Probabilistic Relational Reasoning via Metrics

Arthur Azevedo de Amorim
Carnegie Mellon University
Pittsburgh, PA

Marco Gaboardi
University at Buffalo
Buffalo, NY

Justin Hsu
University of Wisconsin
Madison, WI

Shin-ya Katsumata
National Institute of Informatics
Tokyo, Japan

***Abstract*—The Fuzz programming language by Reed and Pierce uses an elegant linear type system combined with a monad-like type to express and reason about probabilistic sensitivity properties, most notably ε-differential privacy. We show how to extend Fuzz to capture more general *relational* properties of probabilistic programs, with *approximate*, or (ε, δ)-differential privacy serving as a leading example. Our technical contributions are threefold. First, we introduce the categorical notion of *comonadic lifting* of a monad to model composition properties of probabilistic divergences. Then, we show how to express relational properties in terms of sensitivity properties via an adjunction we call the *path construction*. Finally, we instantiate our semantics to model the terminating fragment of Fuzz extended with types carrying information about other divergences between distributions.**

I. Introduction

Over the past decade, differential privacy has emerged as a robust, compositional notion of privacy, imposing rigorous bounds on what database queries reveal about private data. Formally, a probabilistic database query f is ε-differentially private if, given two pairs of *adjacent* databases X_1 and X_2—that is, databases differing in at most one record—we have

$$\mathbb{P}(f(X_i) \in U) \leq e^{\varepsilon}\mathbb{P}(f(X_j) \in U) \quad (i, j \in \{1, 2\}), \tag{1}$$

where U ranges over arbitrary sets of query results. Intuitively, the parameter ε measures how different the result distributions $f(X_1)$ and $f(X_2)$ are—the smaller ε is, the less the output depends on any single record in the input database.

The strengths and applications of differential privacy have prompted the development of a range of verification techniques, in particular approaches based on linear types such as the Fuzz language [30]. These systems exploit the fact that ε-differential privacy is equivalent to a *sensitivity* property, a guarantee that applies to *arbitrary* pairs of databases:

$$\mathsf{MD}(f(X_1), f(X_2)) \leq \varepsilon \cdot d_{DB}(X_1, X_2), \tag{2}$$

where d_{DB} measures how similar the input databases are (for instance, via the Hamming distance on sets), and MD, the *max divergence*, is the smallest value of ε for which (1) holds. In other words, f is ε-Lipschitz continuous, or *ε-sensitive*. Sensitivity has pleasant properties for formal verification; for example, the sensitivity of the composition of two functions is the product of their sensitivities. By leveraging these composition principles, Fuzz can track the sensitivity of a function in its type, reducing a proof of differential privacy for an algorithm to simpler sensitivity checks about its components.

978-1-7281-3608-0/19/$31.00

However, not all distance bounds between distributions can be converted into sensitivity properties. One example is (ε, δ)-differential privacy [11], a relaxation of ε-differential privacy that allows privacy violations with a small probability δ; in return, (ε, δ)-differential privacy can allow significantly more accurate data analyses. Superficially, its definition resembles (1): a query f is (ε, δ)-differentially private if, for all pairs of adjacent input databases X_1 and X_2, we have

$$\mathbb{P}(f(X_i) \in U) \leq e^{\varepsilon}\mathbb{P}(f(X_j) \in U) + \delta \quad (i, j \in \{1, 2\}). \tag{3}$$

Setting $\delta = 0$ recovers the original definition. Introducing the *skew divergence* AD_ε [4], (ε, δ)-privacy is equivalent to a bound $\mathsf{AD}_\varepsilon(f(X_1), f(X_2)) \leq \delta$ for adjacent databases.

Despite the similarity between the two definitions, Fuzz could not handle (ε, δ)-differential privacy, because it cannot be stated directly in terms of function sensitivity. This is possible for ε-differential privacy because (1) can be recast as the bound $\mathsf{MD}(f(X_1), f(X_2)) \leq \varepsilon$, which is equivalent to the sensitivity property (2) because the max divergence is a proper metric satisfying the triangle inequality. In contrast, the skew divergence does *not* satisfy the triangle inequality and does not scale up smoothly when the inputs X_1 and X_2 are farther apart—for instance, an (ε, δ)-private function f usually does not satisfy $\mathsf{AD}_\varepsilon(f(X_1), f(X_2)) \leq 2 \cdot \delta$ when X_1 and X_2 are at distance 2. Similar problems arise for other properties based on distances that violate the triangle inequality, such as the Kullback-Leibler (KL) and χ^2 divergences.

This paper aims to bridge this gap, showing that Fuzz's core can *already* accommodate other quantitative properties, with (ε, δ)-privacy being our motivating application. To do so, we first need a semantics for the probabilistic features of Fuzz. Typically in programming language semantics, probabilistic programs are structured using the probability monad [16, 26]: the return operation produces a deterministic distribution that always yields the same value, while the bind operation samples from a distribution and runs another probabilistic computation. However, monads cannot describe the composition principles supported by many useful metrics on probabilities—though the typing rules for distributions in Fuzz resemble the usual monadic rules [26], there issues related to the context sensitivities. Accordingly, our *first contribution* is a notion of *comonadic lifting* of a monad, which lifts the operations of a monad from a symmetric monoidal closed category (SMCC) to a related *refined* category. We demonstrate our theory by modeling statistical distance and Fuzz's max divergence. We also propose a *graded* variant of liftings to encompass other examples, including the Hellinger distance and the KL and χ^2 divergences.

Our *second contribution* is a *path metric* construction that reduces relational properties such as (ε, δ)-differential privacy to equivalent statements about sensitivity. Concretely, given any reflexive, symmetric relation R on a set X, we define the path metric d_R on X by setting $d_R(x_1, x_2)$ to be the length of the shortest path connecting x_1 and x_2 in the graph corresponding to R. The path construction provides a full and faithful functor from the category RSRel of reflexive, symmetric relations into the category Met of metric spaces and 1-sensitive functions. We also show a right adjoint to the path construction—as Met is a symmetric monoidal closed category and RSRel is a cartesian closed category, this adjunction recalls mixed linear and non-linear models of linear logic [8].

Putting these two pieces together, our *third contribution* is a model of the terminating fragment of Fuzz [30]. We extend the language with new types and typing rules to express and reason about relational properties beyond ε-differential privacy, including (ε, δ)-differential privacy. Our framework can smoothly incorporate the new features by combining the path construction and graded liftings, giving a unified perspective on a class of probabilistic relational properties.

Outline: We begin by reviewing the Fuzz language, the interpretation of its deterministic fragment in the category of metric spaces, and some basic probability theory in Section II. Section III introduces comonadic liftings on monads, and uses them to interpret the terminating fragment of Fuzz with probabilistic constructs.

Shifting gears, Section IV explores how probabilistic properties can be modeled in the category of relations. Section V develops a graded version of our comonadic liftings over relations, to model composition of properties like (ε, δ)-privacy. In Section VI we consider how to transfer liftings between different categories; our leading example is the *path construction*, which moves liftings over relations to liftings over metric spaces. As an application, Section VII extends Fuzz with graded types capable of modeling (ε, δ)-privacy and other sensitivity properties of divergences.

In Section VIII, we sketch how our results can be partially extended to model general recursion in Fuzz, by combining metric CPOs [3] with the probabilistic powerdomain of Jones and Plotkin [17]. While the probabilistic features and liftings pose no problems, the path construction runs into technical difficulties; we leave this extension as a challenging open problem. Finally, we survey related work (Section IX) and conclude (Section X).

II. Preliminaries

To fix notation and terminology, we review here basic concepts of category theory, probabilities and metric spaces. For ease of reference, we include an overview of the deterministic, terminating fragment of the Fuzz language, recalling its semantics based on metric spaces [3].

A. Category Theory

A map of adjunctions [22, Sections IV.1,IV.7] from $\langle L, R, \eta, \varepsilon \rangle : \mathbb{C} \rightharpoonup \mathbb{D}$ to $\langle L', R', \eta', \varepsilon' \rangle : \mathbb{C}' \rightharpoonup \mathbb{D}'$ is a pair of functors $F : \mathbb{C} \to \mathbb{C}', G : \mathbb{D} \to \mathbb{D}'$ satisfying $G \circ L = L' \circ F, F \circ R = R' \circ G$ and $F \circ \eta = \eta' \circ F$. (The last equality is equivalent to $G \circ \varepsilon = \varepsilon' \circ G$.) We write such a map as $(F, G) : (L \dashv R) \to (L' \dashv R')$.

Though our main applications revolve around probabilistic programs, we develop our theory in terms of general monads $\mathcal{T} = (T, \eta, (-)^\dagger)$ presented as Kleisli triples. The operation $(-)^\dagger$, the *Kleisli lifting* of the monad, promotes a morphism $f : X \to TY$ to $f^\dagger : TX \to TY$ and satisfies common unit and associativity laws. We assume that $\mathcal{T}$ is defined on a symmetric monoidal closed category and carries a compatible *strength* $\sigma_{X,Y} : X \otimes TY \to T(X \otimes Y)$ [19], which allows us to define *parameterized* liftings: given $f : X \otimes Y \to TZ$, we define $f^\ddagger : X \otimes TY \to TZ$ as the composite

$$f^\ddagger \triangleq X \otimes TY \xrightarrow{\sigma} T(X \otimes Y) \xrightarrow{f^\dagger} TZ\ . \qquad (4)$$

Lifting the evaluation morphism $ev_{X,TY} : (X \multimap TY) \otimes X \to TY$ of the internal hom $\multimap$ yields the internalized Kleisli lifting

$$\mathrm{kl}^{\mathcal{T}}_{X,Y} : (X \multimap TY) \otimes TX \to TY. \qquad (5)$$

B. Probability Theory

Our running example is the monad DX of discrete probability distributions over a set X—that is, functions $\mu : X \to [0, 1]$ such that $\mu(x) \neq 0$ for at most countably many elements $x \in X$, and $\sum_{x \in X} \mu(x) = 1$. For a subset $U \subseteq X$, we define $\mu(U)$ as $\sum_{x \in U} \mu(x)$. Given an element $x \in X$, we write $\eta(x) \in DX$ for the point mass at x, i.e., $\eta(x)(x') \triangleq 1$ if $x = x'$, otherwise 0. Given $f : X \to DY$, its Kleisli lifting $f^\dagger : DX \to DY$ is defined by sampling a value from its input distribution and feeding that sample to f. Formally,

$$f^\dagger(\mu)(y) \triangleq \sum_{x \in X} f(x)(y)\mu(x) \quad (\mu \in DX, y \in Y).$$

C. Metric Spaces

Let $\mathbb{R}^\infty_{\geq 0}$ be the set of non-negative reals extended with a greatest element ∞. A *metric* on a set X is a function $d : X \times X \to \mathbb{R}^\infty_{\geq 0}$ that satisfies (i) $d(x, x) = 0$, (ii) $d(x, y) = d_X(y, x)$, and (iii) $d(x, z) \leq d(x, y) + d(y, z)$ (the *triangle inequality*). We stipulate that $r \bullet \infty = \infty \bullet r = \infty$ for any $r \in \mathbb{R}^\infty_{\geq 0}$, where $\bullet$ stands for addition or multiplication. A *metric space* is a pair $X = (|X|, d_X)$, where $|X|$ is a carrier set and d_X is a metric on $|X|$.[1] We will often refer to a metric space by its carrier, and we will write d with no subscript when the metric space is clear. A metric space X is *above* a set I if $|X| = I$, that is, $X = (I, d)$ with some metric d on I.

A function $f : X \to Y$ between two metric spaces is r-*sensitive* if $d_Y(f(x_1), f(x_2)) \leq r d_X(x_1, x_2)$ for all pairs of

[1] These conditions technically define an *extended pseudo-metric space*—"extended" because distances may be infinite, and "pseudo" because distinct elements may be at distance 0—but for the sake of brevity we use "metric space" throughout.

Metric Space	Carrier Set	$d(a,b)$
$\mathbb{R}$	$\mathbb{R}$	$\lvert a-b \rvert$
1	$\{\star\}$	0
$r \cdot X$	X	$\begin{cases} r d_X(a,b) & : r \neq \infty \\ \infty & : r = \infty, a \neq b \\ 0 & : r = \infty, a = b \end{cases}$
$X \times Y$	$X \times Y$	$\max(d_X(a_1,b_1), d_Y(a_2,b_2))$
$X \otimes Y$	$X \times Y$	$d_X(a_1,b_1) + d_Y(a_2,b_2)$
$X + Y$	$X \uplus Y$	$\begin{cases} d_X(a,b) & \text{if } a,b \in X \\ d_Y(a,b) & \text{if } a,b \in Y \\ \infty & \text{otherwise} \end{cases}$
$X \multimap Y$	$X \to Y$ Non-exp.	$\sup_{x \in X} d_Y(a(x), b(x))$

Fig. 1: Basic constructions on metric spaces

$$\begin{aligned} e \in E ::= {} & x \mid k \in \mathbb{R} \mid e_1 + e_2 \mid () \mid \lambda x.\, e \mid e_1\ e_2 \mid (e_1, e_2) \\ & \mid \textbf{let}\,(x,y) = e \,\textbf{in}\, e' \mid \langle e_1, e_2 \rangle \mid \pi_i\ e \mid !e \mid \textbf{let}\,!x = e \,\textbf{in}\, e' \\ & \mid \textbf{inl}\, e \mid \textbf{inr}\, e \mid (\textbf{case}\, e \,\textbf{of}\, \textbf{inl}\, x.\, e_l \mid \textbf{inr}\, y.\, e_r) \\ \sigma, \tau ::= {} & \mathbb{R} \mid 1 \mid \sigma \multimap \tau \mid \sigma \otimes \tau \mid \sigma \,\&\, \tau \mid \sigma + \tau \mid !_r \sigma \end{aligned}$$

$$r, s \in \mathbb{R}^{\infty}_{\geq 0} \qquad \Gamma, \Delta ::= \varnothing \mid \Gamma, x :_r \sigma$$

$$r \cdot \varnothing \triangleq \varnothing \qquad r \cdot (\Gamma, x :_s \sigma) \triangleq r \cdot \Gamma, x :_{r \cdot s} \sigma$$

$$\varnothing + \varnothing \triangleq \varnothing \qquad (\Gamma, x :_r \sigma) + (\Delta, x :_s \sigma) \triangleq (\Gamma + \Delta), x :_{r+s} \sigma$$

$$(\Gamma, x :_r \sigma) + \Delta \triangleq (\Gamma + \Delta), x :_r \sigma \qquad (x \notin \Delta)$$

$$\Gamma + (\Delta, x :_s \sigma) \triangleq (\Gamma + \Delta), x :_s \sigma \qquad (x \notin \Gamma)$$

Fig. 2: Fuzz syntax, types, contexts, and context operations

elements $x_1, x_2 \in X$. Thus, the smaller r is, the less the output of a function varies when its input varies. Note that this condition is vacuous when $r = \infty$, so any function between metric spaces is ∞-sensitive. When $r = 1$, we speak of a *non-expansive* function instead. We write $f : X \overset{\text{ne}}{\to} Y$ to mean that f is a non-expansive function from X to Y.

