



Compositional Imprecise Probability

A Solution from Graded Monads and Markov Categories

JACK LIELL-COCK, University of Oxford, United Kingdom

SAM STATON, University of Oxford, United Kingdom

Imprecise probability is concerned with uncertainty about which probability distributions to use. It has applications in robust statistics and machine learning.

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.

Imprecise probability can be modelled in various ways, with the leading monad-based approach using convex sets of probability distributions. This model is not fully compositional because the monad involved is not commutative, meaning it does 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 and relate it with the earlier monadic approach, proving that we obtain tighter bounds on the uncertainty.

CCS Concepts: • Theory of computation → Semantics and reasoning; • Mathematics of computing → Probability and statistics.

Additional Key Words and Phrases: imprecise probability, probabilistic programming, graded monads

ACM Reference Format:

Jack Liell-Cock and Sam Staton. 2025. Compositional Imprecise Probability: A Solution from Graded Monads and Markov Categories. *Proc. ACM Program. Lang.* 9, POPL, Article 54 (January 2025), 31 pages. <https://doi.org/10.1145/3704890>

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; a point such as *p* on the edge between *g* and *b* represents the probability distribution where *r* is impossible (Figure 1a).

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 statistical robustness (e.g. [50, 119]), economics (e.g. [5, 12, 13, 22, 94, 118]), and engineering (e.g. [10, 38, 72, 117]). In machine learning, there has been a recent integration of imprecise probability in Bayesian

Authors' Contact Information: Jack Liell-Cock, University of Oxford, Oxford, United Kingdom, jack.liell-cock@keble.ox.ac.uk; Sam Staton, University of Oxford, Oxford, United Kingdom, sam.staton@cs.ox.ac.uk.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).

ACM 2475-1421/2025/1-ART54

<https://doi.org/10.1145/3704890>

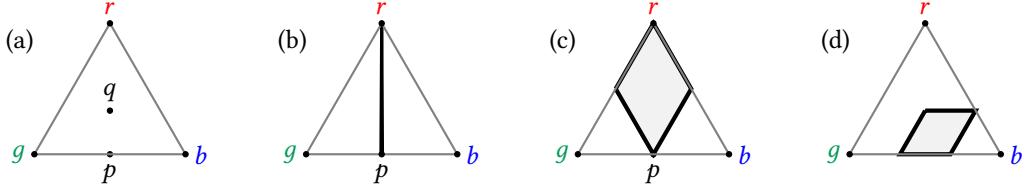


Fig. 1. (a) Five probabilities over the three-point set $\{r, g, b\}$ illustrated as points in the triangle: the three extreme points are the corners; p is the equal odds chance between b and g ; q is the equal odds chance between all three points. (b) A line indicating a convex region between r and 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 [119, Ex. 7.3].

learning (e.g. [19]), reinforcement learning (e.g. [93, 121]), conformal prediction (e.g. [57, 110]), infrabayesianism (e.g. [6, 7, 70]), and the foundations of safe AI [31, 32].

There is already a body of work on semantics models of programming languages with imprecise probability [3, 45–47, 52, 62, 63, 85–89, 96, 115, 116]; we discuss this further in §7. Our contribution is to investigate new models that support our compositional desiderata (§1.1) by naming the non-deterministic choices (§1.2), and to investigate syntax for this graded programming language to streamline equational reasoning. We show that this gives tighter uncertainty bounds than earlier work (Theorem 5.6) and that it is a maximal approach (Theorem 6.2).

1.1 Desiderata: A Language for Imprecise Probability with Compositional Reasoning

To focus on the essence of imprecise probability, we look here at a minimal language, that is, a first-order functional language without first-class functions or recursion. Rather, we have if-then-else statements, sequencing with immutable variable assignment (like [77, 91]), and the following two commands, which both return a boolean value:

- **bernoulli**: a fair Bernoulli choice [14] which draws a ball from some urn containing two balls labelled ‘true’ and ‘false’ and replaces it;
- **knight**: a Knightian choice [65] 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, except to know that the urn is not empty. (These ‘Knightian urns’ are fresh each time, so they can each be used only once. That is, we are not interested in using multiple draws and frequencies to predict their contents.)

