest distance δ such that $[f(x) - \delta, f(x) + \delta]$ and $[g(x) - \delta, g(x) + \delta]$ both contain the images of $[x - \varepsilon, x + \varepsilon]$ through g and f respectively (see Fig. 1a). Then, as shown in Fig. 1b, by letting $\delta = d(g, g)(x, \varepsilon)$, we have that $d(g, g)$ sends the interval $I = [x - \varepsilon, x + \varepsilon]$ onto the interval $[g(x) - \delta, g(x) + \delta]$, which has diameter 2δ , while the image of I has diameter δ , making the triangular law of partial metrics fail.

By contrast, in our model, the distance between two programs $f, g : \mathbb{Real} \rightarrow \mathbb{Real}$ lives in the quantale of monotone maps from *approximate values* of \mathbb{Real} (i.e. closed intervals) to positive reals. More precisely, this distance is the function that maps a closed interval a to the diameter of the smallest interval containing *both* $f(a)$ and $g(a)$. This notion of distance does satisfy all the axioms of a partial metric, as illustrated in Fig. 2. Observe that we no



■ **Figure 2** Our new metric is a partial metric: in the example above it can be seen that $d(f, h) \leq d(f, g) + d(g, h) - d(g, g)$ (with all distances computed in the interval $[x - \varepsilon, x + \varepsilon]$).



■ **Figure 3** The self-distances δ, δ' of $\sin(x)$ and x in a small interval $[-\varepsilon, \varepsilon]$ of 0 are very close.

117 longer depict the “center” of the interval $[x - \varepsilon, x + \varepsilon]$, and that the triangular inequality
 118 works because in summing $d(f, g)$ and $d(g, h)$ the self-distance $d(g, g)$ is counted twice.

119 Note that the distance of f from *itself*, which needs not be (constantly) 0, provides a
 120 measure of the sensitivity of f , since it associates each interval a with the size of the interval
 121 $f(a)$ spanned by f on a (a similar feature is present in differential logical relations).

122 The use of partial metrics with functional distances yields a rich and expressive framework
 123 to reason about contextual transformations. For instance, we can express the closeness of
 124 $\lambda x. \sin(x)$ and $\lambda x.x$ around 0 by the fact that their distance, applied to a small interval $[-\varepsilon, \varepsilon]$
 125 around 0, is very close to the self-distance of $\lambda x. \sin(x)$ on the same interval (as illustrated
 126 in Fig. 3). Moreover, the triangular inequality of partial metrics can be used to infer new
 127 bounds from previously established ones in a compositional way.

128 **Diameter Space over a Cartesian Closed Category** Our approach was devised primarily to
 129 account for transformations in higher-order languages designed for real analysis computation
 130 (like *e.g.* Real PCF [18]). However, diameter spaces can be constructed starting from any
 131 higher-order programming language with a reasonable denotational semantics. In fact, for *any*
 132 cartesian closed category \mathbb{C} , we can construct a cartesian *lax*-closed category $\text{Diam}(\mathbb{C})$, whose
 133 morphisms can be seen as approximate versions of the morphisms of \mathbb{C} . The “lax” preservation
 134 of the cartesian closed structure reflects the fact that, by composing approximations in a
 135 higher-order setting, also their error rates compose (typically, approximating non β -normal
 136 λ -terms will lead to higher error-rates than approximating their β -normal forms).

137 The generality of our construction shows in particular that our partial metric semantics
 138 requires no restrictions (*e.g.* Lipschitz-continuity) on morphisms, and adapts well to the
 139 model one starts with: for instance, the category $\text{Diam}(\text{Set})$ contains a partial metric on
 140 the set of *all* set-theoretic functions from \mathbb{R} to \mathbb{R} , while the categories $\text{Diam}(\text{Eff})$ (where Eff
 141 is the *effective topos* [27]) and $\text{Diam}(\text{Scott})$ show that our approach scales well to a more
 142 computability-minded setting.

143 **2 Generalized Partial Metric Spaces**

144 Partial metric spaces were introduced in the early nineties as a variant of metric spaces in
 145 which self-distances can be non-zero. Such spaces have attracted much attention in program
 146 semantics [8, 9, 33, 37, 36, 26, 23], due to their compatibility with standard constructions
 147 from both domain theory (since their topology is T_0) and usual metric topology (*e.g.* Cauchy
 148 sequences, completeness, Banach-fixed point theorem) [8, 33]. *Generalized partial metric*

149 spaces, *i.e.* partial metric spaces whose metric takes values over an arbitrary quantale [25],
 150 are well-investigated too [29, 28].

151 In this paper we will only be concerned with partial metrics taking values over a *commu-*
 152 *tative integral* quantale [25], of which we recall the definition below.

153 ► **Definition 1.** A commutative integral quantale is a triple $(Q, +, \leq)$ where:

- 154 ■ (Q, \leq) is a complete lattice,
- 155 ■ $(Q, +)$ is a commutative monoid,
- 156 ■ $+$ commutes with arbitrary infs,
- 157 ■ the least element of Q is neutral for $+$.

158 For readability, we have we have reversed the ordering with respect to the conventional
 159 definition, so that for example, $([0, \infty], +, \leq)$ is a commutative integral quantale whose least
 160 element is 0 (as opposed to “ $([0, \infty], +, \geq)$ is a commutative integral quantale whose *largest*
 161 element is 0”, which is what we would get with the usual definition). It is straightforward to
 162 check that for all commutative integral quantales Q, R , the product monoid $Q \times R$ equipped
 163 with the product ordering is also a commutative integral quantale. In addition, for all posets
 164 X , the set of monotone functions from X to Q , equipped with the pointwise monoid operation
 165 and the pointwise ordering, is also a commutative integral quantale. Another example of
 166 commutative integral quantale is given by the lattice of ideals of any commutative ring, with
 167 the product of ideals as the monoid operation.

168 We recall now the definition of a generalized partial metric space:

169 ► **Definition 2.** A generalized partial metric space (*in short, GPMS*) is the data of a set X ,
 170 a commutative integral quantale Q and a function $d : X \times X \rightarrow Q$ such that:

- 171 ■ for all $x, y \in X$, $d(x, x) \leq d(x, y)$,
- 172 ■ for all $x, y \in X$, if $d(x, x) = d(x, y) = d(y, y)$, then $x = y$,
- 173 ■ for all $x, y \in X$, $d(x, y) = d(y, x)$,
- 174 ■ for all $x, y, z \in X$, $d(x, z) + d(y, y) \leq d(x, y) + d(y, z)$.

175 For every metric space (X, d) , the structure $(X, ([0, \infty], +, \leq), d)$ is a GPMS. As is
 176 well-known [8], any real-valued GPMS $(X, [0, \infty], d)$ induces a metric d^* by letting

$$177 \quad d^*(x, y) = 2d(x, y) - d(x, x) - d(y, y) \quad (\star)$$

178 For a more telling and somewhat archetypal example, take any set X and consider the set
 179 $X^{\leq \omega}$ of all sequences of elements of X indexed by an ordinal less than or equal to ω . For all
 180 such sequences s, t , let $d(s, t) = 2^{-n} \in [0, \infty]$, where n is the length of the largest common
 181 prefix to s and t : one can check that $(X^{\leq \omega}, [0, \infty], d)$ is indeed a generalized partial metric
 182 space. In fact, if we interpret the prefixes of a sequence as pieces of partial information,
 183 then we have $d(s, s) = d(s, t)$ if and only if t is a refinement of s (*i.e.* if it contains more
 184 information), and $d(s, s) = 0$ if and only if s is total (*i.e.* if it cannot be refined).

185 One can check that for all partial metric spaces (X, Q, d_X) and (Y, R, d_Y) , $(X \times$
 186 $Y, Q \times R, d_{X \times Y})$ is a generalized partial metric space, where $d_{X \times Y}((x_1, y_1), (x_2, y_2)) =$
 187 $(d_X(x_1, x_2), d_Y(y_1, y_2))$. However, in general, it is not clear how one should define a partial
 188 metric on a function space. In Section 3.2 we introduce a construction to obtain partial
 189 metric spaces on function spaces by generalizing some properties of the standard diameter
 190 function on sets of real numbers.

191 **3** Approximate Programs for the Simply-Typed λ -Calculus over Real

192 To illustrate our construction, we start from a relatively concrete example: we consider a
 193 simply-typed lambda calculus with a base type `Real` and primitives for real numbers, and we
 194 follow the plan outlined in the introduction, which yields for each simple type a notion of
 195 approximate value, approximate function, diameter and distance between programs. Most
 196 definitions are straightforward and intuitive: the interesting, not immediately obvious point
 197 is that our construction does yield a *partial metric* on each type.

198 *Simple types* are defined as follows: `Real` is a simple type; if A and B are simple
 199 types, then $A \rightarrow B$ and $A \times B$ are simple types. For all $n > 0$, we fix a set \mathcal{F}_n of
 200 functions from \mathbb{R}^n to \mathbb{R} . We consider the usual Curry-style simply-typed λ -calculus over
 201 the types defined above (the left and right projection are denoted by $\pi_L : A \times B \rightarrow A$ and
 202 $\pi_R : A \times B \rightarrow B$ respectively, and the constructor for pairs by $\langle -, - \rangle$), enriched with the
 203 following constants: for all $r \in \mathbb{R}$, a constant $r : \text{Real}$; for all $n > 0$ and all $f \in \mathcal{F}_n$, a constant
 204 $f : \text{Real} \rightarrow \dots \rightarrow \text{Real} \rightarrow \text{Real}$. We call this calculus $\text{ST}\lambda\text{C}(\mathcal{F}_n)$, and its terms are simply
 205 called *terms*. We write $t[x_1 := u_1, \dots, x_n := u_n]$ to denote the *simultaneous* substitution
 206 of u_1, \dots, u_n for x_1, \dots, x_n in t . For all types A , we denote by Λ_A the set of closed terms
 207 of type A . The relation of β -reduction is enriched with the following rule, extended to all
 208 contexts: for all $n > 0$, $f \in \mathcal{F}_n$, and $r_1, \dots, r_n \in \mathbb{R}$, $f r_1 \dots r_n \rightarrow_\beta s$, where $s = f(r_1, \dots, r_n)$.
 209 By standard arguments [1], this calculus has the properties of subject reduction, confluence
 210 and strong normalisation.