To illustrate these concepts, consider the set of real numbers $\mathbb{R}$ equipped with the Euclidean metric: $d(x,y) = |x - y|$. The doubling function that maps the real number x to $2x$ is 2-sensitive; more generally, a function that scales a real number by another real number k is $|k|$-sensitive. However, the squaring function that maps each x to x^2 is not r-sensitive for any finite r. The identity function on a metric space is always non-expansive. The definition of ε-differential privacy, as stated in (2), says that the private query f is ε-sensitive.

Figure 1 summarizes basic constructions on metric spaces. Scaling allows us to express sensitivity in terms of non-expansiveness: an r-sensitive function from X to Y is a non-expansive function from the scaled metric space $r \cdot X$ (or simply rX) to Y. The metric on ∞X does not depend on the metric of X—note that we define scaling by ∞ separately from scaling by a finite number to ensure that $d(x,x) = 0$—so we use this notation even when X is a plain set that does not have a metric associated with it. We consider two metrics on products: one combines metrics by taking their maximum, while the other takes their sum. The metric on disjoint unions places their two sides infinitely apart.

Let Met be the category of metric spaces and non-expansive functions. We write $p : \mathsf{Met} \to \mathsf{Set}$ to denote the forgetful functor defined by $pX = |X|$ and $pf = f$. Note that $\mathsf{Set}(X, pY) = \mathsf{Met}(\infty \cdot X, Y)$, so $\infty \cdot (-)$ and p form an adjoint pair. The $\times$ metric—used to interpret the connective $\&$—and $+$ metric yield products and sums in Met, with the expected projections and injections. Since the two sides of a sum are infinitely apart, we can define non-expansive functions by case analysis without reasoning about sensitivity across two different branches. The other metric on products, given by $\otimes$, is needed to make the operations of currying and function application compatible with the metric on non-expansive functions defined above. Formally, Met forms a symmetric monoidal closed category with monoidal structure given by $(\otimes, 1)$ and exponentials given by $\multimap$.

D. The Fuzz Language

Fuzz [30] is a type system for analyzing program sensitivity. The language is a largely standard, call-by-value lambda calculus; Fig. 2 summarizes the syntax, types, contexts, and context operations. Types in Fuzz are interpreted as metric spaces, and function types carry a numeric annotation that describes their sensitivity. The type system tracks the sensitivity of typed terms with respect to each of its bound variables, akin to bounded linear logic [15]. Fig. 3 presents the typing rules of the terminating, deterministic fragment of the language.

In prior work [3], we described a model where each typing derivation $x_1 :_{r_1} \tau_1, \ldots, x_n :_{r_n} \tau_n \vdash e : \sigma$ corresponds to a non-expansive function $[\![e]\!] : r_1[\![\tau_1]\!] \otimes \cdots \otimes r_n[\![\tau_n]\!] \to [\![\sigma]\!]$,[2] where types are interpreted homomorphically using the constructions on metric spaces described thus far. In particular, context splitting corresponds to the family of functions $\delta : [\![\Delta + \Gamma]\!] \overset{\text{ne}}{\to} [\![\Delta]\!] \otimes [\![\Gamma]\!]$ defined as $\delta(x) \triangleq (x_1, x_2)$, where x_1 and x_2 are obtained by removing the components of x that do not appear in Δ and Γ, respectively.

In addition to the probabilistic features that we will cover next, the original Fuzz language also includes general recursive types. These pose challenges related to non-termination, which we return to in Section VIII.

III. A Semantics for Probabilistic Fuzz

In addition to the deterministic constructs, Fuzz offers a monad-like interface for probabilistic programming [26],

[2]Our interpretation of scaling differs slightly from our previous one [3] in that distinct points in scaled spaces ∞X are infinitely apart. This does not affect the validity of the interpretation; in particular, scaling is still associative, and commutes with $\otimes$, $\&$ and $+$.

$$\frac{(x :_r \sigma) \in \Gamma \qquad r \geq 1}{\Gamma \vdash x : \sigma} \ (\text{Var}) \qquad \frac{}{\Gamma \vdash () : 1} \ (1I)$$

$$\frac{k \in \mathbb{R}}{\Gamma \vdash k : \mathbb{R}} \ (\text{Const}) \qquad \frac{\Gamma, x :_1 \sigma \vdash e : \tau}{\Gamma \vdash \lambda x.\, e : \sigma \multimap \tau} \ (\multimap I)$$

$$\frac{\Gamma \vdash e_1 : \sigma \multimap \tau \qquad \Delta \vdash e_2 : \sigma}{\Gamma + \Delta \vdash e_1\ e_2 : \tau} \ (\multimap E)$$

$$\frac{\Gamma \vdash e_1 : \sigma \qquad \Delta \vdash e_2 : \tau}{\Gamma + \Delta \vdash (e_1, e_2) : \sigma \otimes \tau} \ (\otimes I)$$

$$\frac{\Gamma \vdash e : \sigma_1 \otimes \sigma_2 \qquad \Delta, x :_r \sigma_1, y :_r \sigma_2 \vdash e' : \tau}{r\Gamma + \Delta \vdash \mathbf{let}\,(x, y) = e\,\mathbf{in}\,e' : \tau} \ (\otimes E)$$

$$\frac{\Gamma \vdash e_1 : \sigma \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \langle e_1, e_2 \rangle : \sigma \,\&\, \tau} \ (\& I)$$

$$\frac{\Gamma \vdash e : \sigma_1 \,\&\, \sigma_2}{\Gamma \vdash \pi_i\, e : \sigma_i} \ (\& E) \qquad \frac{\Gamma \vdash e : \sigma}{r\Gamma \vdash !e : !_r\sigma} \ (!I)$$

$$\frac{\Gamma \vdash e_1 : !_s\sigma \qquad \Delta, x :_{rs} \sigma \vdash e_2 : \tau}{r\Gamma + \Delta \vdash \mathbf{let}\,!x = e_1\,\mathbf{in}\,e_2 : \tau} \ (!E)$$

$$\frac{\Gamma \vdash e : \sigma}{\Gamma \vdash \mathbf{inl}\,e : \sigma + \tau} \ (+I_l) \qquad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathbf{inr}\,e : \sigma + \tau} \ (+I_r)$$

$$\frac{\begin{array}{c}\Gamma \vdash e : \sigma_1 + \sigma_2 \\ \Delta, x :_r \sigma_1 \vdash e_l : \tau \qquad \Delta, y :_r \sigma_2 \vdash e_r : \tau\end{array}}{r\Gamma + \Delta \vdash \mathbf{case}\,e\,\mathbf{of}\,\mathbf{inl}\,x.\,e_l \mid \mathbf{inr}\,y.\,e_r : \tau} \ (+E)$$

Fig. 3: Fuzz typing rules (deterministic, terminating fragment)

structured around the operations discussed in Section II-B—sampling and producing a deterministic distribution. These operations are typed with the following rules, where $\bigcirc$ is the type constructor of probability distributions:

$$\frac{\Gamma \vdash e_1 : \bigcirc\tau \qquad \Delta, x :_{\infty} \tau \vdash e_2 : \bigcirc\sigma}{\Delta + \Gamma \vdash \mathbf{bind}\,x \leftarrow e_1; e_2 : \bigcirc\sigma} \tag{6}$$

$$\frac{\Gamma \vdash e : \tau}{\infty\Gamma \vdash \mathbf{return}\,e : \bigcirc\tau} \tag{7}$$

The treatment of sensitivities in these rules is dictated by differential privacy. Roughly speaking, sensitivities measure the privacy loss suffered by each input variable when the result of a program is released. Under this reading, the rule for **bind** says that the privacy loss of an input is the sum of the losses for each sub-term. Assuming that the bound variable x has infinite sensitivity in that rule is tantamount to imposing no restrictions on its use. This is possible thanks to the composition properties of differential privacy: the result of a private computation is effectively sanitized and can be used *arbitrarily* without further harming privacy. On the contrary, the scaling factor in the rule for **return** implies that we cannot expect any privacy guarantees when releasing the result of a computation on sensitive data—every input is marked as having infinite privacy loss. In practice, results of differentially private computations must first be obfuscated with random noise (cf. Section III-D), and **return** is only used with non-private inputs and the results of previous private computations obtained by **bind**.

The above typing rules resemble those of standard monadic constructs, except for the ∞ sensitivities. We could be tempted to combine interpretations for the two sub-derivations in the **bind** rule using a monad on metric spaces as follows:

$$[\![\Delta]\!] \otimes [\![\Gamma]\!] \xrightarrow{\lambda[\![e_2]\!] \otimes [\![e_1]\!]} (\infty[\![\tau]\!] \multimap [\![\bigcirc\sigma]\!]) \otimes [\![\bigcirc\tau]\!] \xrightarrow{?} [\![\bigcirc\sigma]\!].$$

In a typical linear monadic calculus, we could just plug in the internal Kleisli lifting in the morphism marked with "?". Here, however, the types do not match up—there is the ∞ factor.

To model the probabilistic features of Fuzz, we need a structure that is similar to a monad but with slightly different types for **return** and **bind**. Our solution lies in the notion of *parameterized comonadic lifting*, which refines the operations of a preexisting monad. In addition to the max divergence originally used in Fuzz, we show how these liftings can be used to model the statistical distance, which can be seen as measuring δ in $(0, \delta)$-differential privacy. Later, we will generalize liftings to handle (ε, δ)-differential privacy.

A. Weakly Closed Monoidal Refinements

The $\otimes$ monoidal structure of Met, which lies at the core of Fuzz's linear analysis, is derived from the cartesian monoidal structure of Set. These categories are related by the forgetful functor p, which is strict monoidal. Following Melliès and Zeilberger [25], we view p as a *refinement* layering metrics on sets.

$$(\mathsf{Met}, 1, \otimes) \underset{\infty\cdot -}{\overset{p}{\rightleftarrows}} \mathsf{Set} \tag{8}$$

The above monoidal structures are closed, but exponentials only match up for discrete metric spaces $\infty \cdot Z$—that is, (p, p) is a map of adjunctions of type

$$(- \otimes (\infty \cdot Z) \dashv (\infty \cdot Z) \multimap -) \to (- \times Z \dashv Z \Rightarrow -).$$

Parameterized comonadic liftings are based on a generalization of this situation:

Definition 1. *A* weakly closed monoidal refinement *of a symmetric monoidal closed category (SMCC)* $(\mathbb{B}, \mathbf{I}, \otimes, \multimap)$ *consists of a symmetric monoidal category* $(\mathbb{E}, \dot{\mathbf{I}}, \dot{\otimes})$ *and an adjunction satisfying the following four conditions:*

$$(\mathbb{E}, \dot{\mathbf{I}}, \dot{\otimes}) \underset{L}{\overset{p}{\rightleftarrows}} \mathbb{B} \tag{9}$$

1) *p is strict symmetric monoidal and faithful;*

2) *the unit of the adjunction is the identity;*
3) *for each* $X \in \mathbb{B}$, $- \dot{\otimes} LX$ *has a right adjoint* $X \dot{\pitchfork} -$;
4) *for each* $X \in \mathbb{B}$, (p, p) *is a map of adjunction of type* $(- \otimes LX \dashv X \dot{\pitchfork} -) \to (- \otimes X \dashv X \multimap -)$.

There are many such refinements. Since $- \times (\infty \cdot X)$ is equal to $- \otimes (\infty \cdot X)$, it also has $\infty \cdot X \multimap -$ as a right adjoint, which yields another example involving Met:

$$(\mathsf{Met}, 1, \times) \underset{\infty\cdot-}{\overset{p}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathsf{Set} \tag{10}$$

We'll see further examples in Section IV when extending Fuzz with (ε, δ)-differential privacy.

Given a weakly closed monoidal refinement as in (9), we write ! for the comonad $L \circ p$. For $X, Y \in \mathbb{E}$ and a morphism $f : pX \to pY$ in $\mathbb{B}$, by $f : X \dot{\to} Y$ we mean that there exists a (necessarily unique) morphism $\dot{f} : X \to Y$ such that $p\dot{f} = f$. Since the unit of the adjunction is the identity, we have 1) $\forall f \in \mathbb{B}(X, pY)$. $f : LX \dot{\to} Y$ and 2) $\forall f \in \mathbb{E}(LX, Y)$. $pf : LX \dot{\to} Y$.

B. Parameterized Comonadic Liftings

Consider this simplified instance of the Fuzz **bind** rule:

$$\dfrac{\overline{y :_1 \bigcirc\tau \vdash y : \bigcirc\tau} \qquad \dfrac{\vdots}{\Delta, x :_\infty \tau \vdash e : \bigcirc\sigma}}{\Delta, y :_1 \bigcirc\tau \vdash \mathbf{bind}\, x \leftarrow y; e : \bigcirc\sigma}$$

Recall that **bind** samples x from $y \notin \Delta$ and computes e. In a set-theoretic semantics that ignores sensitivities, $\bigcirc$ might correspond to the monad D of discrete probability distributions on Set, and we can use the Kleisli lifting of $[\![e]\!]$ to interpret the entire derivation. We can refine this interpretation with metrics by lifting D—that is, finding a functor $\dot{D}$ such that $p \circ \dot{D} = D \circ p$—provided that the following implication holds:

$$f : [\![\Delta]\!] \otimes ![\![\tau]\!] \overset{\text{ne}}{\to} \dot{D}[\![\sigma]\!] \implies f^\ddagger : [\![\Delta]\!] \otimes \dot{D}[\![\tau]\!] \overset{\text{ne}}{\to} \dot{D}[\![\sigma]\!]$$

The notion of parameterized comonadic lifting arises by generalizing this argument to other monads.