For example, consider the following two programs.

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

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

else (if y then r else 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 guaranteed true and y is guaranteed 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. [40, 53, 69]), statistics (e.g. [18, 35, 56]), and probabilistic programming (e.g. [28, 58, 106]), and the connections between them (e.g. [109]). These desiderata are formalized in §2.

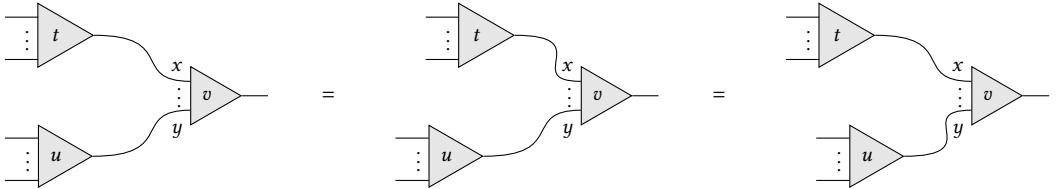
DESIDERATUM 1. *The language should be commutative:*

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

and *affine*:

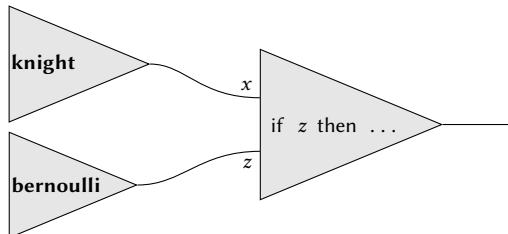
$$(x \leftarrow t ; u) = u \quad (\text{if } x \text{ is not free in } u).$$

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



is not ambiguous. Desideratum 1 is also considered a fundamental aspect of the abstract axiomatization of probability [66] with commutativity amounting to Fubini's theorem in the measure-theoretic setting [69].

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:*

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

where x is not free in b .

```


$$\begin{aligned}
& x \leftarrow \mathbf{knight} ; z \leftarrow \mathbf{bernoulli} ; \\
& \text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{) else (if } x \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Desideratum 1 (commutativity)} \\
& z \leftarrow \mathbf{bernoulli} ; x \leftarrow \mathbf{knight} ; \\
& \text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{) else (if } x \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Desideratum 2} \\
& z \leftarrow \mathbf{bernoulli} ; \text{if } z \text{ then (} x \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } g \text{)} \\
& \qquad \qquad \qquad \text{else (} x \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Alpha renaming} \\
& z \leftarrow \mathbf{bernoulli} ; \text{if } z \text{ then (} x \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } g \text{)} \\
& \qquad \qquad \qquad \text{else (} y \leftarrow \mathbf{knight} ; \text{if } y \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Desideratum 1 (affine)} \\
& z \leftarrow \mathbf{bernoulli} ; \text{if } z \text{ then (} x \leftarrow \mathbf{knight} ; y \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } g \text{)} \\
& \qquad \qquad \qquad \text{else (} x \leftarrow \mathbf{knight} ; y \leftarrow \mathbf{knight} ; \text{if } y \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Desideratum 2} \\
& z \leftarrow \mathbf{bernoulli} ; x \leftarrow \mathbf{knight} ; y \leftarrow \mathbf{knight} ; \\
& \text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{) else (if } y \text{ then } r \text{ else } b \text{)} \\
= & \text{-- Desideratum 1 (commutativity)} \\
& x \leftarrow \mathbf{knight} ; y \leftarrow \mathbf{knight} ; z \leftarrow \mathbf{bernoulli} ; \\
& \text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{) else (if } y \text{ then } r \text{ else } b \text{)}
\end{aligned}$$


```

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

This hoisting equation follows from the standard beta and eta laws for if-then-else statements, which are the basic building blocks for equational reasoning. In models of effects following Moggi’s monadic approach [90, 91], this property follows from the base category being distributive, which has long been argued as desirable (e.g. [24]) and thus assumed in all systems following this approach (e.g. [27, 106]). Nevertheless, we briefly discuss the consequences of dropping Desideratum 2 in §6.1.

