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

Guillaume Geoffroy
University of Bologna, Department of Computer Science and Engineering, Italy
guillaume.geoffroy@unibo.it

Paolo Pistone
University of Bologna, Department of Computer Science and Engineering, 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 $\boldsymbol{\lambda}$-calculus, program metrics, approximate program transformations, partial metric spaces

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.23

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 $\mathtt{C}[t] \rightsquigarrow \mathtt{C}[u]$, which replaces a program t by u *within a context* $\mathtt{C}[\ ]$, as a function of the mismatch between t and u and of the sensitivity of the context $\mathtt{C}[\ ]$ 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 $\mathtt{C}$ will be highly sensitive to how close to 0 these functions are

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

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

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

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

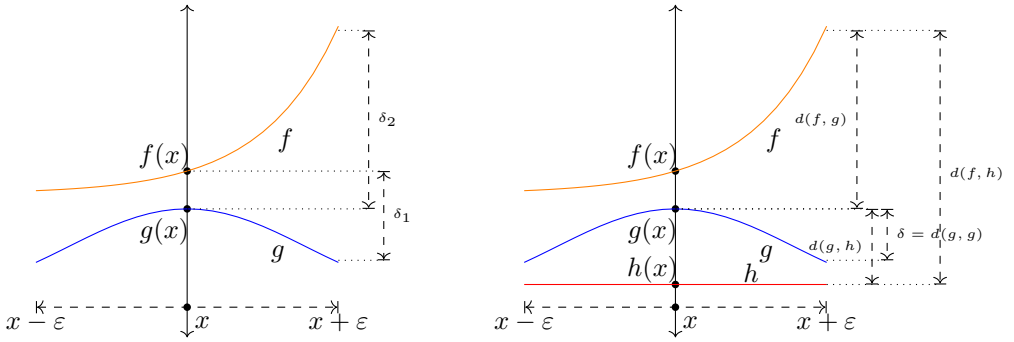
On the other side of the coin, other approaches like [44, 32] are fully contextual and higher-order, but provide, at best, only weak approximations of a standard notion of metric. Nonetheless, these approaches introduce the idea, which we retain here, that program differences must be taken as being themselves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

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

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

In our models a higher-order type A is interpreted as a 4-tuple $(|A|, [\![A]\!], (\!|A|\!), \delta_A)$ called a *diameter space*, where $|A|$ is a set of *exact* values, $[\![A]\!] \subset \mathcal{P}(|A|)$ is a complete lattice of *approximate* values, $(\!|A|\!)$ is a quantale, and $\delta_A : [\![A]\!] \to (\!|A|\!)$ is a function, called *diameter*,

[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 $f, g : \mathbb{R} \to \mathbb{R}$, 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.

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 $d_A : |A| \times |A| \to (\!|A|\!)$ 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 were 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 $f, g : \mathsf{Real} \to \mathsf{Real}$ is taken in the quantale of functions from $\mathbb{R} \times \mathbb{R}_+^\infty$ to $\mathbb{R}_+^\infty$: 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 $f, g : \mathsf{Real} \to \mathsf{Real}$ lives in the quantale of monotone maps from *approximate values* of 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 longer depict the "center" of the interval $[x-\varepsilon, x+\varepsilon]$, and that the triangular inequality works because in summing $d(f, g)$ and $d(g, h)$ the self-distance $d(g, g)$ is counted twice.

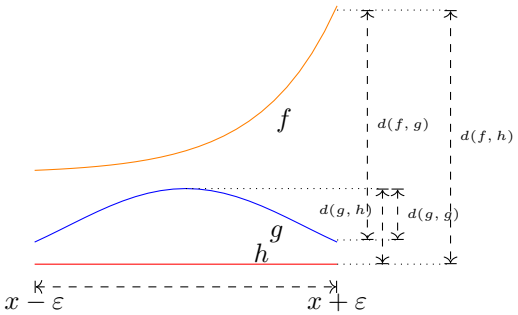


Figure 2 Our new metric is a partial metric: in the example above it can be seen that $d(f,h) \leq d(f,g) + d(g,h) - d(g,g)$ (with all distances computed in the interval $[x-\varepsilon, x+\varepsilon]$).