Definition 2. *A* $\dot{\otimes}$-parameterized !-lifting of $\mathcal{T}$ along a weakly closed monoidal refinement $p : \mathbb{E} \to \mathbb{B}$ *is a mapping* $\dot{T} : |\mathbb{E}| \to |\mathbb{E}|$ *such that 1)* $p\dot{T}X = TpX$*, and 2) for any* $X, Y, Z \in \mathbb{E}$ *and* $\mathbb{C}$*-morphism* f *such that* $f : Z \dot{\otimes} !X \dot{\to} \dot{T}Y$*, its parameterized Kleisli lifting (see* (4)*) satisfies* $f^\ddagger : Z \dot{\otimes} \dot{T}X \dot{\to} \dot{T}Y$.

Every $\dot{\otimes}$-parameterized !-lifting $\dot{T}$ satisfies, for any $X \in \mathbb{E}$,

$$\eta_{pX} : !X \dot{\to} \dot{T}X, \tag{11}$$

which allows us to extend $\dot{T}$ to a functor of type $\mathbb{E} \to \mathbb{E}$.

To model probability distributions in Fuzz, we pick a $\otimes$-parameterized !-lifting $\dot{D}$ of the distribution monad $\mathcal{D}$. The choice of $\dot{D}$ can vary—as we will soon see, one such !-lifting models the max divergence in the original Fuzz. A distribution type $\bigcirc\tau$ is mapped to $\dot{D}[\![\tau]\!]$, and **return** is interpreted using (11) by setting

$$[\![\mathbf{return}\, e]\!] : [\![\infty\Gamma]\!] \to [\![\bigcirc\tau]\!]$$
$$[\![\mathbf{return}\, e]\!] \triangleq \eta_{p[\![\tau]\!]} \circ (\infty \cdot [\![e]\!]).$$

Since ! commutes with $\otimes$, the first factor is well-typed. We interpret **bind** as a parameterized Kleisli lifting:

$$[\![\mathbf{bind}\, x \leftarrow e_1; e_2]\!] : [\![\Delta + \Gamma]\!] \to [\![\bigcirc\tau]\!]$$
$$[\![\mathbf{bind}\, x \leftarrow e_1; e_2]\!] \triangleq [\![e_2]\!]^\ddagger \circ ([\![\Delta]\!] \otimes [\![e_1]\!]) \circ \delta.$$

To illustrate possible variations, the same interpretation works if we choose $\dot{D}$ to be a $\times$-parameterized !-lifting of $\mathcal{D}$: since $Z \otimes !X = Z \times !X$ and because $\mathrm{id}_{pX,pY} : X \otimes Y \overset{\text{ne}}{\to} X \times Y$ is non-expansive, any $\times$-parameterized lifting is also a $\otimes$-parameterized lifting. However, in this case we can also strengthen the typing rule for **bind** as follows:

$$\frac{\Gamma \vdash e_1 : \bigcirc\tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc\sigma}{\Gamma \vdash \mathbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc\sigma}, \tag{12}$$

where the context Γ is shared between e_1 and e_2, yielding an additive variant of the Fuzz bind rule (in the sense of linear logic). Observe that the domain of $[\![e_2]\!]$ is defined as $[\![\Gamma]\!] \otimes ![\![\tau]\!]$, which is equal to $[\![\Gamma]\!] \times ![\![\tau]\!]$. Thus we can apply the parameterized Kleisli lifting to $[\![e_2]\!]$. The interpretation of **bind** $x \leftarrow e_1; e_2$ is then given as the following composite:

$$[\![\mathbf{bind}\, x \leftarrow e_1; e_2]\!] = [\![\Gamma]\!] \xrightarrow{\langle \mathrm{id}, [\![e_1]\!] \rangle} [\![\Gamma]\!] \times \dot{D}[\![\tau]\!] \xrightarrow{[\![e_2]\!]^\ddagger} \dot{D}[\![\sigma]\!] .$$

We will see examples of $\times$-parameterization in Section VII after we introduce graded liftings.

C. Constructing Liftings via Parameterized Assignments

To express the max divergence with a comonadic lifting, we appeal to results of Barthe and Olmedo [4]. Their results are phrased in terms of a notion of *composability*, which we can recast as a sensitivity property.

Lemma 1. *Consider the situation Eq.* (8)*. Let* $\Delta : |\mathsf{Set}| \to |\mathsf{Met}|$ *be a mapping such that* $p\Delta X = DX$*. Then* Δ *satisfies the composability condition [4]:*

$$d_{\Delta Y}(f^\dagger\mu, g^\dagger\nu) \le d_{\Delta X}(\mu, \nu) + \sup_{x \in X} d_{\Delta Y}(f(x), g(x))$$

if and only if the internalized Kleisli lifting *of* $\mathcal{T}$ *is a non-expansive map* $\mathrm{kl}^{\mathcal{D}}_{X,Y} : (\infty \cdot X \multimap \Delta Y) \otimes \Delta X \overset{\text{ne}}{\to} \Delta Y$.

This equivalent formulation can be readily generalized to other weakly closed monoidal refinements.

Definition 3. *A* $\dot{\otimes}$-parameterized assignment of $\mathbb{E}$ on $\mathcal{T}$ *in a weakly closed monoidal refinement* $p : \mathbb{E} \to \mathbb{B}$ *is a mapping* $\Delta : |\mathbb{B}| \to |\mathbb{E}|$ *such that* $p\Delta X = TX$ *and the internalized Kleisli lifting* $\mathrm{kl}^{\mathcal{T}}$ *of* $\mathcal{T}$ *in Eq.* (5) *satisfies*

$$\mathrm{kl}^{\mathcal{T}}_{X,Y} : (X \pitchfork \Delta Y) \dot{\otimes} \Delta X \dot{\to} \Delta Y. \tag{13}$$

Parameterized assignments turn out to be just an alternative presentation of parameterized !-liftings—the former arising from concepts of probability theory, and the latter mimicking the Fuzz typing rules. Formally, we have:

Theorem 1. *Consider a weakly closed monoidal refinement* (9) *and a* $\otimes$*-strong monad* $\mathcal{T}$ *on* $\mathbb{B}$*. There is an equivalence of preorders between*

1) $!\mathbf{Lift}_{\dot\otimes}(\mathcal{T})$*, the subpreorder of* $\mathbf{Ord}(p, T \circ p)$ *consisting of* $\dot\otimes$*-parameterized* $!$*-liftings of* $\mathcal{T}$*; and*
2) $\mathbf{Asign}_{\dot\otimes}(\mathcal{T})$*, the subpreorder of* $\mathbf{Ord}(p, T)$ *consisting of* $\dot\otimes$*-parameterized assignments of* $\mathbb{E}$ *on* $\mathcal{T}$*,*

where, given $F : A \to |\mathbb{B}|$*,* $\mathbf{Ord}(p, F)$ *is the class of mappings* $\{G : A \to |\mathbb{E}| \mid pGX = FX\}$ *ordered by* $G \le G' \iff \forall a \in A \,.\, \mathrm{id}_{Fa} : Ga \dot\to G'a$*.*

D. Max Divergence

Since the max divergence satisfies the composability condition [4], Theorem 1 allows us to derive a corresponding parameterized lifting. In addition to the basic monadic operations, this lifting supports the real-valued *Laplace* distribution, a fundamental building block in differential privacy. We can make a database query differentially private by adding Laplace noise to its result while calibrating the scale of the noise according to the query's sensitivity, as measured in terms of a suitable metric on databases. Its density function is given by

$$L(\mu, b)(x) \triangleq \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right),$$

where $\mu \in \mathbb{R}$ and $b > 0$ are parameters controlling the mean and the scale of the distribution. The Laplace distribution induces a discrete distribution $\hat{L}(\mu, b) \in D\mathbb{R}$ by truncating the sample up to some fixed precision (breaking ties arbitrarily). This new distribution is compatible with the max divergence: it satisfies the following max divergence bound:

$$\mathsf{MD}_{\mathbb{R}}(\hat{L}(\mu, b), \hat{L}(\mu', b)) \le \frac{|\mu - \mu'|}{b}.$$

In other words, the mapping $\mu \mapsto \hat{L}(\mu, b)$ is a b^{-1}-sensitive function from $\mathbb{R}$ to $D\mathbb{R}$ equipped with the max divergence.[3]

Fuzz exposes the Laplace distribution as a primitive

$$\textbf{Laplace}[\varepsilon] : !_\varepsilon \mathbb{R} \multimap \bigcirc \mathbb{R},$$

originally called *add_noise*. We can interpret this as:

$$[\![\textbf{Laplace}[\varepsilon]]\!] \triangleq \lambda x.\, \hat{L}(x, 1/\varepsilon).$$

The full Fuzz language also provides a type $\mathsf{set}\,\tau$ of finite sets with elements drawn from τ, used to model sets of private data ("databases"). This type is equipped with the Hamming distance, which is compatible with the primitives operations on sets, e.g., computing the size, filtering according to a predicate, etc. Extending the interpretation of Fuzz accordingly, a function of type $!_\varepsilon\, \mathsf{set}\,\tau \multimap \bigcirc\mathbb{R}$ corresponds to an ε-sensitive function from databases to distributions over $\mathbb{R}$ equipped with the max divergence. As we discussed in the Introduction, this sensitivity property is equivalent to ε-differential privacy with respect to the adjacency relation relating pairs of databases at Hamming distance at most 1, i.e., databases differing in at most one record.

[3]This follows from ε-privacy of the Laplace mechanism and stability of max divergence under post-processing (see, e.g., Dwork and Roth [10]).

E. Statistical Distance

Barthe and Olmedo [4] show that the composability condition is also valid for the statistical distance:

$$\mathsf{SD}_X(\mu, \nu) \triangleq \frac{1}{2} \sum_{i \in X} |\mu(i) - \nu(i)|.$$

This allows us to extend Fuzz with a new type constructor $\bigcirc^{\mathsf{SD}}$, which we interpret using the statistical distance and its corresponding lifting. In addition to **return** and **bind**, we can soundly incorporate a primitive to compute the Bernoulli distribution, which models a biased coin flip:

$$\textbf{Bernoulli} : \mathbb{R} \multimap \bigcirc^{\mathsf{SD}} \mathbb{B}$$
$$[\![\textbf{Bernoulli}]\!](p)(\textbf{true}) \triangleq \min(\max(p, 0), 1)$$
$$[\![\textbf{Bernoulli}]\!](p)(\textbf{false}) \triangleq \min(\max(1-p, 0), 1).$$

It is straightforward to check that the Bernoulli distribution satisfies the following statistical distance bound:

$$\mathsf{SD}_{\mathbb{B}}([\![\textbf{Bernoulli}]\!](p), [\![\textbf{Bernoulli}]\!](p')) \le |p - p'|,$$

implying that the type stated above is sound.

IV. Relations and (ε, δ)-Differential Privacy

We will now shift gears and consider how to extend Fuzz to handle (ε, δ)-differential privacy. Recall that (ε, δ)-privacy is a *relational* property: a query f satisfies the definition if it maps pairs of related input databases to related output distributions, for suitable notions of "relatedness." What makes this notion challenging for Fuzz is that it cannot be phrased directly as a sensitivity property (except for the special case $\delta = 0$, which we analyzed above). Rather than resorting to an entirely different verification technique, we propose to incorporate relational reasoning into Fuzz by embedding relations into metric spaces.

To warm up, we first show how to define (ε, δ)-differential privacy in terms of a category of relations. Later (Section V), we use this formulation to capture the composition properties of differential privacy with graded versions of parameterized liftings. Then, we consider how to transfer these structures from relations to metric spaces via the *path construction* (Section VI). Finally, we extend Fuzz with grading to support relational properties (Section VII).

A. Differential Privacy in RSRel

We begin by fixing a category of relations to work in. To smooth the eventual transfer to metrics, which are reflexive and symmetric, we work with reflexive and symmetric relations.

Definition 4. *The category* RSRel *of reflexive, symmetric relations has as objects pairs* $X = (|X|, \sim_X)$ *of a carrier set* $|X|$ *and a reflexive, symmetric relation* $\sim_X \subseteq |X| \times |X|$*. We will often use the carrier set* $|X|$ *to refer to* X*, and write* $\sim$ *when the underlying space is clear. A morphism*

$X \to Y$ *is a function from* X *to* Y *that preserves the relation:* $x \sim x' \implies f(x) \sim f(x')$. *For* $X, Y \in \mathsf{RSRel}$ *and* $f : |X| \to |Y|$, *we write* $f : X \stackrel{\text{re}}{\to} Y$ *to mean* $f \in \mathsf{RSRel}(X, Y)$.

The category RSRel has a terminal object, binary products and exponentials ($X, Y \in \mathsf{RSRel}$):

$$1 \triangleq (1, 1 \times 1)$$
$$X \times Y \triangleq (|X| \times |Y|, \{((x, y), (x', y')) \mid x \sim x', y \sim y'\})$$
$$X \Rightarrow Y \triangleq (|X| \Rightarrow |Y|, \{(f, f') \mid \forall x \sim x'.\ f(x) \sim f'(x')\}).$$

hence RSRel is a CCC.[4] The forgetful functor $q : \mathsf{RSRel} \to \mathsf{Set}$, defined by $qX = |X|$ and $qf = f$, has a left adjoint $M : \mathsf{Set} \to \mathsf{RSRel}$ endowing a set X with the diagonal relation. Moreover, q strictly preserves the cartesian closed structure, hence is a weakly closed monoidal refinement of the CCC Set.

$$(\mathsf{RSRel}, 1, \times) \underset{M}{\overset{q}{\rightleftarrows}} \mathsf{Set} \qquad (14)$$

The definition of differential privacy is parameterized by a set db of *databases*, along with a binary *adjacency* relation $adj \subseteq db \times db$, which we assume to be symmetric and reflexive. Conventional choices for db include the set of sets of (or multisets, or lists) of records from some universe of possible data, while adj could relate pairs of databases at symmetric difference at most 1. We recall the original definition here for convenience.

