Quantitative Behavioural Reasoning for Higher-order Effectful Programs: Applicative Distances

Francesco Gavazzo

Quantitative Behavioural Reasoning for Higher-order Effectful Programs: Applicative Distances

Francesco Gavazzo
Università di Bologna & INRIA Sophia Antipolis
Bologna, Italy

Abstract

This paper studies quantitative refinements of Abramsky's applicative similarity and bisimilarity in the context of a generalisation of Fuzz, a call-by-value λ-calculus with a linear type system that can express program sensitivity, enriched with algebraic operations *à la* Plotkin and Power. To do so a general, abstract framework for studying behavioural relations taking values over quantales is introduced according to Lawvere's analysis of generalised metric spaces. Barr's notion of relator (or lax extension) is then extended to quantale-valued relations, adapting and extending results from the field of monoidal topology. Abstract notions of quantale-valued effectful applicative similarity and bisimilarity are then defined and proved to be a compatible generalised metric (in the sense of Lawvere) and pseudometric, respectively, under mild conditions.

Keywords applicative distance, applicative similarity, applicative bisimilarity, Howe's method, algebraic effects, Fuzz, relator

ACM Reference Format:
Francesco Gavazzo. 2018. Quantitative Behavioural Reasoning for Higher-order Effectful Programs: Applicative Distances. In *LICS '18: LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209108.3209149

1 Introduction

Program preorders and equivalences are fundamental concepts in the theory of programming languages since the very birth of the discipline. Such notions are usually defined by means of relations between program phrases aimed to order or identify programs according to their observable *behaviours*, the latter being usually defined by means of a primitive notion of observation such as termination to a given value. We refer to such relations as *behavioural relations*. Well-known behavioural relations for higher-order functional languages include *contextual equivalence/preorder* [28], *applicative (bi)similarity* [1], and *logical relations* [32].

Instead of asking when two programs e and e' are behaviourally similar or equal, a more informative question may be asked, namely how much (behaviourally) different e and e' are. That means that instead of looking at relations relating programs with similar or equal behaviours we look at relations assigning programs a numerical value representing their *behavioural distance*, i.e. a numerical value quantifying the observable differences between their behaviours. The question of quantifying observable differences between programs turned out to be particularly interesting (and challenging) for effectful higher-order languages, where ordinary qualitative (i.e. boolean-valued) equivalences and preorders are too strong. This is witnessed by recent results on behavioural pseudometrics for probabilistic λ-calculi [7, 8] as well as results on semantics of higher-order languages for differential privacy [12, 31]. In the first case one soon realises that programs exhibiting different behaviours only with probability close to zero are fully discriminated by ordinary behavioural relations, whereas in the second case relational reasoning does not provide any information on how much (behavioural) differences between inputs affect differences between outputs.

These problems can be naturally addressed by working with quantitative relations capturing weakened notions of metric such as *generalised metrics* [25] and *pseudometrics* [36]. It is then natural to ask whether and to what extent ordinary behavioural relations can be refined into quantitative relations still preserving their nice properties. Although easy to formulate, answering such a question is far from trivial and requires major improvements in the current theory of behavioural reasoning about programs.

This paper contributes to answering the above question by studying the quantitative refinement of Abramsky's *applicative similarity* and *bisimilarity* [1] for higher-order languages enriched with algebraic effects. Applicative similarity/bisimilarity is a coinductively defined preorder/equivalence relating programs that exhibit similar/equal extensional behaviours. Due to its coinductive nature and to its nice properties, applicative (bi)similarity has been studied for a variety of calculi, both pure and effectful. Notable examples are extensions to nondeterministic [24] and probabilistic [6, 10] λ-calculi, and its more recent extension [9] to λ-calculi with algebraic effects [30]. In [9] an abstract notion of applicative similarity is studied for an untyped λ-calculus enriched with a signature of effect-triggering operation symbols. Operation symbols are interpreted as algebraic operations with respect to a monad T encapsulating the kind of effect such operations produce. Examples are probabilistic choices with the (sub)distribution monad, and nondeterministic choices with the powerset monad. The main ingredient used to extend Abramsky's applicative similarity is the concept of a *relator* [4, 37] for a monad T, i.e. an abstraction meant to capture the possible ways a relation on a set X can be turned into a relation on TX. That allows to define an abstract notion of *effectful* applicative similarity parametric in a relator, and to prove an abstract precongruence theorem stating that effectful applicative similarity is a compatible preorder.

The present work originated from the idea of generalising the theory developed in [9] to relations taking values over quantitative domains (such as the real extended half-line $[0, \infty]$ or the unit interval $[0, 1]$). Such generalisation requires three major improvements in the current theory of effectful applicative (bi)similarity:

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209149

1. The first improvement is to move from boolean-valued relations to relations taking values over quantitative domains such as $[0,\infty]$ or $[0,1]$ in such a way that restricting these domains to the two-element set $\{\mathsf{false},\mathsf{true}\}$ makes the theory collapse to the usual theory of applicative (bi)similarity. For that we rely on Lawvere's analysis [25] of generalised metric spaces and preordered sets as enriched categories. Accordingly, we replace boolean-valued relations with relations taking values over quantales [19] $(\mathsf{V},\leq,\otimes,k)$, i.e. algebraic structures (notably complete lattices equipped with a monoid structure) that play the role of sets of abstract quantities. Examples of quantales include the extended real half-line $([0,\infty],\geq,0,+)$ ordered by the "greater or equal" relation $\geq$ and with monoid structure given by addition, and the extended real half-line $([0,\infty],\geq,0,\max)$ with monoid structure given by binary maximum, as well as any complete Boolean and Heyting algebra. This allows to develop an algebra of quantale-valued relations, V-relations for short, which provides a general framework for studying both behavioural relations and behavioural distances (for instance, an equivalence V-relation instantiates to an ordinary equivalence relation on the boolean quantale $(\{\mathsf{false},\mathsf{true}\},\leq,\wedge,\mathsf{true})$, and to a pseudometric on the quantale $([0,\infty],\geq,0,+)$).
2. The second improvement is the generalisation of the notion of relator to quantale-valued relators, i.e. relators acting on relations taking values over quantales. Perhaps surprisingly, such generalisation is at the heart of the field of *monoidal topology* [19], a subfield of categorical topology aiming to unify ordered, metric, and topological spaces in categorical terms. Central to the development of monoidal topology is the notion of V-relator or V-lax extension of a monad T which, analogously to the notion of relator, is a construction lifting V-relations on a set X to V-relations on TX. Notable examples of V-relators are obtained from the Hausdorff distance (for the powerset monad) and from the Wasserstein-Kantorovich distance [38] (for the distribution monad).
3. The third improvement (on which we will expand more in the next paragraph) is the development of a *compositional* theory of behavioural V-relations (and thus of behavioural distances). As we are going to see, ensuring compositionality in an higher-order setting is particularly challenging due to the ability of higher-order programs to copy their input several times, a feature that allows them to amplify distances between their inputs *ad libitum*.

The result is an abstract theory of behavioural V-relations that allows to define notions of quantale-valued applicative similarity and bisimilarity parametric in a V-relator. The notions obtained generalise the existing notions of real-valued applicative (bi)similarity and can be instantiated to concrete calculi to provide new notions of applicative distance. A remarkable example is the case of probabilistic λ-calculi, where to the best of the author's knowledge a (non-trivial) applicative distance for a universal (i.e. Turing complete) probabilistic λ-calculus is still lacking in the literature (but see Section 6).

The main theorem of this paper states that under suitable conditions on monads and V-relators the abstract notion of quantale-valued applicative similarity is a compatible—i.e. compositional—reflexive and transitive V-relation. Under mild conditions such result extends to quantale-valued applicative bisimilarity, which is thus proved to be a compatible, reflexive, symmetric, and transitive V-relation (i.e. a compatible pseudometric).

In addition to the concrete results obtained for quantale-valued applicative (bi)similarity, the contribution of the present work also relies on introducing and combining several notions and results developed in different fields (such as monoidal topology, coalgebra, and programming language theory) to build an abstract framework for studying quantitative refinements of behavioural relations for higher-order languages whose applications go beyond the present study of applicative (bi)similarity.

Compositionality, distance amplification, and linear types
Once we have understood what is the behavioural distance $\delta(e,e')$ (which, for the sake of this argument, we assume to be a non-negative real number) between two programs e and e', it is natural to ask if and how much such distance is modified when e and e' are used inside a bigger program—i.e. a context—$C[-]$. Indeed we would like to reason about the distance $\delta(C[e],C[e'])$ *compositionally*, i.e. in terms of the distance $\delta(e,e')$.

Compositionality is at the heart of relational reasoning about program behaviours. Informally, compositionality states that observational indistinguishability is preserved by language constructors; formally, a relation is compositional if it is *compatible* with all language constructors, meaning that whenever two programs e and e' are related, then so are the bigger programs $C[e]$ and $C[e']$.

Analogous to the idea that compatible relations are preserved by language constructors, we are tempted to define as compatible those distances that are not increased by language constructors. That is, we would like to say that a behavioural distance δ is compatible if the distance $\delta(C[e],C[e'])$ between $C[e]$ and $C[e']$ is always bounded by the distance $\delta(e,e')$, no matter how $C[-]$ uses e and e'. However, we soon realise that such proposal cannot work: not only how $C[-]$ uses e and e' matters, but also *how much* it uses them does. This phenomenon, called *distance amplification* [8], can be easily observed when dealing with probabilistic languages. Consider the following example for a probabilistic untyped λ-calculus [10] taken from [8]. Let I be the identity combinator and $I\oplus\Omega$ be the program evaluating to I with probability $\frac{1}{2}$, and diverging with probability $\frac{1}{2}$. Assuming we observe the probability of convergence of a program, it speaks by itself that we would expect the behavioural distance $\delta(I,I\oplus\Omega)$ between I and $I\oplus\Omega$ to be $\frac{1}{2}$. However, it is sufficient to consider a family $\{C_n[-]\}_{n\geq 0}$ of contexts that duplicate their input n-times[1] to see that any such context amplifies the observable distance between I and $I\oplus\Omega$: as n grows, the probability of convergence of $C[I\oplus\Omega]$ tends to zero, whereas the one of $C[I]$ remains always equal to one. During its evaluation, every time the context C_n evaluates its inputs the detected distance between the latter is somehow accumulated to the distances previously observed, thus exploiting the *linear*—in opposition to classical—nature of the act of measuring. Such linearity naturally reflects the monoidal closed structure of categories of metric spaces, in opposition with the cartesian closed structure characterising 'classical' (i.e. boolean-valued) observations.

