Compositional imprecise probability

Jack Liell-Cock and Sam Staton
University of Oxford

Draft, April 2024

Abstract

Imprecise probability is concerned with uncertainty about which probability distributions to use. It has applications in robust statistics and elsewhere. Imprecise probability can be modelled in various ways, including by convex sets of probability distributions.

We look at programming language models for imprecise probability. Our desiderata are that we would like our model to support all kinds of composition, categorical and monoidal, in other words, guided by dataflow diagrams. Another equivalent perspective is that we would like a model of synthetic probability in the sense of Markov categories.

There is already a fairly popular monad-based approach to imprecise probability, but it is not fully compositional because the monad involved is not commutative, which means that we do not have a proper monoidal structure. In this work, we provide a new fully compositional account. The key idea is to name the non-deterministic choices. To manage the renamings and disjointness of names, we use graded monads. We show that the resulting compositional model is maximal. We relate with the earlier monad approach, showing that we obtain tighter bounds on the uncertainty.

1 Overview

This paper is about using programming language notations to give compositional descriptions of imprecise probability. For illustration, consider a situation with three outcomes: red (r), green (g) and blue (b). A precise probability distribution can be understood as a point in the triangle: the corner (r) represents 100% certainty of red; the points on the edge between g and b represent the probability distributions where r is impossible (Figure 1a).

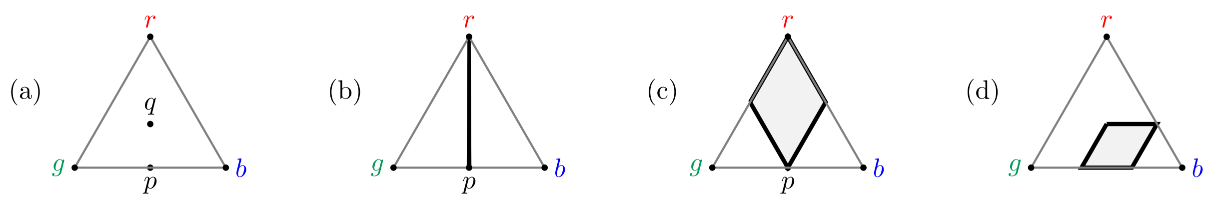


Figure 1: (a) Five probabilities over the three-point set $\{r, g, b\}$ illustrated as points in the triangle: the three extreme points are the corners; p is the equal odds chance between b and g; q is the equal odds chance between all three points. (b) A line indicating a convex region between r and p, which includes q. (c) A convex region which is the convex hull of four points, including r, p and also the equal odds chance between r and b and between r and g. (d) A different convex region, considered in [77, Ex. 7.3].

An *imprecise probability* on three outcomes is a convex region of the triangle (Figure 1b–1d). One interpretation is that if a probability distribution describes a bet, as in the foundations of Bayesianism, then a convex region is a collection of bets that would be reasonable given the current imprecise knowledge. Imprecise probability has a long history in terms of statistical robustness (e.g. [29,77]), recently considered as part of infrabayesianism (e.g. [3,4,44]) and the foundations of safe AI [16].

There is already a body of work on semantics models of programming languages with imprecise probability [2, 24, 25, 26, 30, 38, 39, 55, 56, 57, 58, 59, 63, 74, 75]. Our contribution is to investigate new models that support our compositional desiderata (§1.1) by naming the non-deterministic choices (§1.2). We show that this compares favourably with earlier work (Thm. 1, §1.3) that it is a maximal approach (Thm. 2).

1.1 Desiderata: a language for imprecise probability with compositional reasoning

A first language for describing imprecise probability is a first-order functional language without recursion. Rather, we have if/then/else statements, sequencing with immutable variable assignment (like [50, 60]), and the following two commands, which both return a boolean value:

- **bernoulli**: a fair Bernoulli choice [6] which draws a ball from some urn containing two balls labelled 'true' and 'false', and replaces it;
- **knight**: a Knightian choice [41] which draws a ball from a fresh urn containing balls labelled 'true' and 'false', where the number and ratio of balls are unknown and we have no priors on their distribution.

For example, consider the following two programs.

Example 1.1. Consider the following program, which we argue describes the convex region in Figure 1b:

$x \leftarrow$ **knight** ; $z \leftarrow$ **bernoulli** ;
if z then (if x then return r else return g)
else (if x then return r else return b)

We draw two boolean values, x and z, respectively with Knightian uncertainty and from a fair Bernoulli trial. We then combine these two boolean values using the logic on the second and third lines of the program.

Example 1.2. The following program describes the convex region in Figure 1c:

$x \leftarrow$ **knight** ; $y \leftarrow$ **knight** ; $z \leftarrow$ **bernoulli** ;
if z then (if x then return r else return g)
else (if y then return r else return b)

This time, we draw three boolean values, x, y and z, where y is with Knightian uncertainty too. We then combine these three boolean values using the logic on the second and third lines of the program, which is almost the same except for the use of y when z is false. Decoupling the Knightian uncertainties increases the region of imprecise probability because it allows new outcomes (such as an equal chance between r and b when x is true and y is false) that were impossible in Example 1.1.

Our desiderata for a *compositional account* of a first-order language are the following. We are inspired by recent compositional accounts of probability theory (e.g. [19, 31, 43]), statistics (e.g. [10, 33, 47]), and probabilistic programming (e.g. [15, 34, 68]), and the connections between them (e.g. [70]). These desiderata are formalized in Section 2.

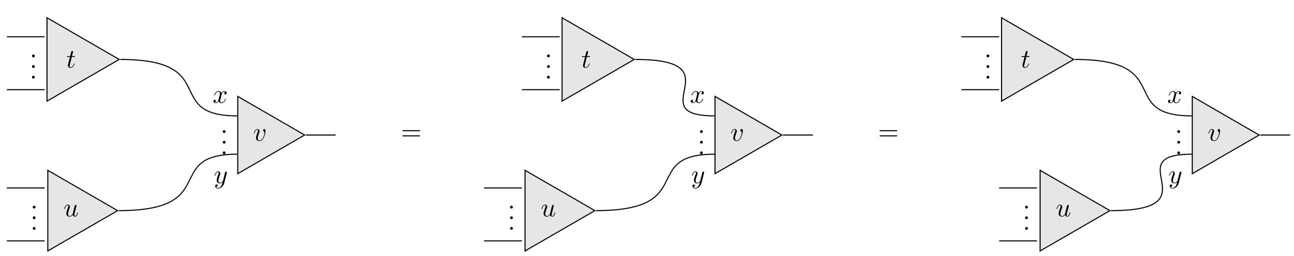
Desideratum 1. *The language should be commutative:*

$$x \leftarrow t\ ;\ y \leftarrow u\ ;\ v \quad = \quad y \leftarrow u\ ;\ x \leftarrow t\ ;\ v \qquad \textit{(if } x \textit{ is not free in } u \textit{ and } y \textit{ not free in } t\textit{)}$$

and affine:

$$(x \leftarrow t\ ;\ u) \quad = \quad u \qquad \textit{(if } x \textit{ is not free in } u\textit{).}$$

This means that we can regard composition graphically, as a data flow graph. For instance, the notation

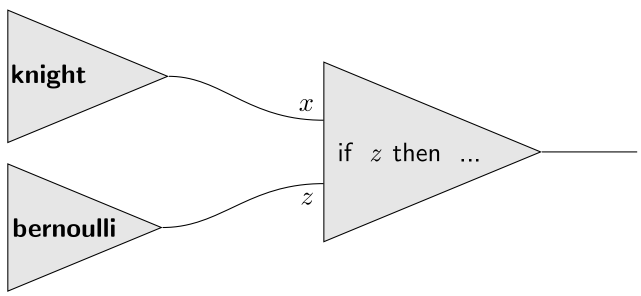


is not ambiguous.

```
   x ← knight ; z ← bernoulli ;
   if z then ( if x then return r else return g ) else ( if x then return r else return b )
= −− Desideratum 1 (commutativity)
   z ← bernoulli ; x ← knight ;
   if z then ( if x then return r else return g ) else ( if x then return r else return b )
= −− Desideratum 2
   z ← bernoulli ; if z then (x ← knight ; if x then return r else return g )
                        else (x ← knight ; if x then return r else return b )
= −− Alpha renaming
   z ← bernoulli ; if z then (x ← knight ; if x then return r else return g )
                        else (y ← knight ; if y then return r else return b )
= −− Desideratum 1 (affine)
   z ← bernoulli ; if z then (x ← knight ; y ← knight ; if x then return r else return g )
                        else (x ← knight ; y ← knight ; if y then return r else return b )
= −− Desideratum 2
   z ← bernoulli ; x ← knight ; y ← knight ;
   if z then ( if x then return r else return g ) else ( if y then return r else return b )
= −− Desideratum 1 (commutativity)
   x ← knight ; y ← knight ; z ← bernoulli ;
   if z then ( if x then return r else return g ) else ( if y then return r else return b )
```

Figure 2: An equational derivation that Examples 1.1 and 1.2 must be equal if Desiderata 1 and 2 are satisfied.

Although this requirement does not hold generally in the presence of memory side effects and mutable variables, we do not have mutable variables here, and it is desirable in a declarative language. For example, we would like to notate the program from Example 1.1 as



Desideratum 2. *The standard equational reasoning about if/then/else should apply, and in particular the following hoisting equation should be allowed:*

$$\mathsf{if}\ b\ \mathsf{then}\ (x \leftarrow t\ ;\ u)\ \mathsf{else}\ (x \leftarrow t\ ;\ v) \quad = \quad x \leftarrow t\ ;\ \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v$$

where x is not free in b.