Definition 5 (Dwork et al. [11]). *Let* $\varepsilon, \delta \in [0, \infty)$. *A randomized computation* $f : db \to DX$ *is* (ε, δ)-differentially private *if for all pairs of adjacent databases* $(d, d') \in adj$ *and subsets of outputs* $S \subseteq X$, *we have:*

$$f(d)(S) \leq \exp(\varepsilon) \cdot f(d')(S) + \delta \quad \text{and}$$
$$f(d')(S) \leq \exp(\varepsilon) \cdot f(d)(S) + \delta.$$

We can track the privacy parameters by attaching the following *indistinguishability relation* to the *codomain* of a differentially private algorithm. Given $\varepsilon, \delta \in [0, \infty)$ and a set X, we define $\mathsf{DPR}(\varepsilon, \delta)(X) \in \mathsf{RSRel}$ by setting

$$\begin{aligned}\mathsf{DPR}(\varepsilon, \delta)(X) \triangleq (DX, \{(\mu, \nu) \mid \forall S \subseteq X\ .\ & \\ &(\mu(S) \leq \exp(\varepsilon) \cdot \nu(S) + \delta) \wedge \\ &(\nu(S) \leq \exp(\varepsilon) \cdot \mu(S) + \delta)\}).\end{aligned}$$

Proposition 1. *A function* $f : db \to DX$ *is* (ε, δ)*-differentially private if and only if* $f : (db, adj) \stackrel{\text{re}}{\to} \mathsf{DPR}(\varepsilon, \delta)(X)$.

Like ε-differential privacy, (ε, δ)-differential privacy behaves well under *sequential composition*.

Theorem 2 (Dwork et al. [12]). *Let* $f : db \to DX$ *and* $g : db \times X \to DY$ *be such that 1)* f *is* (ε, δ)*-differentially private, and 2)* $g(-, x) : db \to DY$ *is* (ε', δ')*-differentially private for every* $x \in X$. *Then the composite function* $d \mapsto g^\ddagger(d, f(d))$ *is* $(\varepsilon + \varepsilon', \delta + \delta')$*-differentially private.*

[4] We note that RSRel has another symmetric monoidal closed structure with tensor product $X \otimes Y = (|X| \times |Y|, \{((x, y), (x, y')) \mid y \sim y'\} \cup \{((x, y), (x', y)) \mid x \sim x'\})$. The functor q is also a weakly closed monoidal refinement of type $(\mathsf{RSRel}, 1, \otimes) \to \mathsf{Set}$.

V. Graded !-Liftings

When reasoning about ε-differential privacy in Met (and in Fuzz), the privacy parameter is reflected in the scale of the *domain* of a non-expansive map, and the composition principle of $(\varepsilon, 0)$-privacy corresponds to composition of non-expansive maps. In RSRel, there is no analogous scaling operation for reasoning about (ε, δ)-privacy. To track these parameters through composition, we will instead use the query *codomain* by extending the liftings of Section III with *monoid grading* (analogously to graded extensions of monads [18, 24, 33]). The generality of these graded counterparts, which abstract the composition properties of Theorem 2, will prove useful later on (Section VII-B), when modeling the composition behavior of other relational properties.

We assume a weakly closed monoidal refinement of a SMCC $(\mathbb{B}, \mathbf{I}, \otimes, \multimap)$, as in (9), fixing a $\otimes$-strong monad over $\mathbb{B}$, and a preordered monoid $(M, \leq, 1, \cdot)$.

Definition 6. *An* M-graded $\dot{\otimes}$-parameterized !-lifting *is a monotone function* $\dot{T} : (M, \leq) \to \mathbf{Ord}(p, T \circ p)$ *(cf. Theorem 1) such that the parameterized Kleisli lifting Eq.* (4) *of* $\mathcal{T}$ *satisfies*

$$f : Z \dot{\otimes} !X \dot{\to} \dot{T}\alpha Y \implies f^\ddagger : Z \dot{\otimes} \dot{T}\beta X \dot{\to} \dot{T}(\beta \cdot \alpha)Y.$$

When $M = 1$, this definition reduces to its non-graded counterpart. Monotonicity of an M-graded $\dot{\otimes}$-parameterized !-lifting $\dot{T}$ means that for any monoid element $\alpha \leq \beta$ and any object $X \in \mathbb{E}$, we have $\dot{T}\alpha X \leq \dot{T}\beta X$. Regarding the unit, $\eta_{pX} : !X \dot{\to} \dot{T}\alpha X$ holds for any $\alpha \in M$ and $X \in \mathbb{E}$. From this, each $\dot{T}\alpha$ extends to an endofunctor over $\mathbb{E}$.

Definition 7. *An* M-graded $\dot{\otimes}$-parameterized assignment of $\mathbb{E}$ on $\mathcal{T}$ *is a monotone function* $\Delta : (M, \leq) \to \mathbf{Ord}(p, T)$ *such that the internalized Kleisli lifting morphism* $\mathrm{kl}^{\mathcal{T}}$ *of* $\mathcal{T}$ *in Eq.* (5) *satisfies*

$$\mathrm{kl}^{\mathcal{T}}_{X,Y} : (X \pitchfork \Delta\alpha Y) \dot{\otimes} \Delta\beta X \dot{\to} \Delta(\beta \cdot \alpha)Y. \qquad (15)$$

Once again, the original definition corresponds to the case $M = 1$. To illustrate this notion, in the weakly closed monoidal refinement (8), an M-graded $\otimes$-parameterized assignment Δ of Met on $\mathcal{D}$ consists of a family of metrics $d_{\Delta\alpha X}$ on DX, indexed by $\alpha \in M$, such that, for any $X, Y \in \mathsf{Set}, \mu, \nu \in DX, f, g : X \to DY$ and $\alpha, \beta \in M$, we have

$$d_{\Delta(\alpha \cdot \beta)Y}(f^\dagger \mu, g^\dagger \nu) \leq d_{\Delta\alpha X}(\mu, \nu) + \sup_{x \in X} d_{\Delta\beta Y}(f(x), g(x)).$$

In this case, assignments encode a family of distances on distributions enjoying the sequential composition theorem for statistical divergences proposed by Barthe and Olmedo [4, Theorem 1] (see also Olmedo [28]).

An instance of a graded assignment is the indistinguishability relation for differential privacy. The following is a consequece of Theorem 2:

Proposition 2. *Let* $\mathbb{R}^+_{\geq 0}$ *be the additive monoid on the set* $\mathbb{R}_{\geq 0}$ *of nonnegative real numbers. In the weakly closed monoidal* *refinement* (14)*, the indistinguishability relation, regarded as a mapping of type* $\mathsf{DPR} : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to |\mathsf{RSRel}|$*, is an* $(\mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0})$*-graded* $\times$*-parameterized assignment of* RSRel *on* $\mathcal{D}$*.*

We have a graded analogue of Theorem 1.

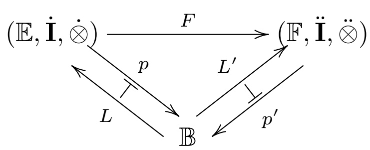
Theorem 3. *Consider a weakly closed monoidal refinement* (9) *and a* $\otimes$*-strong monad* $\mathcal{T}$ *on* $\mathbb{B}$*. Let* M *be a preordered monoid. The following preorders are equivalent:*

1) $!\mathbf{Lift}_{\dot{\otimes}}(\mathcal{T}, M)$*, the preordered class of* M*-graded* $\dot{\otimes}$*-parameterized* !*-liftings of* $\mathcal{T}$ *with the pointwise preorder, and*
2) $\mathbf{Asign}_{\dot{\otimes}}(\mathcal{T}, M)$*, the preordered class of* M*-graded* $\dot{\otimes}$*-parameterized assignments on* $\mathcal{T}$ *with the pointwise preorder.*

VI. Transfers of Assignments

So far, we are able to model relational properties and their composition behavior in RSRel. In this section, we will show how to carry out this reasoning in a different category, namely Met. We introduce this idea abstractly, then give a concrete example called the *path construction* for the special case of RSRel and Met. In our framework, this transfer of structure is induced by a morphism between weakly closed monoidal refinements.

Definition 8. *Consider two weakly closed monoidal refinements of a SMCC* $\mathbb{B}$*, and a functor* $F : \mathbb{E} \to \mathbb{F}$*:*



(16)

F *is a* morphism of weakly closed monoidal refinements *if*

1) F *is strict symmetric monoidal,*
2) $(\mathrm{Id}_{\mathbb{B}}, F) : (L \dashv p) \to (L' \dashv p')$*, and*
3) $(F, F) : (- \mathbin{\dot{\otimes}} LX \dashv X \mathbin{\dot{\pitchfork}} -) \to (- \mathbin{\ddot{\otimes}} L'X \dashv X \mathbin{\ddot{\pitchfork}} -)$ *for each* $X \in \mathbb{B}$*.*

We write $F : (\mathbb{E}, \dot{\mathbf{I}}, \dot{\otimes}, L, p) \to_{\mathbb{B}} (\mathbb{F}, \ddot{\mathbf{I}}, \ddot{\otimes}, L', p')$*.*

Theorem 4. *If* $F : \mathbb{E} \to \mathbb{F}$ *is a morphism of weakly closed monoidal refinements, then* $F \circ -$ *restricts to a monotone function of type* $\mathbf{Asign}_{\dot{\otimes}}(\mathcal{T}, M) \to \mathbf{Asign}_{\ddot{\otimes}}(\mathcal{T}, M)$*.*

We express (ε, δ)-privacy in metric spaces through the *path construction functor* $P : \mathsf{RSRel} \to \mathsf{Met}$. Given an object $X \in \mathsf{RSRel}$, we define a metric on the underlying set by counting the number of times $\sim_X$ must be composed to relate two points. Such metrics are also known as *path metrics*, and the corresponding metric spaces are known as *path-metric spaces*.

Definition 9. *Let* $X \in \mathsf{RSRel}$*. The* path metric *is a metric on* X *defined as follows:* $d(x, x')$ *is the length* k *of the shortest path of elements* $x_0, \ldots, x_k$ *such that* $x_0 \triangleq x$*,* $x_k \triangleq x'$*, and* $x_i \sim x_{i+1}$ *for every* $i \in \{0, \ldots, k-1\}$*. If no such sequence exists, we set* $d(x, x') \triangleq \infty$*. We write* PX *for the corresponding metric space. This definition can be extended to a functor* $\mathsf{RSRel} \to \mathsf{Met}$ *that acts as the identity on morphisms.*

Conversely, we can turn any metric space into an object of RSRel by relating elements at distance at most 1.

Definition 10 (At most one)**.** *Given* $X \in \mathsf{Met}$*, we define* $QX \in \mathsf{RSRel}$ *by*

$$QX = (X, \{(x, x') \mid d(x, x') \leq 1\}), \quad Qf = f.$$

Theorem 5. *The functor* $P : \mathsf{RSRel} \to \mathsf{Met}$ *is fully faithful, and a left adjoint to* $Q : \mathsf{Met} \to \mathsf{RSRel}$*.*

Theorem 6. *The path metric functor* $P : \mathsf{RSRel} \to \mathsf{Met}$ *is a morphism of weakly closed monoidal refinements:*

$$P : (\mathsf{RSRel}, 1, \times, M, q) \to_{\mathsf{Set}} (\mathsf{Met}, 1, \times, \infty \cdot -, p).$$

VII. Metric Semantics of Graded Fuzz

We have all the ingredients we need to extend Fuzz. We fix a preordered monoid $(M, \leq, 1, \cdot)$, augment Fuzz with M-graded monadic types $\bigcirc_\alpha \tau$, adjust the typing rules (7) and (6) to track the grading, and add a grading subsumption rule.

$$\frac{\Gamma \vdash e : \bigcirc_\alpha \tau \qquad \alpha \leq \beta}{\Gamma \vdash e : \bigcirc_\beta \tau} \tag{17}$$

$$\frac{\Gamma \vdash e : \tau}{\infty \cdot \Gamma \vdash \mathbf{return}\, e : \bigcirc_\alpha \tau} \tag{18}$$

$$\frac{\Gamma \vdash e_1 : \bigcirc_\alpha \tau \qquad \Delta, x :_\infty \tau \vdash e_2 : \bigcirc_\beta \sigma}{\Delta + \Gamma \vdash \mathbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc_{\alpha \cdot \beta} \sigma} \tag{19}$$

Here, α and β range over M. The interpretation in Section III can be easily extended when we have an M-graded $\otimes$-parameterized !-lifting $\dot{D}$ of the distribution monad $\mathcal{D}$ along $p : \mathsf{Met} \to \mathsf{Set}$, taking $[\![\bigcirc_\alpha \tau]\!] = \dot{D}\alpha[\![\tau]\!]$.