One earlier approach to a semantic study of a language like this is provided by a convex powerset of distributions monad (e.g. [15, 16, 47, 52, 63, 87, 88]). 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:

$$z \leftarrow \mathbf{bernoulli} ; \text{if } z \text{ then (} x \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } g \text{)} \\
\qquad \qquad \qquad \text{else (} x \leftarrow \mathbf{knight} ; \text{if } x \text{ then } r \text{ else } b \text{)}$$

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 a different urn or the same urn. Perhaps there is one Knightian draw 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 (§4). To do this, we rewrite Example 1.1 by annotating the only Knightian choice with the name a_1 :

$$x \leftarrow \mathbf{knight}(a_1) ; z \leftarrow \mathbf{bernoulli} ; \\
\text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{)}$$

```
else (if  $x$  then  $r$  else  $b$ )
```

We think of this program as giving rise to the convex set in Figure 1b. Now when we try to follow the same equational derivation as in Figure 2, the third program becomes:

```
 $z \leftarrow \text{bernoulli} ; \text{if } z \text{ then } (x \leftarrow \text{knight}(a_1) ; \text{if } x \text{ then } r \text{ else } g)$ 
else ( $x \leftarrow \text{knight}(a_1) ; \text{if } x \text{ then } r \text{ else } b$ )
```

This program is equivalent even if we were to alpha rename x to y on the second branch because they use the same Knightian choice, i.e. one with the same name.

```
 $z \leftarrow \text{bernoulli} ; \text{if } z \text{ then } (x \leftarrow \text{knight}(a_1) ; \text{if } x \text{ then } r \text{ else } g)$  (1)
else ( $y \leftarrow \text{knight}(a_1) ; \text{if } y \text{ then } r \text{ else } b$ )
```

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

```
 $z \leftarrow \text{bernoulli} ; \text{if } z \text{ then } (x \leftarrow \text{knight}(a_1) ; \text{if } x \text{ then } r \text{ else } g)$  (2)
else ( $y \leftarrow \text{knight}(a_2) ; \text{if } y \text{ then } r \text{ else } b$ )
```

From this program, we can continue the derivation as in Figure 2, using affinity and Desideratum 1 to get:

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

which is intuitively what Example 1.2 describes, and gives rise to the convex set in Figure 1c. Hence, by explicitly labelling the Knightian choices, we have carved out a distinction between Examples 1.1 and 1.2.

The idea of naming non-deterministic choices appears in work outside probability (e.g. proved transitions [17]) and probabilistic choices are often named in practical probabilistic programming [114, §6.2] which has already been explored using graded monads [78]. More generally, intensionality in non-determinism is known to be a profitable perspective (e.g. [21, 74]).

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. [91]), namely the reader transformer (e.g. [80]) of the finite distributions monad (e.g. [55, Ch. 2]):

$$T_{2^A}(X) = [2^A \Rightarrow D(X)] \quad (3)$$

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 :

$$\llbracket t \rrbracket_{2^A} = \{p \gg_D t \mid p \in D(2^A)\} \subseteq D(X).$$

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 must account for name clashes. This depends on how we interpret the set A in (3).

We resolve this issue by regarding the monad (3) as a graded monad [60, 61, 92]. This is closely related to the ‘para’ construction (e.g. [39, 48]). The grading is crucial for defining the composition of programs, with the main steps being:

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

$$\gg_T : T_{2^A}(X) \times (X \Rightarrow T_{2^B}(Y)) \rightarrow T_{2^{A \uplus B}}(Y)$$

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

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

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: $\llbracket t \rrbracket_{2^A} = \llbracket T_{2^\iota}(t) \rrbracket_{2^B}$.