The above example shows that if we want to reason compositionally about behavioural distances, then we have to accept that contexts can amplify distances, and thus we should take into account the number of times a program accesses its input. More concretely,

[1] For instance $\{(\lambda x.\underbrace{(xI)\ldots(xI)}_{n})(\lambda y.[-])\}_{n\geq 0}$.

our notion of compatibility allows a context $C[-]$ using its input s times to increase the distance $\delta(e, e')$ between e and e', but of a factor *at most* s. That is, the distance $\delta(C[e], C[e'])$ should be bounded by $s \cdot \delta(e, e')$. Our main result states that quantale-valued applicative (bi)similarity is compatible in this sense. This result allows us to reason about behavioural distances compositionally, so that we can e.g. conclude that the distance between I and $I \oplus \Omega$ is indeed $\frac{1}{2}$ (Example 12).

Reasoning about the number of times programs use (or test) their inputs requires a shift from ordinary languages to refined languages tracking information about the so-called *program sensitivity* [12, 31]. The sensitivity of a program is the 'law' describing how much behavioural differences in outputs are affected by behavioural differences in inputs, and thus provides the abstraction needed to handle distance amplification.

Our refined language is a generalisation of the language Fuzz [12, 31], which we call V-Fuzz. Fuzz is a PCF-like language refining standard λ-calculi by means of a powerful linear type system enriched with sensitivity-indexed 'bang types' that allow to track program sensitivity. Despite being parametric with respect to an arbitrary quantale, the main difference between V-Fuzz and Fuzz is that the former is an effectful calculus parametric with respect to a signature of (algebraic) operation symbols. This allows to consider imperative, nondeterministic, and probabilistic versions of Fuzz, as well as combinations thereof.

Structure of the work After having recalled some necessary mathematical preliminaries, we introduce V-Fuzz and its monadic operational semantics (Section 3). We then introduce (Section 4) the machinery of V-relators showing how it can be successfully instantiated on several examples. In Section 5 we define applicative Γ-similarity, a V-relation generalising effectful applicative similarity parametric with respect to a V-relator Γ, and prove it is a reflexive and transitive V-relation whose kernel induces an abstract notion of applicative similarity. Our main theorem states that under suitable conditions on the V-relator Γ, applicative Γ-similarity is compatible. Finally, we define the notion of applicative Γ-bisimilarity and prove that under mild conditions such notion is a compatible equivalence V-relation (viz. a compatible pseudometric).

2 Preliminaries

In this section we recall some basic definitions and results needed in the rest of the paper. Unfortunately, there is no hope to be comprehensive, and thus we assume the reader to be familiar with basic domain theory [2] (in particular we assume the notions of ω-complete (pointed) partial order, ω-cppo for short, monotone, and continuous functions), basic order theory [11], and basic category theory [27]. In particular, for a monoidal category $\langle \mathbb{C}, I, \otimes \rangle$ we assume the reader to be familiar with the notion of *strong Kleisli triple* (or, equivalently, *strong monad*) [21, 27] $\mathbb{T} = \langle T, \eta, -^* \rangle$. We use the notation $f^* : Z \otimes TX \to TY$ for the strong Kleisli extension of $f : Z \otimes X \to TY$ (and use the same notation for the ordinary Kleisli lifting of $f : X \to TY$, the latter being essentially the subcase of $-^*$ for $Z = I$). We denote by $\mathbb{C}_\mathbb{T}$ the Kleisli category of $\mathbb{T}$. Finally, we recall that every monad on **Set**, the category of sets and functions, is strong (with respect to the cartesian structure).

We also try to follow the notation used in the just mentioned references. As a small difference, we denote by $g \cdot f$ the composition of g with f rather than by $g \circ f$. The reader is referred to this paper's long version [16] for formal details.

2.1 Monads and Algebraic Effects

Following [30] we consider algebraic operations as sources of side effects. Syntactically, algebraic operations are given via a signature Σ consisting of a set of operation symbols (uninterpreted operations) together with their arity (i.e. their number of operands). Semantically, operation symbols are interpreted as algebraic operations on strong monads on **Set**. To any n-ary operation symbol $\mathbf{op} \in \Sigma$ and any set X we associate a map $op_X : (TX)^n \to TX$ such that $op_Y(f^*(z, a_1), \ldots, f^*(z, a_1)) = f^*(z, op_X(a_1, \ldots, a_n))$ holds for any $f : Z \times X \to TY$, $z \in Z, a_i \in TX$.

We also use monads to give operational semantics to V-Fuzz [9]. Intuitively, a program e evaluates to a *monadic value* $v \in T\mathcal{V}$, where $\mathcal{V}$ denotes the set of values. The evaluation of a term is defined as the limit of its "finite evaluations", and thus we need monads to carry a suitable domain structure. Recall that any category $\mathbb{C}$ is ω-cppo-enriched if the hom-set $\mathbb{C}(X, Y)$ carries an ω-cppo-structure, for all objects X, Y, and composition is continuous. A (strong) monad $\mathbb{T}$ is ω-cppo-enriched if $\mathbb{C}_\mathbb{T}$ is. In particular, in **Set** that means that we have an ω-cppo $\langle TX, \sqsubseteq_X, \bot_X \rangle$ for any set X. We also require $f^*(z, \bot_X) = \bot_Y$, for f as above. Finally, we say that $\mathbb{T}$ is Σ-*continuous* if it satisfies the above conditions and operations $op_X : (TX)^n \to TX$ are continuous. See [16, 30] for details.

Example 1. The following are Σ-continuous monads:

1. The partiality monad $(-)_\bot$ mapping a set X to $X_\bot \triangleq X + \{\bot_X\}$. We give $X_\bot$ an ω-cppo structure via $\sqsubseteq_X$ defined by $x \sqsubseteq_X y$ if and only if $x = \bot_X$ or $x = y$.
2. The powerset monad mapping a set to its powerset. We give $\mathcal{P}(X)$ an ω-cppo structure via subset inclusion $\subseteq$. We can consider the signature $\Sigma = \{\oplus\}$ consisting of a single binary operation symbol for pure nondeterministic choice and interpret it as set-theoretic union.
3. The discrete *subdistribution* monad $\mathcal{D}_{\leq 1}$ mapping a set X to $\mathcal{D}(X_\bot)$, where $\mathcal{D}$ denotes the discrete distribution monad. On $\mathcal{D}(X_\bot)$, we define the order $\sqsubseteq_X$ by $\mu \sqsubseteq_X \nu$ if and only if for any $x \in X$, $\mu(x) \leq \nu(x)$ holds. The pair $(\mathcal{D}(X_\bot), \sqsubseteq_X)$ forms an ω-cppo, with bottom element given by the Dirac distribution on $\bot_X$ (the distribution modelling the always zero subdistribution). We can consider the signature $\Sigma = \{\oplus_p \mid p \in \mathbb{Q},\ 0 < p < 1\}$ whose interpretation on the subdistribution monad is defined by $(\mu \oplus_p \nu)(x) \triangleq p \cdot \mu(x) + (1 - p) \cdot \nu(x)$. Restricting to $p \triangleq \frac{1}{2}$ we obtain fair probabilistic choice $\oplus$.

2.2 Relations, Metrics, and Quantales

We now recall basic notions on quantales and quantale-valued relations (V-relations) along the lines of [25]. The reader is referred to [16] and [19] for further details.

Definition 1. *A (unital) quantale* $(\mathsf{V}, \leq, \otimes, k)$, V *for short, consists of a monoid* $(\mathsf{V}, \otimes, k)$ *and a sup-lattice* $(\mathsf{V}, \leq)$ *satisfying the following distributivity laws:*

$$b \otimes \bigvee_{i \in I} a_i = \bigvee_{i \in I} (b \otimes a_i), \qquad (\bigvee_{i \in I} a_i) \otimes b = \bigvee_{i \in I} (a_i \otimes b).$$

The element k is called unit, whereas $\otimes$ is called multiplication of the quantale. Given quantales V, W, *a* quantale lax morphism *is a*

monotone *map* $h : \mathsf{V} \to \mathsf{W}$ *satisfying the following inequalities:*

$$\ell \leq h(k), \qquad h(a) \otimes h(b) \leq h(a \otimes b),$$

where ℓ *is the unit of* W.

It is easy to see that $\otimes$ is monotone in both arguments. We denote top and bottom elements of a quantale by $\top$ and $\bot$, respectively. Moreover, we say that a quantale is commutative if its underlying monoid is, and it is non-trivial if $k \neq \bot$. From now on we tacitly assume quantales to be commutative and non-trivial.

Example 2. The following are examples of quantales:

1. The *boolean quantale* $(2, \leq, \wedge, \mathsf{true})$ where $2 = \{\mathsf{true}, \mathsf{false}\}$ and $\mathsf{false} \leq \mathsf{true}$.
2. The extended real half-line $([0, \infty], \geq, +, 0)$ ordered by the "greater or equal" relation $\geq$ and extended[2] addition as monoid multiplication. We refer to such quantale as the *Lawvere quantale*. Note that in the Lawvere quantale the bottom element is ∞, the top element is 0, whereas infimum and supremum are defined as sup and inf, respectively.
3. Replacing addition with maximum in the Lawevere quantale we obtain the *ultrametric Lawvere quantale* $([0, \infty], \geq, \max, 0)$, which has been used to study generalised ultrametric spaces [33] (note that in the ultrametric Lawvere quantale monoid multiplication and binary meet coincide).
4. Restricting the Lawvere quantale to the unit interval we obtain the *unit interval quantale* $([0, 1], \geq, +, 0)$, where $+$ stands for truncated addition.

In all quantales of Example 2 the unit k coincide with the top element (i.e. $k = \top$). Quantales with such property are called *integral quantales*, and are particularly well-behaved. From now on we tacitly assume quantales to be integral.

V-relations For a quantale V, a V-relation $\alpha : X \nrightarrow Y$ between sets X and Y is a function $\alpha : X \times Y \to \mathsf{V}$. For any set X we can define the identity V-relation $id_X : X \nrightarrow X$ mapping diagonal elements (x, x) to k, and all other elements to $\bot$. Moreover, for V-relations $\alpha : X \nrightarrow Y$ and $\beta : Y \nrightarrow Z$, we can define the composition $\beta \cdot \alpha : X \nrightarrow Z$ by:

$$(\beta \cdot \alpha)(x, z) \triangleq \bigvee_{y \in Y} \alpha(x, y) \otimes \beta(y, z).$$

Composition of V-relations is associative, and id is the unit of composition. As a consequence, we have that sets and V-relations form a category, called V-**Rel**. V-**Rel** is a monoidal category with unit given by the one-element set and tensor product given by cartesian product of sets with $\alpha \otimes \beta : X \times Y \nrightarrow X' \times Y'$ defined pointwise, for $\alpha : X \nrightarrow X'$ and $\beta : Y \to Y'$. Moreover, for all sets X, Y, the hom-set V-**Rel**(X, Y) inherits a complete lattice structure from V according to the pointwise order.