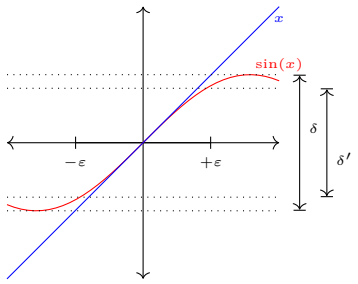


Figure 3 The self-distances δ, δ' of $\sin(x)$ and x in a small interval $[-\varepsilon, \varepsilon]$ of 0 are very close.

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

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

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

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

2 Generalized Partial Metric Spaces

Partial metric spaces were introduced in the early nineties as a variant of metric spaces in which self-distances can be non-zero. Such spaces have attracted much attention in program semantics [8, 9, 33, 37, 36, 26, 23], due to their compatibility with standard constructions from both domain theory (since their topology is T_0) and usual metric topology (e.g. Cauchy sequences, completeness, Banach-fixed point theorem) [8, 33]. *Generalized partial metric spaces*, i.e. partial metric spaces whose metric takes values over an arbitrary quantale [25], are well-investigated too [29, 28].

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

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

- $(Q, \leq)$ *is a complete lattice,*
- $(Q, +)$ *is a commutative monoid,*
- $+$ *commutes with arbitrary* $\inf s$*,*
- *the least element of* Q *is neutral for* $+$*.*

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

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

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

- *for all* $x, y \in X$*,* $d(x, x) \leq d(x, y)$*,*
- *for all* $x, y \in X$*, if* $d(x, x) = d(x, y) = d(y, y)$*, then* $x = y$*,*
- *for all* $x, y \in X$*,* $d(x, y) = d(y, x)$*,*
- *for all* $x, y, z \in X$*,* $d(x, z) + d(y, y) \leq d(x, y) + d(y, z)$*.*

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

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

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

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

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

To illustrate our construction, we start from a relatively concrete example: we consider a simply-typed lambda calculus with a base type Real and primitives for real numbers, and we follow the plan outlined in the introduction, which yields for each simple type a notion of

approximate value, approximate function, diameter and distance between programs. Most definitions are straightforward and intuitive: the interesting, not immediately obvious point is that our construction does yield a *partial metric* on each type.

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

▶ Remark 3. The class of real-valued functions which can be computed in $\mathsf{ST}\lambda\mathsf{C}(\mathcal{F}_n)$ depends on the choice we make for $\mathcal{F}_n$. With suitable choices (see for instance [40, 17, 18]) one can obtain that all programs of type $\mathsf{Real} \to \mathsf{Real}$ compute *continuous* functions[2], that all such programs are *integrable* over closed intervals, or that all such programs are *continuously differentiable*.

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

3.1 Approximate Values and Approximate Programs

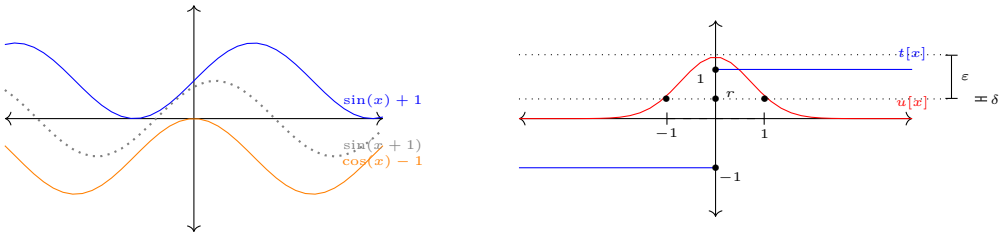
The first step of our construction for $\mathsf{ST}\lambda\mathsf{C}(\mathcal{F}_n)$ is to associate to each simple type A a set $[\![A]\!]$ whose elements are certain sets of programs of type A that we call *approximate values of type* A. A closed term $t \in \Lambda_A$ represents a program with return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

For each simple type A, the set of approximate values $[\![A]\!] \subseteq \mathcal{P}(\Lambda_A)$ is defined inductively as follows:

- $[\![\mathsf{Real}]\!] = \{\{t \in \Lambda_{\mathsf{Real}} \mid \exists r \in I, t \to_\beta^* r\} \mid I \subseteq \mathbb{R} \text{ is a compact interval or } \emptyset \text{ or } \mathbb{R}\}$,
- $[\![A \times B]\!] = \{a \times b \mid a \in [\![A]\!], b \in [\![B]\!]\}$, where $a \times b = \{t \in \Lambda_{A \times B} \mid \pi_L t \in a \text{ and } \pi_R t \in b\}$,
- $[\![A \to B]\!] = \{\{t \in \Lambda_{A \to B} \mid \forall u \in \Lambda_A,\ tu \in I(u)\} \mid I : \Lambda_A \to [\![B]\!]\}$.