211 **► Remark 3.** The class of real-valued functions which can be computed in $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ depends
 212 on the choice we make for \mathcal{F}_n . With suitable choices (see for instance [40, 17, 18]) one can
 213 obtain that all programs of type $\text{Real} \rightarrow \text{Real}$ compute *continuous* functions², that all such
 214 programs are *integrable* over closed intervals, or that all such programs are *continuously*
 215 *differentiable*.

216 In addition to the usual notion of β -equivalence between terms of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$, we will
 217 exploit also a stronger equivalence: given two closed terms t, u of type A , we say that t and u
 218 are *observationally equivalent* and write $t \approx_A u$ if for all terms C such that $x : A \vdash C : \text{Real}$
 219 is derivable, $C[x := t]$ is β -equivalent to $C[x := u]$ (which amounts to saying that they both
 220 β -reduce to the same real number). It is clear that observational equivalence is a congruence
 221 and that two β -equivalent terms are always observationally equivalent.

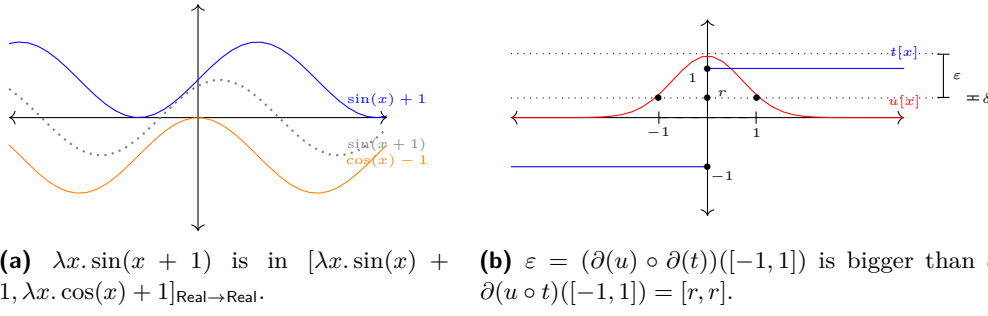
222 **3.1** Approximate Values and Approximate Programs

223 The first step of our construction for $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ is to associate to each simple type A a set
 224 $\llbracket A \rrbracket$ whose elements are certain sets of programs of type A that we call *approximate values of*
 225 *type* A . A closed term $t \in \Lambda_A$ represents a program with return type A and no parameters,
 226 so an approximate value can be thought of as a specification of a program with return type
 227 A and no parameters *up to* a certain degree of error or approximation.

228 For each simple type A , the set of approximate values $\llbracket A \rrbracket \subseteq \mathcal{P}(\Lambda_A)$ is defined inductively
 229 as follows:

- 230 ■ $\llbracket \text{Real} \rrbracket = \{ \{ t \in \Lambda_{\text{Real}} \mid \exists r \in I, t \rightarrow_\beta^* r \} \mid I \subseteq \mathbb{R} \text{ is a compact interval or } \emptyset \text{ or } \mathbb{R} \},$
- 231 ■ $\llbracket A \times B \rrbracket = \{ a \times b \mid a \in \llbracket A \rrbracket, b \in \llbracket B \rrbracket \},$ where $a \times b = \{ t \in \Lambda_{A \times B} \mid \pi_L t \in a \text{ and } \pi_R t \in b \},$
- 232 ■ $\llbracket A \rightarrow B \rrbracket = \{ \{ t \in \Lambda_{A \rightarrow B} \mid \forall u \in \Lambda_A, tu \in I(u) \} \mid I : \Lambda_A \rightarrow \llbracket B \rrbracket \}.$

² Note that for this to be possible, \mathcal{F}_n cannot contain the identity function over `Real`.



■ **Figure 4** Examples of functional approximate values and of approximate programs.

233 The approximate values of type `Real` are sets of closed programs of type `Real` which
 234 essentially coincide with the compact intervals of \mathbb{R} , plus the empty set and \mathbb{R} itself. An
 235 approximate value in $\llbracket A \times B \rrbracket$ is a “rectangle” $a \times b$, with $a \in \llbracket A \rrbracket$ and $b \in \llbracket B \rrbracket$, while an
 236 approximate value in $\llbracket A \rightarrow B \rrbracket$ is uniquely determined by a function I from closed terms
 237 $u \in \Lambda_A$ to approximate values $I(u) \in \llbracket B \rrbracket$.

238 For example, any two terms $t, u \in \Lambda_{\text{Real}}$ with normal forms $q, r \in \mathbb{R}$ induce an approximate
 239 value $[t, u]_{\text{Real}} = \{v \in \Lambda_{\text{Real}} \mid v \rightarrow_{\beta}^* s \wedge (q \leq s \leq r \vee q \geq s \geq r)\}$ of type `Real`. Similarly, any
 240 two terms $t, u \in \Lambda_{\text{Real} \rightarrow \text{Real}}$ induce an approximate value $[t, u]_{\text{Real} \rightarrow \text{Real}} = \{v \in \Lambda_{\text{Real} \rightarrow \text{Real}} \mid$
 241 $\forall r \in \Lambda_{\text{Real}} \ v r \in [tr, ur]_{\text{Real}}\}$. For instance, if $t = \lambda x. \sin(x) + 1$ and $u = \lambda x. \cos(x) - 1$, then
 242 $[t, u]_{\text{Real} \rightarrow \text{Real}}$ contains all closed terms corresponding to maps oscillating between $\sin(x) + 1$
 243 and $\cos(x) + 1$ (e.g. the program $\lambda x. \sin(x + 1)$, as illustrated in Fig. 4a).

244 For all A , the set $\llbracket A \rrbracket$ is a subset of $\mathcal{P}(\Lambda_A)$ closed under arbitrary intersections. We
 245 deduce that $\llbracket A \rrbracket$ has arbitrary meets (given by intersections) and arbitrary joins $\bigvee_{i \in I} a_i =$
 246 $\bigcap \{a \in \llbracket A \rrbracket \mid \forall i \in I \ a_i \subseteq a\}$, and thus $\llbracket A \rrbracket$ is a complete lattice. In particular, for all $t \in \Lambda_A$,
 247 there is a least element of $\llbracket A \rrbracket$ that contains t , which will be denoted by \bar{t} . One can check
 248 that $\bar{t} = \bar{u}$ if and only if $t \approx_A u$.

249 Monotone functions from approximate values to approximate values represent *approximate*
 250 *programs*. They behave like a model of the simply-typed λ -calculus in a weak sense, namely:

- 251 ■ for all monotone functions $\vec{\alpha} \mapsto c[\vec{\alpha}] : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket B \rightarrow C \rrbracket$ and $\vec{\alpha} \mapsto b[\vec{\alpha}] :$
 252 $\llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket B \rrbracket$, we can define a monotone function $\vec{\alpha} \mapsto (c[\vec{\alpha}] \ b[\vec{\alpha}]) = \sup\{\bar{v}\bar{u} \mid$
 253 $v \in c[\vec{\alpha}], u \in b[\vec{\alpha}]\} : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket C \rrbracket$,
- 254 ■ for all monotone functions $\vec{\alpha} \mapsto c[\vec{\alpha}] : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket C \rrbracket$ and all $i \leq n$, we can
 255 define a monotone function $(\alpha_j)_{j \neq i} \mapsto (\lambda \alpha_i. c[\vec{\alpha}]) = \{v \in \Lambda_{A_i \rightarrow C} \mid \forall t_i \in \Lambda_{A_i}, \ v t_i \in$
 256 $c[\alpha_1, \dots, \bar{t}_i, \dots, \alpha_n]\} : \prod_{j \neq i} \llbracket A_j \rrbracket \rightarrow \llbracket A_i \rightarrow C \rrbracket$,

257 and these two constructions are weakly compatible with β -reduction and η -expansion:

258 ► **Proposition 4.** For all monotone functions $(\vec{\alpha}, \beta) \mapsto c[\vec{\alpha}, \beta] : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \times \llbracket B \rrbracket \rightarrow \llbracket C \rrbracket$
 259 and $\vec{\alpha} \mapsto b[\vec{\alpha}] : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket B \rrbracket$, $(\vec{\alpha} \mapsto (\lambda \beta. c[\vec{\alpha}, \beta]) \ b[\vec{\alpha}]) \leq (\vec{\alpha} \mapsto c[\vec{\alpha}, b[\vec{\alpha}]])$, and
 260 for all monotone functions $\vec{\alpha} \mapsto d[\vec{\alpha}] : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket B \rightarrow C \rrbracket$, $(\vec{\alpha} \mapsto \lambda \beta. d[\vec{\alpha}] \ \beta) \geq$
 261 $(\vec{\alpha} \mapsto d[\vec{\alpha}])$, where functions are ordered by pointwise inclusion. In other words, on approxi-
 262 mate programs, β -reduction and η -expansion discard information, and conversely β -expansion
 263 and η -reduction recover some information.