There is a bijection $-^\circ : \mathsf{V}\text{-}\mathbf{Rel}(X, Y) \to \mathsf{V}\text{-}\mathbf{Rel}(Y, X)$ that maps each V-relation α to its dual α° defined by $\alpha^\circ(y, x) \triangleq \alpha(x, y)$. Moreover, we can define the graph functor $\mathcal{G}$ from **Set** to V-**Rel** acting as the identity on sets and mapping each function f to its graph (so that $\mathcal{G}(f)(x, y)$ is equal to k if $y = f(x)$, and $\bot$ otherwise). It is easy to see that since V is non-trivial $\mathcal{G}$ is faithful. In light of this observation we will use the notation $f : X \to Y$ in place of $\mathcal{G}(f) : X \nrightarrow Y$ in V-**Rel**.

[2]We extend ordinary addition as follows: $x + \infty \triangleq \infty \triangleq \infty + x$.

Easy calculations (see [19]) show that we can use the pointfree inequality $\alpha \leq g^\circ \cdot \beta \cdot f$, for $\alpha : X \nrightarrow Y$, $\beta : X' \nrightarrow Y'$, $f : X \to X'$, $g : Y \to Y'$, to compactly express the following generalised non-expansiveness condition[3]: $\forall (x, y) \in X \times Y.\ \alpha(x, y) \leq \beta(f(x), g(y))$.

Among V-relations we are interested in those generalising equivalences and pseudometrics.

Definition 2. *A* V*-relation* $\alpha : X \nrightarrow X$ *is* reflexive *if* $id_X \leq \alpha$*, i.e.* $k \leq \alpha(x, y)$, transitive *if* $\alpha \cdot \alpha \leq \alpha$*, i.e.* $\alpha(x, y) \otimes \alpha(y, z) \leq \alpha(x, z)$*, and* symmetric *if* $\alpha \leq \alpha^\circ$*, i.e.* $\alpha(x, y) \leq \alpha(y, x)$*, for all* $x, y, z \in X$.

We call a reflexive and transitive V-relation a V*-preorder* or *generalised metric* [25], and a reflexive, symmetric, and transitive V-relation a V*-equivalence* or *pseudometric*.

Example 3.

1. We see that 2-**Rel** is the ordinary category **Rel** of sets and relations.
2. On the Lawvere quantale, transitivity gives the usual triangle inequality and V-equivalences are precisely (standard) pseudometrics [36]. Instantiating transitivity on the ultrametric Lawvere quantale we recover the usual strong triangle inequality.

Operations For a signature Σ, we need to specify how operations in Σ interact with V-relations (e.g. how they modify distances), and thus how they interact with quantales.

Definition 3. *Let* Σ *be a signature. A* Σ*-quantale is a quantale* V *equipped with* monotone *operations* $op_\mathsf{V} : \mathsf{V}^n \to \mathsf{V}$*, for each* n*-ary operation* $\mathbf{op} \in \Sigma$*, satisfying the following inequalities:*

$$k \leq op_\mathsf{V}(k, \ldots, k),$$
$$op_\mathsf{V}(a_1, \ldots, a_n) \otimes op_\mathsf{V}(b_1, \ldots, b_n) \leq op_\mathsf{V}(a_1 \otimes b_1, \ldots, a_n \otimes b_n).$$

Example 4. Both in the Lawvere quantale and in the unit interval quantale we can interpret operations $\oplus_p$ from Example 1 as probabilistic choices: $x \oplus_p y \triangleq p \cdot x + (1 - p) \cdot y$. In general, for a quantale V we can interpret $op_\mathsf{V}(a_1, \ldots, a_n)$ both as $a_1 \otimes \ldots \otimes a_n$ and $a_1 \wedge \ldots \wedge a_n$.

Change of Base Functors We model sensitivity of a program as a function giving the 'law' describing how distances between inputs are modified by the program. The notion of *change of base functor* provides a mathematical abstraction to model the concept of sensitivity with respect to an arbitrary quantale.

Definition 4. *A change of base functor [19], CBF for short, between quantales* V, W *is a lax quantale morphism* $h : \mathsf{V} \to \mathsf{W}$ *(see Definition 1). If* $\mathsf{V} = \mathsf{W}$ *we speak of change of base* endofunctors *(CBEs, for short), and denote them by* $s, r \ldots$*. Clearly, every CBE* s *is also a CBF.*

The action $h \circ \alpha$ of a CBF $h : \mathsf{V} \to \mathsf{W}$ on a V-relation $\alpha : X \nrightarrow Y$ is defined by $h \circ \alpha(x, y) \triangleq h(\alpha(x, y))$ (to improve readability we omit brackets). Note that since V is integral, CBFs preserve the unit.

Example 5.

1. Extended[4] real-valued multiplication $c \cdot -$, for $c \in [0, \infty]$, is a CBE on the Lawvere quantale. Functions $c \cdot -$ act as CBEs also on the unit interval quantale (where multiplication is meant to be truncated).
2. Both in the Lawvere quantale and in the unit interval quantale, polynomials P such that $P(0) = 0$ are CBEs.

[3]Recall that when we instantiate V as e.g. the Lawvere quantale we have to *reverse* inequalities.

[4]We extend real-valued multiplication by: $0 \cdot \infty \triangleq 0 \triangleq \infty \cdot 0$, $\infty \cdot x \triangleq \infty \triangleq x \cdot \infty$.

3. Define CBEs $n, \infty : \mathsf{V} \to \mathsf{V}$, for $n < \omega$ by $0(a) \triangleq k$, $(n+1)(a) \triangleq a \otimes n(a)$, and $\infty(a) \triangleq \mathbb{1}$. Note that 1 acts as the identity function.

Lemma 1. *Let* V *be a* Σ*-quantale. CBEs are closed under the following operations (where* $\mathbf{op} \in \Sigma$*):*

$$(s \otimes r)(a) \triangleq s(a) \otimes r(a),$$
$$(r \cdot s)(a) \triangleq r(s(a)),$$
$$(s \wedge r)(a) = s(a) \wedge s(b),$$
$$op_{\mathsf{V}}(s_1, \ldots, s_n)(a) \triangleq op_{\mathsf{V}}(s_1(a), \ldots, s_n(a)).$$

3 The V-fuzz Language

As already observed in the introduction, when dealing with behavioural V-relations a crucial parameter in amplification phenomena is program sensitivity. To deal with such parameter we introduce V-fuzz, a higher-order effectful language generalising Fuzz [12]. As Fuzz, V-Fuzz is characterised by a powerful type system inspired by *bounded linear logic* [18] giving syntactic information on program sensitivity.

Syntax V-fuzz is a *fine-grained call-by-value* [26] *linear* λ-calculus with finite sum and recursive types. In particular, we make a formal distinction between values and computations (which we simply refer to as terms), and use syntactic primitives to returning values (**val**) and sequentially compose computations (via a **let-in** constructor). The syntax of V-Fuzz is parametrised over a signature Σ of operation symbols, a Σ-quantale V, and a family Π of CBEs. From now on we assume Σ, V, and Π to be fixed. Moreover, we assume Π to contain at least CBEs n, ∞ in Example 5 and to be closed under operations in Lemma 1. Types, values, and terms of V-Fuzz are defined in Figure 1, where t denotes a type variable, I is a *finite* set (whose elements are denoted by $\hat{\imath}, \hat{\jmath}, \ldots$), and s is in Π.

$$\begin{aligned}
\sigma &::= t \mid \textstyle\sum_{i \in I} \sigma_i \mid \sigma \multimap \sigma \mid \mu t.\sigma \mid !_s\sigma.\\
v &::= x \mid \lambda x.e \mid \langle \hat{\imath}, v \rangle \mid \mathbf{fold}\ v \mid !v.\\
e &::= \mathbf{val}\ v \mid ve \mid \mathbf{case}\ v\ \mathbf{of}\ \{\langle i, x\rangle \to e_i\} \mid \mathbf{let}\ x = e\ \mathbf{in}\ e\\
&\quad \mid \mathbf{case}\ v\ \mathbf{of}\ \{!x \to e\} \mid \mathbf{case}\ v\ \mathbf{of}\ \{\mathbf{fold}\ x \to e\} \mid \mathbf{op}(e, \ldots, e).
\end{aligned}$$

Figure 1. Types, values, and terms of V-Fuzz.

Free and bound variables in terms and values are defined as usual. We work with equivalence classes of terms modulo renaming and tacitly assume conventions on bindings. Moreover, we denote by $w[v/x]$ and $e[x := v]$ the *value* and *term* obtained by capture-avoiding substitution of the *value* v for x in w and e, respectively (see [9] for details).

Similar conventions hold for types. In particular, we denote by $\sigma[\tau/t]$ the result of capture-avoiding substitution of type τ for the type variable t in σ. Finally, we write $\mathbf{0}$ for the empty sum type, $\mathbf{1}$ for $\mathbf{0} \multimap \mathbf{0}$, and nat for $\mu t.\mathbf{1} + t$. We denote the numeral n by $\underline{n}$.

V-Fuzz type system is essentially based on judgments of the form $x_1 :_{s_1} \sigma_1, \ldots, x_n :_{s_n} \sigma_n \vdash e : \sigma$, where $s_1, \ldots, s_n$ are CBEs. The informal meaning of such judgment is that on input x_i ($i \leq n$), the term e has sensitivity s_i. That is, e amplifies the (behavioural) distance $\alpha(v_i, w_i)$ between two input values v_i, w_i of *at most* a factor s_i; symbolically, $s_i \circ \alpha(v_i, w_i) \leq \alpha(e[x_i := v_i], e[x_i := w_i])$.

$$\frac{s \leq 1}{\Gamma, x :_s \sigma \vdash^{\mathsf{v}} x : \sigma} \qquad \frac{\Gamma_1 \vdash e_1 : \sigma \quad \cdots \quad \Gamma_n \vdash e_n : \sigma}{op_{\mathsf{V}}(\Gamma_1, \ldots, \Gamma_n) \vdash \mathbf{op}(e_1, \ldots, e_n) : \sigma}$$

$$\frac{\Gamma, x :_1 \sigma \vdash e : \tau}{\Gamma \vdash^{\mathsf{v}} \lambda x.e : \sigma \multimap \tau} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : \sigma \multimap \tau \quad \Delta \vdash^{\mathsf{v}} w : \sigma}{\Gamma \otimes \Delta \vdash vw : \tau}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \sigma_{\hat{\imath}}}{\Gamma \vdash^{\mathsf{v}} \langle \hat{\imath}, v \rangle : \sum_{i \in I} \sigma_i} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : \sum_{i \in I} \sigma_i \quad \Delta, x :_s \sigma_i \vdash e_i : \tau \quad (\forall i \in I)}{s \cdot \Gamma \otimes \Delta \vdash \mathbf{case}\ v\ \mathbf{of}\ \{\langle i, x\rangle \to e_i\} : \tau}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \sigma}{\Gamma \vdash \mathbf{val}\ v : \sigma} \qquad \frac{\Gamma \vdash e : \sigma \quad \Delta, x :_s \sigma \vdash f : \tau}{(s \wedge 1) \cdot \Gamma \otimes \Delta \vdash \mathbf{let}\ x = e\ \mathbf{in}\ f : \tau}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \sigma}{s \cdot \Gamma \vdash^{\mathsf{v}} !v : !_s\sigma} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : !_r\sigma \quad \Delta, x :_{s \cdot r} \sigma \vdash e : \tau}{s \cdot \Gamma \otimes \Delta \vdash \mathbf{case}\ v\ \mathbf{of}\ \{!x \to e\} : \tau}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \sigma[\mu t.\sigma/t]}{\Gamma \vdash^{\mathsf{v}} \mathbf{fold}\ v : \mu t.\sigma} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : \mu t.\sigma \quad \Delta, x :_s \sigma[\mu t.\sigma/t] \vdash e : \tau}{s \cdot \Gamma \otimes \Delta \vdash \mathbf{case}\ v\ \mathbf{of}\ \{\mathbf{fold}\ x \to e\} : \tau}$$

Figure 2. Typing rules.