The approximate values of type Real are sets of closed programs of type Real which essentially coincide with the compact intervals of $\mathbb{R}$, plus the empty set and $\mathbb{R}$ itself. An approximate value in $[\![A \times B]\!]$ is a "rectangle" $a \times b$, with $a \in [\![A]\!]$ and $b \in [\![B]\!]$, while an approximate value in $[\![A \to B]\!]$ is uniquely determined by a function I from closed terms $u \in \Lambda_A$ to approximate values $I(u) \in [\![B]\!]$.

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



(a) $\lambda x.\sin(x+1)$ is in $[\lambda x.\sin(x)+1, \lambda x.\cos(x)+1]_{\mathsf{Real}\to\mathsf{Real}}$.

(b) $\varepsilon = (\partial(u)\circ\partial(t))([-1,1])$ is bigger than $\delta = \partial(u\circ t)([-1,1]) = [r,r]$.

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

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

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

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

- for all monotone functions $\vec{\alpha} \mapsto c[\vec{\alpha}] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![B \to C]\!]$ and $\vec{\alpha} \mapsto b[\vec{\alpha}] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![B]\!]$, we can define a monotone function $\vec{\alpha} \mapsto (c[\vec{\alpha}]\ b[\vec{\alpha}]) = \sup\{\overline{vu} \mid v \in c[\vec{\alpha}], u \in b[\vec{\alpha}]\} : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![C]\!]$,
- for all monotone functions $\vec{\alpha} \mapsto c[\vec{\alpha}] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![C]\!]$ and all $i \le n$, we can define a monotone function $(\alpha_j)_{j\neq i} \mapsto (\lambda\alpha_i.\ c[\vec{\alpha}]) = \{v \in \Lambda_{A_i\to C} \mid \forall t_i \in \Lambda_{A_i},\ vt_i \in c[\alpha_1, \ldots, \overline{t_i}, \ldots, \alpha_n]\} : \prod_{j\neq i}[\![A_j]\!] \to [\![A_i \to C]\!]$,

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

▶ **Proposition 4.** *For all monotone functions* $(\vec{\alpha}, \beta) \mapsto c[\vec{\alpha}, \beta] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \times [\![B]\!] \to [\![C]\!]$ *and* $\vec{\alpha} \mapsto b[\vec{\alpha}] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![B]\!]$, $(\vec{\alpha} \mapsto (\lambda\beta.\ c[\vec{\alpha}, \beta])\ b[\vec{\alpha}]) \le (\vec{\alpha} \mapsto c[\vec{\alpha}, b[\vec{\alpha}]])$, *and for all monotone functions* $\vec{\alpha} \mapsto d[\vec{\alpha}] : [\![A_1]\!] \times \ldots \times [\![A_n]\!] \to [\![B \to C]\!]$, $(\vec{\alpha} \mapsto \lambda\beta.\ d[\vec{\alpha}]\ \beta) \ge (\vec{\alpha} \mapsto d[\vec{\alpha}])$, *where functions are ordered by pointwise inclusion. In other words, on approximate programs,* β*-reduction and* η*-expansion* discard *information, and conversely* β*-expansion and* η*-reduction* recover *some information.*

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

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 $\partial(t) : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![B]\!]$ 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 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 $\partial(t) : [\![\mathsf{Real}]\!] \to [\![\mathsf{Real}]\!]$ sends the interval $[-\pi, \pi]_{\mathsf{Real}}$ into $[0, 2]_{\mathsf{Real}}$.

▶ 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 $\partial(u)(\partial(t)(a)) = [-1, 1] \supsetneq [r, r] = \partial(u[t/x])(a)$.