The key element is that the injective renaming ι induces a surjection $2^\iota : 2^B \rightarrow 2^A$ between the spaces of Knightian choices. We abstract and generalize by allowing arbitrary surjections $2^B \rightarrow 2^A$, further by allowing sets other than 2^A , and further still by allowing surjective stochastic maps rather than surjections. Then quotienting via these regradings leads to the canonical connection of the monad with convex sets of distributions (Proposition 5.5).

In the rest of the paper, we start from the setting of Markov categories (§2) rather than monads, since they are a minimal framework suited to our minimal language. We make the connection with monads in §3, introducing the formalism for named choices in §4.

As an aside, we note that probability monads too can often be regarded as sort-of reader monads (e.g. [11, 28, 34, 103, 104, 111, 113]), since probability distributions $D(X)$ can be described by random variables $\Omega \rightarrow X$, 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.

1.2.3 An Algebraic Perspective. There is a large literature on combining probability with non-determinism, some of which takes the perspective of algebraic theories. In the context of algebraic effects, the commands **bernoulli** and **knight** are generic effects [99] for probabilistic and Knightian choice, with which we can define two binary operators:

$$(t +_{0.5} u) \stackrel{\text{def}}{=} \text{if } \mathbf{bernoulli} \text{ then } t \text{ else } u \quad (t \oplus u) \stackrel{\text{def}}{=} \text{if } \mathbf{knight} \text{ then } t \text{ else } u$$

We provide a more detailed discussion of this perspective in §7.2 and §7.3. In brief, Desideratum 1 states that these operators commute with each other (12) and are idempotent (13). Desideratum 2 is always assumed in algebraic effects as a consequence of the algebraicity property. Symmetry of the operators is also desirable but is incompatible with the desiderata (17). Our graded approach to Knightian choice (§1.2.2) connects to graded algebraic theories [73, §3.1] which amounts to

naming the non-deterministic binary operators and allows for symmetry of the operators up to regrading (22).

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 5.6 (§5): *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 convex powerset of distributions monad (e.g. [15, 16, 47, 52, 63, 87, 88]). We have an ‘op-lax’ functor

$$R : \mathbf{Imp} \rightarrow \mathbf{Kl}(\mathbf{CP}).$$

from our locally graded category \mathbf{Imp} (§4.3) to the Kleisli category of the convex powerset of distributions monad \mathbf{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 composition using the Kleisli category of the \mathbf{CP} monad (see also the Example in §5.4).

Note that this could not be a proper functor because we would then have a quotient theory in violation of the maximality theorem (Theorem 6.2). But an op-lax functor is beneficial as an interpretation of giving a tighter bound.

Theorem 6.2 (§6): *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.

2 Markov Categories and Programming Language Syntax

Markov categories have been proposed as a synthetic foundation of probability [40], but they also form a basic framework for equational reasoning about probabilistic programming (e.g. [2, 109]). As we show, distributive Markov categories precisely capture our Desiderata (§1.1). Markov categories are connected to monad-based approaches too, as we discuss in §3.

We begin by recalling notions of Markov categories and how we may interpret programming language syntax in them (§2.1), and then generalize them to the locally graded setting (§2.2).

Definition 2.1. A *monoidal category* is a category C equipped with a functor $\otimes : C \times C \rightarrow C$ 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. [81]). It is strict if the isomorphisms are equalities. A *symmetric* monoidal category is moreover equipped with isomorphisms $\sigma_{X,Y} : X \otimes Y \cong Y \otimes X$ that satisfy $\sigma_{Y,X} = \sigma_{X,Y}^{-1}$ and additional 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 $X \rightarrow I$ for all X .