An *environment* Γ is a sequence $x_1 :_{s_1} \sigma_1, \ldots, x_n :_{s_n} \sigma_n$ of distinct identifiers with associated closed types and CBEs (we denote the empty environment by $\emptyset$). We can lift operations on CBEs in Lemma 1 to environments as follows:

$$r \cdot \Gamma = x_1 :_{r \cdot s_1} \sigma_1, \ldots, x_n :_{r \cdot s_n} \sigma_n,$$
$$\Gamma \otimes \Delta = x_1 :_{s_1 \otimes r_1} \sigma_1, \ldots, x_n :_{s_n \otimes r_n} \sigma_n,$$
$$op_{\mathsf{V}}(\Gamma^1, \ldots, \Gamma^m) = x_1 :_{op_{\mathsf{V}}(s_1^1, \ldots, s_1^m)} \sigma_1, \ldots, x_n :_{op_{\mathsf{V}}(s_n^1, \ldots, s_n^m)} \sigma_n,$$

for $\Gamma = x_1 :_{s_1} \sigma_1, \ldots, x_n :_{s_n} \sigma_n$, $\Delta = x_1 :_{r_1} \sigma_1, \ldots, x_n :_{r_n} \sigma_n$, and $\Gamma^i = x_1 :_{s_1^i} \sigma_1, \ldots, x_n :_{s_n^i} \sigma_n$. Note that the above operations are defined for environments having the same structure (i.e. differing only on CBEs). This is not a real restriction since we can always add the missing identifiers $y :_k \sigma$, where k is the constant function returning the unit of the quantale (but see [31]).

The type system for V-Fuzz is defined in Figure 2. The system is based on two kinds of judgment (exploiting the fine-grained style of the calculus): judgments of the form $\Gamma \vdash^{\mathsf{v}} v : \sigma$ for values and judgments of the form $\Gamma \vdash e : \sigma$ for terms. We denote by $\mathcal{V}_\sigma$ and Λ_σ for the set of closed values and terms of type σ, respectively.

Example 6.
1. Instantiating V-Fuzz with $\Sigma \triangleq \emptyset$, the Lawvere quantale, and CBEs $\Pi = \{c \cdot - \mid c \in [0, \infty]\}$ we obtain the original Fuzz [31] (provided we add a basic type for real numbers). We can also add nondeterminism via a binary nondeterminism choice operation $\oplus$.
2. We define the language P-Fuzz as the instantiation of V-Fuzz with a fair probabilistic choice operation $\oplus$, the unit interval quantale $([0,1], \geq, +, 0)$, and CBEs $\Pi = \{c \cdot - \mid c \in [0, \infty]\}$ (as usual we are actually referring to truncated multiplication). We interpret $\oplus$ in $[0,1]$ as in Example 4.

Typing rules for V-Fuzz are similar to those of Fuzz (e.g. in the variable rule we require $s \leq 1$, meaning that the open value x can access x at least once) with the exception of the rule for sequencing where we apply sensitivity $s \wedge 1$ to the environment Γ even if the sensitivity of x in f is s. Consider the following instance of the sequencing rule on the Lawvere quantale:

$$\frac{x :_1 \sigma \vdash e : \sigma \quad y :_0 \sigma \vdash f : \tau}{x :_{\max(0,1) \cdot 1} \sigma \vdash \mathbf{let}\ y = e\ \mathbf{in}\ f : \tau}$$

$$
\begin{aligned}
|e|_0^\sigma &\triangleq \bot_{\mathcal{V}_\sigma}\\
|\mathbf{val}\ v|_{n+1}^\sigma &\triangleq \eta_{\mathcal{V}_\sigma}(v)\\
|(\lambda x.e)v|_{n+1}^\sigma &\triangleq |e[x := v]|_n^\sigma\\
|\mathbf{case}\ \langle i, v\rangle\ \mathbf{of}\ \{\langle i, x\rangle \to e_i\}|_{n+1}^\sigma &\triangleq |e_i[x := v]|_n^\sigma\\
|\mathbf{case}\ (\mathbf{fold}\ v)\ \mathbf{of}\ \{\mathbf{fold}\ x \to e\}|_{n+1}^\sigma &\triangleq |e[x := v]|_n^\sigma\\
|\mathbf{case}\ !v\ \mathbf{of}\ \{!x \to e\}|_{n+1}^\sigma &\triangleq |e[x := v]|_n^\sigma\\
|\mathbf{let}\ x = e\ \mathbf{in}\ f|_{n+1}^\sigma &\triangleq (|f[x := -]|_n^{\tau,\sigma})^*|e|_n^\tau\\
|\mathbf{op}(e_1, \dots, e_k)|_{n+1}^\sigma &\triangleq op_{\mathcal{V}_\sigma}(|e_1|_n^\sigma, \dots, |e_k|_n^\sigma)
\end{aligned}
$$

Figure 3. Approximation evaluation semantics.

where f is a closed term of type τ and thus we can assume it to have sensitivity 0 on all variables. According to our informal intuition, e has sensitivity 1 on input x, meaning that (i) e can possibly detect (behavioural) differences between input values v, w, and (ii) e cannot amplify their behavioural distance of a factor bigger than 1. Formally, point (ii) states that we have the inequality $\alpha(v, w) \geq \alpha(e[x := v], e[x := w])$, where α denotes a suitable behavioural $[0, 1]$-relation. On the contrary, f is closed term and thus has sensitivity 0 on any input, meaning that it cannot detect any observable difference between input values. In particular, for all values v, w we have $\alpha(f[y := v], f[y := w]) = \alpha(f, f) = 0$ (provided that α is reflexive). Replacing $\max(0, 1)$ with 0 in the above rule (i.e. $s \wedge 1$ with s in the general case) would allow to infer the judgment $x :_0 \sigma \vdash \mathbf{let}\ y = e\ \mathbf{in}\ f : \tau$, and thus to conclude $\alpha(\mathbf{let}\ y = e[x := v]\ \mathbf{in}\ f, \mathbf{let}\ y = e[x := w]\ \mathbf{in}\ f) = 0$. The latter equality is unsound as evaluating $\mathbf{let}\ y = e[x := v]\ \mathbf{in}\ f$ (resp. $\mathbf{let}\ y = e[x := w]\ \mathbf{in}\ f$) requires to *first* evaluate $e[x := v]$ (resp. $e[x := w]$) thus making observable differences between v and w detectable (see also Section 5 for a formal explanation).

Example 7. For every type σ we have the term I defined as $\mathbf{val}\ (\lambda x.\mathbf{val}\ x)$ of type $\sigma \multimap \sigma$ as well as the purely divergent term $\Omega \triangleq \omega!(\mathbf{fold}\ \omega)$ of type σ, where $\omega \in \Lambda_{!_\infty(\mu t.!_\infty t \multimap \sigma) \multimap \sigma}$ is defined as $\lambda x.\mathbf{case}\ x\ \mathbf{of}\ \{!y \to \mathbf{case}\ y\ \mathbf{of}\ \{\mathbf{fold}\ z \to z!(\mathbf{fold}\ z)\}\}$.

Operational Semantics We give V-Fuzz monadic operational (notably evaluation) semantics in the style of [9]. Let $\mathbb{T} = \langle T, \eta, -^*\rangle$ be a Σ-continuous monad. Operational semantics is defined by means of an evaluation function $|-|^\sigma$ indexed over closed types, associating to any term in Λ_σ a monadic value in $T\mathcal{V}_\sigma$. The evaluation function $|-|^\sigma$ is itself defined by means of the family of functions $\{|-|_n^\sigma\}_{n<\omega}$ defined in Figure 3. Indeed $|-|_n^\sigma$ is a function from $\mathcal{V}_\sigma$ to $T\mathcal{V}_\sigma$.

Let us expand on the definition of $|\mathbf{let}\ x = e\ \mathbf{in}\ f|_{n+1}^\sigma$. Since $\mathbf{let}\ x = e\ \mathbf{in}\ f \in \Lambda_\sigma$, there must be derivable judgments $\emptyset \vdash e : \tau$ and $x :_s \tau \vdash f : \sigma$. As a consequence, for any $v \in \mathcal{V}_\tau$, we have $|f[x := v]|_n^\sigma \in T\mathcal{V}_\sigma$. This induces a function $|f[x := -]|_n^{\tau,\sigma}$ from $\mathcal{V}_\tau$ to $T\mathcal{V}_\sigma$ whose Kleisli extension can be applied to $|e|_n^\tau \in T\mathcal{V}_\tau$.

Finally, it is easy to see that $(|e|_n)_{n<\omega}$ forms an ω-chain in $T\mathcal{V}_\sigma$, so that we can define $|-|^\sigma : \Lambda_\sigma \to T\mathcal{V}_\sigma$ by $|e|^\sigma \triangleq \bigsqcup_{n<\omega} |e|_n^\sigma$. In order to improve readability we oftentimes omit type superscripts in $|e|^\sigma$. We also notice that because op is continuous and $\mathbb{T}$ is ω-cppo-enriched, $|-|^\sigma$ is itself continuous, meaning that we can instantiate the equalities in Figure 3 with $|-|^\sigma$ in place of $|-|_n^\sigma$ (and $=$ in place of $\triangleq$).

4 V-relators and V-relation Lifting

In this section we generalise the abstract theory of relators [4, 37] to V-relators [19]. Informally, a V-relator for a set endofunctor T is an abstraction meant to capture the possible ways a V-relation on a set X can be (nicely) turned into a V-relation on TX; formally, a V-relator extends T from **Set** to V-**Rel**, laxly.