3.2 A Partial Metric on Each Type

So far, we have associated each type A of $\mathsf{ST}\lambda\mathsf{C}(\mathcal{F}_n)$ with a complete lattice $[\![A]\!] \subseteq \mathcal{P}(\Lambda_A)$ of approximate values of type A, and each typed program $t : A \to B$ 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 $\mathsf{ST}\lambda\mathsf{C}(\mathcal{F}_n)$, 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 $(\langle\!|A|\!\rangle, \leq_A, +_A)$ of *distances of type A*:

- $(\langle\!|\mathsf{Real}|\!\rangle, \leq_{\mathsf{Real}}, +_{\mathsf{Real}}) = ([0, \infty], \leq, +)$,
- $\langle\!|A \times B|\!\rangle = \langle\!|A|\!\rangle \times \langle\!|B|\!\rangle$,
- $\langle\!|A \to B|\!\rangle = \mathrm{Poset}([\![A]\!], \langle\!|B|\!\rangle)$.

where, for two posets Q, R, $\mathrm{Poset}(Q, R)$ denotes the set of monotone functions from Q to R. Observe that the quantale $\langle\!|A \to B|\!\rangle$ is a set of functions over the approximate values of A.

For all simple types A, we now define a *distance function* $d_A : \Lambda_A \times \Lambda_A \to \langle\!|A|\!\rangle$:

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

It would be tempting to define $d_{A \to B}(t, u)(a)$ simply as $\sup\{d_B(tv, uw) \mid v, w \in a\}$, but then the axiom "$d_{A \to B}(t, t) \leq d_{A \to B}(t, u)$" 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 $(\Lambda_A / \approx_A, \langle\!|A|\!\rangle, d_A)$ is a generalized partial metric space. To this end, we define for all simple types A a monotone *diameter function* $\delta_A : [\![A]\!] \to \langle\!|A|\!\rangle$ 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-modular* on intersecting approximate values (henceforth, *quasi-sub-modular* – see Proposition 7): this generalizes the fact that, on the (real-valued) metric space $\mathbb{R}$, the diameter is *modular* over intersecting closed intervals (see Fig. 5).

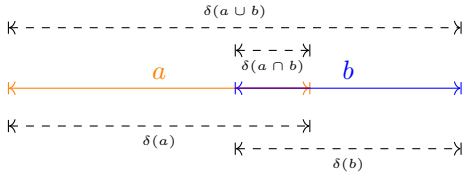


Figure 5 The diameter function is *modular* over intersecting real intervals: $\mathsf{diam}(a \cup b) + \mathsf{diam}(a \cap b) = \mathsf{diam}(a) + \mathsf{diam}(b)$ for all $a, b \in [\mathbb{R}]$ such that $a \cap b \neq \emptyset$. This property is at the heart of our generalization of diameters.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

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

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

This leads then to the following:

▶ **Proposition 7** (δ_A is quasi-sub-modular)**.** *For all simple types A and all $a, b \in [\![A]\!]$ such that $a \wedge b \neq \emptyset$, $\delta(a \wedge b) + \delta(a \vee b) \leq \delta(a) + \delta(b)$.*

Proof. We proceed by induction on types.

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

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

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

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

▶ **Corollary 8.** *For all simple types A, $(\Lambda_A / \approx_A, (\!|A|\!), d_A)$ is a generalized partial metric space, that is to say:*

1. *for all $t, u \in \Lambda_A$, $d_A(t, t) \leq d_A(t, u)$,*
2. *for all $t, u \in \Lambda_A$, if $d_A(t, t) = d_A(t, u) = d_A(u, u)$, then $t \approx_A u$,*
3. *for all $t, u \in \Lambda_A$, $d_A(t, u) = d_A(u, t)$,*
4. *for all $t, u, v \in \Lambda_A$, $d_A(t, v) + d_A(u, u) \leq d_A(t, u) + d_A(u, v)$.*

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

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

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

4 Computing Program Distances using Partial Metrics

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

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

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

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

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

Suppose $t : (A \times A \to A) \to \mathsf{Nat} \to (A \to A) \to A$, for $n \geq 1$, is a term such that $th\mathtt{n}f$ computes the n-times iteration of h as follows: $th\mathtt{0}f = h\langle f\mathtt{0}, f\mathtt{0}\rangle$ and $th(\mathtt{n}+\mathtt{1})f = h\langle th\mathtt{n}f, f(\mathtt{n}+\mathtt{1})\rangle$. Let $\mathsf{Perf}^k(t)$, the k-th perforation of t, be the program $(\mathsf{Perf}^k(t))h\mathtt{n}f = t(\lambda x.(h^{(k)}x))\lfloor\mathtt{n}\rfloor_\mathtt{k}(\lambda x.f(x * \mathtt{k})$, where $\lfloor n\rfloor_k$ indicates the least $m \leq n$ such that m is divisible by k, and $x * \mathtt{k}$ is the multiplication of x by k.