One earlier approach to a semantic study of a language like this is provided by a convex powerset of distributions monad (e.g. [7, 8, 26, 30, 39, 57, 58]). This does not satisfy the desiderata for compositional reasoning. In fact, no semantic model satisfying the desiderata can allow Examples 1.1 and 1.2 to be distinguished, as we show in Figure 2. The key issue is with the third program in Figure 2:

```
z ← bernoulli ; if z then (x ← knight ; if x then return r else return g )
                     else (x ← knight ; if x then return r else return b )
```

This program draws a boolean value with Knightian uncertainty on each of the branches of the if statement. The paradox arises in whether each choice comes from different urns or the same urn. Perhaps there is one Knightian urn that is used in both branches. Or perhaps we draw a boolean value from a new Knightian urn on the second branch. Our proposed solution is to make this distinction explicit.

1.2 Resolution: named Knightian choices

To satisfy both desiderata, **our proposal** is to name each Knightian choice (Section 3). To do this, we rewrite Example 1.1 by annotating the only Knightian choice with the name a_1:

$x \leftarrow \textbf{knight}(a_1)\ ;\ z \leftarrow \textbf{bernoulli}\ ;$
$\texttt{if}\ z\ \texttt{then}\ (\texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ g)\ \texttt{else}\ (\texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ b)$

We think of this program as giving rise to the convex set in Figure 1(b). This is then the same as the program where y describes the outcome of the same Knightian choice, i.e. one with the same name:

$x \leftarrow \textbf{knight}(a_1)\ ;\ y \leftarrow \textbf{knight}(a_1)\ ;\ z \leftarrow \textbf{bernoulli}\ ;$
$\texttt{if}\ z\ \texttt{then}\ (\texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ g)\ \texttt{else}\ (\texttt{if}\ y\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ b)$

but it is different from the program where y describes a different Knightian choice, i.e. one with a different name:

$x \leftarrow \textbf{knight}(a_1)\ ;\ y \leftarrow \textbf{knight}(a_2)\ ;\ z \leftarrow \textbf{bernoulli}\ ;$
$\texttt{if}\ z\ \texttt{then}\ (\texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ g)\ \texttt{else}\ (\texttt{if}\ y\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ b)$

which is intuitively what Example 1.2 describes, and gives rise to the convex set in Figure 1(c). Now when we try to follow the same equational derivation as in Figure 2, the third program becomes:

$z \leftarrow \textbf{bernoulli}\ ;\ \texttt{if}\ z\ \texttt{then}\ (x \leftarrow \textbf{knight}(a_1)\ ;\ \texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ g)$
$\texttt{else}\ (x \leftarrow \textbf{knight}(a_1)\ ;\ \texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ b)$

which conveys the same Knightian value is used on each of the branches of the if statement. This can no longer derive the program:

$z \leftarrow \textbf{bernoulli}\ ;\ \texttt{if}\ z\ \texttt{then}\ (x \leftarrow \textbf{knight}(a_1)\ ;\ \texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ g)$
$\texttt{else}\ (x \leftarrow \textbf{knight}(a_2)\ ;\ \texttt{if}\ x\ \texttt{then}\ \texttt{return}\ r\ \texttt{else}\ \texttt{return}\ b)$

which explicitly uses a different Knightian choice on the else branch.

The idea of naming non-deterministic choices appears in work outside probability (e.g. proved transitions, [9]) and probabilistic choices are often named in practical probabilistic programming [73, §6.2] which has already been explored using graded monads [51]. More generally, intensionality in non-determinism is known to be a profitable perspective (e.g. [12, 46]).

1.2.1 Named Knightian choices via a reader monad

The set up with named Knightian choices is consistent with Desiderata 1 and 2, which we can show by building a monad (e.g. [60]), namely the reader transformer (e.g. [52]) of the finite distributions monad (e.g. [32, Ch. 2]):

$$T_{2^A}(X) = [2^A \Rightarrow D(X)] \tag{1}$$

where X is the set of outcomes, A is the set of names required, and D is the finite distributions monad. Then the Knightian choices are interpreted by reading, and the Bernoulli choices use the distributions monad. This combined monad is well known to be commutative and affine. Thus both desiderata are satisfied.

We can recover a convex set of probability distributions from any $t \in T_{2^A}(X)$ by pushing forward all the possible probability distributions on 2^A. Formally, we can express this using the monadic bind ($\gg\!=_D$, Kleisli composition) of D:

$$[\![t]\!]_{2^A} = \{p \gg\!=_D t \mid p \in D(2^A)\} \subseteq D(X).$$

1.2.2 Grading to account for renamings

A remaining concern with named Knightian choices is that we ought to take seriously name-space issues in composition. When composing programs with named Knightian choices, we may wish to avoid name clashes. This is dependent on how we interpret the set A in (1).

We resolve this issue by regarding the monad (1) as a graded monad [36, 37, 61]. This is closely related to the 'para' construction (e.g. [18, 27]). Some of the crucial steps are as follows:

- Any injection $\iota : A \to B$ induces a renaming of programs using names A to programs using names B, and indeed a natural map $T_{2^\iota} : T_{2^A}(X) \to T_{2^B}(X)$;
- We can regard monadic bind (Kleisli composition) in T as operating on distinct sets of names:

  $$\gg\!=_T\ :\ T_{2^A}(X) \times (X \Rightarrow T_{2^B}(Y)) \to T_{2^{A \uplus B}}(Y)$$

  Thus a computation using names A is sequenced with a computation using names B to build a computation that involves names $(A \uplus B)$.

- This monad is graded-monoidal too, via a map

$$T_{2^A}(X) \times T_{2^B}(Y) \to T_{2^{(A \uplus B)}}(X \times Y)$$

which juxtaposes computations using names A and B to give a computation using $(A \uplus B)$.

- The induced convex set of distributions is invariant under renaming: $[\![t]\!]_{2^A} = [\![T_{2^\iota}(t)]\!]_{2^B}$;

The crucial element is that the injective renaming ι induces a surjection $2^\iota : 2^B \to 2^A$ between the spaces of Knightian choices. We abstract and generalize by allowing arbitrary surjections $2^B \to 2^A$, further by allowing sets other than 2^A, and further still by allowing surjective stochastic maps rather than surjections.

As an aside, we note that probability monads too can often be regarded as sort-of reader monads (e.g. [5, 15, 66, 67, 71, 72]), since probability distributions $D(X)$ can be described by random variables $\Omega \to X$, for some base probability sample space Ω. Thus we could regard our monad $T(X)$ as a quotient of

$$[(\Omega \times \Xi) \Rightarrow X]$$

where Ω is a sample space for Bernoulli probability and $\Xi = 2^A$ is a sample space for Knightian uncertainty. In this work, we will quotient by the 'law' of random variables in Ω, so that the usual equational reasoning about Bernoulli probability is valid.

1.3 Results about quotienting our theory

The names for the Knightian choices in our language appear to be additional intensional information, and the reader monad does not quotient this away. For this reason we show two results about the equational theory. First, we connect our approach to the convex powerset of distributions monad, showing that our bounds are tighter. Second, we show it is maximal — no further quotient is possible.

Theorem 1 (§4): Improved bounds on uncertainty In our resulting language, every closed term describes a convex set of distributions. We thus establish a connection to the non-compositional approach that uses the Kleisli category of the convex powerset of distributions monad (e.g. [7, 8, 26, 30, 39, 57, 58]). We have an 'op-lax' functor

$$R : \mathbf{ImP} \to \mathrm{Kl}(\mathrm{CP}).$$

from our locally graded category **ImP** (§3.2) to the Kleisli category of the convex powerset of distributions monad CP. Being an op-lax functor means that

$$R(g \circ f) \subseteq R(g) \circ R(f),$$

i.e. composition in our category gives a tighter bound on the Knightian uncertainty than the composition using the Kleisli category of the convex powerset of distributions monad.

Note that this could not be a proper functor because we would then have a quotient theory in violation of the maximality theorem (Theorem 2). But an op-lax functor is beneficial as an interpretation of giving a tighter bound.

Theorem 2 (§5): Maximality. Our language also gives rise to a compositional theory of equality. We prove our equational theory is maximal in that we can add no further equations on open terms without equating different convex sets of distributions or compromising the compositional structure.

Further detail: two quotients In slightly more detail, we consider two candidates from the literature for quotients of a graded monad. They are general methods, but appear in our language as follows. Notice that an open term in our language contains both names and variables: names for Knightian choices, and free variables standing for ordinary values that might be substituted later. There are two ways to quotient the names away:

Quotient A Following [20, 21] we could equate two open terms if, for every valuation of the free variables, there is a renaming that equates them.

Quotient B Following [27], we could equate two open terms if there is a renaming such that for every valuation of the free variables they are made equal.

For closed terms with no free variables, the two approaches are the same and give rise to a convex set of distributions (Prop. 4.3).

Quotient A does not directly give a compositional theory in our setting: the criteria of [20] are not satisfied. Nonetheless, the construction of [20] can be adjusted, giving the op-lax functor of Theorem 1 rather than a monad morphism.