Definition 5. *For a set endofucunctor T a* V-relator *for T is a mapping $(\alpha : X \nrightarrow Y) \mapsto (\Gamma\alpha : TX \nrightarrow TY)$ satisfying conditions* (V-rel 1)-(V-rel 4). *We say that Γ is conversive if it additionally satisfies condition* (V-rel 5).

$$1_{TX} \leq \Gamma(1_X), \tag{V-rel 1}$$

$$\Gamma\beta \cdot \Gamma\alpha \leq \Gamma(\beta \cdot \alpha), \tag{V-rel 2}$$

$$Tf \leq \Gamma f, \quad (Tf)^\circ \leq \Gamma f^\circ, \tag{V-rel 3}$$

$$\alpha \leq \beta \implies \Gamma\alpha \leq \Gamma\beta, \tag{V-rel 4}$$

$$\Gamma(\alpha^\circ) = (\Gamma\alpha)^\circ. \tag{V-rel 5}$$

Conditions (V-rel 1), (V-rel 2), and (V-rel 4) are rather standard. Condition (V-rel 3) states that V-relators behave in the expected way on functions. Before giving examples of V-relators it is useful to observe that the collection V-relators is closed under specific operations.

Proposition 1. *Let T, U be set endofunctors. Then:*

1. *If Γ and Δ are V-relators for T and U, respectively, then $\Delta \cdot \Gamma$ defined by $(\Delta \cdot \Gamma)\alpha \triangleq \Delta\Gamma\alpha$ is a V-relator for UT.*
2. *If $\{\Gamma\}_{i\in I}$ is a family of V-relators for T, then $\bigwedge_{i\in I}\Gamma_i$ defined by $(\bigwedge_{i\in I}\Gamma_i)\alpha \triangleq \bigwedge_{i\in I}\Gamma_i\alpha$ is a V-relator for T.*
3. *If Γ is a V-relator for T, then Γ° defined by $\Gamma^\circ\alpha \triangleq (\Gamma\alpha^\circ)^\circ$ is a V-relator for T.*
4. *For any V-relator Γ, $\Gamma \wedge \Gamma^\circ$ is the greatest conversive V-relator smaller than Γ.*

Example 8. Let us consider the monads in Example 1 regarded as functors.

1. For the partiality functor $(-)_\perp$ define the V-relator $(-)_\perp$ by:
$$\alpha_\perp(x, y) \triangleq \alpha(x, y), \quad \alpha_\perp(\perp_X, y) \triangleq k, \quad \alpha_\perp(x, \perp_Y) = \perp\!\!\!\perp,$$
where $x \in X, y \in Y, y \in Y_\perp$, and $\alpha : X \nrightarrow Y$. The V-relation $\alpha_\perp$ generalises the usual notion of *simulation* for partial computations. Similarly, $\alpha_{\perp\perp} \triangleq \alpha_\perp \wedge ((\alpha^\circ)_\perp)^\circ$ generalises the usual notion of *bisimulation* for partial computation.
2. For the powerset functor $\mathcal{P}$ define the V-relator H (called Hausdorff lifting) and its conversive counterpart $H^s \triangleq H \wedge H^\circ$ by $H\alpha(\mathcal{X}, \mathcal{Y}) \triangleq \bigwedge_{x\in\mathcal{X}} \bigvee_{y\in\mathcal{Y}} \alpha(x, y)$. If we instantiate V as the Lawvere quantale, then H^s gives the usual Hausdorff lifting of distances on a set X to distances on $\mathcal{P}X$, whereas for $\mathsf{V} = 2$ we recover the usual notion of *(bi)simulation* for unlabelled transition systems.
3. For the full distribution functor $\mathcal{D}$ we define a $[0, 1]$-relator (with respect to the unit interval quantale) using the so-called *Wasserstein-Kantorovich lifting* [38]. For $\mu \in \mathcal{D}(X), \nu \in \mathcal{D}(Y)$, the set $\Omega(\mu, \nu)$ of *couplings* of μ and ν is the set of joint distributions $\omega \in \mathcal{D}(X \times Y)$ such that $\mu = \sum_{y\in Y} \omega(-, y)$ and $\nu = \sum_{x\in X} \omega(x, -)$. For a $[0, 1]$-relation $\alpha : X \nrightarrow Y$ define:
$$W\alpha(\mu, \nu) \triangleq \inf_{\omega\in\Omega(\mu,\nu)} \sum\nolimits_{x,y} \alpha(x, y) \cdot \omega(x, y).$$

$W\alpha(\mu,\nu)$ attains its infimum and has a dual characterisation.

Proposition 2. *Let $\mu \in \mathcal{D}(X), \nu \in \mathcal{D}(Y)$ be countable distributions and $\alpha : X \nrightarrow Y$ be a $[0,1]$-relation. Then:*

$$\begin{aligned} W\alpha(\mu,\nu) &= \min\{\sum_{x,y} \alpha(x,y)\cdot\omega(x,y) \mid \omega \in \Omega(\mu,\nu)\} \\ &= \max\{\sum_x a_x\cdot\mu(x) + \sum_y b_y\cdot\nu(y) \\ &\qquad \mid a_x + b_y \le \alpha(x,y), a_x, b_y \text{ bounded}\}, \end{aligned}$$

where a_x, b_y bounded means that there exist $\bar{a},\bar{b}\in\mathbb{R}$ such that $\forall x.\ a_x \le \bar{a}$, and $\forall y.\ b_y \le \bar{b}$.

The above proposition is a direct consequence of the Duality Theorem for *countable* transportation problems [22] (Theorem 2.1 and 2.2). Using Proposition 2 we can show that W indeed defines a $[0,1]$-relator (see [16]). Finally, we can compose the Wasserstein lifting W with the V-relator $(-)_\perp$ of point 1 obtaining the (non-conversive) $[0,1]$-relator $W_\perp$ for the countable subdistribution functor $\mathcal{D}_{\le 1}$.

V-relators for Strong Monads In previous paragraphs we defined V-relators for functors. Since we model effects through strong monads it is natural to work with V-relators for strong monads.

Definition 6. *Let $\mathbb{T} = \langle T, \eta, -^*\rangle$ be a strong monad on* **Set***, and Γ be a V-relator for T (regarded as a functor). We say that Γ is an L-continuous*[5] *V-relator for $\mathbb{T}$ if it satisfies the following conditions for any CBE $s \le 1$.*

$$\alpha \le \eta^\circ_Y \cdot \Gamma\alpha \cdot \eta_X, \qquad \text{(Lax unit)}$$

$$\gamma \otimes (s\circ\alpha) \le g^\circ\cdot\Gamma\beta\cdot f \implies \gamma\otimes(s\circ\Gamma\alpha) \le (g^*)^\circ\cdot\Gamma\beta\cdot f^*, \qquad (L\text{-Strong lax bind})$$

The condition $s \le 1$ reflects the presence of $s \wedge 1$ in the typing rule for sequencing. Also notice that by taking $s \triangleq 1$, conditions (Lax unit) and (L-Strong lax bind) are equivalent to requiring unit, multiplication, and strength of $\mathbb{T}$ to be non-expansive.

Example 9. It is easy to check that V-relators for the partiality and the powerset monads satisfy conditions in Definition 6. Using Proposition 2 it is possible to show that also the Wasserstein lifting(s) W and $W_\perp$ do, although this is less trivial (see [16]).

Finally, if $\mathbb{T}$ is Σ-continuous we require V-relators for $\mathbb{T}$ to be compatible with the Σ-continuous structure.

Definition 7. *Let $\mathbb{T}$ be a Σ-continuous monad, V be a Σ-quantale, and Γ be a V-relator for $\mathbb{T}$. We say that Γ is* Σ-compatible *and* inductive *if the following inequalities hold:*

$$op_{\mathsf{V}}(\Gamma\alpha(u_1,y_1),\dots\Gamma\alpha(u_n,y_n)) \le \Gamma\alpha(op_X(u_1,\dots,u_n), op_Y(y_1,\dots,y_n)),$$
$$k \le \Gamma\alpha(\perp_X, y),$$
$$\bigwedge_n \Gamma\alpha(x_n,y) \le \Gamma\alpha(\bigsqcup_n x_n, y).$$

for any ω-chain $(x_n)_{n<\omega}$ and elements $u_1,\dots,u_n$ in TX, elements $y, y_1,\dots,y_n \in TY$, n-ary operation symbol $\mathbf{op}\in\Sigma$, and V-relation $\alpha : X \nrightarrow Y$.

In particular, if Γ is inductive and $a \le \Gamma\alpha(x_n, y))$ holds for any $n<\omega$, then $a \le \Gamma\alpha(\bigsqcup_{n<\omega} x_n, y)$.

[5] Instantiating V as the Lawvere quantale, we see that condition (L-Strong lax bind) is requiring Lipshitz continuity of multiplication and strength of $\mathbb{T}$.

Example 10. Easy calculations show that $(-)_\perp$ and H are inductive and Σ-compatible. Using results from [38] and [5] (Lemma 5.2) it is possible to show that $W_\perp$ is inductive, the relevant inequality being $W_\perp\alpha(\sup_n \mu_n, \nu) \le \sup_n W_\perp\alpha(\mu_n,\nu)$. Proving Σ-compatibility is straightforward.

From V-relators to 2-relators Before applying the abstract theory of V-relators to V-Fuzz we show how a V-relator induces a canonical 2-relator (this will be useful in the next section). Consider the maps:

$$\varphi : \mathsf{V} \to 2 \qquad \psi : 2 \to \mathsf{V}$$
$$k \mapsto \text{true},\ a \mapsto \text{false} \qquad \text{true} \to k,\ \text{false} \to \perp\!\!\!\perp$$

We immediately see that φ and ψ are CBFs. We associate to every V-relation α its kernel 2-relation $\varphi\circ\alpha$ and to any 2-relation $\mathcal{R}$ the V-relation $\psi\circ\mathcal{R}$. Similarly, we can associate to each V-relator Γ the 2-relator $\Delta_\Gamma\mathcal{R} \triangleq \varphi\circ\Gamma(\psi\circ\mathcal{R})$. Finally, we say that Γ is compatible with φ if $\Delta_\Gamma(\varphi\circ\alpha) = \varphi\circ\Gamma\alpha$ holds for any $\alpha : X \nrightarrow Y$.

Example 11. All V-relators in Example 8 are compatible with φ. Moreover, it is easy to see that Δ_Γ coincide with standard 2-relators for (applicative) simulation [9]. For instance, $\Delta_W\mathcal{R}(\mu,\nu)$ holds if and only if there exists a coupling $\omega\in\Omega(\mu,\nu)$ such that $\omega(x,y) > 0$ implies $\mathcal{R}(x,y) = \text{true}$, for all x, y. The latter is nothing but the usual probabilistic relation lifting via couplings [23].

5 Behavioural V-relations

In this section we extend the relational theory developed in e.g. [24] for higher-order functional languages to V-relations for V-Fuzz. Following [29] we refer to such relations as *λ-term V-relations*. Among such V-relations we define applicative Γ-similarity and bisimilarity, the generalisation of Abramsky's applicative (bi)similarity to both algebraic effects and V-relations, and prove that under suitable conditions they are both compatible. As usual we assume a signature Σ, a Σ-quantale V, a collection of CBEs Π (according to Section 3), and a Σ-continuous (strong) monad $\mathbb{T}$ to be fixed. We also assume V-relators to satisfy all requirements given in Section 4.