264 **Proof.** Without loss of generality, we can assume $n = 0$. Let $v \in \lambda \beta. c[\beta]$ and $u \in b$. By
 265 definition, $tu \in c[\bar{u}]$, so $\bar{t}u \subseteq c[\bar{u}] \subseteq c[b]$. Therefore, $(\lambda \beta. c[\beta]) \ b \subseteq b$. Let $v \in d$. For all
 266 $u \in \Lambda_B$, by definition, $vu \in d\bar{u}$. Therefore, $v \in \lambda \beta. d \ \beta$. ◀

267 Beyond theoretical aspects (which will be made clearer in Section 5) Proposition 4 is also

important in practice because it implies that if we compute an approximation of a program from approximations of its parts and then simplify the resulting approximate program using β -reduction and η -expansion, what we obtain is still a valid approximation of the original program.

We can define a weak embedding from terms into approximate programs, by mapping each term to its tightest approximation: for all terms t such that $\alpha_1 : A_1, \dots, \alpha_n : A_n \vdash t : B$, we define a monotone function $\partial(t) : \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket \rightarrow \llbracket B \rrbracket$ by $\partial(t)(a_1, \dots, a_n) = \sup\{\overline{tu_1 \dots u_n} \mid u_1 \in a_1, \dots, u_n \in a_n\}$.

► Remark 5. The map ∂ is constant on classes of observational equivalence, and one can check that it is weakly compatible with the constructions of the λ -calculus, in particular:

- $\partial(\alpha_i)(a_1, \dots, a_n) = a_i$,
- $\partial(tu)(a_1, \dots, a_n) \subseteq \partial(t)(a_1, \dots, a_n) \partial(u)(a_1, \dots, a_n)$,
- $\partial(\lambda\beta.t)(a_1, \dots, a_n) \subseteq \lambda\beta. \partial(t)(\beta, a_1, \dots, a_n)$.

This map $\partial(t)$ can be taken as a measure of the *sensitivity* of t , as it maps an interval a , that is a quantifiably uncertain input, to a quantifiably uncertain output $\partial(t)(a)$. For instance, if we take the term $t[x] = \sin(x) + 1$ above, then $\partial(t) : \llbracket \text{Real} \rrbracket \rightarrow \llbracket \text{Real} \rrbracket$ sends the interval $[-\pi, \pi]_{\text{Real}}$ into $[0, 2]_{\text{Real}}$.

► Remark 6. When composing two maps $\partial(t)$ and $\partial(u)$, we might obtain a worse approximation than by computing $\partial(t[u/x])$ directly. For instance, let $t[x]$ and $u[x]$ be, respectively, the discontinuous and Gaussian functions illustrated in Fig. 4b. If a is the interval $[-1, +1]$, then $\partial(t)(a) = [-1, 1]$, and since $u[x := -1] = u[x := 1] \simeq_\beta r$ for some $0 < r < 1$, we deduce that $\partial(u)(\partial(t)(a)) = [-1, 1] \not\supseteq [r, r] = \partial(u[t/x])(a)$.

3.2 A Partial Metric on Each Type

So far, we have associated each type A of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ with a complete lattice $\llbracket A \rrbracket \subseteq \mathcal{P}(\Lambda_A)$ of approximate values of type A , and each typed program $t : A \rightarrow B$ with an approximate program $\partial(t)$ (in fact, a monotone function) from approximate values of type A to approximate values of type B . We will now exploit this structure to define, for each type A of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$, a generalized partial metric on the closed (exact) programs of type A .

The first step is to define, for every simple type A , a commutative integral quantale $(\llbracket A \rrbracket, \leq_A, +_A)$ of *distances of type A*:

- $(\llbracket \text{Real} \rrbracket, \leq_{\text{Real}}, +_{\text{Real}}) = ([0, \infty], \leq, +)$,
- $\llbracket A \times B \rrbracket = \llbracket A \rrbracket \times \llbracket B \rrbracket$,
- $\llbracket A \rightarrow B \rrbracket = \text{Poset}(\llbracket A \rrbracket, \llbracket B \rrbracket)$.

where, for two posets Q, R , $\text{Poset}(Q, R)$ denotes the set of monotone functions from Q to R . Observe that the quantale $\llbracket A \rightarrow B \rrbracket$ is a set of functions over the approximate values of A .

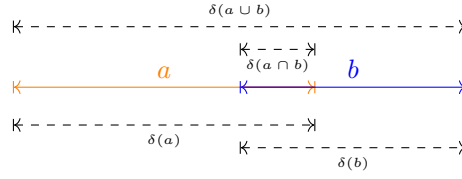
For all simple types A , we now define a *distance function* $d_A : \Lambda_A \times \Lambda_A \rightarrow \llbracket A \rrbracket$:

- $d_{\text{Real}}(t, u) = |r - s|$, where r, s are the unique elements of \mathbb{R} such that $t \rightarrow_\beta^* r$ and $u \rightarrow_\beta^* s$,
- $d_{A \times B}(t, u) = (d_A(\pi_L t, \pi_L u), d_B(\pi_R t, \pi_R u))$,
- $d_{A \rightarrow B}(t, u) = a \mapsto \sup\{d_B(rv, sw) \mid r, s \in \{t, u\}, v, w \in a\}$.

It would be tempting to define $d_{A \rightarrow B}(t, u)(a)$ simply as $\sup\{d_B(tv, uw) \mid v, w \in a\}$, but then the axiom “ $d_{A \rightarrow B}(t, t) \leq d_{A \rightarrow B}(t, u)$ ” of partial metric spaces would fail.

The maps d_A are clearly compatible with observational equivalence (*i.e.* if $a \approx_A a'$ and $b \approx_A b'$, then $d_A(a, b) = d_A(a', b')$).

Our objective is now to prove that $(\Lambda_A / \approx_A, \llbracket A \rrbracket, d_A)$ is a generalized partial metric space. To this end, we define for all simple types A a monotone *diameter function* $\delta_A : \llbracket A \rrbracket \rightarrow \llbracket A \rrbracket$



■ **Figure 5** The diameter function is *modular* over intersecting real intervals: $\text{diam}(a \cup b) + \text{diam}(a \cap b) = \text{diam}(a) + \text{diam}(b)$ for all $a, b \in \mathbb{R}$ such that $a \cap b \neq \emptyset$. This property is at the heart of our generalization of diameters. Observe that this property fails when $a \cap b$ is empty.

313 by $\delta_A(a) = \sup\{d_A(t, u) \mid t, u \in a\}$. The key to our objective will be to prove that δ_A is *sub-*
 314 *modular* on intersecting approximate values (henceforth, *quasi-sub-modular* – see Proposition
 315 7): this generalizes the fact that, on the (real-valued) metric space \mathbb{R} , the diameter is *modular*
 316 over intersecting closed intervals (see Fig. 5).

317 First, one can check that for all $t, u \in \Lambda_A$, $\delta_A(\bar{t} \vee \bar{u}) = d_A(t, u)$, and that:

- 318 ■ $\delta_{\text{Real}}(a) = \sup\{s - r \mid s, r \in \mathbb{R} \text{ such that } s, r \in a\}$,
- 319 ■ $\delta_{A \times B}(p) = (\delta_A(\sup\{\overline{\pi_L t} \mid t \in p\}) \delta_B(\sup\{\overline{\pi_R t} \mid t \in p\}))$,
- 320 ■ $\delta_{A \rightarrow B}(b) = a \mapsto \delta_B(\sup\{\overline{vt} \mid t \in a, v \in b\})$.

321 This leads then to the following:

322 ► **Proposition 7** (δ_A is quasi-sub-modular). *For all simple types A and all $a, b \in \llbracket A \rrbracket$ such*
 323 *that $a \wedge b \neq \emptyset$, $\delta(a \wedge b) + \delta(a \vee b) \leq \delta(a) + \delta(b)$.*

324 **Proof.** We proceed by induction on types.

325 Let $a, b \in \llbracket \text{Real} \rrbracket$ such that $a \wedge b \neq \emptyset$. Let $I = \{r \in \mathbb{R} \mid r \in a\}$ and $J = \{s \in \mathbb{R} \mid s \in b\}$:
 326 then I (respectively, J , $I \cap J$, $I \cup J$) is either \mathbb{R} or a non-empty compact interval of \mathbb{R} ,
 327 and its length in the usual sense is equal to $\delta_{\text{Real}}(a)$ (respectively, $\delta_{\text{Real}}(b)$, $\delta_{\text{Real}}(a \wedge b)$,
 328 $\delta_{\text{Real}}(a \vee b)$). Note that the only reason we know that $I \cup J$ is an interval is because
 329 $a \wedge b \neq \emptyset$ implies $I \cap J \neq \emptyset$. The length of an interval of \mathbb{R} is equal to its Lebesgue measure,
 330 therefore $\text{length}(I \cap J) + \text{length}(I \cup J) = \text{length}(I) + \text{length}(J)$, so $\delta_{\text{Real}}(a \wedge b) + \delta_{\text{Real}}(a \vee b) =$
 331 $\delta_{\text{Real}}(a) + \delta_{\text{Real}}(b)$.

332 Let $a, b \in \llbracket A_L \times A_R \rrbracket$ such that $a \wedge b \neq \emptyset$. For all $c \in \llbracket A_L \times A_R \rrbracket$, let $c_L = \sup\{\overline{\pi_L t} \mid t \in c\}$
 333 and $c_R = \sup\{\overline{\pi_R t} \mid t \in c\}$. One can check that $(a \wedge b)_L = a_L \wedge b_L$, $(a \wedge b)_R = a_R \wedge b_R$,
 334 $(a \vee b)_L = a_L \vee b_L$ and $(a \vee b)_R = a_R \vee b_R$, so $\delta(a \wedge b) + \delta(a \vee b) = (\delta(a_L \wedge b_L) + \delta(a_L \vee$
 335 $b_L), \delta(a_R \wedge b_R) + \delta(a_R \vee b_R)) \leq (\delta(a_L) + \delta(b_L), \delta(a_R) + \delta(b_R)) = \delta(a) + \delta(b)$.