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

2.1 Markov Categories

Definition 2.2. [40] A *Markov category* is a semicartesian category such that every object is equipped with a commutative comonoid structure, that is, a map $\text{copy}_X : X \rightarrow X \otimes X$ that is symmetric and associative and has the terminal map $X \rightarrow I$ as a unit.

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

A *distributive* Markov category [2] is a Markov category that has coproducts such that the canonical maps $X \otimes Z + Y \otimes Z \rightarrow (X + Y) \otimes Z$ are isomorphisms and the coproduct injections $X \rightarrow X + Y \leftarrow Y$ are deterministic.

A typical example of a distributive Markov category is the category **FinStoch** of stochastic matrices (Definition 4.2).

An ordinary distributive category [20, 24] 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. [109]). 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 $t : \Gamma \rightarrow B$ is regarded as a term $\Gamma \vdash t : B$. We note

$$(t, u) \quad \text{for} \quad \Gamma \xrightarrow{\text{copy}} \Gamma \otimes \Gamma \xrightarrow{t \otimes u} A \otimes B \quad (4)$$

$$x \leftarrow t ; u \quad \text{for} \quad \Gamma \xrightarrow{\text{copy}} \Gamma \otimes \Gamma \xrightarrow{\Gamma \otimes t} \Gamma \otimes A \xrightarrow{u} B \quad (5)$$

$$\text{if } t \text{ then } u \text{ else } v \quad \text{for} \quad \Gamma \xrightarrow{\text{copy}} \Gamma \otimes \Gamma \xrightarrow{t \otimes \Gamma} (1 + 1) \otimes \Gamma \cong \Gamma + \Gamma \xrightarrow{[u, v]} B \quad (6)$$

In this way, given interpretations of **bernoulli** and **knight** as morphisms $1 \rightarrow \text{Bool}$, we can interpret the programs from Examples 1.1 and 1.2.

2.2 Graded Markov Categories

Definition 2.3. Let \mathbb{G} be a 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 $\gamma \in \mathbb{G}$ a set $\mathbf{C}_\gamma(X, Y)$ of morphisms, agreeing with \mathbf{C}_I when $\gamma = I$;
- for each morphism $f : \varepsilon \rightarrow \gamma$ in \mathbb{G} , a function $f^\star : \mathbf{C}_\gamma(X, Y) \rightarrow \mathbf{C}_\varepsilon(X, Y)$, ‘regrading’;
- a family of maps $\circ : \mathbf{C}_\gamma(X, Y) \times \mathbf{C}_\varepsilon(Y, Z) \rightarrow \mathbf{C}_{\gamma \otimes \varepsilon}(X, Z)$, ‘composition’;
- a family of maps $\otimes : \mathbf{C}_\gamma(X, X') \times \mathbf{C}_\varepsilon(Y, Y') \rightarrow \mathbf{C}_{\gamma \otimes \varepsilon}(X \otimes Y, X' \otimes Y')$, ‘monoidal product’

all such that composition is natural (regrading followed by composition is equal to composition followed by regrading) and associative up to the associativity of \mathbb{G} (see e.g. [120, §1.2], [76], [44, App. B]), monoidal product of morphisms is also natural and associative and unital up-to the associativity and unit structure of \mathbb{G} , the monoidal interchange law holds up to grade symmetry, and the induced function

$$\mathbf{C}_\gamma(X + Y, Z) \rightarrow \mathbf{C}_\gamma(X, Z) \times \mathbf{C}_\gamma(Y, Z) \quad (7)$$

is a bijection (e.g. [120, p. 35] for the dual product case).

To illustrate what we mean by ‘up to’ regrading in this definition in more detail, for instance, the interchange law says that

$$((h_1 \circ g_1) \otimes (h_2 \circ g_2)) = f^\star((h_1 \otimes h_2) \circ (g_1 \otimes g_2))$$

where $g_i \in \mathbf{C}_{\gamma_i}(X_i, Y_i)$, $h_i \in \mathbf{C}_{\varepsilon_i}(Y_i, Z_i)$, and $f : (\gamma_1 \otimes \varepsilon_1) \otimes (\gamma_2 \otimes \varepsilon_2) \cong (\gamma_1 \otimes \gamma_2) \otimes (\varepsilon_1 \otimes \varepsilon_2)$ is the coherence isomorphism in \mathbb{G} .