Quotient B does not satisfy Desideratum 2 (commuting if-then-else). Informally, it would allow us to rename on the 'then' branch but not the 'else' branch, which is inconsistent with Desideratum 2. Nonetheless, it could be a useful approach in a metalanguage for combining models that do not need a general if-then-else construction. For this reason, Quotient B is not a counterexample to Theorem 2.

2 Rudiments of graded Markov categories and graded monads

We now recall notions of Markov category (§2.1) and relative monads (§ 2.2), recasting them in the locally graded setting. We show how to pass between the concepts (Prop. 2.5, 2.6) and we relate them to notions from enriched category theory (§2.3). For brevity, in this section we focus on definitions and in the next section (§3) we focus on examples, rather than interleaving them.

Definition 2.1. A *monoidal category* is a category $\mathcal{C}$ equipped with a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ and an object I together with associativity and unitor isomorphisms (e.g. $(X \otimes Y) \otimes Z \cong X \otimes (Y \otimes Z)$) that satisfy coherence conditions (e.g. [53]). It is strict if the isomorphisms are equalities. A *symmetric* monoidal category is moreover equipped with isomorphisms $\sigma_{X,Y} : X \otimes Y \cong Y \otimes X$ such that $\sigma_{Y,X} = \sigma^{-1}_{X,Y}$ and satisfying coherence conditions.

A *semicartesian category* is a symmetric monoidal category in which the monoidal unit is a terminal object. That is, there is exactly one morphism $X \to I$ for all X.

A semicartesian category has projections $X \otimes Y \to X \otimes I \cong X$, but it is weaker than a full categorical product because there need not be a natural diagonal $X \to X \otimes X$.

2.1 Graded Markov categories

Definition 2.2. [19] A *Markov category* is a semicartesian category such that every object is equipped with a commutative comonoid structure, that is, a map $\mathsf{copy}_X : X \to X \otimes X$ that is symmetric and associative and has the terminal map $X \to I$ as a unit.

A morphism $f : X \to Y$ in a Markov category is *deterministic* if it commutes with the copy map ($\mathsf{copy}_Y \circ f = (f \otimes f) \circ \mathsf{copy}_X$).

A *distributive* Markov category [1] is a Markov category that has coproducts such that the canonical maps $X \otimes Z + Y \otimes Z \to (X + Y) \otimes Z$ are isomorphisms and the coproduct injections $X \to X + Y \leftarrow Y$ are deterministic.

A typical example of a distributive Markov category is the category **FinStoch** of stochastic matrices (Def. 3.2).

An ordinary distributive category [11, 14] is a distributive Markov category in which every morphism is deterministic. A typical example is the category **FinSet** of finite sets.

Programming syntax. We can use programming language syntax for composition in a distributive Markov category (see also e.g. [70]). The objects of the category are regarded as types, with *bool* regarded as the object $1 + 1$. If $\Gamma = (x_1 : A_1) \otimes \cdots \otimes (x_n : A_n)$ then a morphism $t : \Gamma \to B$ is regarded as a term $\Gamma \vdash t : B$. We notate

(t, u)	for	$\Gamma \xrightarrow{\mathsf{copy}} \Gamma \otimes \Gamma \xrightarrow{t \otimes u} A \otimes B$
$x \leftarrow t\,;\, u$	for	$\Gamma \xrightarrow{\mathsf{copy}} \Gamma \otimes \Gamma \xrightarrow{\Gamma \otimes t} \Gamma \otimes A \xrightarrow{u} B$
$\mathsf{if}\ t\ \mathsf{then}\ u\ \mathsf{else}\ v$	for	$\Gamma \xrightarrow{\mathsf{copy}} \Gamma \otimes \Gamma \xrightarrow{t \otimes \Gamma} (1 + 1) \otimes \Gamma \cong \Gamma + \Gamma \xrightarrow{[t,u]} B$

In this way, given interpretations of **bernoulli** and **knight**, we can interpret the programs from Examples 1.1 and 1.2.

Definition 2.3. Let $\mathbb{G}$ be an semicartesian category. A *graded distributive Markov category* $\mathbf{C}$ is given by

- a distributive Markov category $\mathbf{C}_I$, but moreover,
- for each pair of objects and each grade $a \in \mathbb{G}$ a set $\mathbf{C}_a(X, Y)$ of morphisms, agreeing with $\mathbf{C}_I$ when $a = I$;
- for each morphism $f : b \to a$, a function $\mathbf{C}_a(X, Y) \to \mathbf{C}_b(X, Y)$;

- a family of maps $\circ : \mathbf{C}_a(X,Y) \times \mathbf{C}_b(Y,Z) \to \mathbf{C}_{a\otimes b}(X,Z)$;
- a family $\otimes : \mathbf{C}_a(X,X') \times \mathbf{C}_b(Y,Y') \to \mathbf{C}_{a\otimes b}(X\otimes Y, X'\otimes Y')$

all such that composition is natural and associative up to the associativity of $\mathbb{G}$ (see e.g. [78, §1.2], [49], [23, App. B]), monoidal product of morphisms is also natural and has associators and symmetric braidings up-to the structure of $\mathbb{G}$, and such that the induced function $\mathbf{C}_a(X+Y,Z) \to \mathbf{C}_a(X,Z) \times \mathbf{C}_a(Y,Z)$ is a bijection (e.g. [78, p. 36]).

See Proposition 3.3 for our example of a graded Markov category. We note that since $\mathbb{G}$ is semicartesian, there are canonical projections $a \otimes b \to a$, and so we can regard any morphism at grade a as a morphism at grade $(a\otimes b)$.

2.2 Monads and graded relative affine monads

It is well-established that notions of computation can be modelled by monads [60], including probability and non-determinism [35,57,58]. In this section, we introduce the flavours of monads relevant to this work and establish their correspondence to Markov categories. We present them in the Kleisli triple setting because this is more conducive to their use in programming languages.

Definition 2.4. A *strong monad* over a cartesian closed category $\mathbf{C}$ is for each $X \in \mathbf{C}$ an object $T(X) \in \mathbf{C}$ and a morphism

$$\eta_X : I_{\mathbf{C}} \to [X, T(X)],$$

and a family of morphisms

$$(-)^* : [X, T(Y)] \to [T(X), T(Y)]$$

such that for generalised elements f and g of $[X,T(Y)]$ and $[Y,T(Z)]$, the following equations hold:

$$\begin{aligned} f &= f^* \circ \eta_X \\ id_{T(X)} &= (\eta_X)^* \\ g^* \circ f^* &= (g^* \circ f)^*. \end{aligned} \tag{2}$$

The left-strength $t : X \times T(Y) \to T(X\times Y)$ is induced by the canonical action $(\eta_Y \circ -)^* : [X,Y] \to [T(X),T(Y)]$ [42].

The monad is *commutative* if the following diagram commutes, where $\hat{t}$ is the induced right strength from the symmetry of the cartesian product.

$$\begin{tikzcd} & T(X)\times T(Y) \arrow[dl, "\hat{t}"'] \arrow[dr, "t"] & \\ T(X\times T(Y)) \arrow[dr, "t^*"'] & & T(T(X)\times Y) \arrow[dl, "\hat{t}^*"] \\ & T(X\times Y) & \end{tikzcd}$$

It is *affine* if the unique map $T(1) \to 1$ is an isomorphism. A typical example is $\mathbf{C} = \mathbf{Set}$, and $T = D$ is the finite probability distribution monad [32].

Let $\mathcal{J}$ be a category with finite products and consider a finite product preserving functor $J : \mathcal{J} \to \mathbf{C}$. A *relative strong monad* T on J is a functor $T : \mathcal{J} \to \mathbf{C}$, along with a J-relative unit

$$\eta_X : I \to [J(X), T(X)]$$

natural in $X \in \mathcal{J}$, and a family of J-relative Kleisli extensions

$$(-)^* : [J(X), T(Y)] \to [T(X), T(Y)]$$

natural in $X, Y \in \mathcal{J}$, and such that (2) holds for f and g generalised elements of $[J(X),T(Y)]$ and $[J(Y),T(Z)]$. A typical example is $\mathbf{C} = \mathbf{Set}$, and $\mathcal{J} = \mathbf{FinSet}$, with J the evident embedding, and $T = DJ$.

Let $(\mathbb{G}, \otimes, I)$ be a monoidal category. A *graded strong monad* is a functor $T : \mathbb{G} \to [\mathbf{C}, \mathbf{C}]$, with unit

$$\eta_X : I \to [X, T_I(X)]$$

natural in $X \in \mathcal{J}$, and a family of Kleisli extensions

$$(-)^*_{a,b} : [X, T_b(Y)] \to [T_a(X), T_{a\otimes b}(Y)]$$

natural in $X, Y \in \mathcal{J}$, and such that for f and g generalised elements of $[X, T_a(Y)]$ and $[Y, T_b(Z)]$, the following equations hold:

$$\begin{aligned} f &= T_\lambda \circ (f)^*_{I,a} \circ \eta_X, \\ id_{T_b(X)} &= T_\rho \circ (\eta_X)^*_{b,I}, \\ (g)^*_{c\otimes a,b} \circ (f)^*_{c,a} &= T_\alpha \circ ((g)^*_{a,b} \circ f)^*_{c,a\otimes b}, \end{aligned} \tag{3}$$

where $\lambda : I \otimes a \to a$, $\rho : b \otimes I \to b$, and $\alpha : c \otimes (a \otimes b) \to (c \otimes a) \otimes b$ are the left unitor, right unitor, and associator of $\mathbb{G}$, respectively.