336 Let $f, g \in \llbracket A \rightarrow B \rrbracket$ and $a \in \llbracket A \rrbracket$. For all $h \in \llbracket A \rightarrow B \rrbracket$, let $ha = \sup\{\overline{vt} \mid v \in h, t \in a\}$.
 337 One can check that $(f \wedge g)a \subseteq (fa) \wedge (ga)$ and $(f \vee g)a = (fa) \vee (ga)$. As a result,
 338 $(\delta(f \wedge g) + \delta(f \vee g))(a) \leq \delta((fa) \wedge (ga)) + \delta((fa) \vee (ga)) \leq \delta(fa) + \delta(ga) = (\delta(f) + \delta(g))(a)$. ◀

339 It is well-known [39] that any function $\delta : L \rightarrow [0, \infty]$ on a lattice L that is monotone
 340 and *sub-modular* induces a pseudo-metric $d : L \times L \rightarrow [0, \infty]$ by letting $d^*(a, b) = 2\delta(a \vee$
 341 $b) - \delta(a) - \delta(b)$. In fact, one can decompose this construction: first, one defines a partial
 342 pseudometric d on L by $d(a, b) = \delta(a \vee b)$, and then d^* is just the distance given by equation
 343 (\star): $d^*(a, b) = 2d(a, b) - d(a, a) - d(b, b)$. We can use this way of reasoning to establish that
 344 the maps d_A are indeed partial metrics:

345 ► **Corollary 8.** *For all simple types A , $(\Lambda_A / \approx_A, (\llbracket A \rrbracket), d_A)$ is a generalized partial metric*
 346 *space, that is to say:*

- 347 1. for all $t, u \in \Lambda_A$, $d_A(t, t) \leq d_A(t, u)$,
- 348 2. for all $t, u \in \Lambda_A$, if $d_A(t, t) = d_A(t, u) = d_A(u, u)$, then $t \approx_A u$,

- 349 3. for all $t, u \in \Lambda_A$, $d_A(t, u) = d_A(u, t)$,
 350 4. for all $t, u, v \in \Lambda_A$, $d_A(t, v) + d_A(u, u) \leq d_A(t, u) + d_A(u, v)$.

351 **Proof.** As mentioned above, for all $t, u \in \Lambda_A$, $d_A(t, u) = \delta_A(\bar{t} \vee \bar{u})$, which immediately gives
 352 point 3. Since δ_A is monotone and $\bar{t} \vee \bar{t} \leq \bar{t} \vee \bar{u}$, we also get point 1.

353 One can check (by induction on types) that the restriction of δ_A to the ideal generated
 354 by the \bar{t} (for $t \in \Lambda_A$) is *strictly* monotone. Therefore, if $d_A(t, t) = d_A(t, u) = d_A(u, u)$, *i.e.*
 355 $\delta_A(\bar{t}) = \delta_A(\bar{t} \vee \bar{u}) = \delta_A(\bar{u})$, then $\bar{t} = \bar{t} \vee \bar{u} = \bar{u}$, so $t \approx_A u$.

356 The triangular inequality is an immediate consequence of the quasi-sub-modularity of δ_A :
 357 $d(t, v) + d(u, u) = \delta(\bar{t} \vee \bar{v}) + \delta(\bar{u}) \leq \delta((\bar{t} \vee \bar{u}) \vee (\bar{u} \vee \bar{v})) + \delta((\bar{t} \vee \bar{u}) \wedge (\bar{u} \vee \bar{v})) \leq \delta(\bar{t} \vee \bar{u}) + \delta(\bar{u} \vee \bar{v}) =$
 358 $d(t, u) + d(u, v)$. \blacktriangleleft

359 4 Computing Program Distances using Partial Metrics

360 In the previous section we showed how to associate each simple type A with a partial metric
 361 d_A over the closed terms of type A . We now illustrate through a few basic examples how
 362 the higher-order and metric features of this semantics can be used to formalize contextual
 363 reasoning about program differences.

364 To make our examples more realistic, we will consider some natural extensions of
 365 $\text{ST}\lambda\text{C}(\mathcal{F}_n)$. It is not difficult to see that all constructions from Section 3 still work if
 366 we add to $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ some new base types. For example, we can add to our language a type
 367 Nat for natural numbers, indicating for each $n \in \mathbb{N}$, the corresponding normal forms of Nat
 368 as \mathbf{n} . A natural choice is to let $\llbracket \text{Nat} \rrbracket = \{\{t \mid \exists n \in a \ t \rightsquigarrow \mathbf{n}\} \mid a \text{ finite subset of } \mathbb{N} \text{ or } a = \mathbb{N}\}$,
 369 $\llbracket \text{Nat} \rrbracket = [0, \infty]$ and $d_{\text{Nat}}(t, u) = |n - m|$, where $t \rightarrow_{\beta}^* \mathbf{n}$ and $u \rightarrow_{\beta}^* \mathbf{m}$.

370 Moreover, our constructions scale well also to extensions of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ obtained by adding
 371 new program constructors, as soon as these do not compromise the existence and uniqueness
 372 of normal forms (since the fact that closed programs of type Real have a normal form plays
 373 an important role to define $\llbracket \text{Real} \rrbracket$). For instance, if we suppose that all programs of type
 374 $\text{Real} \rightarrow \text{Real}$ in $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ are either differentiable or integrable (see Remark 3), we can
 375 consider extension of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ with differential or integral operators, as in Real PCF [17, 18].

376 We start with a classical example from approximate computing that we adapt from [44].

377 **► Example 9 (Loop perforation).** We work in the extension of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ with a type Nat .
 378 We discuss a transformation that replaces a program t which performs n iterations by a
 379 program which only performs the iterations $0, k, 2k, 3k, \dots$, each repeated k times.

380 Suppose $t : (A \times A \rightarrow A) \rightarrow \text{Nat} \rightarrow (A \rightarrow A) \rightarrow A$, for $n \geq 1$, is a term such that
 381 $th\mathbf{n}f$ computes the n -times iteration of h as follows: $th0f = h\langle f0, f0 \rangle$ and $th(\mathbf{n} + 1)f =$
 382 $h\langle th\mathbf{n}f, f(\mathbf{n} + 1) \rangle$. Let $\text{Perf}^k(t)$, the k -th perforation of t , be the program $(\text{Perf}^k(t))h\mathbf{n}f =$
 383 $t(\lambda x. (h^{(k)}x)) \llbracket \mathbf{n} \rrbracket_k (\lambda x. f(x * \mathbf{k}))$, where $\llbracket \mathbf{n} \rrbracket_k$ indicates the least $m \leq \mathbf{n}$ such that m is divisible
 384 by k , and $x * \mathbf{k}$ is the multiplication of x by k .

385 To compute the distance $d_A(v_n, w_n)$ between $v_n = th\mathbf{n}f$ and its perforation $w_n =$
 386 $\text{Perf}^k(t)h\mathbf{n}f$ we can reason as follows:

- 387 i. v_n performs n -iterations while w_n performs $k \llbracket \mathbf{n} \rrbracket_k \leq n$ iterations, and we can compute
 388 $d_A(v_n, v_{(k \llbracket \mathbf{n} \rrbracket_k)})$ as the diameter of $\partial(t)\partial(h)(\llbracket k \llbracket \mathbf{n} \rrbracket_k, \mathbf{n} \rrbracket_{\text{Nat}})\partial(f)$.
 389 ii. If n is divisible by k , then for $i \leq n$, at the i -th iteration of v_n the function f is applied
 390 to \mathbf{i} , while at the i -th iteration of w_n , f is applied to $\llbracket \mathbf{i} \rrbracket_k$. Now, the error of replacing
 391 $f\mathbf{i}$ by $f \llbracket \mathbf{j} \rrbracket_k$, with \mathbf{i}, \mathbf{j} in some $a \in \llbracket \text{Nat} \rrbracket$, is accounted for by the approximate program
 392 $c[y] = \partial(f)(y - k)$, where $y - k = y \vee \{u - \mathbf{k} \mid u \in y\}$. We deduce then that $d_A(v_n, w_n)$ is
 393 bounded by the diameter of $\partial(t)\partial(h)\bar{\pi}(\lambda y. c[y])$.

394 iii. From the fact that $w_n = w_{(k \cdot \lfloor n \rfloor_k)}$ and the triangular inequality of the partial metric d_A
 395 we deduce $d_A(v_n, w_n) = d_A(v_n, w_{(k \cdot \lfloor n \rfloor_k)}) \leq d_A(v_n, v_{(k \cdot \lfloor n \rfloor_k)}) + d_A(v_{(k \cdot \lfloor n \rfloor_k)}, w_{(k \cdot \lfloor n \rfloor_k)}) -$
 396 $d_A(v_{(k \cdot \lfloor n \rfloor_k)}, v_{(k \cdot \lfloor n \rfloor_k)})$

397 From facts i.-iii. we deduce an explicit bound for $d_A(v_n, w_n)$ in terms of $\partial(t), \partial(f)$ and n :
 398 $d_A(v_n, w_n) \leq \delta_A(\partial(t)\partial(h)([k \lfloor n \rfloor_k, n]_{\text{Nat}})\partial(f)) + \delta_A(\partial(t)\partial(h)\bar{\pi}(\lambda y.\partial(f)(y - k))) - \delta_A(\partial(t)\partial(h)\bar{\pi}\partial(f))$.