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

- **i.** v_n performs n-iterations while w_n performs $k\lfloor n\rfloor_k \leq n$ iterations, and we can compute $d_A(v_n, v_{(k\lfloor n\rfloor_k)})$ as the diameter of $\partial(t)\partial(h)([k\lfloor n\rfloor_k, n]_{\mathsf{Nat}})\partial(f)$.
- **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 to $\mathtt{i}$, while at the i-th iteration of w_n, f is applied to $\lfloor i\rfloor_k$. Now, the error of replacing $f\mathtt{i}$ by $f\lfloor\mathtt{j}\rfloor_k$, with $\mathtt{i}, \mathtt{j}$ in some $a \in [\![\mathsf{Nat}]\!]$, is accounted for by the approximate program $c[y] = \partial(f)(y-k)$, where $y - k = y \vee \{u - \mathtt{k} \mid u \in y\}$. We deduce then that $d_A(v_n, w_n)$ is bounded by the diameter of $\partial(t)\partial(h)\overline{\mathtt{n}}(\lambda y.c[y])$.
- **iii.** From the fact that $w_n = w_{(k\cdot\lfloor n\rfloor_k)}$ and the triangular inequality of the partial metric d_A we deduce $d_A(v_n, w_n) = d_A(v_n, w_{(k\cdot\lfloor n\rfloor_k)}) \leq d_A(v_n, v_{(k\cdot\lfloor n\rfloor_k)}) + d_A(v_{(k\cdot\lfloor n\rfloor_k)}, w_{(k\cdot\lfloor n\rfloor_k)}) - d_A(v_{(k\cdot\lfloor n\rfloor_k)}, v_{(k\cdot\lfloor n\rfloor_k)})$

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: $d_A(v_n, w_n) \leq \delta_A(\partial(t)\partial(h)([k\lfloor n\rfloor_k, n]_{\mathsf{Nat}})\partial(f)) + \delta_A(\partial(t)\partial(h)\overline{\mathtt{n}}(\lambda y.\partial(f)(y-k))) - \delta_A(\partial(t)\partial(h)\overline{\mathtt{n}}\partial(f))$.

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

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

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

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

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

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

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

$$\begin{aligned} d_{\mathsf{Real}}(\mathsf{I}_{[0,r]}(t), \mathsf{R}^n_{[0,r]}(u)) &\leq d_{\mathsf{Real}}(\mathsf{I}_{[0,r]}(t), \mathsf{R}^n_{[0,r]}(t)) + d_{\mathsf{Real}}(\mathsf{R}_{[0,r]}(t), \mathsf{R}^n_{[0,r]}(u)) \\ &\leq \varepsilon_t(r) + \eta_{t,u}(r) \cdot |r| \end{aligned}$$

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

$$\begin{aligned} & d_{\mathsf{Real}\to\mathsf{Real}}(\lambda x.\mathsf{I}_{[0,x]}(t), \lambda x.\mathsf{R}^n_{[0,x]}(u))(a) \\ \leq\ & d_{\mathsf{Real}\to\mathsf{Real}}(\lambda x.\mathsf{I}_{[0,x]}(t), \lambda x.\mathsf{R}^n_{[0,x]}(t))(a) + d_{\mathsf{Real}\to\mathsf{Real}}(\lambda x.\mathsf{R}_{[0,x]}(t), \lambda x.\mathsf{R}^n_{[0,x]}(u))(a) \\ & \qquad - d_{\mathsf{Real}\to\mathsf{Real}}(\lambda x.\mathsf{R}_{[0,x]}(t), \lambda x.\mathsf{R}^n_{0,x]}(t))(a) \\ \leq\ & d_{\mathsf{Real}\to\mathsf{Real}}(v, v)(a) + \big(d_{\mathsf{Real}\to\mathsf{Real}}(t, u)(a) - d_{\mathsf{Real}\to\mathsf{Real}}(t, t)(a)\big) \cdot \delta_{\mathsf{Real}}(a) \end{aligned}$$