A *graded relative strong monad* T on J is a functor $T : \mathbb{G} \to [\mathcal{J}, \mathbf{C}]$, along with a J-relative unit

$$\eta_X : I_{\mathbf{C}} \to [J(X), T_I(X)]$$

natural in $X \in \mathcal{J}$, and a family of J-relative Kleisli extensions

$$(-)^*_{a,b} : [J(X), T_b(Y)] \to [T_a(X), T_{a\otimes b}(Y)]$$

natural in $X, Y \in \mathcal{J}$, and such that (3) holds for f and g generalised elements of $[J(X), T_a(Y)]$ and $[J(Y), T_b(Z)]$. The graded left-strength t_a is induced by the action $T_\rho \circ (\eta_Y \circ -)^*_{a,I} : [X, Y] \to [T_a(X), T_a(Y)]$. The monad is *commutative* if $\mathbb{G}$ is symmetric monoidal and the following diagram commutes, where $\sigma : a \otimes b \to b \otimes a$ is the symmetric coherence isomorphism of $\mathbb{G}$.

$$\begin{tikzcd}
 & T_a(X) \times T_b(Y) \arrow[dl, "\hat{t}_a"'] \arrow[dr, "t_b"] & \\
T_a(X \times T_b(Y)) \arrow[d, "(t_b)^*_{a,b}"'] & & T_b(T_a(X) \times Y) \arrow[d, "(\hat{t}_a)^*_{b,a}"] \\
T_{a\otimes b}(X \times Y) \arrow[rr, "T_\sigma"] & & T_{b\otimes a}(X \times Y)
\end{tikzcd}$$

It is *affine* if the unique map $T_I(1) \to 1$ is an isomorphism.

Proposition 2.5. *If T is a graded commutative affine relative monad on a distributive category, then its Kleisli category (e.g. [22, 45]) is a graded Markov category:*

- *The objects are the same as $\mathcal{J}$;*
- *The morphisms in $\mathrm{Kl}(T)_a(X, Y)$ are the morphisms $J(X) \to T_a(X)$ in $\mathbf{C}$;*
- *Composition is via the Kleisli extension.*

Proposition 2.6. *Any graded Markov category induces a graded commutative affine relative monad, by*

- *$\mathcal{J} = \mathbf{C}_{I,\mathrm{det}}$, the distributive category of I-graded deterministic maps*
- *The underlying category is $\mathrm{FP}(\mathcal{J}^{\mathrm{op}}, \mathbf{Set})$, the finite product-preserving contravariant presheaves on $\mathcal{J}$;*
- *$J : \mathcal{J} \to \mathrm{FP}(\mathcal{J}^{\mathrm{op}}, \mathbf{Set})$ is the Yoneda embedding;*
- *$T : \mathbb{G} \to [\mathcal{J}, \mathrm{FP}(\mathcal{J}^{\mathrm{op}}, \mathbf{Set})]$ is given by*
$$T_a(Y)(X) = \mathbf{C}_a(X, Y)$$

Proof. Note. In both cases, the proof amounts to expanding the definitions. The constructions are similar to [65, §7]. See also [1, Prop. 13] for the non-graded case. □

(We conjecture that Propositions 2.5–2.6 are part of a biequivalence between graded distributive Markov categories and commutative affine graded relative monads. We do not pursue this here because we will not need the generality of the biequivalence in what follows.)

2.3 Connection with enriched categories

To show that the concepts in this section are canonical, we connect with the theory of enriched categories. Let $\mathcal{V}$ be a symmetric monoidal closed category with limits and colimits, that is moreover semicartesian. Recall (e.g. [40]) that a $\mathcal{V}$-enriched category $\mathbf{C}$ is given by a collection of objects, and for each pair of objects X, Y of $\mathbf{C}$, a 'hom-object' $\mathbf{C}(X, Y)$ in $\mathcal{V}$. Composition is a morphism $\mathbf{C}(X, Y) \otimes \mathbf{C}(Y, Z) \to \mathbf{C}(X, Z)$ in $\mathcal{V}$. We can also define $\mathcal{V}$-enriched monoidal categories, by requiring the functor $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ to be a monoidal category. And $\mathcal{V}$-enriched coproducts require a natural isomorphism

$$\mathbf{C}(X_1 + \cdots + X_n, Y) \cong \mathbf{C}(X_1, Y) \times \cdots \times \mathbf{C}(X_n, Y)$$

between objects of $\mathcal{V}$. Any enriched category has an underlying *ordinary category* $\mathbf{C}_0$, which has the same objects but with a hom-*set* given by $\mathbf{C}_0(X, Y) = \mathcal{V}(I, \mathbf{C}(X, Y))$. This ordinary category inherits monoidal, limit and colimit structure from $\mathbf{C}$.

Definition 2.7 (e.g. [62])**.** A *$\mathcal{V}$-enriched Markov category* is a $\mathcal{V}$-enriched symmetric monoidal category such that the monoidal unit is terminal: $\mathbf{C}(A, I) \cong 1$, and such that the underlying symmetric monoidal category is equipped with the structure of a Markov category (i.e. a comonoid structure in the underlying ordinary category).

A $\mathcal{V}$-enriched Markov category is moreover *distributive* if it has $\mathcal{V}$-coproducts that distribute over the monoidal structure, and such that the coproduct injections are deterministic, in the sense of the underlying ordinary category.

For any semicartesian category $\mathbb{G}$, recall the category of functors $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$. This extends $\mathbb{G}$ to a good 'cosmos' for enrichment since

- $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$ embeds $\mathbb{G}$ fully and faithfully (i.e. essentially as a full subcategory), via the Yoneda embedding $y(a) = \mathbb{G}(-, a)$.
- $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$ has all limits and colimits, computed pointwise.
- $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$ has a semicartesian structure such that the Yoneda embedding is a symmetric monoidal functor. This is given by Day convolution [17], and has the following universal property: for functors $F, G, H \in [\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$, to give a natural transformation $F \otimes G \to H$ is to give a natural family of functions $F(a) \times G(b) \to H(a \otimes b)$.
- $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$ is moreover monoidal closed.

Proposition 2.8. *To give an $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$-enriched distributive Markov category is to give a $\mathbb{G}$-graded distributive Markov category.*

Notes. This follows from the characterization of locally $\mathbb{G}$-graded categories as $[\mathbb{G}^{\mathrm{op}}, \mathbf{Set}]$-enriched categories (e.g. [23, 49, 78]), and then translating Definition 2.7 across this correspondence to arrive at Definition 2.3. □

The correspondence between graded monads and enriched monads is also well understood (e.g. [54]).

2.4 Discussion

Recent work by Perrone [62] has considered enriched Markov categories to obtain an abstract view of the distance between probabilities, which allows for an abstract development of entropy. We note that the enriching category $\mathcal{V} = \mathsf{Div}$ in [62] is indeed semicartesian. The full theory of enriched Markov categories perhaps deserves a more thorough analysis.

3 A graded Markov category for probability and non-determinism

We recall ordinary Markov categories for finite probability (§3.1). We then consider a generic construction for graded Markov categories, and instantiate it in our setting, obtaining the graded Markov category **ImP** (for 'Imprecise Probability', §3.2). We conclude this section with a worked example (§3.3). In the subsequent sections (§4–5) we relate this graded Markov category with convex sets of distributions.

3.1 Ordinary Markov categories for probability

Definition 3.1. A *probability vector* $p \in \mathbb{R}^n$ is a sequence of non-negative numbers that sum to 1. We write $D(n)$ for the probability vectors of length n.

The set $D(n)$ is always a convex set: for any $r \in [0,1]$ and $p, q \in D(n)$, the convex combination $r \cdot p + (1-r) \cdot q$ is again a probability vector in $D(n)$. We write $p +_r q$ as shorthand for such a convex combination. Every probability vector in $D(n)$ arises via convex combinations of the Dirac vectors δ_i, for $i = 1 \dots n$, where $\delta_1 = (1, 0, 0, 0 \dots)$, $\delta_2 = (0, 1, 0, 0 \dots)$ and so on.

A matrix of real numbers $f \in \mathbb{R}^{n \times m}$ is called *stochastic* if each column is a probability vector. This is equivalent to requiring that as a linear map, it preserves the property of being a probability vector, i.e. if $p \in D(n)$ then $(f\,p) \in D(m)$. In fact, every function $D(n) \to D(m)$ that preserves convex structure arises from a stochastic matrix in this way. We call such a function a *convex map*.

Definition 3.2 (e.g. [19], Ex. 2.5)**.** The category **FinStoch** of finite stochastic matrices has as objects natural numbers, and morphisms $m \to n$ stochastic matrices in $\mathbb{R}^{n \times m}$. Composition is matrix multiplication, and the identity morphism is the unit diagonal matrix.

This can be made into a symmetric monoidal category, with monoidal structure on objects given by multiplication of numbers, and on morphisms by the Kronecker product of matrices. It is semicartesian where the terminal object is 1, and there is a unique stochastic matrix with one row. This is moreover a Markov category, with $\mathsf{copy}_n : n \to n \otimes n$ given by the three-dimensional diagonal (in $\mathbb{R}^{(n \times n) \times n}$).