399 We now show how the partial metric semantics can be used to reason about basic
 400 approximation techniques from numerical analysis.

401 ► **Example 10** (Taylor approximation). We assume that all programs of type $\text{Real} \rightarrow \text{Real}$ in
 402 $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ are differentiable and that for all n , program $t : \text{Real} \rightarrow \text{Real}$ and real number r ,
 403 we can define a term $T^n(t, r) : \text{Real} \rightarrow \text{Real}$ computing the n -th truncated Taylor polynomial
 404 of t at r . The distance $d_{\text{Real} \rightarrow \text{Real}}(t, T^n(t, 0))$ is the map associating an interval a with the
 405 diameter of the smallest interval containing the image of a under both t and $T^n(t, 0)$. This
 406 value will approximately converge to the self-distance of t when a is a small interval of 0,
 407 and will tend to diverge when a contains points which are far enough from 0.

408 For example, if t is the function $t = \lambda x. \sin(x)$, and a is an interval of 0, then using
 409 standard analytic reasoning we can compute a bound $d_{\text{Real} \rightarrow \text{Real}}(t, T^n(t, 0))(a) \leq \frac{\delta_{\text{Real}}(a)^{n+1}}{(n+1)!}$,
 410 which tends to 0 as the diameter of a tends to 0.

411 Observe that if, instead, we used the sup-distance $d_{\text{sup}}(t, u) = \sup\{d_{\text{Real}}(tr, ur) \mid r \in$
 412 $\Lambda_{\text{Real}}\}$, then we could not reason as above, since the sup-distance between $\lambda x. \sin(x)$ and its
 413 truncated Taylor polynomials is infinite.

414 ► **Example 11** (Integral approximation). We now assume that all functions in \mathcal{F}_n are integrable
 415 and that we have (see [18]) at our disposal a program $\lambda f x. I_{[0,x]}(f) : (\text{Real} \rightarrow \text{Real}) \rightarrow \text{Real} \rightarrow$
 416 Real such that $I_{[0,r]}(t)$ computes (a precise enough approximation of) the definite integral
 417 $\int_0^{|r|} tx \, dx$. In many contexts we might prefer to replace the expensive computation of
 418 $I_{[0,r]}(t)$ by the (more economical but less precise) computation of a finite Riemann sum
 419 $R_{[0,r]}^n(t) = \sum_{i=1}^n (tx_i) \cdot |r|/n$, where $x_i = i \cdot |r|/n$.

420 Suppose now that, in order to approximate the integral of some computationally expensive
 421 program t on $[0, r]$, we replace t by some more efficient program u which, over $[0, r]$, is very
 422 close to t . Let $\varepsilon_t(r)$ indicate the distance between the true integral of t over $[0, r]$ and $R_{[0,r]}^n(t)$,
 423 and moreover let $\eta_{t,u}(r)$ be the diameter of $\partial(t)([0, r]) \vee \partial(u)([0, r])$.

424 Using the metric structure of Real we can then bound the error we incur in by replacing
 425 the true integral of t with the Riemann sum of u . In fact, by standard calculation we can
 426 compute the bound $d_{\text{Real}}(R_{[0,r]}^n(t), R_{[0,r]}^n(u)) \leq d_{\text{Real} \rightarrow \text{Real}}(t, u)([0, r]) \cdot |r| = \eta_{t,u}(r) \cdot |r|$. Then,
 427 using the triangular inequality of the standard metric on Real we deduce

$$428 \quad d_{\text{Real}}(I_{[0,r]}(t), R_{[0,r]}^n(u)) \leq d_{\text{Real}}(I_{[0,r]}(t), R_{[0,r]}^n(t)) + d_{\text{Real}}(R_{[0,r]}(t), R_{[0,r]}^n(u))$$

$$429 \quad \leq \varepsilon_t(r) + \eta_{t,u}(r) \cdot |r|$$

431 Using the partial metric on $\text{Real} \rightarrow \text{Real}$, we can also derive a bound expressing how much
 432 the error above is *sensitive to changes of r* . First, using standard analytic techniques (under
 433 suitable assumptions for t and its derivatives) one can find a program $v : \text{Real} \rightarrow \text{Real}$ such
 434 that vr computes an upper bound for $\varepsilon_t(r)$. Then, using the triangular inequality of the

435 partial metric on $\text{Real} \rightarrow \text{Real}$ we deduce, for all interval a , the following bound:

$$\begin{aligned}
436 \quad & d_{\text{Real} \rightarrow \text{Real}}(\lambda x. I_{[0,x]}(t), \lambda x. R_{[0,x]}^n(u))(a) \\
437 \quad & \leq d_{\text{Real} \rightarrow \text{Real}}(\lambda x. I_{[0,x]}(t), \lambda x. R_{[0,x]}^n(t))(a) + d_{\text{Real} \rightarrow \text{Real}}(\lambda x. R_{[0,x]}(t), \lambda x. R_{[0,x]}^n(u))(a) \\
438 \quad & \quad - d_{\text{Real} \rightarrow \text{Real}}(\lambda x. R_{[0,x]}(t), \lambda x. R_{[0,x]}^n(t))(a) \\
439 \quad & \leq d_{\text{Real} \rightarrow \text{Real}}(v, v)(a) + (d_{\text{Real} \rightarrow \text{Real}}(t, u)(a) - d_{\text{Real} \rightarrow \text{Real}}(t, t)(a)) \cdot \delta_{\text{Real}}(a) \\
440
\end{aligned}$$

441 5 Diameter Space Models Over a Cartesian Closed Category

442 The examples from the last section relied on the fact that our partial metric semantics scales
443 well to extensions of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ with new base types and new program constructors. In this
444 section we justify this fact in more general terms. In fact, we show that the constructions
445 from Section 3 can be reproduced starting from any model of the simply-typed λ -calculus.

446 First, we need a suitable notion of model of the simply-typed λ -calculus to start with.
447 Traditionally, one uses *cartesian closed categories*: cartesian categories where, for all objects
448 A , the functor $A \times -$ has a right adjoint (the *exponential* functor). However, since many
449 usual examples are in fact *poset-enriched* categories (*e.g.* Scott domains and continuous
450 functions, coherent spaces and stable functions), and since any (locally small) category can
451 be poset-enriched by using equality as the ordering, we will consider instead *cartesian closed*
452 *poset-enriched categories*. To give a counterpart to Proposition 4, we also need a notion of
453 “weak” model of the simply-typed λ -calculus: since poset-enriched categories are a particular
454 case of 2-categories (with a unique 2-arrow from f to g if and only if $f \leq g$), we follow Hilken
455 [24] and consider cartesian categories where, for all objects A , the functor $A \times -$ has a lax
456 right adjoint (the *lax-exponential* functor).

457 Products and exponentials, when they exist, are necessarily unique up to unique iso-
458 morphism: thus, traditionally, a cartesian closed category is defined as a category in which
459 all finite products and exponentials exist, rather than a category *equipped* with products
460 and exponentials (*i.e.* it is a category with a given *property*, rather than a category with
461 additional *structure*). However, this is not the case for lax-exponentials, so for consistency
462 we will adopt the “structure” picture in both cases. Adapting Hilken’s definitions [24] to the
463 simpler case of poset-enriched categories, we obtain:

464 ► **Definition 12.** *Let $(\mathbb{C}, \times, 1)$ be a cartesian poset-enriched category. An exponential*
465 *(respectively, a lax-exponential) on \mathbb{C} is the data of a map exp from $\text{Ob}(\mathbb{C} \times \mathbb{C})$ to $\text{Ob}(\mathbb{C})$*
466 *and two families of monotone maps $(\text{ev}_{W,X,Y} : \mathbb{C}(W, \text{exp}(X, Y)) \rightarrow \mathbb{C}(W \times X, Y))$ and*
467 *$(\lambda_{W,X,Y} : \mathbb{C}(W \times X, Y) \rightarrow \mathbb{C}(W, \text{exp}(X, Y)))$ such that:*

- 468 ■ $\text{ev}_{W,X,Y}$ and $\lambda_{W,X,Y}$ are natural with respect to W ,
- 469 ■ for all $g \in \mathbb{C}(W \times X, Y)$, $\text{ev}(\lambda(g)) = g$ (respectively, $\text{ev}(\lambda(g)) \leq g$),
- 470 ■ for all $f \in \mathbb{C}(W, \text{exp}(X, Y))$, $f = \lambda(\text{ev}(f))$ (respectively, $f \leq \lambda(\text{ev}(f))$).

471 One can check that this definition makes exp a functor (respectively, a lax-functor)
472 from $\text{Ob}(\mathbb{C}^{\text{op}} \times \mathbb{C})$ to $\text{Ob}(\mathbb{C})$ (with $\text{exp}(f, g)$ defined as $\lambda(g \circ \text{ev}(\text{id}) \circ (\text{id} \times f))$). In addition,
473 this definition implies that ev and λ are natural, in the sense that $\text{ev}(\text{exp}(\alpha, \beta) \circ f \circ \gamma) =$
474 $\beta \circ \text{ev}(f) \circ (\gamma \times \alpha)$ and $\text{exp}(\alpha, \beta) \circ \lambda(g) \circ \gamma = \lambda(\beta \circ g \circ (\gamma \times \alpha))$ (respectively, lax-natural [24], in the
475 sense that $\text{ev}(\text{exp}(\alpha, \beta) \circ f \circ \gamma) \leq \beta \circ \text{ev}(f) \circ (\gamma \times \alpha)$ and $\text{exp}(\alpha, \beta) \circ \lambda(g) \circ \gamma \leq \lambda(\beta \circ g \circ (\gamma \times \alpha))$).