If instead we have a M-graded $\times$-parameterized !-lifting $\dot{D}$ of $\mathcal{D}$ along $p : \mathsf{Met} \to \mathsf{Set}$, we can again interpret the monadic type with $[\![\bigcirc_\alpha \tau]\!] = \dot{D}\alpha[\![\tau]\!]$. However, we can replace (19) with a stronger rule for **bind**:

$$\frac{\Gamma \vdash e_1 : \bigcirc_\alpha \tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc_\beta \sigma}{\Gamma \vdash \mathbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc_{\alpha \cdot \beta} \sigma} \tag{20}$$

A. Modeling (ε, δ)-Differential Privacy in Graded Fuzz

We have seen that differential privacy can be expressed in RSRel, with the sequential composition property providing a graded assignment that can be transferred to Met through the path metric. By Proposition 3 and Theorem 3, in the weakly closed monoidal refinement (14) we have $\mathsf{DPR} \in \mathbf{Asign}_{\dot{\times}}(\mathcal{D}, \mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0})$. From Theorem 4, the path construction maps DPR to a graded $\dot{\times}$-parameterized assignment of Met on $\mathcal{D}$:

$$P \circ \mathsf{DPR} \in \mathbf{Asign}_{\dot{\times}}(\mathcal{D}, \mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0})$$

in the weakly closed monoidal refinement (10). Below we identify $P \circ \mathsf{DPR}$ as a $\mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0}$-graded $\dot{\times}$-parameterized !-lifting of $\mathcal{D}$. By posing

$$[\![\bigcirc_{(\varepsilon,\delta)} \tau]\!] \triangleq P(\mathsf{DPR}(\varepsilon, \delta)([\![\tau]\!])),$$

the following specializations of (17), (18), and (20) are sound:

$$\frac{\Gamma \vdash e : \bigcirc_{(\varepsilon,\delta)}\tau \qquad \varepsilon \leq \varepsilon' \qquad \delta \leq \delta'}{\Gamma \vdash e : \bigcirc_{(\varepsilon',\delta')}\tau}$$

$$\frac{\Gamma \vdash e : \tau}{\infty \cdot \Gamma \vdash \textbf{return}\, e : \bigcirc_{(\varepsilon,\delta)}\tau}$$

$$\frac{\Gamma \vdash e_1 : \bigcirc_{(\varepsilon,\delta)}\tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc_{(\varepsilon',\delta')}\sigma}{\Gamma \vdash \textbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc_{(\varepsilon+\varepsilon',\delta+\delta')}\sigma}$$

Much like we did for the standard !-liftings in Sections III-D and III-E, we can add primitive distributions to the system. The basic building block for (ε, δ)-differential privacy is the real-valued *Gaussian* or *normal* distribution. Given a mean $\mu \in \mathbb{R}$ and a variance $\sigma \in \mathbb{R}$, this distribution has density function

$$N(\mu, \sigma)(x) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right).$$

A result from the theory of differential privacy states that if we have a numeric query $q : db \to \mathbb{R}$ whose results differ by at most 1 on adjacent databases, then adding noise drawn from $N(0, \sigma)$ to the query's result yields an (ε, δ)-differentially private algorithm as long as $\sigma \geq s(\varepsilon, \delta) \triangleq 2\ln(1.25/\delta)/\varepsilon$ (see, e.g., Dwork and Roth [10, Theorem A.1]). Like we did for the Laplace distribution, we may discretize the result of the noised query to any fixed precision while preserving privacy, yielding a distribution we call $\hat{N}(\mu, \sigma)$. The function $\lambda x.\, \hat{N}(x, \sigma) : \mathbb{R} \to D\mathbb{R}$ can then be interpreted in RSRel:

$$\lambda x.\, \hat{N}(x, s(\varepsilon, \delta)) : Q\mathbb{R} \overset{\text{re}}{\to} \mathsf{DPR}(\varepsilon, \delta)(\mathbb{R})$$

Unfolding definitions, $Q\mathbb{R}$ relates pairs of real numbers that are at most 1 apart under the standard Euclidean distance. The path construction gives a non-expansive map between path-metric spaces:

$$\lambda x.\, \hat{N}(x, s(\varepsilon, \delta)) : PQ\mathbb{R} \overset{\text{ne}}{\to} P(\mathsf{DPR}(\varepsilon, \delta)(\mathbb{R}))$$

The metric space $PQ\mathbb{R}$ is $\mathbb{R}$ but with the metric rounded up to the nearest integer; we introduce a corresponding Fuzz type $\lceil\mathbb{R}\rceil$. Then, we can introduce a new Fuzz term **Gaussian**$[\varepsilon, \delta]$ for $\varepsilon, \delta > 0$ with type

$$\textbf{Gaussian}[\varepsilon, \delta] : \lceil\mathbb{R}\rceil \multimap \bigcirc_{(\varepsilon,\delta)}\mathbb{R}$$

and the interpretation

$$[\![\textbf{Gaussian}[\varepsilon, \delta]]\!] \triangleq \lambda x.\, N(x, s(\varepsilon, \delta)).$$

Typing (ε, δ)-Differential Privacy: We can now capture (ε, δ)-privacy via Fuzz types. Consider the judgment:

$$\vdash e : \mathsf{db} \multimap \bigcirc_{(\varepsilon,\delta)}\tau$$

where db is interpreted as the path-metric space $P(db, adj)$. Note that if $\mathsf{db} = \mathsf{set}\,\sigma$ is the space of sets of σ and we take the Hamming distance as the metric $d_D B$ on this space (as we did in previous examples), then (db, d_{DB}) is automatically a path metric space for the relation relating any two databases at Hamming distance at most 1. We have a non-expansive map

$$[\![e]\!] : P(db, adj) \overset{\text{ne}}{\to} P(\mathsf{DPR}(\varepsilon, \delta)([\![\tau]\!])).$$

Since the path functor P is full and faithful (Theorem 5), we have a relation-preserving map

$$[\![e]\!] : (db, adj) \overset{\text{re}}{\to} \mathsf{DPR}(\varepsilon, \delta)([\![\tau]\!])$$

in RSRel. By Proposition 1, this map satisfies (ε, δ)-privacy. We give two examples to demonstrate the type system. Consider the type $db \multimap \mathbb{R}$, typically used to model 1-*sensitive queries*. Applying PQ, we find that we can model 1-sensitive queries with the type $\lceil db\rceil \multimap \lceil\mathbb{R}\rceil$. Now $\lceil db\rceil$ rounds up the metric on db to the nearest integer, but since db is already a path metric space, $\lceil db\rceil$ and db have the same denotations. Thus, 1-sensitive queries can be interpreted as type $\mathsf{db} \multimap \lceil\mathbb{R}\rceil$.

Let q_1 and q_2 be 1-sensitive queries of type $\mathsf{db} \multimap \lceil\mathbb{R}\rceil$, and consider the program two_q:

$$\begin{aligned}\lambda db.\, &\textbf{bind}\, a_1 \leftarrow \textbf{Gaussian}[\varepsilon, \delta](q_1(db));\\ &\textbf{bind}\, a_2 \leftarrow \textbf{Gaussian}[\varepsilon, \delta](q_2(db));\\ &\textbf{return}(a_1 + a_2)\end{aligned}$$

This program evaluates the first query q_1 and adds Gaussian noise to the answer, evaluates the second query q_2 and adds more Gaussian noise, and finally returns the sum of the two noisy answers. By applying the typing rules for the Gaussian distribution, along with the graded monadic rules, we can derive the following type:

$$\vdash two\_q : \mathsf{db} \multimap \bigcirc_{(2\varepsilon,2\delta)}\mathbb{R}$$

Though the database db is used twice in the program, it has sensitivity 1 in the final type. This accounting follows from the bind rule, which allows the contexts of its premises to be shared. The fact that the database is used twice is instead tracked through the grading on the codomain, where the privacy parameters (ε, δ) sum up. By soundness of the type system, two_q is $(2\varepsilon, 2\delta)$-differentially private.

Other types can capture variants of differential privacy. For example, suppose added the queries first and noised just once:

$$\lambda db.\, \textbf{let}\, s \leftarrow q_1(db) + q_2(db); \textbf{Gaussian}[\varepsilon, \delta](s)$$

We use standard syntactic sugar for let bindings; call this program two_q'. We can derive the following type:

$$\vdash two\_q' : !_2\mathsf{db} \multimap \bigcirc_{(\varepsilon,\delta)}\mathbb{R}$$

This type is not equivalent to the type for two_q. However, we can still interpret it in terms of differential privacy. In general, consider the following judgment:

$$\vdash e : !_2\mathsf{db} \multimap \bigcirc_{(\varepsilon,\delta)}\tau$$

By soundness, the interpretation is non-expansive:

$$[\![e]\!] : 2 \cdot P(db, adj) \overset{\text{ne}}{\to} P(\mathsf{DPR}(\varepsilon, \delta)([\![\tau]\!])).$$

Though the scaling of a path metric is not necessarily a path metric, we can still give a meaning to this judgment. For any two input databases $(x, x') \in adj$, the distance between $[\![e]\!]x$ and $[\![e]\!]x'$ in $P(\mathsf{DPR}(\varepsilon, \delta)([\![\tau]\!]))$ is at most 2. Suppose that the distance is exactly 2 (smaller distances yield stronger privacy bounds). Then, there must exist an intermediate distribution $y \in D|[\![\tau]\!]|$ such that

$$[\![e]\!]x \sim_{\mathsf{DPR}(\varepsilon,\delta)([\![\tau]\!])} y \sim_{\mathsf{DPR}(\varepsilon,\delta)([\![\tau]\!])} [\![e]\!]x'.$$

Unfolding definitions, a small calculation shows that the output distributions must be related by

$$[\![e]\!]x \sim_{\mathsf{DPR}(2\varepsilon,(1+\exp(\varepsilon))\delta)([\![\tau]\!])} [\![e]\!]x'.$$

Hence we have a relation-preserving map

$$[\![e]\!] : (db, adj) \xrightarrow{\text{re}} \mathsf{DPR}(2\varepsilon, (1+\exp(\varepsilon))\delta)([\![\tau]\!])$$

and by Proposition 1, the map $[\![e]\!]$ and our program two_q' satisfy $(2\varepsilon, (1+\exp(\varepsilon))\delta)$-differential privacy.

B. Modeling Other Divergences

The (ε, δ)-differential privacy property belongs to a broader class of probabilistic relational properties: two related inputs lead to two output distributions that are a bounded distance apart, as measured by some divergence on distributions. By varying the divergence, these properties can capture different notions of probabilistic sensitivity. To support a grading, the divergences must be composable in a certain sense.

Definition 11. *Let $H = (\mathbb{R}^{\infty}_{\geq 0}, \leq, u, \bullet)$ be a partially ordered monoid over the non-negative extended reals. A family of divergences d_X on DX indexed by sets X is H-*composable *if for any $f, g : X \to DY$ and $\mu, \nu \in DX$, we have*

$$d_Y(f^\dagger(\mu), g^\dagger(\nu)) \leq d_X(\mu,\nu) \bullet \sup_{x\in X} d_Y(f(x), g(x)).$$

Previously, Barthe and Olmedo [4] proposed *weak* and *strong composability* to study sequential composition properties for the class of f-divergences. (The skew divergence in (ε, δ)-differential privacy is an example of an f-divergence.) These notions coincide when working with full distributions rather than sub-distributions, as in our settings. Given any family of composable divergences, we can build a corresponding *graded* !-lifting of RSRel on $\mathcal{D}$.

Theorem 7. *Let d_X be an H-composable family of divergences on DX and $q : \mathsf{RSRel} \to \mathsf{Set}$ be the forgetful functor. Define a mapping $R(d)$ by*

$$R(d)(\delta)(X) \triangleq (DX, \{(\mu,\nu) \mid d_X(\mu,\nu) \leq \delta, d_X(\nu,\mu) \leq \delta\}).$$

Then $R(d)$ is a monotone mapping of type $(\mathbb{R}^{\infty}_{\geq 0}, \leq) \to \mathbf{Ord}(q, D)$, and is an H-graded $\times$-parameterized assignment of RSRel on $\mathcal{D}$.

As usual, we identify $R(d)$ and the corresponding H-graded $\times$-parameterized !-lifting of $\mathcal{D}$ along $q : \mathsf{RSRel} \to \mathsf{Set}$.

We extend Fuzz with two examples of composable divergences, briefly sketching the composition rules, graded assignment structure, and Fuzz typing rules. We present further examples in Appendix F.

KL Divergence: The *Kullback-Leibler (KL)* divergence, also known as *relative entropy*, measures the difference in information between two distributions. For discrete distributions over X, it is defined as:

$$\mathsf{KL}_X(\mu,\nu) \triangleq \sum_{i\in X} \mu(i) \log \frac{\mu(i)}{\nu(i)},$$

where summation terms with $\mu(i) = \nu(i) = 0$ are defined to be 0, and terms with $\mu(i) > \nu(i) = 0$ are defined to be ∞. This divergence is reflexive, but it is not symmetric and does not satisfy the triangle inequality. (It is not immediately obvious, but the KL divergence is always non-negative.) We can define a family of relations that models pairs of distributions at bounded KL divergence. For any $\alpha \in \mathbb{R}$ and set X, we define:

$$\begin{aligned}\mathsf{KLR}(\alpha)(X) &\triangleq R(\mathsf{KL})(\alpha)(X)\\ &= (DX, \{(\mu,\nu) \mid \mathsf{KL}_X(\mu,\nu) \leq \alpha, \mathsf{KL}_X(\nu,\mu) \leq \alpha\})\end{aligned}$$

Note that α need not be an integer—it can be any real number. This relation is reflexive and symmetric, hence an object in RSRel. Barthe and Olmedo [4, Proposition 5] show that KL is H-composable for $H = (\mathbb{R}, \leq, 0, +)$, so $\mathsf{KLR}(\alpha)(X)$ is a H-graded $\times$-parameterized assignment of RSRel on $\mathcal{D}$ by Theorem 7. By applying the path construction, we get a H-graded $\times$-parameterized assignment of Met on $\mathcal{D}$ which we can use to capture KL divergence as a graded distribution type:

$$[\![\bigcirc^{\mathsf{KL}}_\alpha \tau]\!] \triangleq P(\mathsf{KLR}(\alpha)([\![\tau]\!]))$$

For **bind**, for instance, we obtain the following typing rule:

$$\frac{\Gamma \vdash e_1 : \bigcirc^{\mathsf{KL}}_\alpha \tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc^{\mathsf{KL}}_\beta \sigma}{\Gamma \vdash \mathbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc^{\mathsf{KL}}_{\alpha+\beta} \sigma}$$