The Markov category **FinStoch** moreover has a distributive structure. The coproduct of objects is given by addition, and the coproduct of morphisms by block matrices (concatenating the columns).

The monad view on **FinStoch** is as follows. First, we consider the embedding $J : \mathbf{FinSet} \to \mathbf{Set}$. We then regard D (Def. 3.1) as a J-relative monad $D \colon \mathbf{FinSet} \to \mathbf{Set}$, which is affine and commutative. In fact there is an ordinary monad D' on **Set**, comprising finitely supported probability distributions (e.g. [32, Ch. 2]), and $D = D'J$. The distributive Markov category **FinStoch** can then be regarded as the Kleisli category for this relative monad.

3.2 The graded Markov category ImP

We first introduce a general construction for graded Markov categories. This is a variation on the Para construction [18], also called monoidal indeterminates [27]. Via the connections between Markov categories and commutative affine relative monads (§2.2), it is equivalently a graded version of the reader monad transformer [52].

Proposition 3.3. *Let $\mathbb{G}$ be a semicartesian subcategory of a distributive Markov category $\mathbf{C}$. There is a graded distributive Markov category with the same objects as $\mathbf{C}$ and with the hom-sets*

$$\mathbf{C}_a(X, Y) = \mathbf{C}(a \otimes X, Y).$$

The reindexing is given by composition: if $f \in \mathbb{G}(b, a)$ and $g \in \mathbf{C}_a(X, Y)$

$$f^*(g) = g \circ (f \otimes X) \in \mathbf{C}_b(X, Y).$$

For the composition of $f \in \mathbf{C}_a(X, Y)$ and $g \in \mathbf{C}_b(Y, Z)$,

$$(g \circ f) = \left(a \otimes b \otimes X \cong b \otimes a \otimes X \xrightarrow{b \otimes f} b \otimes Y \xrightarrow{g} Z\right) \in \mathbf{C}_{a \otimes b}(X, Z).$$

The monoidal product of $f \in \mathbf{C}_a(X, X')$ and $g \in \mathbf{C}_b(Y, Y')$ is given by

$$(f \otimes g) = \left(a \otimes b \otimes X \otimes Y \cong a \otimes X \otimes b \otimes Y \xrightarrow{f \otimes g} X' \otimes Y'\right) \in \mathbf{C}_{a \otimes b}(X \otimes Y, X' \otimes Y').$$

Definition 3.4. A stochastic matrix $f \in \mathbf{FinStoch}(m, n)$ is *surjective* if for every $j \in [1, n]$ there exists $i \in [1, m]$ such that f_i is the Dirac distribution at j. In other words, the induced function $D(m) \to D(n)$ is surjective. Let $\mathbf{FinStoch}_{\mathrm{Surj}}$ be the category of natural numbers and surjective stochastic matrices. This is a semicartesian monoidal subcategory of **FinStoch**.

Definition 3.5. The graded distributive Markov category **ImP** is the $\mathbf{FinStoch}_{\mathrm{Surj}}$-graded version of **FinStoch**, according to Proposition 3.3.

This graded distributive Markov category supports both finite probability and finite non-determinism.

- For binary probabilistic choice with bias r, we consider the morphism in $\mathbf{ImP}_1(1,2)$ given by the column vector $\begin{pmatrix} r \\ 1-r \end{pmatrix}$.
- For a binary non-deterministic (i.e. Knightian) choice, we consider the morphism in $\mathbf{ImP}_2(1,2)$ given by the unit diagonal matrix.

We can extend the above notions of probabilistic and non-deterministic choice between elements of a finite set n by considering probability vectors (in $\mathbf{ImP}_1(1,n)$) and unit diagonal matrices (in $\mathbf{ImP}_n(1,n)$) respectively.

Remark: We could have considered a subcategory of $\mathbf{FinStoch}_{\mathrm{Surj}}$ as the grading. One example is finite sets and (deterministic) surjective functions. Another example is the subcategory where the objects are of the form 2^A and where we only consider the surjections $2^B \to 2^A$ induced by injections $A \to B$ (connecting with nominal sets [64] and the notation for grading considered in Section 1.2; in this case the semicartesian monoidal structure amounts to disjoint union, $A \uplus B$.). We leave for future work the question of to what extent the following results depend on this particular choice of grading.

3.3 Example calculation with ImP

Example 3.6. Consider the scenarios from Examples 1.1 and 1.2 where we draw boolean values with Knightian uncertainty and from fair Bernoulli trials and combine them using different program logic. We denote outcomes as probability vectors of length three, representing the chance of r, g, and b, respectively. Example 1.1 is the morphism

$$(g+h)\circ f = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \\ 0 & 0.5 \end{pmatrix} \in \mathbf{ImP}_2(1,3),$$

where f denotes the conditional on the fair Bernoulli trial, and g and h are the conditionals on the Knightian choices in each branch, respectively.

$$f = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \in \mathbf{ImP}_1(1,2) \qquad g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \in \mathbf{ImP}_2(1,3) \qquad h = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbf{ImP}_2(1,3)$$

On the other hand, Example 1.2 is the morphism

$$(\mathbf{FinStoch}_{\mathrm{Surj},\pi_1}(g) + \mathbf{FinStoch}_{\mathrm{Surj},\pi_2}(h))\circ f = \begin{pmatrix} 1 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix} \in \mathbf{ImP}_4(1,3),$$

where f, g, and h denote the same conditional statements, but now we lift the grading of g and h to 4 via the projections $\pi_1, \pi_2 \in \mathbf{FinStoch}_{\mathrm{Surj}}(2\times 2, 2)$ to account for the decoupling of their Knightian uncertainties.

4 Convex sets of distributions and an op-lax functor

In this section we recall the properties of convex powersets of distributions (see also [7,30,39,57,58] and elsewhere). We then connect our category $\mathbf{ImP}$ (§3) with convex powersets, via the Kan extension method of [20] (§4.2) and then show that this yields an op-lax functor (§4.3), which means that our category is more conservative with uncertainty bounds. We begin by recalling some basic properties of convex sets of distributions.

Definition 4.1. A subset S of $D(n)$ is *convex* if it is closed under convex combinations: if $p, q \in S$ then for any $r \in [0,1]$ we have $p +_r q \in S$.

A convex subset S of $D(n)$ is *finitely generated* if there is a finite sequence $p_1 \dots p_m \in S$ such that every element of S can be achieved by convex combinations of the p_i's. In other words, $S = \{q \cdot (p_1 \dots p_m) \mid q \in D(m)\}$, with the p_i's regarded as column vectors and q regarded as a row vector.

Lemma 4.2. *For any convex map $f : D(m) \to D(n)$ between the sets of probability vectors, the image of f is a convex subset of $D(n)$.*

Moreover, such convex subsets of $D(n)$ are finitely generated, and every finitely generated convex set arises in this way.

Proof. Suppose $q, q' \in \mathsf{image}(f)$, and let $r \in [0,1]$. So we must have $p, p \in D(m)$ such that $f(p) = q$ and $f(p') = q'$. Then

$$q +_r q' = f(p) +_r f(p') = f(p +_r p'),$$

the last step because f is a convex map, and so we see that $q +_r q' \in \mathsf{image}(f)$.

The set $\mathsf{image}(f)$ is generated by $f(\delta_i)$ for $i = 1 \dots m$. Conversely if a set S is generated by $p_1 \dots p_m$, regarded as column vectors, then the matrix $(p_1 \dots p_m) \in \mathbf{FinStoch}(m,n)$ determines a map $f : D(m) \to D(n)$ such that $\mathsf{image}(f) = S$. □

4.1 Convex powersets of distributions

We write $\mathrm{CP}(n)$ for the set of convex subsets of $D(n)$, and $\mathrm{CP}_{\mathrm{fin}}(n)$ for the finitely generated convex subsets of $D(n)$. Both support convex combinations: if $r \in [0,1]$ and $S, T \in \mathrm{CP}(n)$ then

$$S +_r T \overset{\text{def}}{=} \{p +_r q \mid p \in S, q \in T\} \in \mathrm{CP}(n).$$

There is moreover an ordering given by subset, and the join is a convex closure of the union:

$$S \vee T \overset{\text{def}}{=} \{p +_r q \mid r \in [0,1], p \in S, q \in T\}.$$

Proposition 4.3. *There is a family of functions $\phi_{m,n} : \mathbf{ImP}_m(1,n) \to \mathrm{CP}_{\mathrm{fin}}(n)$, that takes $f \in \mathbf{ImP}_m(1,n)$ to its image $\mathsf{image}(f) \in \mathrm{CP}_{\mathrm{fin}}(n)$, and the family is natural in $m \in \mathbf{FinStoch}_{\mathrm{Surj}}$ and $n \in \mathbf{Set}$.*

Proof. First, the fact that the image of f is convex is Lemma 4.2. For naturality in m, suppose $g \in \mathbf{FinStoch}_{\mathrm{Surj}}(m', m)$. Then naturality in m amounts to the fact that

$$\mathsf{image}(f \circ g) = \mathsf{image}(f)$$

which is true since g is surjective. For naturality in n, suppose $h \in \mathbf{Set}(n, n')$. Then naturality amounts to the fact that

$$\mathsf{image}(D(h) \circ f) = \mathrm{CP}(h)(\mathsf{image}(f))$$

which is true because taking an image of f after postcomposition with $D(h)$ is the same as a pointwise application of $D(h)$ to the image of f. □

4.2 Connection to Kan extensions

Fritz and Perrone [20, 21] propose a method to extract a monad from a graded monad, by taking the left Kan extension. They provide criteria for when this process works and induces a monad morphism. This process *cannot work* entirely for our situation, for the following reason. First we note that we can interpret **bernoulli** and **knight** as the following elements of $\mathrm{CP}(2)$:

- **bernoulli** is $\{\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)\} +_{0.5} \{\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\} \;=\; \{\left(\begin{smallmatrix}0.5\\0.5\end{smallmatrix}\right)\}$;
- **knight** is $\{\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)\} \vee \{\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\} \;=\; \{\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\}$.