5 Diameter Space Models Over a Cartesian Closed Category

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

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

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

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

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

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

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

▶ **Definition 13.** *A* $\mathbb{C}$-diameter space *A is the data of*

- *an object* $|A|$ *of* $\mathbb{C}$*. The poset* $\mathbb{C}(1, |A|)$ *will be denoted by* Λ_A*;*
- *a set* $[\![A]\!]$ *of downwards-closed subsets of* Λ_A *that is closed under arbitrary intersections. In particular,* $[\![A]\!]$ *is a complete lattice whose meet is given by intersection, and for all* $t \in \Lambda_A$*, there is a least element of* $[\![A]\!]$ *that contains* t*, which will be denoted by* $\overline{t}$*;*
- *a commutative integral quantale* $((\!|A|\!), +, \leq)$*;*

- *a monotone function* $\delta_A : [\![A]\!] \to (\!|A|\!)$ *such that*

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

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

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

▶ **Example 14.** If $\mathbb{C}$ is the category whose objects are the simple types from Section 3 and whose morphisms are the (open) terms modulo β-equivalence, then for all simple types A, $(A, [\![A]\!], (\!|A|\!), \delta_A)$ defines a $\mathbb{C}$-diameter space.

Following Section 3, for all $\mathbb{C}$-diameter spaces A and B, we define a $\mathbb{C}$-diameter space $A \times B$ such that $|A \times B| = |A| \times |B|$ and a $\mathbb{C}$-diameter space $\exp(A, B)$ such that $|\exp(A, B)| = \exp(|A|, |B|)$:

- $[\![A \times B]\!] = \{a \times b \mid a \in [\![A]\!], b \in [\![B]\!]\}$, where $a \times b = \{t \in \mathbb{C}(1, |A| \times |B|) \mid \pi_L \circ t \in a \text{ and } \pi_R \circ t \in b\}$,
- $(\!|A \times B|\!) = (\!|A|\!) \times (\!|B|\!)$,
- $\delta_{A\times B}(c) = (\delta_A(\{\pi_L \circ t \mid t \in c\}), \delta_B(\{\pi_R \circ t \mid t \in c\}))$,
- $[\![\exp(A, B)]\!] = \{\{t \in \mathbb{C}(1, \exp(|A|, |B|)) \mid \forall u \in \Lambda_A,\ \mathrm{ev}(t) \circ u \in I(u)\} \mid I \in \mathrm{Poset}(\Lambda_A, [\![B]\!])\}$,
- $(\!|\exp(A, B)|\!) = \mathrm{Poset}([\![A]\!], (\!|B|\!))$,
- $\delta_{\exp(\mathsf{A},\mathsf{B})}(c) = a \mapsto \delta_B\left(\sup\left\{\overline{\mathrm{ev}(v) \circ t} \mid t \in a, v \in c\right\}\right)$.

We need a counterpart to Proposition 4. As explained above, we obtain this by organizing the $\mathbb{C}$-diameter spaces as a cartesian poset-enriched category with a lax-exponential. First, we need to define a notion of morphisms between two $\mathbb{C}$-diameter spaces A and B (which represent *approximate programs*). By analogy with Section 3, these will be monotone functions from $[\![A]\!]$ to $[\![B]\!]$; however, in order to actually obtain a cartesian category (which was not an issue in Section 3), we will need to add an extra condition:

▶ **Definition 15.** *We denote by* $\mathrm{Diam}(\mathbb{C})$ *the poset-enriched category defined as follows:*

- *the objects of* $\mathrm{Diam}(\mathbb{C})$ *are the* $\mathbb{C}$*-diameter spaces,*
- *for all* $\mathbb{C}$*-diameter spaces* A *and* B*,* $\mathrm{Diam}(\mathbb{C})(A, B)$ *is the set of all monotone functions* $\varphi : [\![A]\!] \to [\![B]\!]$ *such that there exists* $f \in \mathbb{C}(|A|, |B|)$ *such that for all* $t \in \Lambda_A$*,* $f \circ t \in \varphi\left(\overline{t}\right)$ *(ordered by pointwise inclusion).*

One can check that the operation $- \times -$ defined above on $\mathbb{C}$-diameter spaces is a cartesian product in $\mathrm{Diam}(\mathbb{C})$. In addition, one can check that there exists in $\mathrm{Diam}(\mathbb{C})$ a terminal object $1_{\mathrm{Diam}(\mathbb{C})}$ such that $\left|1_{\mathrm{Diam}(\mathbb{C})}\right| = 1_{\mathbb{C}}$. In other words, $\mathrm{Diam}(\mathbb{C})$ is cartesian. Here too, we denote by $\langle -, - \rangle$ the pairing transformation and by π_L and π_R the projections.