Like we did for the max divergence of differential privacy, we can also introduce primitive distributions and typing rules into Fuzz. For instance, a standard fact in probability theory is that the standard Normal distribution satisfies the bound

$$\mathsf{KL}_{\mathbb{R}}(N(\mu_1, 1), N(\mu_2, 1)) \leq (\mu_1 - \mu_2)^2$$

If we again discretize this continuous distribution to $\hat{N}(\mu, 1)$ and interpret the primitive term $[\![\mathbf{Normal}]\!] = \lambda x.\, \hat{N}(x, 1)$, the following typing rule is sound:

$$\frac{}{\Gamma \vdash \mathbf{Normal} : \lceil \mathbb{R} \rceil \multimap \bigcirc^{\mathsf{KL}}_1 \mathbb{R}}$$

χ^2 Divergence: For discrete distributions on X, the χ^2 divergence is defined as

$$\mathsf{XD}_X(\mu,\nu) \triangleq \sum_{i\in X} \frac{(\mu(i) - \nu(i))^2}{\nu(i)},$$

where summation terms with $\mu(i) = \nu(i) = 0$ are defined to be 0, and terms with $\mu(i) > \nu(i) = 0$ are defined to be ∞. Note that this divergence is not symmetric and does not satisfy the triangle inequality. We define a family of reflexive symmetric relations that models pairs of distributions at bounded χ^2-divergence. For any $\alpha \geq 0$ and set X, we pose

$$\mathsf{XDR}(\alpha)(X) \triangleq R(\mathsf{XD})(\alpha)(X) = (DX, \{(\mu, \nu) \mid \mathsf{XD}_X(\mu, \nu) \leq \alpha, \mathsf{XD}_X(\nu, \mu) \leq \alpha\})$$

Olmedo [28, Theorem 5.4] shows that XD is H-composable for the monoid $H = (\mathbb{R}, \leq, 0, +_\chi)$, where $\alpha +_\chi \beta = \alpha + \beta + \alpha\beta$. Hence $\mathsf{XDR}(\alpha)(X)$ is a H-graded $\times$-parameterized assignment of RSRel on $\mathcal{D}$ by Theorem 7. By applying the path construction, we get a H-graded $\times$-parameterized assignment of Met on $\mathcal{D}$ which we can use to interpret a graded distribution type capturing χ^2-divergence:

$$[\![\bigcirc^{\mathsf{XD}}_\alpha \tau]\!] \triangleq P(\mathsf{XDR}(\alpha)([\![\tau]\!]))$$

The corresponding typing rule for **bind** becomes:

$$\frac{\Gamma \vdash e_1 : \bigcirc^{\mathsf{XD}}_\alpha \tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc^{\mathsf{XD}}_\beta \sigma}{\Gamma \vdash \textbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc^{\mathsf{XD}}_{\alpha +_\chi \beta} \sigma}$$

C. Further Extensions

Internalizing Group Privacy: Differential privacy compares the results of a program when run on two input databases at distance 1. These guarantee can sometimes be extended to cover pairs of inputs at distance k, so-called called *group privacy* guarantees. Roughly speaking, an algorithm is said to be $(\varepsilon(k), \delta(k))$-differentially private for groups of size k if for any two inputs at distance k, the output distributions satisfy the divergence bound for $(\varepsilon(k), \delta(k))$-differential privacy.

For standard ε-differential privacy, group privacy is straightforward: an ε-private program is automatically $k \cdot \varepsilon$-private for groups of size k. This clean, linear scaling of the privacy parameters is the fundamental reason why the original Fuzz language fits ε-differential privacy. In fact, group privacy with linear scaling is arguably a more accurate description of the properties captured by Fuzz—it just so happens that this seemingly stronger property coincides with ε-privacy.

In general, however, group privacy guarantees are not so clean. For (ε, δ)-differential privacy, the parameters also degrade when inputs are farther apart, but this degradation is not linear. In a sense, our perspective generalizes (ε, δ)-differential privacy to group privacy, a notion that better matches the linear nature of Fuzz. For instance, the type $!_2\mathsf{db} \multimap \bigcirc_{(\varepsilon,\delta)}\mathbb{R}$ in the last example represents the group privacy guarantee when (ε, δ)-private algorithms are applied to groups of size 2. While we can explicitly compute the corresponding privacy parameters, this unfolded form seems unwieldy to accommodate in Fuzz.

Handling Advanced Composition: The typing rules we have seen so far capture two aspects of (ε, δ)-differential privacy: primitives such as the Gaussian mechanism and sequential composition via the bind rule. In practice, (ε, δ)-privacy is often needed to apply the *advanced composition theorem* [13]. While standard composition simply adds up the (ε, δ) parameters, the advanced version allows trading off the growth of ε with the growth of δ. By picking a δ that is slightly larger than the sum of the individual δ parameters, advanced composition ensures a significantly slower growth in ε.

Unlike the case of standard composition, the growth of the indices in advanced composition is not given by a monoid operation, so it is typically applied to blocks of n programs rather than two programs at a time. We can express this pattern in Fuzz by adding a family of higher-order primitives $(AC_n)_{n\in\mathbb{N}}$, where AC_n applies advanced composition for exactly n iterations. The type of these primitives is

$$!_\infty(!_\infty\tau \multimap \mathsf{db} \multimap \bigcirc_{(\varepsilon,\delta)}\tau) \multimap (!_\infty\tau \multimap \mathsf{db} \multimap \bigcirc_{(\varepsilon^*,\delta^*)}\tau),$$

where ε^*, δ^* are as in the advanced composition theorem:

$$\varepsilon^* = \varepsilon\sqrt{2n\ln(1/\delta')} + n\varepsilon(\exp(\varepsilon) - 1) \qquad \delta^* = n * \delta + \delta'$$

for any $\delta' \in (0, 1)$.

VIII. Handling Non-Termination

Most of our development would readily generalize to the full Fuzz language, which includes general recursive types (and hence also non-terminating expressions). In prior work [3], we modeled the deterministic fragment of Fuzz with metric CPOs—ordered metric spaces that support definitions of non-expansive functions by general recursion. We can extend this work to encompass probabilistic features by endowing the Jones-Plotkin probabilistic powerdomain [17] with metrics, much like was done in Section III. Briefly, the order on the probabilistic powerdomain $\mathcal{E}(X)$ is given by: $\mu \sqsubseteq \nu$ if and only if for any Scott-open set U of the CPO X, $\mu(U) \leq \nu(U)$ holds. The statistical distance and max divergence are all defined continuously and pointwise from the probabilities $\mu(U)$, and they satisfy the compatibility conditions required for metric CPOs. The proofs that these distances form liftings of the probabilistic powerdomain generalize by replacing sums over countable sets with integrals.

While the simple distances pose no major problem, the same cannot be said about the path metric construction. A natural attempt to generalize relations to CPOs is to require *admissibility*: relations should be closed under limits of chains to support recursive function definitions. Unfortunately, the notion of admissibility is not well-behaved with respect to relation composition: the composite of two admissible relations may not be admissible. This is an obstacle when defining the path metric, since a path of length n in the graph induced by the relation is simply a pair of points related by its n-fold composition. Roughly, because admissible relations fail to compose, the path construction does not yield metric CPOs in general, and does not form a morphism of refinements.

The situation can be partially remedied by categorical arguments. Both the category of reflexive, symmetric admissible relations and the category of metric CPOs can be characterized as fibrations over the category of CPOs with suitably complete fibers. This allows us to define an analog of the path construction abstractly as the left adjoint of the Q functor of Section VI, which builds the "at most one" relation. However, this construction does not inherit the pleasant properties of the path metric on sets and functions. More precisely, the proof of Lemma 3 shown in the Appendix, which is instrumental for showing soundness of the bind rule for (ε, δ)-differential privacy, does not carry over.

IX. Related Work

Language-Based Techniques for Differential Privacy: Owing to its clean composition properties, differential privacy has been a fruitful target for formal verification. Our results build upon Fuzz [30], a linear type system for differential privacy that has subsequently been extended with sized types [14] and algorithmic typechecking [2, 9].

Adaptive Fuzz [36] is a recent extension that features an outer layer for constructing and manipulating Fuzz programs—for instance, using program transformations and partial evaluation—before calling the typechecker and running the query. By tracking privacy externally, and not in the type system, Adaptive Fuzz supports many composition principles for (ε, δ)-differential privacy, such as the advanced composition theorem and adaptive variants called privacy filters. Our work expresses (ε, δ)-privacy and basic composition in the type system, rather than using a two-level design.

Until recently, the only type system we were aware of that could capture (ε, δ)-privacy was HOARe^2 [5], a relational type system that has been extended to handle other f-divergences [6]. Compared to our proposal, one drawback is that it only provides guarantees when private algorithms are applied to inputs that are at most a fixed distance apart. In contrast, our sensitivity-based approach can reason about private functions applied to inputs at arbitrary distances.

Duet [27], a more recent design, proposes a two-layer type system for handling (ε, δ)-privacy: one layer tracks sensitivity, analogously to Fuzz, whereas the other layer tracks the ε and δ parameters through composition. The relationship between this system and ours is not yet clear, but we speculate that there might be a connection between its two layers and the path adjunction, the inner sensitivity layer corresponding to Met, and the outer layer corresponding to RSRel.

Recently, variations of differential privacy have been proposed for designing mechanisms with better accuracy. These variations are motivated by properties of continuous distributions and can be characterized through a span lifting [32]. We hope to adapt our approach to reason about these notions of privacy over discrete distributions, as we have for the Laplace mechanism. An interesting problem for future work would be to extend our semantics to continuous distributions, perhaps by leveraging recent advances in probabilistic semantics [35].

Verification of Probabilistic Relational Properties: The last decade has seen significant developments in verification for probabilistic relational properties other than differential privacy. Our work is most closely related to techniques for reasoning about f-divergences [4, 28]. Recent work by Barthe et al. [7] develops a program logic for reasoning about a probabilistic notion of sensitivity based on couplings and the Kantorovich metric. Barthe et al. [7] identified path metrics as a useful concept for formal verification, in connection with the path coupling proof technique. Our work uses path metrics for a different purpose: interpreting relational properties as function sensitivity.

The path adjunction can also be defined as a general construction on enriched categories [20]. Given a monoidal category $\mathcal{V}$ with coproducts, the forgetful functor $\mathcal{V} - \mathsf{Cat} \to \mathcal{V} - \mathsf{Graph}$ has a left adjoint generalizing the construction of the free category on a graph. When $\mathcal{V} = ([0, \infty]^{\geq}, +, 0)$, a $\mathcal{V}$-category is a metric space without the symmetry axiom and a $\mathcal{V}$-graph is a weighted graph. The at-most-one relation of Definition 10 yields a further adjunction between $\mathcal{V}$-graphs and the category of reflexive relations. The composition of these two adjunctions restricted to true symmetric metric spaces and symmetric relations is precisely the path adjunction.

Categorical Semantics for Metrics and Probabilities: Our constructions build on a rich literature in categorical semantics for metric spaces and probability theory. In prior work [3], we modeled the non-probabilistic fragment of the Fuzz language using the concept of a metric CPO; we have adapted this model of the terminating fragment of the language to handle probabilistic sampling. Sato [31] introduced a *graded relational lifting* of the Giry monad for the semantics of relational Hoare logic for the verification of (ε, δ)-differential privacy with continuous distributions. Our graded liftings are similar to his graded liftings, but the precise relationship is not yet clear. Reasoning about metric properties remains an active area of research [23, 29].

X. Conclusion and Future Directions

We have extended the Fuzz programming language to handle probabilistic relational properties beyond ε-differential privacy, including (ε, δ)-differential privacy and other properties based on composable f-divergences. We introduced the categorical notion of parameterized lifting to reason about (ε, δ)-differential privacy in a compositional way. Finally, we cast relational properties as sensitivity properties through a path metric construction.

There are several natural directions for future work. Most concretely, the interaction between the path metric construction and non-termination remains poorly understood. While the differential privacy literature generally does not consider non-terminating computations, extending our results to CPOs would complete the picture. More speculatively, it could be interesting to understand the path construction through calculi that include adjunctions as type constructors [21]. This perspective could help smooth the interface between relational and metric reasoning.

Acknowledgment

Katsumata was supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603) and JSPS KAKENHI (Grant-in-Aid for Scientific Research (C)) Grant Number JP15K00014. This work was also supported by a Facebook TAV award. Marco Gaboardi was partially supported by NSF under grant #1718220.

References

[1] Thorsten Altenkirch, James Chapman, and Tarmo Uustalu. Monads need not be endofunctors. *Logical Methods in Computer Science*, 11(1), 2015.

[2] Arthur Azevedo de Amorim, Marco Gaboardi, Emilio Jesús Gallego Arias, and Justin Hsu. Really natural linear indexed type-checking. In *Symposium on Implementation and Application of Functional Programming Languages (IFL), Boston, Massachusetts*, pages 5:1–5:12. ACM, 2014.

[3] Arthur Azevedo de Amorim, Marco Gaboardi, Justin Hsu, Shin-ya Katsumata, and Ikram Cherigui. A semantic account of metric preservation. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Paris, France*, pages 545–556. ACM, 2017.

[4] Gilles Barthe and Federico Olmedo. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In *International Colloquium on Automata, Languages and Programming (ICALP), Riga, Latvia*, volume 7966 of *Lecture Notes in Computer Science*, pages 49–60. Springer-Verlag, 2013.

[5] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. Higher-order approximate relational refinement types for mechanism design and differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Mumbai, India*, pages 55–68. ACM, 2015.

[6] Gilles Barthe, Gian Pietro Farina, Marco Gaboardi, Emilio Jesús Gallego Arias, Andy Gordon, Justin Hsu, and Pierre-Yves Strub. Differentially private bayesian programming. In *ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria*, pages 68–79. ACM, 2016.

[7] Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. Proving expected sensitivity of probabilistic programs. *Proceedings of the ACM on Programming Languages*, 2(POPL):57:1–57:29, December 2017.

[8] Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *International Workshop on Computer Science Logic (CSL), Kazimierz, Poland*, volume 933 of *Lecture Notes in Computer Science*, pages 121–135. Springer-Verlag, 1994.

[9] Loris D'Antoni, Marco Gaboardi, Emilio Jesús Gallego Arias, Andreas Haeberlen, and Benjamin Pierce. Sensitivity analysis using type-based constraints. In *Workshop on Functional Programming Concepts in Domain-Specific Languages (FPCDSL)*, pages 43–50. ACM, 2013.