476 For the rest of this section, we fix a cartesian poset-enriched category $(\mathbb{C}, \times, 1)$ (we denote
477 by $\langle -, - \rangle$ the pairing transformation and by π_L and π_R the projections) and an exponential
478 $(\text{exp}, \text{ev}, \lambda)$ on \mathbb{C} . The morphisms of this category represent *exact programs*, so they play the
479 role of the terms from Section 3.

- 480 ► **Definition 13.** A \mathbb{C} -diameter space A is the data of
- 481 ■ an object $|A|$ of \mathbb{C} . The poset $\mathbb{C}(1, |A|)$ will be denoted by Λ_A ;
 - 482 ■ a set $\llbracket A \rrbracket$ of downwards-closed subsets of Λ_A that is closed under arbitrary intersections. In particular, $\llbracket A \rrbracket$ is a complete lattice whose meet is given by intersection, and for all
 - 483 $t \in \Lambda_A$, there is a least element of $\llbracket A \rrbracket$ that contains t , which will be denoted by \bar{t} ;
 - 484 ■ a commutative integral quantale $(\llbracket A \rrbracket, +, \leq)$;
 - 485 ■ a monotone function $\delta_A : \llbracket A \rrbracket \rightarrow \llbracket A \rrbracket$ such that

$$\forall a, b \in \llbracket A \rrbracket \text{ s.t. } a \wedge b \neq \emptyset, \delta(a \wedge b) + \delta(a \vee b) \leq \delta(a) + \delta(b),$$

486 and such that for all $t, u \in \Lambda_A$, if $\delta_A(\bar{t}) = \delta_A(\bar{t} \vee \bar{u})$, then $\bar{t} = \bar{t} \vee \bar{u}$.

487 The role of the condition $a \wedge b \neq \emptyset$ is illustrated by Fig. 5.

- 488 ► **Example 14.** If \mathbb{C} is the category whose objects are the simple types from Section 3 and
- 489 whose morphisms are the (open) terms modulo β -equivalence, then for all simple types A ,
- 490 $(A, \llbracket A \rrbracket, \llbracket A \rrbracket, \delta_A)$ defines a \mathbb{C} -diameter space.

491 Following Section 3, for all \mathbb{C} -diameter spaces A and B , we define a \mathbb{C} -diameter space $A \times B$

492 such that $|A \times B| = |A| \times |B|$ and a \mathbb{C} -diameter space $\exp(A, B)$ such that $|\exp(A, B)| =$

493 $\exp(|A|, |B|)$:

- 494 ■ $\llbracket A \times B \rrbracket = \{a \times b \mid a \in \llbracket A \rrbracket, b \in \llbracket B \rrbracket\}$, where $a \times b = \{t \in \mathbb{C}(1, |A| \times |B|) \mid \pi_L \circ t \in$
- 495 $a \text{ and } \pi_R \circ t \in b\}$,
- 496 ■ $\llbracket A \times B \rrbracket = \llbracket A \rrbracket \times \llbracket B \rrbracket$,
- 497 ■ $\delta_{A \times B}(c) = (\delta_A(\{\pi_L \circ t \mid t \in c\}), \delta_B(\{\pi_R \circ t \mid t \in c\}))$,
- 498 ■ $\llbracket \exp(A, B) \rrbracket = \{\{t \in \mathbb{C}(1, \exp(|A|, |B|)) \mid \forall u \in \Lambda_A, \text{ev}(t) \circ u \in I(u)\} \mid I \in \text{Poset}(\Lambda_A, \llbracket B \rrbracket)\}$,
- 499 ■ $\llbracket \exp(A, B) \rrbracket = \text{Poset}(\llbracket A \rrbracket, \llbracket B \rrbracket)$,
- 500 ■ $\delta_{\exp(A, B)}(c) = a \mapsto \delta_B \left(\sup \left\{ \overline{\text{ev}(v) \circ t} \mid t \in a, v \in c \right\} \right)$.

501 We need a counterpart to Proposition 4. As explained above, we obtain this by organizing

502 the \mathbb{C} -diameter spaces as a cartesian poset-enriched category with a lax-exponential. First,

503 we need to define a notion of morphisms between two \mathbb{C} -diameter spaces A and B (which

504 represent *approximate programs*). By analogy with Section 3, these will be monotone functions

505 from $\llbracket A \rrbracket$ to $\llbracket B \rrbracket$; however, in order to actually obtain a cartesian category (which was not

506 an issue in Section 3), we will need to add an extra condition:

- 507 ► **Definition 15.** We denote by $\text{Diam}(\mathbb{C})$ the poset-enriched category defined as follows:
- 508 ■ the objects of $\text{Diam}(\mathbb{C})$ are the \mathbb{C} -diameter spaces,
 - 509 ■ for all \mathbb{C} -diameter spaces A and B , $\text{Diam}(\mathbb{C})(A, B)$ is the set of all monotone functions
 - 510 $\varphi : \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket$ such that there exists $f \in \mathbb{C}(|A|, |B|)$ such that for all $t \in \Lambda_A$, $f \circ t \in \varphi(\bar{t})$
 - 511 (ordered by pointwise inclusion).

512 One can check that the operation $- \times -$ defined above on \mathbb{C} -diameter spaces is a cartesian

513 product in $\text{Diam}(\mathbb{C})$. In addition, one can check that there exists in $\text{Diam}(\mathbb{C})$ a terminal

514 object $1_{\text{Diam}(\mathbb{C})}$ such that $|1_{\text{Diam}(\mathbb{C})}| = 1_{\mathbb{C}}$. In other words, $\text{Diam}(\mathbb{C})$ is cartesian. Here too,

515 we denote by $\langle -, - \rangle$ the pairing transformation and by π_L and π_R the projections.

516 Now, following Section 3, we can complete the definition of the lax-exponential: let

517 A, B, C be \mathbb{C} -diameter spaces,

- 518 ■ for all $\varphi \in \text{Diam}(\mathbb{C})(A, \exp(B, C))$, we define $\text{ev}_{A, B, C}(\varphi) \in \text{Diam}(\mathbb{C})(A \times B, C)$ by
- 519 $\text{ev}_{A, B, C}(\varphi)(p) = \sup \left\{ \overline{\text{ev}(v) \circ u} \mid v \in \varphi(\pi_L(p)), u \in \pi_R(p) \right\}$,

23:14 A Partial Metric Semantics of Higher-Order Programs

520 ■ for all $\psi \in \text{Diam}(\mathbb{C})(A \times B, C)$, we define $\lambda_{A,B,C}(\psi) \in \text{Diam}(\mathbb{C})(A, \text{exp}(B, C))$ by
 521 $\lambda_{A,B,C}(\psi)(a) = \{v \in \Lambda_{\text{exp}(B,C)} \mid \forall u \in \Lambda_B, \text{ev}(v) \circ u \in \psi(a \times \bar{u})\}$.

522 ► **Proposition 16.** *The triple $(\text{exp}, \text{ev}, \lambda)$ is a lax-exponential on $\text{Diam}(\mathbb{C})$.*

523 **Proof.** Naturality with respect to A is immediate.

524 Let $p = a \times b \in \llbracket A \times B \rrbracket$. For all $v \in \lambda(\psi)(a)$ and $u \in b$, by definition $\text{ev}(v) \circ u \in$
 525 $\psi(a \times \bar{u}) \subseteq \psi(p)$. Therefore, $\text{ev}(\lambda(\psi))(p) \subseteq p$.

526 Let $a \in \llbracket A \rrbracket$ and $v \in \varphi(a)$. For all $u \in \Lambda_B$, by definition, $\text{ev}(v) \circ u \in \lambda(\varphi)(a \times \bar{u})$, so
 527 $v \in \lambda(\text{ev}(\varphi))(a)$. ◀

528 As in Section 3, we can find a kind of weak embedding from \mathbb{C} to $\text{Diam}(\mathbb{C})$. Namely, for
 529 all \mathbb{C} -diameter spaces A and B , we define a monotone map $\partial : \mathbb{C}(|A|, |B|) \rightarrow \text{Diam}(\mathbb{C})(A, B)$
 530 by $\partial(f)(a) = \sup\{\overline{f \circ t} \mid t \in a\}$. The following compatibility result is immediate and offers a
 531 counterpart to Remark 6:

532 ► **Proposition 17.** *For all \mathbb{C} -diameter spaces A, B, C , all $f \in \mathbb{C}(|A|, |B|)$ and all $g \in$
 533 $\mathbb{C}(|B|, |C|)$, $\partial(g \circ f) \leq \partial(g) \circ \partial(f)$. In addition, $\partial(\text{id}_{|A|}) = \text{id}_A$.*

534 One way to reformulate this result is that ∂ induces an oplax-functor from the category
 535 with the same objects as $\text{Diam}(\mathbb{C})$ and the same morphisms as \mathbb{C} , to $\text{Diam}(\mathbb{C})$.