Now, following Section 3, we can complete the definition of the lax-exponential: let A, B, C be $\mathbb{C}$-diameter spaces,

- for all $\varphi \in \mathrm{Diam}(\mathbb{C})(A, \exp(B, C))$, we define $\mathrm{ev}_{A,B,C}(\varphi) \in \mathrm{Diam}(\mathbb{C})(A \times B, C)$ by $\mathrm{ev}_{A,B,C}(\varphi)(p) = \sup\left\{\overline{\mathrm{ev}(v) \circ u} \mid v \in \varphi(\pi_L(p)), u \in \pi_R(p)\right\}$,
- for all $\psi \in \mathrm{Diam}(\mathbb{C})(A \times B, C)$, we define $\lambda_{A,B,C}(\psi) \in \mathrm{Diam}(\mathbb{C})(A, \exp(B, C))$ by $\lambda_{A,B,C}(\psi)(a) = \{v \in \Lambda_{\exp(B,C)} \mid \forall u \in \Lambda_B,\ \mathrm{ev}(v) \circ u \in \psi(a \times \overline{u})\}$.

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

Proof. Naturality with respect to A is immediate.

Let $p = a \times b \in [\![A \times B]\!]$. For all $v \in \lambda(\psi)(a)$ and and $u \in b$, by definition $\mathrm{ev}(u) \circ u \in \psi(a \times \overline{u}) \subseteq \psi(p)$. Therefore, $\mathrm{ev}(\lambda(\psi))(p) \subseteq p$.

Let $a \in [\![A]\!]$ and $v \in \varphi(a)$. For all $u \in \Lambda_B$, by definition, $\mathrm{ev}(v) \circ u \in \lambda(\varphi)(a \times \overline{u})$, so $v \in \lambda(\mathrm{ev}(\varphi))(a)$. ◀

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

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

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

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

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

- *for all $f \in \mathbb{C}(|A|, \exp(|B|, |C|))$, $\partial(\mathrm{ev}(f)) \leq \mathrm{ev}(\partial(f))$,*
- *for all $g \in \mathbb{C}(|A| \times |B|, |C|)$, $\partial(\lambda(g)) \leq \lambda(\partial(g))$.*

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

▶ **Proposition 19.** *For all $\mathbb{C}$-diameter spaces A, $(\Lambda_A / \approx_A, (\!|A|\!), d_A)$ is a generalized partial metric space.*

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

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

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

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

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

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

6 Conclusions

Related Work. As stated in the introduction, differential logical relations [32] are a primary source of inspiration for our approach. A related, but more syntactic approach to approximate program transformations is that of Westbrook and Chauduri [44], who use a System F-based

type system with a type of real numbers and an explicit distinction between exact and approximate programs. Most examples of contextual reasoning from [44] can be reformulated in our framework (as the case of loop perforation discussed in Section 4).

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

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

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

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

References

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=zqkXKMNXVi0C`.

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.

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. `doi:10.1007/978-3-642-29420-4_3`.

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. doi:10.1016/0304-3975(80)90045-6.

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. doi:10.1145/3009837.3009890.

6 Christel Baier and Mila E. Majster-Cederbaum. Denotational semantics in the cpo and metric approach. *Theoretical Computer Science*, 135(2):171–220, 1994. doi:10.1016/0304-3975(94)00046-8.

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. doi:10.1145/2103656.2103670.

8 Michael Bukatin, Ralph Kopperman, Steve Matthews, and Homeira Pajoohesh. Partial metric spaces. *American Mathematical Monthly - AMER MATH MON*, 116:708–718, October 2009. doi:10.4169/193009709X460831.

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.

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.

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.

12 Maria Manuel Clementino, Dirk Hofmann, and Isar Stubbe. Exponentiable functors between quantaloid-enriched categories. *Applied Categorical Structures*, 17(1):91–101, 2009. doi:10.1007/s10485-007-9104-5.

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. doi:10.1109/LICS.2015.64.

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.

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.

16 Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004. doi:10.1016/j.tcs.2003.09.013.

17 Pietro Di Gianantonio and Abbas Edalat. A language for differentiable functions. In Frank Pfenning, editor, *Foundations of Software Science and Computation Structures*, pages 337–352, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

18 Abbas Edalat and Martín Hötzel Escardó. Integration in real pcf. *Information and Computation*, 160(1):128–166, 2000. doi:10.1006/inco.1999.2844.