[10] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[11] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *IACR International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), Saint Petersburg, Russia*, volume 4004, pages 486–503. Springer-Verlag, 2006.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *IACR Theory of Cryptography Conference (TCC), New York, New York*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer-Verlag, 2006.

[13] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *IEEE Symposium on Foundations of Computer Science (FOCS), Las Vegas, Nevada*, pages 51–60. IEEE, 2010.

[14] Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Rome, Italy*, pages 357–370. ACM, 2013.

[15] Jean-Yves Girard, Andre Scedrov, and Philip Scott. Bounded linear logic. *Theoretical Computer Science*, 97 (1):1–66, 1992.

[16] Michèle Giry. *A categorical approach to probability theory*, volume 915 of *Lecture Notes in Mathematics*, pages 68–85. Springer-Verlag, 1982.

[17] Claire Jones and Gordon D. Plotkin. A probabilistic powerdomain of evaluations. In *IEEE Symposium on Logic in Computer Science (LICS), Asilomar, California*, pages 186–195. IEEE, 1989.

[18] Shin-ya Katsumata. Parametric effect monads and semantics of effect systems. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), San Diego, California*, pages 633–646. ACM, 2014.

[19] Anders Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, 23:113–120, 1970.

[20] Francis William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43(1):135–166, Dec 1973.

[21] Daniel R. Licata and Michael Shulman. Adjoint logic with a 2-category of modes. In *Logical Foundations of Computer Science*, volume 9537 of *Lecture Notes in Computer Science*, pages 219–235. Springer-Verlag, 2016.

[22] Saunders Mac Lane. *Categories for the Working Mathematician (Second Edition)*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.

[23] Radu Mardare, Prakash Panangaden, and Gordon Plotkin. Quantitative algebraic reasoning. In *IEEE Symposium on Logic in Computer Science (LICS), New York, New York*, pages 700–709. IEEE, 2016.

[24] Paul-André Melliès. The parametric continuation monad.

Mathematical Structures in Computer Science, 27(5): 651–680, 2017.

[25] Paul-André Melliès and Noam Zeilberger. Functors are type refinement systems. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Mumbai, India*, pages 3–16. ACM, 2015.

[26] Eugenio Moggi. Computational lambda-calculus and monads. In *IEEE Symposium on Logic in Computer Science (LICS), Asilomar, California*, pages 14–23. IEEE, 1989.

[27] Joseph P. Near, David Darais, Chike Abuah, Tim Stevens, Pranav Gaddamadugu, Lun Wang, Neel Somani, Mu Zhang, Nikhil Sharma, Alex Shan, and Dawn Song. Duet: An Expressive Higher-order Language and Linear Type System for Statically Enforcing Differential Privacy. http://david.darais.com/assets/papers/duet/duet.pdf, 2019. Preprint; accessed April 2019.

[28] Federico Olmedo. *Approximate Relational Reasoning for Probabilistic Programs*. PhD thesis, Universidad Politécnica de Madrid, 2014.

[29] Gordon Plotkin. Towards a metric semantics for probabilistic programming, 2017. Invited talk at Workshop on Probabilistic Programming Semantics (PPS).

[30] Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *ACM SIGPLAN International Conference on Functional Programming (ICFP), Baltimore, Maryland*, pages 157–168. ACM, 2010.

[31] Tetsuya Sato. Approximate relational hoare logic for continuous random samplings. *Electronic Notes in Theoretical Computer Science*, 325:277–298, 2016.

[32] Tetsuya Sato, Gilles Barthe, Marco Gaboradi, Justin Hsu, and Shin-ya Katsumata. Approximate span liftings: Compositional semantics for relaxations of differential privacy. In *IEEE Symposium on Logic in Computer Science (LICS), Vancouver, British Columbia*, 2019. URL https://arxiv.org/abs/1710.09010.

[33] A.L. Smirnov. Graded monads and rings of polynomials. *Journal of Mathematical Sciences*, 151(3):3032–3051, 2008.

[34] Tarmo Uustalu. Strong relative monads (extended abstract). In B. Jacobs, M. Niqui, J. Rutten, and A. Silva, editors, *Short Contributions of 10th Int. Wksh. on Coalgebraic Methods in Comput. Sci., CMCS '10 (Paphos, March 2010)*, pages 23–24. 2010.

[35] Matthijs Vákár, Ohad Kammar, and Sam Staton. A domain theory for statistical probabilistic programming. *PACMPL*, 3:36:1–36:29, 2019.

[36] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth, and Benjamin C. Pierce. A framework for adaptive differential privacy. In *ACM SIGPLAN International Conference on Functional Programming (ICFP), Oxford, England*, pages 10:1–10:29. ACM, 2017.

APPENDIX

A. Parameterized L-Relative Liftings

Relative monads [1] generalize ordinary monads by allowing the unit and Kleisli lifting operations to depend on another functor, akin to the role of the comonad ! in (11) and Definition 2. It is natural to wonder if parameterized liftings are related to relative monads, but the two notions are actually distinct. While parameterized liftings were designed to model a bind rule under a context of variables, the Kleisli lifting of a relative monad can only handle bind in empty contexts, which corresponds to setting Z to $\dot{\mathbf{I}}$ in Definition 2.

In the classical case, we can parameterize bind by combining the Kleisli lifting of a monad with a strength. While a notion of strength also exists for relative monads [34], it was introduced with a different purpose in mind, as an analogue of *arrows* in functional programming languages. We add extra conditions to relative monads so that they are above T, and so that the parameterization is taken into account.

Definition 12. *A* $\dot{\otimes}$-parameterized L-relative lifting *of* $\mathcal{T}$ *along* $p : \mathbb{E} \to \mathbb{B}$ *is a mapping* $\Delta : |\mathbb{B}| \to |\mathbb{E}|$ *such that 1)* $p\Delta X = TX$*, and 2) the parameterized Kleisli lifting* (4) *of* $\mathcal{T}$ *satisfies*

$$f : Z \dot{\otimes} LX \dot{\to} \Delta Y \implies f^{\ddagger} : Z \dot{\otimes} \Delta X \dot{\to} \Delta Y$$

Every $\dot{\otimes}$-parameterized L-relative lifting Δ of $\mathcal{T}$ satisfies 1) $\eta_X : LX \dot{\to} \Delta X$ and 2) $f : LX \dot{\to} \Delta Y$ implies $f^{\dagger} : \Delta X \dot{\to} \Delta Y$. From the faithfulness of p, these two properties imply that Δ is a L-relative monad.

The graded variant of $\dot{\otimes}$-parameterized L-relative lifting is defined as follows:

Definition 13. *An* M-graded $\dot{\otimes}$-parameterized L-relative lifting *is a monotone function* $\Delta : (M, \leq) \to \mathbf{Ord}(p, T)$ *such that the parameterized Kleisli lifting Eq.* (4) *of* $\mathcal{T}$ *satisfies*

$$f : Z \dot{\otimes} LX \dot{\to} \Delta\alpha Y \implies f^{\ddagger} : Z \dot{\otimes} \Delta\beta X \dot{\to} \Delta(\beta \cdot \alpha)Y.$$

This has the indistinguishability relation as an instance.

Proposition 3. *Let* $\mathbb{R}^{+}_{\geq 0}$ *be the additive monoid of nonnegative real numbers. Then in the weakly closed monoidal refinement* (14)*, the indistinguishability relation, regarded as a mapping of type* $\mathsf{DPR} : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to |\mathsf{RSRel}|$*, satisfies*

$$\mathsf{DPR} \in \mathbf{RLift}_{\times}(\mathcal{D}, \mathbb{R}^{+}_{\geq 0} \times \mathbb{R}^{+}_{\geq 0}).$$

B. Proof of Theorem 1 and Theorem 3

We show the non-graded version (Theorem 1); the graded version (Theorem 3) is similar.

Let $\Delta \in \mathbf{RLift}_{\dot{\otimes}}(\mathcal{T})$. We define $\dot{T}_{\Delta} X = \Delta p X$. We show that it is a $\dot{\otimes}$-parameterized !-lifting, that is,

$$f : X \dot{\otimes} LpY \dot{\to} \Delta pZ \implies f^{\ddagger} : X \dot{\otimes} \Delta pY \dot{\to} \Delta pZ.$$

This is true by the assumption.

Conversely, let $\dot{T} \in !\mathbf{Lift}_{\dot{\otimes}}(\mathcal{T})$. We define $\Delta_{\dot{T}} I = \dot{T} L I$. We show that it is a $\dot{\otimes}$-parameterized L-relative lifting, that is,

$$f : X \dot{\otimes} LI \dot{\to} \dot{T}LJ \implies f^{\ddagger} : X \dot{\otimes} \dot{T}LI \dot{\to} \dot{T}LJ.$$

This is also true from $!LI = LI$ and the assumption.

We show that the above processes are equivalence of preorders. We first have

$$\Delta_{\dot{T}_{\Delta}} I = \dot{T}_{\Delta} LI = \Delta pLI = \Delta I.$$

Next, we have $\dot{T}_{\Delta_{\dot{T}}} X = \Delta_{\dot{T}} pX = \dot{T}LpX \leq \dot{T}X$. We show the converse $\dot{T}X \leq \dot{T}LpX$. Since $\dot{T}$ is a !-lifting, we have $\eta_{pLpX} : !LpX \dot{\to} \dot{T}LpX$. From $pL = \mathrm{Id}$, we have $\eta_{pX} : !X \dot{\to} \dot{T}LpX$. As $\dot{T}$ is a $\dot{\otimes}$-parameterized !-lifting, we obtain ${\eta_{pX}}^{\dagger} = \mathrm{id}_{TpX} : \dot{T}X \dot{\to} \dot{T}LpX$, that is, $\dot{T}X \leq \dot{T}LpX$. Therefore $\dot{T} \simeq \dot{T}_{\Delta_{\dot{T}}}$ holds in the preorder $!\mathbf{Lift}_{\dot{\otimes}}(\mathcal{T})$.

This finishes the equivalence of $!\mathbf{Lift}_{\dot{\otimes}}(\mathcal{T})$ and $\mathbf{RLift}_{\dot{\otimes}}(\mathcal{T})$.

Next, suppose that $\Delta \in \mathbf{Asign}_{\dot{\otimes}}(\mathcal{T})$. We infer:

$$\dfrac{\dfrac{\dfrac{f : X \dot{\otimes} FI \dot{\to} \Delta J}{\lambda(f) : X \dot{\to} I \pitchfork \Delta J}}{\lambda(f) \dot{\otimes} \mathrm{id}_{TJ} : X \dot{\otimes} \Delta I \dot{\to} (I \pitchfork \Delta J) \dot{\otimes} \Delta I}}{ev^{\ddagger} \circ (\lambda(f) \dot{\otimes} \mathrm{id}_{TJ}) : X \dot{\otimes} \Delta I \dot{\to} \Delta J}$$

At the last step, we use the $\dot\otimes$-strong assignment. Now

$$\begin{aligned} ev^\ddagger \circ (\lambda(f) \,\dot\otimes\, \mathrm{id}_{TJ}) &= \mu \circ T(ev) \circ \theta \circ (\lambda(f) \,\dot\otimes\, T\mathrm{id}_J) \\ &= \mu \circ T(ev \circ \lambda(f) \,\dot\otimes\, \mathrm{id}_J) \circ \theta \\ &= \mu \circ T(f) \circ \theta \\ &= f^\ddagger. \end{aligned}$$

Therefore $\Delta \in \mathbf{RLift}_{\dot\otimes}(\mathcal{T})$.

Conversely, suppose that $\Delta \in \mathbf{RLift}_{\dot\otimes}(\mathcal{T})$. Since p is the map of adjunction from $- \,\dot\otimes\, LI \dashv I \,\dot\pitchfork\, -$ to $- \,\dot\otimes\, I \dashv I \,\dot{\multimap}\, -$, we have

$$ev : (I \,\dot\pitchfork\, \Delta J) \,\dot\otimes\, LI \,\dot\to\, \Delta J.$$

Therefore we obtain

$$ev^\ddagger : (I \,\dot\pitchfork\, \Delta J) \,\dot\otimes\, \Delta I \,\dot\to\, \Delta J,$$

that is, $\Delta \in \mathbf{Asign}_{\dot\otimes}(\mathcal{T})$. We conclude that $\mathbf{RLift}_{\dot\otimes}(\mathcal{T}) = \mathbf{Asign}_{\dot\otimes}(\mathcal{T})$.

C. Proof of Proposition 3

Before the proof, we introduce an auxiliary concept abstracting the composition properties of divergences studied in differential privacy as Theorem 2. This concept is valid when the symmetric monoidal structure $(\dot{\mathbf{I}}, \dot\otimes)$ assumed on $\mathbb{E}$ in (9) is cartesian.

Definition 14. *Assume that in* (9) *the symmetric monoidal structure on* $\mathbb{E}$ *is cartesian (hence we denote it by* $(\dot 1, \dot\times)$*). An* M-graded sequentially composable family of $\mathbb{E}$-objects *above* $\mathcal{T}$ *is a monotone function* $\Delta : (M, \le) \to \mathbf{Ord}(p, T)$ *such that for any* $f : Z \,\dot\to\, \Delta\alpha X$ *and* $g : Z \,\dot\times\, LX \,\dot\to\, \Delta\beta Y$*, we have*

$$g^\ddagger \circ \langle \mathrm{id}_{pZ}, f\rangle : Z \,\dot\to\, \Delta(\alpha\cdot\beta)Y.$$

the subpreorder of $[(M, \le), \mathbf{Ord}(p, T)]$ *consisting of* M*-graded sequentially composable families of* $\mathbb{E}$*-objects above* $\mathcal{T}$ *is denoted by* $\mathbf{Comp}(\mathcal{T}, M)$*.*

Let us see how this definition expands in the weakly closed monoidal refinement (14), and a monad $\mathcal{T}$ on Set:

$$(\mathsf{RSRel}, \dot 1, \dot\times) \underset{M}{\overset{q}{\rightleftarrows}} \mathsf{Set} \circlearrowleft \mathcal{T} \quad (q \dashv M)$$

An M-graded sequentially composabile family of RSRel-objects assigns to each set TX and $\alpha, \beta \in M$, a reflexive, symmetric relation $\sim^m_X$ satisfying:

$$\begin{aligned} &(\forall z \sim z' \;.\; f(z) \sim^\alpha_X f(z')) \\ &\wedge (\forall z \sim z', x \;.\; g(z, x) \sim^\beta_Y g(z', x)) \\ &\implies \forall z \sim z' \;.\; g^\ddagger(z, f(z)) \sim^{\alpha\cdot\beta}_Y g^\ddagger(z', f(z')) \end{aligned}$$

Theorem 8. $\mathbf{Asign}_{\dot\times}(\mathcal{T}, M) = \mathbf{Comp}(\mathcal{T}, M)$.