536 One can check that ∂ preserves products, in the sense that $\partial(\langle f, g \rangle) = \langle \partial(f), \partial(g) \rangle$,
 537 $\partial(\pi_L) = \pi_L$ and $\partial(\pi_R) = \pi_R$. In addition ∂ is weakly compatible with the exponential, which
 538 corresponds to Remark 5:

539 ► **Proposition 18.** *Let A, B, C be \mathbb{C} -diameter spaces,*

540 ■ *for all $f \in \mathbb{C}(|A|, \text{exp}(|B|, |C|))$, $\partial(\text{ev}(f)) \leq \text{ev}(\partial(f))$,*

541 ■ *for all $g \in \mathbb{C}(|A| \times |B|, |C|)$, $\partial(\lambda(g)) \leq \lambda(\partial(g))$.*

542 Finally, following Section 3, for all \mathbb{C} -diameter spaces A and all $t, u \in \Lambda_A$, we write $t \approx_A u$
 543 if $\bar{t} = \bar{u}$. In addition, we define a function $d_A : \Lambda_A \times \Lambda_A \rightarrow \llbracket A \rrbracket$ by $d_A(t, u) = \delta_A(\bar{t} \vee \bar{u})$. Then
 544 the same arguments as in Corollary 8 show that:

545 ► **Proposition 19.** *For all \mathbb{C} -diameter spaces A , $(\Lambda_A / \approx_A, \llbracket A \rrbracket, d_A)$ is a generalized partial
 546 metric space.*

547 One can check that what is described in Section 3 is indeed an instance of this construction.
 548 Here are a couple more examples:

549 ► **Example 20.** We can take $\mathbb{C} = \text{Set}$ (with the morphisms ordered by equality): $\text{Diam}(\text{Set})$
 550 contains an object Real_{Set} that represents the real numbers with their standard metric
 551 and the compact intervals (plus \emptyset and \mathbb{R}) as approximate values, namely $|\text{Real}_{\text{Set}}| = \mathbb{R}$,
 552 $\llbracket \text{Real}_{\text{Set}} \rrbracket = \{\text{the compact intervals}, \emptyset, \mathbb{R}\}$, $\llbracket \text{Real}_{\text{Set}} \rrbracket = [0, \infty]$ and $\delta_{\text{Real}_{\text{Set}}}(I) = \text{length}(I)$.

553 In this case, $|\text{exp}(\text{Real}_{\text{Set}}, \text{Real}_{\text{Set}})|$ is the set of all functions from \mathbb{R} to \mathbb{R} , so $d_{\text{Real}_{\text{Set}}}$ defines
 554 a partial metric on all such functions.

555 ► **Example 21.** We can take $\mathbb{C} = \text{Eff}$, the effective topos [27]: Eff contains an object \mathbb{R}_{Eff}
 556 of recursive reals, and we can define an object Real_{Eff} in $\text{Diam}(\text{Eff})$ by $|\text{Real}_{\text{Eff}}| = \mathbb{R}_{\text{Eff}}$,
 557 $\llbracket \text{Real}_{\text{Eff}} \rrbracket = \{I \cap \mathbb{R}_{\text{Eff}} \mid I \in \llbracket \text{Real}_{\text{Set}} \rrbracket\}$, $\llbracket \text{Real}_{\text{Eff}} \rrbracket = [0, \infty]$ and $\delta_{\text{Real}_{\text{Eff}}}(I) = \text{length}(I)$.

558 In this case, $|\text{exp}(\text{Real}_{\text{Eff}}, \text{Real}_{\text{Eff}})|$ is the set of all recursive functions from Real_{Eff} to
 559 Real_{Eff} , so $d_{\text{Real}_{\text{Eff}}}$ defines a partial metric on all such functions.

560 ► **Example 22.** We can take $\mathbb{C} = \text{Scott}$, the poset-enriched category of Scott domains and
 561 continuous functions. It contains an object representing the reals: $\mathbb{R}_{\text{Scott}} = (\mathbb{R} \cup \{\perp\}, \sqsubseteq)$,
 562 with $r \sqsubseteq s$ iff $r = s$ or $r = \perp$. Again, we can define in $\text{Diam}(\text{Scott})$ an object $\text{Real}_{\text{Scott}}$ that
 563 represents the real numbers with their standard metric, and this defines a partial metric
 564 on $|\exp(\text{Real}_{\text{Scott}}, \text{Real}_{\text{Scott}})|$, the set of all Scott continuous functions from $\mathbb{R}_{\text{Scott}}$ to $\mathbb{R}_{\text{Scott}}$,
 565 which are essentially the partial functions from \mathbb{R} to \mathbb{R} .

566 6 Conclusions

567 **Related Work** As stated in the introduction, differential logical relations [32] are a primary
 568 source of inspiration for our approach. A related, but more syntactic approach to approximate
 569 program transformations is that of Westbrook and Chauduri [44], who use a System F-based
 570 type system with a type of real numbers and an explicit distinction between exact and
 571 approximate programs. Most examples of contextual reasoning from [44] can be reformulated
 572 in our framework (as the case of loop perforation discussed in Section 4).

573 The literature on program pseudo-metrics is vast. A major distinction can be made
 574 between those approaches in which metrics account for *extensional* aspects of programs (like
 575 ours), and approaches in which metrics are used to characterize more *intensional* aspects. To
 576 the first family belong all metric models developed for reasoning about differential privacy
 577 [35, 3, 7], probabilistic computation [13, 14] and co-inductive models [16, 43, 11, 42]. To the
 578 second class belong approaches like [19] which recovers the Scott model of PCF through a
 579 ultrametric semantics, and most models based on partial metric spaces [9, 33], which rely on
 580 a correspondence between continuous Scott domains and the T_0 topology of partial metrics.

581 From a more mathematical viewpoint, [12] discusses a characterization of exponentiable
 582 GPMS, showing that no such category can both be cartesian closed and contain the standard
 583 metric on \mathbb{R} . This result seems to add further evidence of the necessity of considering
 584 metrics over varying quantales in order to model higher-order languages. Finally, the elegant
 585 categorical approach to GPMS based on *quantaloid-enriched categories* from [26] seems to
 586 provide the relevant structure to develop explicit typing rules for our approximate programs.

587 **Future Work** The approach we presented lends itself to further extensions and general-
 588 izations. First, we would like to investigate the interpretation of more type constructions
 589 than those of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$ (e.g. coproducts, recursive types, effects). Moreover, we would like
 590 to explore the possibility of exploiting the structure of the category $\text{Diam}(\mathbb{C})$ to construct
 591 new and more refined notions of approximations. For example (we work in $\text{Diam}(\text{Set})$ for
 592 simplicity), starting from the “standard” set of approximate values \mathcal{I} on $\mathbb{R}^{X \times X}$ (with elements
 593 of \mathcal{I} being families of compact intervals $U_{x,x'} \subseteq \mathbb{R}$ indexed by elements of X and X'), one
 594 can define a new family $\Delta^*\mathcal{I}$ of approximate values for \mathbb{R}^X by “pulling back” the exact map
 595 $\Delta : \mathbb{R}^X \rightarrow \mathbb{R}^{X \times X}$ defined by $\Delta f(x, x') = f(x') - f(x)$, i.e. letting $\Delta^*\mathcal{I} = \{\Delta^{-1}(a) \mid a \in \mathcal{I}\}$.
 596 The new approximate values then correspond to sets of functions $f \in \mathbb{R}^X$ with a controlled
 597 variation, that is, such that $f(x') - f(x)$ is bounded by some family of intervals $U_{x,x'} \in \mathcal{I}$.

598 Another interesting research direction concerns probabilistic extensions of $\text{ST}\lambda\text{C}(\mathcal{F}_n)$.
 599 Probabilistic metrics [15, 30, 13, 14] have been the object of much research in recent years, due
 600 to the relevance of metric reasoning in some areas of computer science in which probabilistic
 601 computation plays a key role (e.g. in cryptography [22] and machine learning [31]). A
 602 convenient starting point seems to be the recent generalization of probabilistic (generalized)
 603 metric spaces to the partial metric case [23].

604 — References

- 605 1 S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum. *Handbook of Logic in Computer Science: Volume 2. Background: Computational Structures*. Handbook of Logic in Computer Science. Clarendon Press, 1992. URL: <https://books.google.fr/books?id=zqkXKMNXViOC>.
- 606 2 Mario Alvarez-Picallo and C.-H. Luke Ong. Change actions: models of generalised differentiation. In *FOSSACS 2019*, volume 11425 of *LNCS*, pages 45–61, 2019.
- 607 3 Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: On the trade-off between utility and information leakage. In *Proceedings of the 8th International Conference on Formal Aspects of Security and Trust, FAST-11*, pages 39–54, Berlin, Heidelberg, 2011. Springer-Verlag. URL: https://doi.org/10.1007/978-3-642-29420-4_3, doi:10.1007/978-3-642-29420-4_3.
- 608 4 A. Arnold and M. Nivat. Metric interpretations of infinite trees and semantics of non deterministic recursive programs. *Theoretical Computer Science*, 11(2):181 – 205, 1980. URL: <http://www.sciencedirect.com/science/article/pii/0304397580900456>, doi:[https://doi.org/10.1016/0304-3975\(80\)90045-6](https://doi.org/10.1016/0304-3975(80)90045-6).
- 609 5 Arthur Azevedo de Amorim, Marco Gaboardi, Justin Hsu, Shin-ya Katsumata, and Ikram Cherigui. A semantic account of metric preservation. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages - POPL 2017*. ACM Press, 2017. URL: <http://dx.doi.org/10.1145/3009837.3009890>, doi:10.1145/3009837.3009890.
- 610 6 Christel Baier and Mila E. Majster-Cederbaum. Denotational semantics in the cpo and metric approach. *Theoretical Computer Science*, 135(2):171 – 220, 1994. URL: <http://www.sciencedirect.com/science/article/pii/0304397594000468>, doi:[https://doi.org/10.1016/0304-3975\(94\)00046-8](https://doi.org/10.1016/0304-3975(94)00046-8).
- 611 7 Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Béguelin. Probabilistic relational reasoning for differential privacy. In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '12*. ACM Press, 2012. URL: <http://dx.doi.org/10.1145/2103656.2103670>, doi:10.1145/2103656.2103670.
- 612 8 Michael Bukatin, Ralph Kopperman, Steve Matthews, and Homeira Pajooesh. Partial metric spaces. *American Mathematical Monthly - AMER MATH MON*, 116:708–718, 10 2009. doi:10.4169/193009709X460831.
- 613 9 Michael A. Bukatin and Joshua S. Scott. Towards computing distances between programs via scott domains. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science*, pages 33–43, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- 614 10 Y. Cai, P.G. Giarrusso, T. Rendel, and K. Ostermann. A theory of changes for higher-order languages: incrementalizing λ -calculi by static differentiation. *ACM SIGPLAN Not.*, 49:145–155, 2014.
- 615 11 Konstantinos Chatzikokolakis, Daniel Gebler, Catuscia Palamidessi, and Lili Xu. Generalized bisimulation metrics. In Paolo Baldan and Daniele Gorla, editors, *CONCUR 2014 - Concurrency Theory*, pages 32–46, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- 616 12 Maria Manuel Clementino, Dirk Hofmann, and Isar Stubbe. Exponentiable functors between quantaloid-enriched categories. *Applied Categorical Structures*, 17(1):91–101, 2009. URL: <https://doi.org/10.1007/s10485-007-9104-5>, doi:10.1007/s10485-007-9104-5.
- 617 13 Raphaëlle Crubillé and Ugo Dal Lago. Metric reasoning about λ -terms: The affine case. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, page 633?644, USA, 2015. IEEE Computer Society. URL: <https://doi.org/10.1109/LICS.2015.64>, doi:10.1109/LICS.2015.64.
- 618 14 Raphaëlle Crubillé and Ugo Dal Lago. Metric reasoning about λ -terms: The general case. In Hongseok Yang, editor, *Programming Languages and Systems*, pages 341–367, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.
- 619 15 J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, pages 413–422, July 2002. doi:10.1109/LICS.2002.1029849.