19 Martín Hötzen Escardó. A metric model of PCF. In *Workshop on Realizability Semantics and Applications*, 1999.

20 Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *Proceedings of the 40th annual ACM*

SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '13. ACM Press, 2013. doi:10.1145/2429069.2429113.

21 Francesco Gavazzo. Quantitative behavioural reasoning for higher-order effectful programs: Applicative distances. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, page 452?461, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3209108.3209149.

22 Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

23 Jialiang He, Hongliang Lai, and Lili Shen. Towards probabilistic partial metric spaces: Diagonals between distance distributions. *Fuzzy Sets and Systems*, 370:99–119, 2019. Theme: Topology and Metric Spaces. doi:10.1016/j.fss.2018.07.011.

24 Barnaby P. Hilken. Towards a proof theory of rewriting: the simply typed 2λ-calculus. *Theoretical Computer Science*, 170(1):407–444, 1996. doi:10.1016/S0304-3975(96)80713-4.

25 Dirk Hofmann, Gavin J Seal, and W Tholen. *Monoidal Topology: a Categorical Approach to Order, Metric and Topology*. Cambridge University Press, New York, 2014.

26 Dirk Hofmann and Isar Stubbe. Topology from enrichment: the curious case of partial metrics. *Cahiers de Topologie et Géométrie DIfférentielle Catégorique, LIX*, 4:307–353, 2018.

27 J.M.E. Hyland. The effective topos. In A.S. Troelstra and D. [van Dalen], editors, *The L. E. J. Brouwer Centenary Symposium*, volume 110 of *Studies in Logic and the Foundations of Mathematics*, pages 165–216. Elsevier, 1982. doi:10.1016/S0049-237X(09)70129-6.

28 Gunther Jäger and T. M. G. Ahsanullah. Characterization of quantale-valued metric spaces and quantale-valued partial metric spaces by convergence. *Applied General Topology*, 19(1):129–144, 2018.

29 Ralph Kopperman, Steve Matthews, and Homeira Pajoohesh. Partial metrizability in value quantales. *Applied General Topology*, 5(1):115–127, 2004.

30 Dexter Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22(3):328–350, 1981. doi:10.1016/0022-0000(81)90036-2.

31 Andreas Krause, Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research (JMLR)*, 9:2761–2801, December 2008.

32 Ugo Dal Lago, Francesco Gavazzo, and Akira Yoshimizu. Differential logical relations, part I: the simply-typed case. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 111:1–111:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.111.

33 S. G. Matthews. Partial metric topology. *Annals of the New York Academy of Sciences*, 728(1):183–197, 1994. doi:10.1111/j.1749-6632.1994.tb44144.x.

34 Sparsh Mittal. A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4), 2016.

35 Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. *SIGPLAN Not.*, 45(9):157–168, September 2010. doi:10.1145/1932681.1863568.

36 Bessem Samet, Calogero Vetro, and Francesca Vetro. From metric spaces to partial metric spaces. *Fixed Point Theory and Applications*, 2013(1):5, 2013. doi:10.1186/1687-1812-2013-5.

37 M. P. Schellekens. The correspondence between partial metrics and semivaluations. *Theoretical Computer Science*, 315(1):135–149, 2004.

38 Stelios Sidiroglou-Douskos, Sasa Misailovic, Henry Hoffmann, and Martin Rinard. Managing performance vs. accuracy trade-offs with loop perforation. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ESEC/FSE ?11, pages 124–134, New York, NY, USA, 2011. Association for Computing Machinery.

39 D. A. Simovici. On submodular and supermodular functions on lattices and related structures. In *2014 IEEE 44th International Symposium on Multiple-Valued Logic*, pages 202–207, May 2014. doi:10.1109/ISMVL.2014.43.

40 Paul Taylor. A lambda calculus for real analysis. *Journal of Logic and Analysis*, 2(5):1–115, 2010.

41 Franck van Breugel. An introduction to metric semantics: operational and denotational models for programming and specification languages. *Theoretical Computer Science*, 258(1):1–98, 2001. doi:10.1016/S0304-3975(00)00403-5.

42 Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic transition systems. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 421–432, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

43 Franck van Breugel and James Worrell. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science*, 331(1):115–142, 2005. Automata, Languages and Programming. doi:10.1016/j.tcs.2004.09.035.

44 Edwin Westbrook and Swarat Chaudhuri. A semantics for approximate program transformations, 2013. URL: https://arxiv.org/abs/1304.5531.