Definition 8. *1. A* closed *λ-term V-relation $\alpha = (\alpha^\Lambda, \alpha^{\mathsf{V}})$ associates to each closed type σ, binary V-relations $\alpha^{\mathsf{V}}_\sigma, \alpha^\Lambda_\sigma$ on closed values and terms inhabiting it, respectively.*

2. An open *λ-term V-relation α associates to each (term) sequent $\Gamma \vdash \sigma$ a V-relation $\Gamma \vdash \alpha(-,-) : \sigma$ on terms inhabiting it, and to each value sequent $\Gamma \vdash^{\mathsf{V}} \sigma$ a V-relation $\Gamma \vdash^{\mathsf{V}} \alpha(-,-) : \sigma$ on values inhabiting it. We require open λ-term V-relations to be closed under weakening, i.e. for any environment Δ we require:*

$$(\Gamma \vdash \alpha(e,f) : \sigma) \le (\Gamma\otimes\Delta \vdash \alpha(e,f) : \sigma),$$
$$(\Gamma \vdash^{\mathsf{V}} \alpha(v,w) : \sigma) \le (\Gamma\otimes\Delta \vdash^{\mathsf{V}} \alpha(v,w) : \sigma).$$

Since the syntactic shape of expressions determines whether we are dealing with terms or values, oftentimes we will omit superscripts from λ-term V-relations.

Both the discrete and the indiscrete V-relations are open λ-term V-relations. Moreover, the collection of open λ-term V-relations carries a complete lattice structure (with respect to the pointwise order). Finally, we can always extend a closed λ-term V-relation $\alpha = (\alpha^\Lambda, \alpha^{\mathsf{V}})$ to an open one.

Definition 9. *Let $\Gamma \triangleq x_1 :_{s_1} \sigma_1, \dots, x_n :_{s_n} \sigma_n$ be an environment. For values $\vec{v} \triangleq v_1,\dots,v_n$ we write $\vec{v} : \Gamma$ if for any $i \le n$, $\emptyset \vdash^{\mathsf{V}} v_i : \sigma_i$*

holds. Given a closed λ-term V-relation $\alpha = (\alpha^{\Lambda}, \alpha^{\mathcal{V}})$ we define its open extension α^{o} as follows[6]*:*

$$\Gamma \vdash \alpha^{o}(e,f) : \tau \triangleq \bigwedge\nolimits_{\vec{v}:\Gamma} \alpha^{\Lambda}_{\tau}(e[\vec{x} := \vec{v}], f[\vec{x} := \vec{v}])$$
$$\Gamma \vdash^{\mathsf{v}} \alpha^{o}(v,w) : \tau \triangleq \bigwedge\nolimits_{\vec{u}:\Gamma} \alpha^{\mathcal{V}}_{\tau}(v[\vec{u}/\vec{x}], w[\vec{u}/\vec{x}]).$$

We now define *applicative* Γ*-similarity.*

Definition 10. *Let Γ be a V-relator and $\alpha = (\alpha^{\Lambda}, \alpha^{\mathcal{V}})$ be a closed λ-term V-relation. Define the closed λ-term V-relation $[\alpha] = ([\alpha]^{\Lambda}, [\alpha]^{\mathcal{V}})$ as follows:*

$$[\alpha]^{\Lambda}_{\sigma}(e,f) \triangleq \Gamma\alpha^{\mathcal{V}}_{\sigma}(|e|,|f|),$$
$$[\alpha]^{\mathcal{V}}_{\sigma\multimap\tau}(v,w) \triangleq \bigwedge\nolimits_{u\in\mathcal{V}_{\sigma}} \alpha^{\Lambda}_{\tau}(vu, wu),$$
$$[\alpha]^{\mathcal{V}}_{\sum_{i\in I}\sigma_i}(\langle \hat{\imath}, v\rangle, \langle \hat{\imath}, w\rangle) \triangleq \alpha^{\mathcal{V}}_{\sigma_i}(v,w),$$
$$[\alpha]^{\mathcal{V}}_{\sum_{i\in I}\sigma_i}(\langle \hat{\imath}, v\rangle, \langle \hat{\jmath}, w\rangle) \triangleq \perp\!\!\!\perp,$$
$$[\alpha]_{\mu t.\sigma}(\mathbf{fold}\ v, \mathbf{fold}\ w) \triangleq \alpha_{\sigma[\mu t.\sigma/t]}(v,w),$$
$$[\alpha]_{!_s\sigma}(!v, !w) \triangleq (s \circ \alpha_{\sigma})(v,w).$$

(notice that the definition of $[\alpha]^{\mathcal{V}}$ is by case analysis on $\emptyset \vdash^{\mathsf{v}} v, w : \sigma$). A λ-term V-relation α is an applicative Γ-simulation if $\alpha \leq [\alpha]$.

The clause for $\sigma \multimap \tau$ generalises the usual applicative clause, whereas the clause for $!_s\sigma$ 'scale' $\alpha^{\mathcal{V}}_{\sigma}$ by s. It is easy to see that the above definition induces a map $\alpha \mapsto [\alpha]$ on the complete lattice of closed λ-term V-relations. Moreover, such map is monotone since both Γ and CBEs are.

Definition 11. *Define applicative Γ-similarity δ as the greatest fixed point of $\alpha \mapsto [\alpha]$. That is, δ is the greatest (closed) λ-term V-relation satisfying the equation $\alpha = [\alpha]$ (such greatest solution exists by the Knaster-Tarski Theorem).*

Applicative Γ-similarity comes with an associated coinduction principle: for any closed λ-term V-relation α, if $\alpha \leq [\alpha]$, then $\alpha \leq \delta$.

Example 12. Instantiating Definition 11 with the Wasserstein lifting $W_{\perp}$ we obtain the quantitative analogue of *probabilistic applicative similarity* [10] for P-Fuzz. In particular, for two terms $e, f \in \Lambda_{\sigma}$, $\delta(e,f)$ is (for readability we omit subscripts):

$$\min_{\omega\in\Omega(|e|,|f|)} \sum_{v,w\in\mathcal{V}} \omega(v,w)\cdot\delta^{\mathcal{V}}(v,w) + \sum_{v\in\mathcal{V}} \omega(v,\perp)\cdot\delta^{\mathcal{V}}_{\perp}(v,\perp)$$
$$+ \sum_{w\in\mathcal{V}} \omega(\perp,w)\cdot\delta^{\mathcal{V}}_{\perp}(\perp,w) + \omega(\perp,\perp)\cdot\delta^{\mathcal{V}}_{\perp}(\perp,\perp).$$

The above formula can be simplified observing that we have $\delta^{\mathcal{V}}_{\perp}(\perp,\perp) = 0$, $\delta^{\mathcal{V}}_{\perp}(v,\perp) = 1$, and $\delta^{\mathcal{V}}_{\perp}(\perp,w) = 0$ by the very definition of $\delta_{\perp}$. We immediately notice that δ is adequate in the following sense: for all terms $e, f \in \Lambda_{\sigma}$ we have the inequality

$$\sum |e| - \sum |f| \leq \delta^{\Lambda}(e,f),$$

where $\sum |e|$ is the probability of convergence of e, i.e. $\sum_{v\in\mathcal{V}} |e|(v)$, and subtraction is actually truncated subtraction.

Let us now consider terms $I, \Omega \in \Lambda_{\sigma\multimap\sigma}$ of Example 7. We claim that $\delta^{\Lambda}(I, I\oplus\Omega) = \frac{1}{2}$. By adequacy we immediately see that $\frac{1}{2} \leq \delta^{\Lambda}(I, I\oplus\Omega)$. We prove $\delta^{\Lambda}(I, I\oplus\Omega) \leq \frac{1}{2}$. Let $v \triangleq \lambda x.\mathbf{val}\ x$ and consider the coupling ω defined by:

$$\omega(v,v) = \frac{1}{2}, \quad \omega(v,\perp) = \frac{1}{2}$$

and zero for the rest. Indeed ω is a coupling of $|I|$ and $|I\oplus\Omega|$. Moreover, by the very definition of δ and $W_{\perp}$ we have:

$$\delta^{\Lambda}(I, I\oplus\Omega) \leq \omega(v,v)\cdot\delta^{\mathcal{V}}(v,v) + \omega(v,\perp).$$

The right hand side of the above inequality gives exactly $\frac{1}{2}$, provided that $\delta^{\mathcal{V}}(v,v) = 0$. This indeed holds in full generality.

Proposition 3. *Applicative Γ-similarity δ is a reflexive and transitive λ-term V-relation.*

Proof sketch. The proof is by coinduction using the lax equations characterising V-relators (Definition 5) and CBEs (Definition 4). □

In light of Example 11 we can look at the kernel of δ and recover well-known notions of (relational) applicative similarity (properly generalised to V-Fuzz).

Proposition 4. *Define applicative Δ_{Γ}-similarity $\preceq$ by instantiating Definition 10 with the 2-relator Δ_{Γ} and replacing the clause for types of the form $!_s\sigma$ as follows: $!v\ \mathcal{R}_{!_s\sigma}\ !w$ implies $(\varphi\cdot s\cdot\psi)\circ\mathcal{R}_{\sigma}(v,w)$. Then the kernel $\varphi\circ\delta\circ\delta$ coincide with $\preceq$.*

Note that if $\mathcal{R}_{\sigma}(v,w)$ holds, then so does $(\varphi\cdot s\cdot\psi)\circ\mathcal{R}_{\sigma}(v,w)$, but the vice-versa does not necessarily hold (take e.g $s \triangleq 0$).

Finally, we introduce the notion of *compatibility* which captures a form of Lipshitz-continuity with respect to V-Fuzz constructors. It is useful to follow [24] and define compatibility via the notion of *compatible refinement.*

Definition 12. *The* compatible refinement *$\hat{\alpha}$ of an open λ-term V-relation α is defined by:*

$$(\Gamma \vdash \hat{\alpha}(e,f) : \sigma) \triangleq \bigvee\{a \mid \Gamma \models a \leq \hat{\alpha}(e,f) : \sigma\},$$
$$(\Gamma \vdash^{\mathsf{v}} \hat{\alpha}(v,w) : \sigma) \triangleq \bigvee\{a \mid \Gamma \models^{\mathsf{v}} a \leq \hat{\alpha}(v,w) : \sigma\},$$

where judgments $\Gamma \models a \leq \hat{\alpha}(e,f) : \sigma$ and $\Gamma \models^{\mathsf{v}} a \leq \hat{\alpha}(v,w) : \sigma$ are inductively defined for $a \in \mathsf{V}$, $\Gamma \vdash e, f : \sigma$, and $\Gamma \vdash^{\mathsf{v}} v, w : \sigma$ by rules in Figure 4. We say that α is compatible if $\hat{\alpha} \leq \alpha$.