- 656 16 Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for
657 labelled markov processes. *Theoretical Computer Science*, 318(3):323 – 354, 2004. URL:
658 <http://www.sciencedirect.com/science/article/pii/S0304397503006042>, doi:<https://doi.org/10.1016/j.tcs.2003.09.013>.
659
- 660 17 Pietro Di Gianantonio and Abbas Edalat. A language for differentiable functions. In Frank
661 Pfenning, editor, *Foundations of Software Science and Computation Structures*, pages 337–352,
662 Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- 663 18 Abbas Edalat and Martín Hötzel Escardó. Integration in real pcf. *Information and Compu-*
664 *tation*, 160(1):128–166, 2000. URL: <http://www.sciencedirect.com/science/article/pii/S0890540199928444>, doi:<https://doi.org/10.1006/inco.1999.2844>.
665
- 666 19 Martín Hötzen Escardó. A metric model of PCF. In *Workshop on Realizability Semantics and*
667 *Applications*, 1999.
- 668 20 Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce.
669 Linear dependent types for differential privacy. In *Proceedings of the 40th annual ACM*
670 *SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '13*. ACM
671 Press, 2013. URL: <http://dx.doi.org/10.1145/2429069.2429113>, doi:10.1145/2429069.
672 2429113.
- 673 21 Francesco Gavazzo. Quantitative behavioural reasoning for higher-order effectful programs:
674 Applicative distances. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic*
675 *in Computer Science*, LICS '18, page 452?461, New York, NY, USA, 2018. Association for
676 Computing Machinery. URL: <https://doi.org/10.1145/3209108.3209149>, doi:10.1145/
677 3209108.3209149.
- 678 22 Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System*
679 *Sciences*, 28(2):270 – 299, 1984. URL: <http://www.sciencedirect.com/science/article/pii/0022000084900709>, doi:[https://doi.org/10.1016/0022-0000\(84\)90070-9](https://doi.org/10.1016/0022-0000(84)90070-9).
680
- 681 23 Jialiang He, Hongliang Lai, and Lili Shen. Towards probabilistic partial metric spaces:
682 Diagonals between distance distributions. *Fuzzy Sets and Systems*, 370:99 – 119, 2019. Theme:
683 Topology and Metric Spaces. URL: <http://www.sciencedirect.com/science/article/pii/S0165011418304421>, doi:<https://doi.org/10.1016/j.fss.2018.07.011>.
684
- 685 24 Barnaby P. Hilken. Towards a proof theory of rewriting: the simply typed 2λ -
686 calculus. *Theoretical Computer Science*, 170(1):407 – 444, 1996. URL: <http://www.sciencedirect.com/science/article/pii/S0304397596807134>, doi:[https://doi.org/10.1016/S0304-3975\(96\)80713-4](https://doi.org/10.1016/S0304-3975(96)80713-4).
687
688
- 689 25 Dirk Hofmann, Gavin J Seal, and W Tholen. *Monoidal Topology: a Categorical Approach to*
690 *Order, Metric and Topology*. Cambridge University Press, New York, 2014.
- 691 26 Dirk Hofmann and Isar Stubbe. Topology from enrichment: the curious case of partial metrics.
692 *Cahiers de Topologie et Géométrie Différentielle Catégorique*, LIX, 4:307–353, 2018.
- 693 27 J.M.E. Hyland. The effective topos. In A.S. Troelstra and D. [van Dalen], edi-
694 tors, *The L. E. J. Brouwer Centenary Symposium*, volume 110 of *Studies in Logic and*
695 *the Foundations of Mathematics*, pages 165 – 216. Elsevier, 1982. URL: <http://www.sciencedirect.com/science/article/pii/S0049237X09701296>, doi:[https://doi.org/10.1016/S0049-237X\(09\)70129-6](https://doi.org/10.1016/S0049-237X(09)70129-6).
696
697
- 698 28 Gunther Jäger and T. M. G. Ahsanullah. Characterization of quantale-valued metric spaces and
699 quantale-valued partial metric spaces by convergence. *Applied General Topology*, 19(1):129–144,
700 2018.
- 701 29 Ralph Kopperman, Steve Matthews, and Homeira Pajoohesh. Partial metrizable in value
702 quantales. *Applied General Topology*, 5(1):115–127, 2004.
- 703 30 Dexter Kozen. Semantics of probabilistic programs. *Journal of Computer and System*
704 *Sciences*, 22(3):328 – 350, 1981. URL: <http://www.sciencedirect.com/science/article/pii/0022000081900362>, doi:[https://doi.org/10.1016/0022-0000\(81\)90036-2](https://doi.org/10.1016/0022-0000(81)90036-2).
705

- 706 31 Andreas Krause, Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular
707 observation selection. *Journal of Machine Learning Research (JMLR)*, 9:2761–2801, December
708 2008.
- 709 32 Ugo Dal Lago, Francesco Gavazzo, and Akira Yoshimizu. Differential logical relations, part I:
710 the simply-typed case. In *46th International Colloquium on Automata, Languages, and Program-*
711 *ming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 111:1–111:14, 2019. URL: <https://doi.org/10.4230/LIPIcs.ICALP.2019.111>, doi:10.4230/LIPIcs.ICALP.2019.111.
- 712 33 S. G. Matthews. Partial metric topology. *Annals of the New York Academy of Sciences*,
713 728(1):183–197, 1994. doi:10.1111/j.1749-6632.1994.tb44144.x.
- 714 34 Sparsh Mittal. A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4),
715 2016.
- 716 35 Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus
717 for differential privacy. *SIGPLAN Not.*, 45(9):157–168, September 2010. URL: <https://doi.org/10.1145/1932681.1863568>, doi:10.1145/1932681.1863568.
- 718 36 Bessem Samet, Calogero Vetro, and Francesca Vetro. From metric spaces to partial metric
719 spaces. *Fixed Point Theory and Applications*, 2013(1):5, 2013. URL: <https://doi.org/10.1186/1687-1812-2013-5>, doi:10.1186/1687-1812-2013-5.
- 720 37 M. P. Schellekens. The correspondence between partial metrics and semivaluations. *Theoretical*
721 *Computer Science*, 315(1):135–149, 2004.
- 722 38 Stelios Sidiroglou-Douskos, Sasa Misailovic, Henry Hoffmann, and Martin Rinard. Manag-
723 ing performance vs. accuracy trade-offs with loop perforation. In *Proceedings of the 19th*
724 *ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software*
725 *Engineering, ESEC/FSE '11*, pages 124–134, New York, NY, USA, 2011. Association for
726 Computing Machinery.
- 727 39 D. A. Simovici. On submodular and supermodular functions on lattices and related structures.
728 In *2014 IEEE 44th International Symposium on Multiple-Valued Logic*, pages 202–207, May
729 2014. doi:10.1109/ISMVL.2014.43.
- 730 40 Paul Taylor. A lambda calculus for real analysis. *Journal of Logic and Analysis*, 2(5):1–115,
731 2010.
- 732 41 Franck van Breugel. An introduction to metric semantics: operational and denotational
733 models for programming and specification languages. *Theoretical Computer Science*, 258(1):1 –
734 98, 2001. URL: <http://www.sciencedirect.com/science/article/pii/S0304397500004035>,
735 doi:[https://doi.org/10.1016/S0304-3975\(00\)00403-5](https://doi.org/10.1016/S0304-3975(00)00403-5).
- 736 42 Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic
737 transition systems. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors,
738 *Automata, Languages and Programming*, pages 421–432, Berlin, Heidelberg, 2001. Springer
739 Berlin Heidelberg.
- 740 43 Franck van Breugel and James Worrell. A behavioural pseudometric for probabilistic tran-
741 sition systems. *Theoretical Computer Science*, 331(1):115 – 142, 2005. Automata, Lan-
742 guages and Programming. URL: <http://www.sciencedirect.com/science/article/pii/S0304397504006711>, doi:<https://doi.org/10.1016/j.tcs.2004.09.035>.
- 743 44 Edwin Westbrook and Swarat Chaudhuri. A semantics for approximate program transforma-
744 tions. <https://arxiv.org/abs/1304.5531>, 2013.
- 745
746
747
748