This construction CP extends to a monad on **Set** [30]. Therefore, we can follow through the derivation of Figure 2 to see that the graded monad *cannot* be commutative (apparently contradicting [30, Lemma 5.2]) since the convex sets in Fig. 1(b)–(c) are different. (For another argument, note that CP contains two binary idempotent symmetric operations, $\vee$ and $+_{0.5}$, and see e.g. [76].)

By contrast, **ImP** (§3.2) *does* satisfy our desiderata (§1.1). So there cannot be a monad morphism between them. Nonetheless, the Kan extension of our graded Markov category **ImP**, regarded as a graded monad via Proposition 2.6, does give the finitely-generated convex powerset monad CP as a functor, just not as a monad.

Proposition 4.4. *The family* $\phi_{m,n} : \mathbf{ImP}_m(1,n) \to \mathrm{CP}_{\mathrm{fin}}(n)$ *exhibits* $\mathrm{CP}_{\mathrm{fin}} : \mathbf{FinSet} \to \mathbf{Set}$ *as the Kan extension of*

$$\mathbf{ImP}_{(-)}(1,=) : \mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}} \to [\mathbf{FinSet}, \mathbf{Set}]$$

along the unique functor $\mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}} \to 1$.

$$\begin{tikzcd}
\mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}} \arrow[r, "!"] \arrow[dr, "\mathbf{ImP}_{(-)}(1,=)"'] & 1 \arrow[d, "\mathrm{CP}_{\mathrm{fin}}"] \\
& {[\mathbf{FinSet}, \mathbf{Set}]}
\end{tikzcd}$$

Proof. Kan extensions in $[\mathbf{FinSet}, \mathbf{Set}]$ can be computed pointwise, and for any $n \in \mathbf{FinSet}$ the Kan extension of $\mathbf{ImP}_{(-)}(1,n) : \mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}} \to \mathbf{Set}$ along $\mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}} \to 1$ is simply the colimit of the functor. Thus it suffices to show that the canonical function

$$\Phi : \operatorname*{colim}_{m \in \mathbf{FinStoch}_{\mathrm{Surj}}^{\mathrm{op}}} \mathbf{ImP}_m(1,n) \to \mathrm{CP}_{\mathrm{fin}}(n)$$

(induced by ϕ) is a bijection. This function Φ is given by $\Phi[m, f \in \mathbf{FinStoch}(m,n)] = \mathsf{image}(f)$. To see that it is surjective we recall that every finitely generated convex set is the image of some convex function $D(m) \to D(n)$ (Lemma 4.2). To see that it is injective we suppose that $\mathsf{image}(f) = \mathsf{image}(f')$, for $f \in \mathbf{FinStoch}(m,n)$ and $f' \in \mathbf{FinStoch}(m',n)$. We need to show that $[m,f] = [m',f']$ in the colimit. It suffices to find m'' with $h \in \mathbf{FinStoch}(m,n)$ and surjections $g \in \mathbf{FinStoch}_{\mathrm{Surj}}(m,m'')$ and $g' \in \mathbf{FinStoch}_{\mathrm{Surj}}(m',m'')$:

$$\begin{tikzcd}
m \arrow[dr, "g"] \arrow[ddr, bend right, "f"'] & & m' \arrow[dl, "g'"'] \arrow[ddl, bend left, "f'"] \\
& m'' \arrow[d, "h"'] & \\
& n &
\end{tikzcd}$$

The finitely generated convex set $\mathsf{image}(f) = \mathsf{image}(f')$ must have a unique convex hull, and we let m'' be the number of extremal points of the convex hull, which are uniquely determined. We construct g by noting that $f(i)$ must be a convex combination from the m'' extremal points, and so we let $g(i)$ be the probability vector corresponding to that combination. We construct g' from f' similarly. To see that g is surjective we note that since f is surjective onto its image we must have points in m that map onto the extremal points, and hence onto all the points of m'' via g. Similarly, g' is surjective. ☐

4.3 An op-lax functor

Definition 4.5. The construction $\mathrm{CP}_{\mathrm{fin}}$ extends to a relative monad. The unit morphism $\eta_n : n \to \mathrm{CP}_{\mathrm{fin}}(n)$ picks out the singleton set containing the Dirac vector, $\eta_n(i) = \{\delta_i\}$. The Kleisli extension takes a function $f : m \to \mathrm{CP}_{\mathrm{fin}}(n)$ to a function $f^* : \mathrm{CP}_{\mathrm{fin}}(m) \to \mathrm{CP}_{\mathrm{fin}}(n)$ given by

$$f^*(X) = \bigvee_{x \in ext(X)} \sum_{i \in m} x_i \cdot f(i);$$

where *ext* takes the extreme points of the finitely generated convex subset.

From this structure, we build a Kleisli category as usual.

- The objects of $\mathrm{Kl}(\mathrm{CP}_{\mathrm{fin}})$ are natural numbers.
- The morphisms $m \to n$ are functions $m \to \mathrm{CP}_{\mathrm{fin}}(n)$.
- The identity morphism is the unit η. Composition of g and f is given by $g^* \circ f$.

In fact, this category is order-enriched. That is to say, the hom-sets $\mathrm{Kl}(\mathrm{CP}_{\mathrm{fin}})$ have a natural partial order structure given by $f \leq g$ if for all i, $f(i) \subseteq g(i)$. Composition is thus monotone.

We now extend the quotient of Proposition 4.3 to an identity-on-objects op-lax functor $\mathbf{ImP} \to \mathrm{Kl}(\mathrm{CP}_{\mathrm{fin}})$.

Theorem 1. *Consider the assignment of a morphism $f \in \mathbf{ImP}_a(m,n)$ to $R(f) : m \to \mathrm{CP}_{\mathrm{fin}}(n)$ given by $R(f)(i) = \mathsf{image}(f(-,i))$. This defines an op-lax functor*

$$\mathbf{ImP} \to \mathrm{Kl}(\mathrm{CP}_{\mathrm{fin}})$$

Proof notes. It is straightforward that $R(\mathsf{id}) = \mathsf{id}$. It remains to show that $R(g \circ f) \subseteq R(g) \circ R(f)$. Since we will show that R preserves finite coproducts, it is sufficient to first suppose that the domain of f is 1. So consider $f \in \mathbf{ImP}_a(1,m)$ and $g \in \mathbf{ImP}_b(m,n)$. So $(g \circ f) \in \mathbf{FinStoch}(a \times b, n)$. We must show that for all $(i,j) \in (a \times b)$, the probability vector $(g \circ f)(i,j)$ is in

$$(R(g) \circ R(f))() = R(g)^*(\mathsf{image}(f)) \in \mathrm{CP}(n).$$

To show this, we note that the grade of $(g \circ f)$ is $(a \times b)$, but we can also consider an alternative kind of composite $(g * f)$ with a bigger grade $(a \times b^m)$. This is given by

$$(g * f) = \Big(a \times b^m \xrightarrow{f} m \times b^m \xrightarrow{(eval, proj_1)} b \times m \xrightarrow{g} n\Big);$$

where the middle arrow is the evident function between sets regarded as a stochastic matrix. Contrast with

$$(g \circ f) = \Big(a \times b \xrightarrow{f} m \times b \xrightarrow{swp} b \times m \xrightarrow{g} n\Big).$$

The function $(a \times b) \to (a \times b^m)$ that copies b is an injection and exhibits

$$\mathsf{image}(g \circ f) \subseteq \mathsf{image}(g * f)$$

Moreover, we have that

$$\mathsf{image}(g * f) = R(g) \circ R(f).$$

The intuitive point is that in $(g * f)$, for each possible intermediate m we are allowed to use different choices of b, but in $(g \circ f)$, each possible intermediate m will use the same choices of b.

To see that R preserves coproducts we note that on objects it is immediate, and expanding the definitions shows that the coproduct injections and copairings are exactly preserved by R. $\square$

4.4 Discussion and example

Example 4.6. We once again revisit the scenarios from Examples 1.1 and 1.2 where boolean values are drawn with Knightian uncertainty or from fair Bernoulli trials and combined using different program logic. Consider the morphism denoting a fair Bernoulli trial (f from Example 3.6),

$$f = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \in \mathbf{ImP}_1(1,2),$$

and a morphism that employs Knightian uncertainty on each of its inputs ($g + h$ from Example 3.6),

$$g = \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \in \mathbf{ImP}_2(2,3).$$

Then $R(f) : 1 \to \mathrm{CP}_{\mathrm{fin}}(2)$ maps the singleton set to

$$\left\{ \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \right\},$$

$R(g) : 2 \to \mathrm{CP}_{\mathrm{fin}}(3)$ maps the two-element set to

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\} \text{ and } \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\},$$

and, following Example 3.6, $R(g \circ f) : 1 \to \mathrm{CP}_{\mathrm{fin}}(3)$ maps the singleton set to

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix} \right\}.$$

This is the convex subset in Figure 1(b) if we consider the probability vectors as giving the corresponding chances of outcomes r, g, and b.