It is easy to see that if α is compatible, then we indeed have the desired inequalities. For instance, compatibility implies:

$$(s\wedge 1)\circ(\Gamma \vdash \alpha(e,e') : \sigma) \otimes (\Delta, x :_s \sigma \vdash \alpha(f,f') : \tau)$$
$$\leq (s\wedge 1)\cdot\Gamma\otimes\Delta \vdash \alpha(\mathbf{let}\ x = e\ \mathbf{in}\ f, \mathbf{let}\ x = e'\ \mathbf{in}\ f') : \tau$$

Notice that the presence of $s\wedge 1$, instead of s, ensures that for terms like $e \triangleq \mathbf{let}\ x = I\ \mathbf{in}\ \underline{0}$ and $e' \triangleq \mathbf{let}\ x = \Omega\ \mathbf{in}\ \underline{0}$, the distance $\alpha(e,e')$ is determined *before* sequencing (which captures the idea that although $\underline{0}$ will not 'use' any input, I and Ω will be still evaluated, thus producing observable differences between e and e'). In fact, if we replace $s\wedge 1$ with s, then by taking $s \triangleq 0$ compatibility would imply $\alpha(e,e') = k$, which is clearly unsound.

Howe's Method To prove compatibility of applicative Γ-similarity we design a generalisition of the so-called Howe's method [20] combining and extending ideas from [7] and [9].

Definition 13. *The* Howe's extension *α^{H} of an open λ-term V-relation α is defined as the least solution to the equation $\beta = \alpha\cdot\hat{\beta}$.*

It is easy to see that compatible refinement $\hat{-}$ is monotone, and thus so is the map Φ_{α} defined by $\Phi_{\alpha}(\beta) \triangleq \alpha\cdot\hat{\beta}$. As a consequence, we can define α^{H} as the least fixed point of Φ_{α}. Since open extension $-^{o}$ is monotone as well, we can define the Howe's extension of a closed λ-term V-relation α as $(\alpha^{o})^{H}$.

[6] The superscript is the letter 'o' (for open), and should not be confused with $\circ$ which we use for the map $-^{\circ}$ sending a V-relation to its dual.

$$\frac{}{\Gamma, x :_s \sigma \models k \leq \hat{\alpha}(x,x) : \sigma} \qquad \frac{a_1 \leq \Gamma_1 \vdash \alpha(e_1, f_1) : \sigma \quad \cdots \quad a_n \leq \Gamma_n \vdash \alpha(e_n, f_n) : \sigma}{op_{\mathsf{V}}(\Gamma_1, \ldots, \Gamma_n) \models op_{\mathsf{V}}(a_1, \ldots, a_n) \leq \hat{\alpha}(\mathbf{op}(e_1, \ldots, e_n), \mathbf{op}(e_1, \ldots, e_n)) : \sigma}$$

$$\frac{a \leq \Gamma, x :_1 \sigma \vdash \alpha(e, f) : \tau}{\Gamma \models^{\mathsf{v}} a \leq \hat{\alpha}(\lambda x.e, \lambda x.f) : \sigma \multimap \tau} \qquad \frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(v, v') : \sigma \multimap \tau \quad b \leq \Delta \vdash^{\mathsf{v}} \alpha(w, w') : \sigma}{\Gamma \otimes \Delta \models a \otimes b \leq \hat{\alpha}(vw, v'w') : \tau}$$

$$\frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(v, w) : \sigma_{\hat{\imath}}}{\Gamma \models^{\mathsf{v}} a \leq \hat{\alpha}(\langle \hat{\imath}, v\rangle, \langle \hat{\imath}, w\rangle) : \sum_{i \in I} \sigma_i} \qquad \frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(\langle \hat{\imath}, v\rangle, \langle \hat{\imath}, w\rangle) : \sum_{i \in I} \sigma_i \quad b_i \leq \Delta, x :_{s_i} \sigma_i \vdash \leq \alpha(e_i, f_i) : \tau \quad (\forall i \in I)}{s \cdot \Gamma \otimes \Delta \models s(a) \otimes b_{\hat{\imath}} \leq \hat{\alpha}(\mathbf{case}\ \langle \hat{\imath}, v\rangle\ \mathbf{of}\ \{\langle i, x\rangle \to e_i\}, \mathbf{case}\ \langle \hat{\imath}, w\rangle\ \mathbf{of}\ \{\langle i, x\rangle \to f_i\}) : \tau}$$

$$\frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(v, w) : \sigma}{\Gamma \models a \leq \hat{\alpha}(\mathbf{val}\ v, \mathbf{val}\ w) : \sigma} \qquad \frac{a \leq \Gamma \vdash \alpha(e, e') : \sigma \quad b \leq \Delta, x :_s \sigma \vdash \alpha(f', f') : \tau}{(s \wedge 1) \cdot \Gamma \otimes \Delta \models (s \wedge 1)(a) \otimes b \leq \hat{\alpha}(\mathbf{let}\ x = e\ \mathbf{in}\ f, \mathbf{let}\ x = e'\ \mathbf{in}\ f') : \tau}$$

$$\frac{a \leq \Gamma \models \alpha(v, w) : \sigma}{s \cdot \Gamma \models^{\mathsf{v}} s(a) \leq \alpha(!v, !w) : !_s \sigma} \qquad \frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(v, w) : !_r \sigma \quad b \leq \Delta, x :_{s \cdot r} \sigma \vdash \alpha(e, f) : \tau}{s \cdot \Gamma \otimes \Delta \models s(a) \otimes b \leq \hat{\alpha}(\mathbf{case}\ v\ \mathbf{of}\ \{!x \to e\}, \mathbf{case}\ w\ \mathbf{of}\ \{!x \to f\}) : \tau}$$

$$\frac{a\Gamma \vdash^{\mathsf{v}} \alpha(v, w) : \sigma[\mu t.\sigma/t]}{\Gamma \models^{\mathsf{v}} a \leq \hat{\alpha}(\mathbf{fold}\ v, \mathbf{fold}\ w) : \mu t.\sigma} \qquad \frac{a \leq \Gamma \vdash^{\mathsf{v}} \alpha(v, w) : \mu t.\sigma \quad b \leq \Delta, x :_s \sigma[\mu t.\sigma/t] \vdash b \leq \alpha(e, f) : \tau}{s \cdot \Gamma \otimes \Delta \models s(a) \otimes b \leq \hat{\alpha}(\mathbf{case}\ v\ \mathbf{of}\ \{\mathbf{fold}\ x \to e\}, \mathbf{case}\ w\ \mathbf{of}\ \{\mathbf{fold}\ x \to f\}) : \tau}$$

Figure 4. Compatible refinement.

The Howe's extension of an open λ-term V-preorder α is compatible and bigger than α. In particular, Proposition 3 implies that $(\delta^o)^H$ is compatible and bigger than δ^o. Moreover, Howe's extension enjoys another remarkable property, namely substitutivity.

Definition 14. *An open λ-term V-relation α is value substitutive if for all well-typed values $\Gamma, x :_s \sigma \vdash^{\mathsf{v}} v, w : \tau, \emptyset \vdash^{\mathsf{v}} u : \sigma$, and terms $\Gamma, x :_s \sigma \vdash e, f : \tau$ we have:*

$$(\Gamma, x :_s \sigma \vdash^{\mathsf{v}} \alpha(v, w) : \tau) \leq (\Gamma \vdash \alpha(v[u/x], w[u/x]) : \tau),$$
$$(\Gamma, x :_s \sigma \vdash \alpha(e, f) : \tau) \leq (\Gamma \vdash \alpha(e[x := u], f[x := u]) : \tau).$$

Lemma 2 (Substitutivity)**.** *Let α be a value substitutive λ-term V-preorder. For all values, $\Gamma, x :_s \sigma \vdash^{\mathsf{v}} u, z : \tau$ and $\emptyset \vdash v, w : \sigma$, and terms $\Gamma, x :_s \sigma \vdash e, f : \tau$, let $\underline{a} \triangleq \emptyset \vdash^{\mathsf{v}} \alpha^H(v, w) : \sigma$. Then:*

$$(\Gamma, x :_s \sigma \vdash^{\mathsf{v}} \alpha^H(u, z) : \tau) \otimes s(\underline{a}) \leq \Gamma \vdash^{\mathsf{v}} \alpha^H(u[v/x], z[w/x]) : \tau,$$
$$(\Gamma, x :_s \sigma \vdash \alpha^H(e, f) : \tau) \otimes s(\underline{a}) \leq \Gamma \vdash \alpha^H(e[x := v], f[x := w]) : \tau.$$

Notice that the open extension of any closed λ-term V-relation is value-substitutive. We can finally state the so-called *Key Lemma*.

Lemma 3 (Key Lemma)**.** *Let α be a reflexive and transitive applicative Γ-simulation. Then the Howe's extension of α restricted to closed terms/values in an applicative Γ-simulation.*

Proof sketch. The proof is non-trivial and due to space constraints it cannot be included here. A detailed account is given in [16]. Let us write α^H for the Howe's extension of α restricted to closed terms/values. By induction on n one shows that for any $n \geq 0$, $(\alpha^H)^{\wedge}_{\sigma}(e, f) \leq \Gamma(\alpha^H)^{\mathsf{V}}_{\sigma}(|e|_n, |f|)$ holds for all terms $e, f \in \Lambda_\sigma$. Since Γ is inductive, the above inequality indeed gives the thesis. The base case follows again by inductivity of Γ, whereas the inductive step requires a case analysis on the structure of e. The crucial case is sequencing, where we rely on condition (L-Strong lax bind). □

From the Key Lemma it directly follows our main result.

Theorem 4 (Compatibility)**.** *Applicative Γ-similarity is compatible.*

Corollary 1 (Metric Preservation (cf. [12]))**.** *For any environment $\Gamma \triangleq x_1 :_{s_1} \sigma, \ldots, x_n :_{s_n} \sigma$, values $\bar{v}, \bar{w} : \Gamma$, and $\Gamma \vdash e : \sigma$ we have:*

$$s_1 \circ \delta^{\mathsf{V}}_{\sigma_1}(v_1, w_1) \otimes \cdots \otimes s_n \circ \delta^{\mathsf{V}}_{\sigma_n}(v_n, w_n) \leq \delta^{\wedge}_{\sigma}(e[\vec{x} := \vec{v}], e[\vec{x} := \vec{w}]).$$

Applicative* Γ-*bisimilarity In this last paragraph we define applicative Γ-*bisimilarity* and show that under suitable conditions on CBEs and operations we obtain a *compatible behavioural pseudometric*. Such conditions are stricter than those necessary to prove compatibility of applicative Γ-similarity. Nevertheless, easy calculations show that they are met by all concrete examples we have considered in this paper.

In the rest of this paragraph we assume CBEs to be monotone *monoid (homo)morphisms*, i.e. we modify Definition 4 requiring the equalities $h(k) = \ell$ and $h(a \otimes b) = h(a) \otimes h(b)$ to hold. We also assume operations op_{V} to be *quantale (homo)morphism*, i.e. to preserves unit, multiplication, and joins. In light of Example 8 we give the following definition.