Proof. Let $\Delta \in \mathbf{Comp}(\mathcal{T}, M)$. That is, the following implication holds:

$$\begin{aligned} &f : Z \,\dot\to\, \Delta\alpha X \wedge g : Z \,\dot\times\, LX \,\dot\to\, \Delta n Y \\ &\implies g^\ddagger \circ \langle \mathrm{id}_{pZ}, f\rangle : Z \,\dot\to\, \Delta(m\cdot n)Y \end{aligned}$$

We instantiate the premise of the sequential composition condition with the following data (X, Y are left unchanged)

$$\begin{aligned} Z &= (X \,\dot\pitchfork\, \Delta\beta Y) \,\dot\times\, \Delta\alpha X \\ f &= \pi_1 : Z \,\dot\to\, \Delta\alpha X \\ g &= \lambda(ev \circ \langle \pi_1 \circ \pi_1, \pi_2\rangle) : Z \,\dot\times\, LX \,\dot\to\, \Delta\beta Y. \end{aligned}$$

We then obtain

$$g^\ddagger \circ \langle \mathrm{id}, f\rangle : (X \,\dot\pitchfork\, \Delta\beta Y) \,\dot\times\, \Delta\alpha X \,\dot\to\, \Delta(\alpha\cdot\beta)Y,$$

and we have $\mathrm{kl} = g^\ddagger \circ \langle \mathrm{id}, f\rangle$. Therefore $\Delta \in \mathbf{Asign}_{\dot\times}(\mathcal{T}, M)$.

Conversely, suppose that $\Delta \in \mathbf{Asign}_{\dot\times}(\mathcal{T}, M)$. We have the following construction of morphisms in $\mathbb{E}$:

$$\dfrac{\dfrac{\dfrac{g : Z \mathbin{\dot\times} LX \dot\to \Delta\beta Y}{\lambda(g) : Z \dot\to X \mathbin{\dot\pitchfork} \Delta\beta Y} \qquad f : Z \dot\to \Delta\alpha X}{\langle \lambda(g), f\rangle : Z \dot\to (X \mathbin{\dot\pitchfork} \Delta\beta Y) \mathbin{\dot\times} \Delta\alpha X}}{g^\ddagger \circ \langle \mathrm{id}, f\rangle = \mathrm{kl} \circ \langle \lambda(g), f\rangle : Z \dot\to \Delta(\alpha \cdot \beta) Y}$$

Therefore $\Delta \in \mathbf{Comp}(\mathcal{T}, M)$. □

As a result, the sequential composability (Theorem 2) of differential privacy is equivalent to that the indistinguishability relation is a graded sequentially composable family:

$$\mathsf{DPR} \in \mathbf{Comp}(\mathcal{D}, \mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0}) = \mathbf{Asign}_\times(\mathcal{D}, \mathbb{R}^+_{\geq 0} \times \mathbb{R}^+_{\geq 0})$$

where $\mathbb{R}^+_{\geq 0}$ is the additive monoid of nonnegative real numbers.

D. Properties of the Path Adjunction

As a basic sanity check, converting the path metric of a relation back into a relation with the at-most-one operation yields the original relation.

Proposition 4. *For any $X \in \mathsf{RSRel}$, $QPX = X$.*

Composing in the opposite order PQX does not yield X in general. Consider, for instance, a metric space $X = (\{x, y\}, d)$ where $d(x, y) = 2$. However, P and Q do form an adjoint pair.

Proposition 5. *For any $X \in \mathsf{RSRel}$ and $Y \in \mathsf{Met}$, we have $\mathsf{Met}(PX, Y) = \mathsf{RSRel}(X, QY)$. In particular, the functors P and Q form an adjoint pair $P \dashv Q$, whose unit is identity morphism.*

Proof. Let us show that the first term is contained in the second. Suppose we have a morphism $f : PX \to Y$. Take x and x' in X such that $x \sim x'$. We have $d(x, x') \leq 1$ in PX, thus $d(f(x), f(x')) \leq 1$ by non-expansiveness—that is, $f(x) \sim f(x')$ in QY by definition. This shows that f is also a morphism in $X \to QY$.

To conclude, we just need to show the reverse inclusion. Suppose we have a morphism $f : X \to QY$. Showing that f is a non-expansive function $PX \to Y$ is equivalent to showing that, given a path $x_0 \sim \cdots \sim x_k$ of length k in X, $d(x_0, x_k) \leq k$. Since $f : X \to QY$, we know that for every $i < k$ the relation $f(x_i) \sim_{QY} f(x_{i+1})$ holds; by definition, this means that $d(f(x_i), f(x_{i+1})) \leq 1$. Applying the triangle inequality $k - 1$ times, we conclude $d(f(x_0), f(x_k)) \leq k$. □

Since the unit is identity, we conclude that P is *full and faithful*. In the current setting, this fact can be phrased as follows:

Corollary 1. *Let X and Y be objects of RSRel. Then $f : X \overset{\mathrm{re}}{\to} Y$ if and only if $f : PX \overset{\mathrm{ne}}{\to} PY$.*

The adjunction $P \dashv Q$ preserves much—but not all—of the structure in RSRel and Met. Combined with the previous results, these properties yield a proof of Theorem 6. We summarize these properties below. First, discrete spaces are preserved.

Lemma 2. *For any set X, $P(\infty \cdot X) = \infty \cdot X \in \mathsf{Met}$, and $Q(\infty \cdot X) = \infty \cdot X \in \mathsf{RSRel}$.*

The path functor $P : \mathsf{RSRel} \to \mathsf{Met}$ strictly preserves all products (including infinite ones), and symmetric monoidal structure.

Lemma 3. *Let $(X_i)_{i \in I}$ be a family of objects of RSRel. Then $P(\prod_i X_i) = \prod_i PX_i$. The metric on a general product of metric spaces is defined by taking the supremum of all the metrics.*

Proof. As the underlying sets are equal, it suffices to show that the metrics are equal. Let d_P be the metric associated with $P(\prod_i X_i)$, and let $d_\prod$ be the metric associated with $\prod_i PX_i$. Since P is a functor, we know that the identity function is a morphism in $P(\prod_i X_i) \to \prod_i PX_i$; that is, $d_\prod(x, x') \leq d_P(x, x')$ for all families $x, x' \in \prod_i X_i$. To conclude, we just need to show the opposite inequality. If $d_\prod(x, x') = \infty$, we are done; otherwise, there exists some $k \in \mathbb{N}$ such that $d_{PX_i}(x_i, x'_i) \leq k$ for every $i \in I$. Since all the relations associated with the X_i are reflexive, we know that for every i there must exist a path of related elements $x_i = x_{i,0} \sim \cdots \sim x_{i,k} = x'_i$ of length exactly k, by padding one of the ends with reflexivity edges. Therefore, there exists a path of related families $(x_i)_{i \in I} = (x_{i,0})_{i \in I} \sim \cdots \sim (x_{i,k})_{i \in I} = (x'_i)_{i \in I}$ of length k in $\prod_i X_i$. By the definition of the path metric, this means that $d_P(x, x') \leq k$, and thus $d_P(x, x') \leq d_\prod(x, x')$, as we wanted to show. □

Lemma 4. *For every $X, Y \in \mathsf{RSRel}$, $P(X \otimes Y) = PX \otimes PY$.*

Proof. Once again, it suffices to show that both metrics are equal.

For all pairs (x, y) and (x', y') in $X \times Y$, there are paths between x and x' and between y and y' whose summed length is at most k if and only if there is a path between (x, y) and (x', y') of length at most k in $X \otimes Y$. For the "only if" direction, if there are such paths $(x_i)_{i \le k_x}$ and $(y_i)_{i \le k_y}$, then the sequence

$$(x_0, y_0), (x_1, y_0), \ldots, (x_{k_x}, y_0), (x_{k_x}, y_1), \ldots, (x_{k_x}, y_{k_y})$$

is a path from (x, y) and (x', y') in $X \otimes Y$. Conversely, suppose that we have a path from (x, y) to (x', y') of length at most k in $X \otimes Y$. We can show by induction on the length of the path that there are paths from x to x' and from y and y' with total length at most k. The base case is trivial. If there is a hop, by the definition of the relation for $X \otimes Y$, this hop only adds one unit to the length of the path on X or to the path on Y, which allows us to conclude. □

E. Proof of Theorem 4

Let $\Delta : (M, \le) \to \mathbf{Ord}(p, T)$ be an M-graded $\dot{\otimes}$-parameterized assignment of $\mathbb{E}$ on $\mathcal{T}$. Since $p \circ F = p'$, $F \circ \Delta$ is a monotone function of type $(M, \le) \to \mathbf{Ord}(p', T)$. By applying F to the internal Kleisli lifting morphism Eq. (5), we obtain

$$\frac{kl : (X \pitchfork \Delta m Y) \dot{\otimes} \Delta n X \dot{\to} \Delta(n \cdot m) Y}{kl : (X \ddot{\pitchfork} F(\Delta m Y)) \ddot{\otimes} F(\Delta n X) \dot{\to} F(\Delta(n \cdot m) Y)}$$

This concludes that $F \circ \Delta : (M, \le) \to \mathbf{Ord}(p', T)$ is a $\ddot{\otimes}$-parameterized assignment.

F. Modeling Other Divergences in Graded Fuzz

Hellinger Distance: The *Hellinger* distance is a standard measure of similarity between distributions originating from statistics. For discrete distributions over X, the Hellinger distance is defined as:

$$\mathsf{HD}_X(\mu, \nu) \triangleq \sqrt{\frac{1}{2} \sum_{i \in X} |\sqrt{\mu(i)} - \sqrt{\nu(i)}|^2}.$$

This distance is a proper metric: it satisfies reflexivity, symmetry, and triangle inequality. However, its behavior under composition means that we cannot model it with a standard assignment structure. Instead, we will use a graded assignment. For any $\alpha \ge 0$ and set X, we define:

$$\begin{aligned} \mathsf{HDR}(\alpha)(X) &\triangleq R(\mathsf{HD})(\alpha)(X) \\ &= (DX, \{(\mu, \nu) \mid \mathsf{HD}_X(\mu, \nu) \le \alpha, \mathsf{HD}_X(\nu, \mu) \le \alpha\}) \end{aligned}$$

Note that this is a reflexive and symmetric relation, hence an object in RSRel. Barthe and Olmedo [4, Proposition 5] show the following composition principle:

$$\mathsf{HD}_Y(f^\dagger \mu, g^\dagger \nu)^2 \le \mathsf{HD}_X(\mu, \nu)^2 + \sup_{x \in X} \mathsf{HD}_Y(f(x), g(x))^2$$

Note that this is for *squared* Hellinger distance. Taking square roots, we have the following composition property for standard Hellinger distance:

$$\mathsf{HD}_Y(f^\dagger \mu, g^\dagger \nu) \le \sqrt{\mathsf{HD}_X(\mu, \nu)^2 + \sup_{x \in X} \mathsf{HD}_Y(f(x), g(x))^2}$$

Hence, HD is H-composable for the monoid $H = (\mathbb{R}, \le, 0, +_2)$, where $\alpha +_2 \beta = \sqrt{\alpha^2 + \beta^2}$ and $\mathsf{HDR}(\alpha)(X)$ has the structure of a H-graded $\times$-parameterized assignment of RSRel on $\mathcal{D}$ by Theorem 7. By applying the path construction to this assignment, we get a H-graded $\times$-parameterized assignment of Met on $\mathcal{D}$ with which we can then interpret a graded distribution type capturing Hellinger distance:

$$[\![\bigcirc^{\mathsf{HD}}_\alpha \tau]\!] \triangleq P(\mathsf{HDR}(\alpha)(|[\![\tau]\!]|))$$

The typing rule for **bind** is then:

$$\frac{\Gamma \vdash e_1 : \bigcirc^{\mathsf{HD}}_\alpha \tau \qquad \Gamma, x :_\infty \tau \vdash e_2 : \bigcirc^{\mathsf{HD}}_\beta \sigma}{\Gamma \vdash \mathbf{bind}\, x \leftarrow e_1; e_2 : \bigcirc^{\mathsf{HD}}_{\alpha +_2 \beta} \sigma}$$

Like the other divergences, we can introduce primitives and typing rules capturing the Hellinger distance. For example, the natural-valued *Poisson* distribution models the number of events occurring in some time interval, if the events happen independently and at constant rate. Given a parameter $\alpha : \mathbb{N}$, this distribution has the following probability mass function:

$$P(\alpha)(n) = \frac{\alpha^n e^{-\alpha}}{n!}$$

It is known that the Poisson distribution satisfies the following Hellinger distance bound:

$$\mathsf{HD}_{\mathbb{N}}(P(\alpha), P(\alpha')) = 1 - \exp\left(-\frac{|\sqrt{\alpha} - \sqrt{\alpha'}|^2}{2}\right) \leq \frac{1}{2}|\sqrt{\alpha} - \sqrt{\alpha'}|^2$$

Given α and α' at most 1 apart, this Hellinger distance is at most $1/2$. Hence, we may introduce a type **Poisson** and interpret it as $[\![\textbf{Poisson}]\!] = \lambda x.\, P(x)$, and the following typing rule is sound:

$$\frac{}{\Gamma \vdash \textbf{Poisson} : \lceil \mathbb{R} \rceil \multimap \bigcirc^{\mathsf{HD}}_{1/2} \mathbb{N}}$$