On the other hand, by composing g with f after mapping them into $\mathrm{Kl}(\mathrm{CP}_{\mathrm{fin}})$, we lose the ability to distinguish which outcomes were related via the same Knightian choices. So the morphism $R(g) \circ R(f) : 1 \to \mathrm{CP}_{\mathrm{fin}}(3)$ maps the singleton element to

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix} \right\},$$

which is the convex subset given in Figure 1(c). Thus, $R(g \circ f) \subsetneq R(g) \circ R(f)$.

Therefore, by accounting for corresponding choices of Knightian uncertainty within morphism compositions, our category **ImP** obtains tighter bounds on the imprecise probabilities.

5 Maximality as a commutative theory

In Proposition 4.3 we gave a family of maps ϕ that convert our compositional imprecise probability into convex sets of probability distributions. These maps are not injective, but in Theorem 2 we show that they cannot be any more injective, in the following sense: We cannot quotient the hom-sets of **ImP** without either losing the connection with convex sets (and hence statistics) or losing the monoidal or distributive structure (and hence the compositionality desiderata of §1.1).

Definition 5.1. Let $\mathbb{G}$ be a semicartesian category. Let $\mathbf{C}$ and $\mathbf{D}$ be $\mathbb{G}$-graded distributive Markov categories. A *graded distributive Markov functor* $F : \mathbf{C} \to \mathbf{D}$ is given by a mapping from the objects of $\mathbf{C}$ to the objects of $\mathbf{D}$ and a family of mappings from $\mathbf{C}_a(X, Y) \to \mathbf{D}_a(F(X), F(Y))$, strictly preserving the composition, monoidal and coproduct structure, and the copy maps.

Note In view of §2.1, we note that a graded distributive Markov functor is the same thing as the existing notion of strict distributive monoidal functor between distributive monoidal enriched categories (e.g. [40]), together with the requirement that the copy maps are preserved, which is in common with the ordinary Markov category literature [19]. We could also formulate this in terms of monad morphisms, following Section 2.2.

Theorem 2. *Let* $\mathbf{C}$ *be* $\mathbf{FinStoch}_{\mathrm{Surj}}$*-graded distributive Markov category with a graded distributive Markov functor* $F : \mathbf{ImP} \to \mathbf{C}$ *and a natural family of functions*

$$\psi_{m,n} : \mathbf{C}_m(1, n) \to \mathrm{CP}_{\mathrm{fin}}(n)$$

such that

$$\phi_{m,n} : \mathbf{ImP}_m(1, n) \to \mathrm{CP}_{\mathrm{fin}}(n)$$

(Proposition 4.3) factors through ψ*. Then* F *is faithful: if* $F(f) = F(g)$ *in* $\mathbf{C}$ *then also* $f = g$ *in* **ImP**.

Proof. Since F preserves finite coproducts, it is sufficient to suppose the domain of f and g is 1. That is, let $f, g \in \mathbf{ImP}_m(1, n)$ and suppose $\phi_{m,n}$ factors as

$$\mathbf{ImP}_m(1, n) \xrightarrow{F_{m,n}} \mathbf{C}_m(1, n) \xrightarrow{\psi_{m,n}} \mathrm{CP}_{\mathrm{fin}}(n).$$

Let $d \in \mathbf{ImP}_m(1, m)$ be the evident tuple of Diracs. Define $\iota \in \mathbf{ImP}_1(m, n+m)$ and $\jmath \in \mathbf{ImP}_1(n, m+n)$ as the lifting of the injections $m \to m + n \leftarrow n$ via postcomposition with the unit of D. Since F is a graded distributive Markov functor and $F_{m,n}(f) = F_{m,n}(g)$,

$$F_{m,m+n}(\jmath \circ f +_{0.5} \iota \circ d) = F_{m,m+n}(\jmath \circ g +_{0.5} \iota \circ d)$$

where for $h, k : X \to D(Y)$, we define $(h +_r k)(x) = h(x) +_r k(x)$. Applying ψ gives

$$\phi_{m,m+n}(\jmath \circ f +_{0.5} \iota \circ d) = \phi_{m,m+n}(\jmath \circ g +_{0.5} \iota \circ d) \tag{4}$$

Now, for all $i \in m$, $(\jmath \circ f +_{0.5} \iota \circ d)(i)$ are independent because they each use a different dimension. They are all extremal vertices on the convex hull $\phi_{n,m+n}(\jmath \circ f +_{0.5} \iota \circ d)$. Moreover, they must be the same vertices as $(\jmath \circ g +_{0.5} \iota \circ d)(i)$ for respective $i \in m$ because the convex hulls are the same (4). Therefore,

$$\jmath \circ f +_{0.5} \iota \circ d = \jmath \circ g +_{0.5} \iota \circ d.$$

We can recover f and g as for any $i \in m, j \in n$,

$$f(i)(j) = 2 \times (\jmath \circ f +_{0.5} \iota \circ d)(i)(j),$$
$$g(i)(j) = 2 \times (\jmath \circ g +_{0.5} \iota \circ d)(i)(j).$$

So $f = g$. □

6 Summary and outlook

We have shown that by taking a graded perspective and naming Knightian choices we can obtain a compositional account of Bernoulli and Knightian uncertainty together. The account gives a refined bound on the uncertainty (Theorem 1) and is maximal among the compositional accounts (Theorem 2).

There are a number of future directions. A first question is how to accommodate iteration. The convex sets considered in this article are all finitely generated, but if we allow iterative programs that have an unbounded number of Knightian choices, this leads to a more general class of convex sets.

The concerns about iteration hold even if we restrict to finite outcome spaces, and thus far we have focused on this for simplicity. Much work on programming semantics for imprecise probability has focused beyond finite outcome spaces, and it will be interesting to revisit this from our perspective: this includes domain theoretic structures (e.g. [25, 38, 39, 75]) and metric structures (e.g. [57, 58]).

It would be interesting to compare to another recent compositional framework combining unknowns with probability by Stein and Samuelson, currently focusing on Gaussians [69].

Our approach is based on random elements, and so is the quasi-Borel-space probability monad (e.g. [28, 72]), so this might be a good approach to accommodating function spaces.

On the more practical side, an open question is how to perform statistical inference in a probabilistic programming language with imprecise probability.

Going beyond statistics, it is possible that there are other scenarios where this approach is useful: making a theory compositional by using a graded theory (for a first purely speculative example, the issues with amb outlined in [48]).

Acknowledgements It has been helpful to talk to many people about this work. The 2023/2024 Aria workshops emphasised aspects of imprecise probability as discussion topics (and so it was helpful to discuss with davidad, Dylan Braithwaite, Elena Di Lavore, Alex Lew, Owen Lynch, Sean Moss, Zenna Tavares, and others). It has been helpful to discuss with colleagues in Oxford (particularly Paolo Perrone, who is running an adjoint school project on uncertainty in Markov categories, and some time ago Kwok Ho Cheung regarding his thesis work [13]).

Research supported by Clarendon Scholarship, ERC Consolidator Grant BLAST, and AFOSR Project FA9550-21-1-0038.

References

[1] N. Ackerman, C. Freer, Y. Kaddar, J. Karwowski, S. Moss, D. Roy, S. Staton, and H. Yang. Probabilistic programming interfaces for random graphs: Markov categories, graphons, and nominal sets. In *Proc. POPL 2024*, 2024.

[2] A. Adjé, O. Bouissou, J. Goubault-Larrecq, E. Goubault, and S. Putot. Static analysis of programs with imprecise probabilistic inputs. In *Proc. VSTTE 2013*, pages 22–47, 2013.

[3] A. Appel and V. Kosoy. Basic inframeasure theory. LessWrong, August 2020. `https://www.lesswrong.com/posts/YAa4qcMyoucRS2Ykr/basic-inframeasure-theory`.

[4] A. Appel and V. Kosoy. Inframeasures and domain theory. LessWrong, March 2021. `https://www.lesswrong.com/posts/vrbidMiczaoHBhZGp/inframeasures-and-domain-theory`.

[5] T. Barker. A monad for randomized algorithms. In *Proc. MFPS 2016*, 2016.

[6] J. Bernoulli. *Ars conjectandi, opus posthumum.* Thurneysen, 1713.

[7] F. Bonchi, A. Sokolova, and V. Vignudelli. The theory of traces for systems with nondeterminism and probability. In *Proc. LICS 2019*, 2019.

[8] F. Bonchi, A. Sokolova, and V. Vignudelli. Presenting convex sets of probability distributions by convex semilattices and unique bases. In *Proc. CALCO 2021*, CALCO 2021.

[9] G. Boudol and I. Castellani. Flow models of distributed computations: three equivalent semantics for CCS. *Inform. Comput.*, 114:247–314, 1994.

[10] D. Braithwaite, J. Hedges, and T. St Clere Smithe. The compositional structure of Bayesian inference. In *Proc. MFCS 2023*, 2023.

[11] A. Carboni, S. Lack, and R. F. C. Walters. Introduction to extensive and distributive categories. *J. Pure Appl. Algebra*, 84(2):145–158, 1993.