Definition 15. *Recall Proposition 1. Define applicative* Γ-bisimilarity *γ as applicative $(\Gamma \wedge \Gamma^{\circ})$-similarity.*

Proposition 3 implies that γ is reflexive and transitive. Moreover, if CBEs preserve binary meet (a condition satisfied by all our examples), i.e. $s(a) \wedge s(b) = s(a \wedge b)$ for any CBE s in Π, then γ is also symmetric, and thus a pseudometric. Finally we observe that γ is the greatest λ-term V-relation α such that both α and α° are applicative Γ-simulation.

Proving compatibility of γ is not straightforward, and requires a variation of the so-called *transitive closure trick* [29]. First of all we notice that we cannot apply the Key Lemma on γ since $\Gamma \wedge \Gamma^{\circ}$ being conversive is, in general, not inductive. To overcome this problem, we follow [35] and observe that applicative Γ-bisimilarity is the greatest *symmetric* applicative Γ-*simulation*. We can now prove that under suitable conditions on CBEs γ is compatible.

Theorem 5. *Say that a CBE s is* finitely continuous, *if $s \neq \infty$ implies $s(\bigvee A) = \bigvee \{s(a) \mid a \in A\}$, for any set $A \subseteq \mathsf{V}$. Then if CBEs in Π are finitely continuous, then γ is compatible.*

Proof sketch. See [16] for a detailed proof. By Key Lemma γ^H is a compatible Γ-simulation. Since CBEs are are finitely continuous the transitive closure of γ^H is itself a compatible *symmetric* Γ-simulation. We conclude compatibility of γ by coinduction. □

Finally, we notice that all concrete CBEs considered in this work are finitely continuous. We can then rely on Theorem 5 to come

up with concrete notions of compatible applicative Γ-bisimilarity. Notably, we obtain compatible pseudometrics for Fuzz[7] and *P*-Fuzz.

6 Related Work

Several works have been done in the past years on quantitative (metric) reasoning in the context of programming language semantics. In particular, several authors have used (cartesian) categories of *ultrametric spaces* as a foundation for denotational semantics of both concurrent [3, 13] and sequential programming languages [15]. Bisimilarity-based distances have been proposed for probabilistic *first-order* calculi [14], where, due to the absence of higher-order features, amplification phenomena do not occur (but see [17]). A different approach is investigated in [12] where a denotational semantics combining ordinary metric spaces and domains is given to *pure* (i.e. without effects) Fuzz. The main theorem of [12] is a denotational version of the so-called *metric preservation* [31] (whose original proof requires a suitable *step-indexed metric logical relation*). Our Corollary 1 is the operational counterpart of such result generalised to arbitrary algebraic effects.

A different but deeply related line of research has been recently proposed in [7, 8], where coinductive, operationally-based distances have been studied for probabilistic λ-calculi. In particular, in [7] a notion of applicative distance based on the Wasserstein lifting is proposed for a probabilistic *affine* λ-calculus. Restricting to affine programs only makes the calculus strongly normalising and removes copying capabilities of programs by construction. That way programs cannot amplify distances between their inputs and therefore are forced to behave as non-expansive functions. This limitation is overcome in [8], where a coinductive notion of distance is proposed for a full linear λ-calculus, and distance trivialisation phenomena are studied in depth. The price to pay for such generality is that the distance proposed is not applicative, but a trace distance somehow resembling environmental bisimilarity [34].

7 Conclusion

In this work we have introduced an abstract framework for studying quantale-valued behavioural relations for higher-order effectful languages. Such framework has been instantiated to define the quantitative refinements of Abramsky's applicative similarity and bisimilarity for V-Fuzz, a universal λ-calculus with a linear type system tracking program sensitivity enriched with algebraic effects. Our main theorems state that under suitable conditions the quantitative notions of applicative similarity and bisimilarity obtained are a compatible generealised metric and pseudometric, respectively. These results can be instantiated to obtain compatible pseudometrics for several concrete calculi.

A future research direction is to use the abstract framework developed to study behavioural distances different from applicative distances. In particular, investigating contextual distances [16], denotationally-based distances [12], and distances based on logical relations [31] are interesting topics for further research.

Acknowledgments

The author would like to thank Ugo Dal Lago, Raphaëlle Crubillé, Paul Levy, and the anonymous reviewers for the many useful comments and suggestions. Special thanks also goes to Alex Simpson and Niels Voorneveld for many insightful discussions about the topic of this work. The author is partially supported by the ANR projects 16CE250011 REPAS and 14CE250005 ELICA.

[7] Formally, we should extend our definitions adding a basic type for real numbers and primitives for arithmetical operations, but that is straightforward.

References

[1] S. Abramsky. 1990. The Lazy Lambda Calculus. (1990), 65–117.
[2] S. Abramsky and A. Jung. 1994. Domain Theory. In *Handbook of Logic in Computer Science*. Clarendon Press, 1–168.
[3] A. Arnold and M. Nivat. 1980. Metric Interpretations of Infinite Trees and Semantics of non Deterministic Recursive Programs. *Theor. Comput. Sci.* 11 (1980), 181–205.
[4] M. Barr. 1970. Relational algebras. *Lect. Notes Math.* 137 (1970), 39–55.
[5] P. Clément and W. Desch. 2008. Wasserstein metric and subordination. (2008).
[6] R. Crubillé and U. Dal Lago. 2014. On Probabilistic Applicative Bisimulation and Call-by-Value λ-Calculi. In *Proc. of ESOP 2014*. 209–228.
[7] R. Crubillé and U. Dal Lago. 2015. Metric Reasoning about λ-Terms: The Affine Case. In *Proc. of LICS 2015*. 633–644.
[8] R. Crubillé and U. Dal Lago. 2017. Metric Reasoning About λ-Terms: The General Case. In *Proc. of ESOP 2017*. 341–367.
[9] U. Dal Lago, F. Gavazzo, and P.B. Levy. 2017. Effectful applicative bisimilarity: Monads, relators, and Howe's method. In *Proc. of LICS 2017*. 1–12.
[10] U. Dal Lago, D. Sangiorgi, and M. Alberti. 2014. On coinductive equivalences for higher-order probabilistic functional programs. In *Proc. of POPL 2014*. 297–308.
[11] B.A. Davey and H.A. Priestley. 1990. *Introduction to lattices and order*. Cambridge University Press.
[12] A.A. de Amorim, M. Gaboardi, J. Hsu, S. Katsumata, and I. Cherigui. 2017. A semantic account of metric preservation. In *Proc. of POPL 2017*. 545–556.
[13] J.W. de Bakker and J.I. Zucker. 1982. Denotational Semantics of Concurrency. In *STOC*. 153–158.
[14] W. Du, Y. Deng, and D. Gebler. 2016. Behavioural Pseudometrics for Nondeterministic Probabilistic Systems. In *Proc. of SETTA 2016*. 67–84.
[15] M.H. Escardo. 1999. A metric model of PCF. In *Workshop on Realizability Semantics and Applications*.
[16] F. Gavazzo. 2018. Quantitative Behavioural Reasoning for Higher-order Effectful Programs: Applicative Distances (Long Version). https://arxiv.org/abs/1801.09072
[17] D. Gebler, Larsen. K.G., and S. Tini. 2016. Compositional bisimulation metric reasoning with Probabilistic Process Calculi. *LMCS* 12, 4 (2016).
[18] J-Y. Girard, A. Scedrov, and P.J. Scott. 1992. Bounded Linear Logic: A Modular Approach to Polynomial-Time Computability. *Theor. Comput. Sci.* 97 (1992), 1–66.
[19] D. Hofmann, G.J. Seal, and W. Tholen (Eds.). 2014. *Monoidal Topology. A Categorical Approach to Order, Metric, and Topology*. Number 153 in Encyclopedia of Mathematics and its Applications. Cambridge University Press.
[20] D.J. Howe. 1996. Proving Congruence of Bisimulation in Functional Programming Languages. *Inf. Comput.* 124, 2 (1996), 103–112.
[21] A. Kock. 1972. Strong functors and monoidal monads. *Archiv der Mathematik* 23 (1972), 113–120.
[22] K.O. Kortanek and M. Yamasaki. 1995. Discrete infinite transportation problems. *Discrete Applied Mathematics* 58 (1995), 19–33.
[23] A. Kurz and J. Velebil. 2016. Relation lifting, a survey. *J. Log. Algebr. Meth. Program.* 85, 4 (2016), 475–499.
[24] S.B. Lassen. 1998. *Relational Reasoning about Functions and Nondeterminism*. Ph.D. Dissertation. Dept. of Computer Science, University of Aarhus.
[25] F.W. Lawvere. 1973. Metric spaces, generalized logic, and closed categories. *Rend. Sem. Mat. Fis. Milano* 43 (1973), 135–166.
[26] P.B. Levy, J. Power, and H. Thielecke. 2003. Modelling Environments in Call-by-Value Programming Languages. *Inf. Comput.* 185, 2 (2003), 182–210.
[27] S. MacLane. 1971. *Categories for the Working Mathematician*. Springer-Verlag.
[28] J. Morris. 1969. *Lambda Calculus Models of Programming Languages*. Ph.D. Dissertation. MIT.
[29] A.M. Pitts. 2011. Howe's Method for Higher-Order Languages. In *Advanced Topics in Bisimulation and Coinduction*, D. Sangiorgi and J. Rutten (Eds.). Cambridge University Press, 197–232.
[30] G.D. Plotkin and J. Power. 2001. Adequacy for Algebraic Effects. In *Proc. of FOSSACS 2001*. 1–24.
[31] J. Reed and B.C. Pierce. 2010. Distance makes the types grow stronger: a calculus for differential privacy. In *Proc. of ICFP 2010*. 157–168.
[32] J.C. Reynolds. 1983. Types, Abstraction and Parametric Polymorphism. In *IFIP Congress*. 513–523.
[33] J.J.M.M. Rutten. 1996. Elements of Generalized Ultrametric Domain Theory. *Theor. Comput. Sci.* 170, 1-2 (1996), 349–381.
[34] D. Sangiorgi, N. Kobayashi, and E. Sumii. 2011. Environmental bisimulations for higher-order languages. *ACM Trans. Program. Lang. Syst.* 33, 1 (2011), 5:1–5:69.
[35] A. Simpson and N. Voorneveld. 2018. Behavioural equivalence via modalities for algebraic effects. In *Proc. of ESOP 2018*.
[36] L.A. Steen and J.A. Seebach. 1995. *Counterexamples in Topology*. Dover Publications.
[37] A.M. Thijs. 1996. *Simulation and fixpoint semantics*. Rijksuniversiteit Groningen.
[38] C. Villani. 2008. *Optimal Transport: Old and New*. Springer Berlin Heidelberg.