[12] G. L. Cattani, M. P. Fiore, and G. Winskel. A theory of recursive domains with applications to concurrency. In *Proc. LICS 1998*, 1998.

[13] K.-H. Cheung. *Distributive Interaction of Algebraic Effects.* PhD thesis, University of Oxford Department of Computer Science, 2017.

[14] J. R. B. Cockett. Introduction to distributive categories. *Math. Struct. Comput. Sci.*, 3(3):277–307, 1993.

[15] S. Dash, Y. Kaddar, H. Paquet, and S. Staton. Affine monads and lazy structures for Bayesian programming. In *Proc. POPL 2023*, 2023.

[16] D. 'davidad' Dalrymple. Safeguarded AI: constructing safety by design. ARIA Programme Thesis, January 2024. `https://www.aria.org.uk/wp-content/uploads/2024/01/ARIA-Safeguarded-AI-Programme-Thesis-V1.pdf`.

[17] B. Day. On closed categories of functors. In *Proc. Midwest Category Seminar IV*, volume 137 of *Lect. Notes Math.* Springer, 1969.

[18] B. Fong, D. Spivak, and R. Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '19. IEEE Press, 2021.

[19] T. Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Adv. Math.*, 370, 2020.

[20] T. Fritz and P. Perrone. A criterion for Kan extensions of lax monoidal functors. arxiv:1809.10481, 2018.

[21] T. Fritz and P. Perrone. A probability monad as the colimit of spaces of finite samples. *Theory and Applications of Categories*, 34, 2019.

[22] M. Gaboardi, S. Katsumata, D. Orchard, and T. Sato. Graded hoare logic and its categorical semantics. In *Proc. ESOP 2021*, 2021.

[23] B. Gavranovic. *Fundamental Components of Deep Learning: A category-theoretic approach.* PhD thesis, Strathclyde University, 2024. arXiv:2403.13001.

[24] J. Goubault-Larrecq. Continuous previsions. In *Proc. CSL 2007*, 2007.

[25] J. Goubault-Larrecq. Prevision domains and convex powercones. In *Proc. FOSSACS 2008*, 2008.

[26] A. Goy and D. Petrisan. Combining probabilistic and non-deterministic choice via weak distributive laws. In *LICS 2020*, 2020.

[27] C. Hermida and R. Tennent. Monoidal indeterminates and categories of possible worlds. *Theoretical Computer Science*, 430, 2012.

[28] C. Heunen, O. Kammar, S. Staton, and H. Yang. A convenient category for higher-order probability theory. In *Proc. LICS 2017*, 2017.

[29] P. J. Huber. *Robust statistics.* Wiley, 1981.

[30] B. Jacobs. Coalgebraic trace semantics for combined possibilitistic and probabilistic systems. In *Proc. CMCS 2008*, 2008.

[31] B. Jacobs. From probability monads to commutative effectuses. *J. Log. Algebr. Methods Program.*, 94:200–237, 2018.

[32] B. Jacobs. Structured probabilistic reasoning. Available from the author's homepage, July 2023. Draft book.

[33] B. Jacobs, A. Kissinger, and F. Zanasi. Causal inference by string diagram surgery. In *Proc. FOSSACS 2019*, 2019.

[34] X. Jia, B. Lindenhovius, M. W. Mislove, and V. Zamdzhiev. Commutative monads for probabilistic programming languages. In *Proc. LICS 2021*, pages 1–14, 2021.

[35] C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *Fourth Annual Symposium on Logic in Computer Science*, pages 186–195, June 1989.

[36] O. Kammar and G. D. Plotkin. Algebraic foundations for effect-dependent optimisations. In *Proc. POPL 2012*, pages 349–360, 2012.

[37] S. Katsumata. Parametric effect monads and semantics of effect systems. In *Proc. POPL 2014*, 2014.

[38] K. Keimel. Topological cones: Foundations for a domain theoretical semantics combining probability and nondeterminism. In *MFPS 2005*, 2005.

[39] K. Keimel and G. D. Plotkin. Mixed powerdomains for probability and nondeterminism. *Log. Methods Comput. Sci.*, 13, 2017.

[40] G. M. Kelly. *Basic Concepts of Enriched Category Theory.* CUP, 1982.

[41] F. H. Knight. *Risk, uncertainty and profit.* Houghton Mifflin, 1921.

[42] A. Kock. Strong functors and monoidal monads. *Arch. Math*, 23(1):113–120, Dec. 1972.

[43] A. Kock. Commutative monads as a theory of distributions. *Theory Appl. Categ.*, 26(4):97–131, 2012.

[44] V. Kosoy and A. Appel. Infra-Bayesian physicalism: A formal theory of naturalized induction. AI Alignment Forum, November 2021. Available at `https://www.alignmentforum.org/posts/gHgs2e2J5azvGFatb/infra-bayesian-physicalism-a-formal-theory-of-naturalized`.

[45] S. Kura. Graded algebraic theories. In *Proc. FOSSACS 2020*, 2020.

[46] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *Proc. LICS 2013*, 2013.

[47] E. D. Lavore and M. Román. Evidential decision theory via partial Markov categories. In *Proc. LICS 2023*, 2023.

[48] P. B. Levy. Amb breaks well-pointedness, ground amb doesn't. In *Proc. MFPS 2007*, 2007.

[49] P. B. Levy. Locally graded categories. Slides available from the author's webpage, 2019.

[50] P. B. Levy, J. Power, and H. Thielecke. Modelling environments in call-by-value programming languages. *Inform. Comput.*, 185, 2003.

[51] A. K. Lew, M. F. Cusumano-Towner, B. Sherman, M. Carbin, and V. K. Maninghka. Trace types and denotational semantics for sound programmable inference in probabilistic languages. In *Proc. POPL 2020*, 2020.

[52] S. Liang, P. Hudak, and M. Jones. Monad transformers and modular interpreters. In *Proc. POPL 1995*, 1995.

[53] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1998.

[54] D. McDermott and T. Uustalu. Flexibly graded monads and graded algebras. In *Proc. MPC 2022*, 2022.

[55] A. K. McIver and C. Morgan. Partial correctness for probabilistic demonic programs. *Theoret. Comput. Sci.*, 266:513–541, 2001.

[56] M. Mio. Upper-expectation bisimilarity and Łukasiewicz mu-calculus. In *Proc. FoSSaCS 2014*, 2014.

[57] M. Mio, R. Sarkis, and V. Vignudelli. Combining nondeterminism, probability, and termination: Equational and metric reasoning. In *Proc. LICS 2021*, 2021.

[58] M. Mio and V. Vignudelli. Monads and quantitative equational theories for nondeterminism and probability. In *Proc. CONCUR 2020*, 2020.

[59] M. W. Mislove, J. Ouaknine, and J. Worrell. Axioms for probability and nondeterminism. In *Proc. EXPRESS 2003*, 2003.

[60] E. Moggi. Notions of computation and monads. *Information and Computation*, 1991.

[61] D. Orchard, P. Wadler, and H. E. III. Unifying graded and parameterised monads. In *Proc. MSFP 2020*, 2020.

[62] P. Perrone. Markov categories and entropy. *IEEE Transactions on Information Theory*, 2023.

[63] D. Petrisan and R. Sarkis. Semialgebras and weak distributive laws. In *Proc. MFPS 2021*, 2021.

[64] A. M. Pitts. *Nominal Sets: names and symmetry in computer science*. CUP, 2013.

[65] J. Power. Generic models for computational effects. *Theor. Comput. Sci.*, 364(2):254–269, 2006.

[66] D. Shiebler. Categorical Stochastic Processes and Likelihood. *Compositionality*, 3, Apr. 2021.

[67] A. Simpson. Probability sheaves and the Giry monad. In *Proc. CALCO 2017*, 2017.

[68] S. Staton. Commutative semantics for probabilistic programming. In *Proc. ESOP 2017*, pages 855–879, 2017.

[69] D. Stein and R. Samuelson. A categorical treatment of open linear systems, March 2024. arxiv:2403.03934.

[70] D. Stein and S. Staton. Compositional semantics for probabilistic programs with exact conditioning. In *Proc. LICS 2021*, 2021.

[71] Z. Tavares, J. Koppel, X. Zhang, R. Das, and A. Solar-Lezama. A language for counterfactual generative models. In *Proc. ICML 2021*, pages 10173–10182, 2021.

[72] M. Vákár, O. Kammar, and S. Staton. A domain theory for statistical probabilistic programming. *Proc. ACM Program. Lang.*, 3(POPL):36:1–36:29, 2019.

[73] J.-W. van de Meent, B. Paige, H. Yang, and F. Wood. An introduction to probabilistic programming. *arXiv e-print 1809.10756*, 2018.

[74] G. van Heerdt, J. Hsu, J. Ouaknine, and A. Silva. Convex language semantics for nondeterministic probabilistic automata. In *Proc. ICTAC 2018*, 2018.

[75] D. Varacca and G. Winskel. Distributing probability over non-determinism. *Math. Structures Comput. Sci.*, pages 87–113, 2006.

[76] Various authors. Eckmann-Hilton argument. nLab, April 2024. `https://ncatlab.org/nlab/show/Eckmann-Hilton+argument#variation`.

[77] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, 1991.

[78] R. Wood. *Indical methods for relative categories*. PhD thesis, Dalhousie University, 1976.