The reader familiar with enriched category theory may see Proposition 3.5 for a brief equivalent definition.

See Proposition 4.3 for our example of a graded Markov category. We note that since \mathbb{G} is semicartesian, there are canonical projections $\gamma \otimes \varepsilon \rightarrow \gamma$, and so we can regard any morphism at grade γ as a morphism at grade $\gamma \otimes \varepsilon$.

2.3 Graded Programming Syntax

We propose to use programming language syntax for composition in a graded distributive Markov category, as in a distributive Markov category. Our syntax and interpretation are similar to other languages for grades and effect systems (e.g. [82, §2.3]). The objects of the category are again regarded as types with Bool and contexts Γ interpreted as before. Then a morphism at grade γ , e.g. $t \in \mathbf{C}_\gamma(\Gamma, B)$, is regarded as a term at grade γ , written $\Gamma \vdash t : B \& \gamma$. We can build a simple internal language for the graded category using typing rules such as the following:

$$\frac{}{\Gamma, x : A \vdash x : A \& I} \quad \frac{\Gamma \vdash u : A \& \gamma \quad \Gamma, x : A \vdash t : B \& \varepsilon}{\Gamma \vdash x \leftarrow u; t : B \& \gamma \otimes \varepsilon}$$

$$\frac{\Gamma \vdash t : A \& \gamma \quad f : \varepsilon \rightarrow \gamma}{\Gamma \vdash c_f[t] : A \& \varepsilon} \quad \frac{\Gamma \vdash b : \text{Bool} \& \gamma \quad \Gamma \vdash u : A \& \varepsilon \quad \Gamma \vdash t : A \& \varepsilon}{\Gamma \vdash \text{if } b \text{ then } u \text{ else } t : A \& \gamma \otimes \varepsilon}$$

The notation $c_f[t]$ is a regrading or coercion, corresponding to the functorial action on hom-sets $f^* : \mathbf{C}_\gamma(\Gamma, A) \rightarrow \mathbf{C}_\varepsilon(\Gamma, A)$ in the graded Markov category.

There are also typing rules for product types and sums more generally, but since we won't need them in examples, we omit them here. We do use a three-element type in the illustrations, $\text{Three} = 1 + 1 + 1$, with constructors

$$\Gamma \vdash r : \text{Three} \& I \quad \Gamma \vdash g : \text{Three} \& I \quad \Gamma \vdash b : \text{Three} \& I$$

For the language of imprecise probability with named Knightian choices, we suppose that there is a grade $\widehat{a} \in \mathbb{G}$ that represents the grading by a single name a . We then consider the following commands:

$$\Gamma \vdash \mathbf{bernoulli} : \text{Bool} \& I \quad \Gamma \vdash \mathbf{knight}(a) : \text{Bool} \& \widehat{a}$$

These will be interpreted via specific morphisms in $\mathbf{C}_I(I, \text{Bool})$ and $\mathbf{C}_{\widehat{a}}(I, \text{Bool})$ respectively.

2.3.1 Example and Regrading by Coherence Isomorphisms. We then have the following derivable type, for a program from §1.2:

$$\begin{aligned} \Gamma \vdash x \leftarrow \mathbf{knight}(a_1) ; y \leftarrow \mathbf{knight}(a_2) ; z \leftarrow \mathbf{bernoulli} ; \\ \text{if } z \text{ then (if } x \text{ then } r \text{ else } g \text{) else (if } y \text{ then } r \text{ else } b \text{) : Three} \& (\widehat{a}_1 \otimes \widehat{a}_2 \otimes I \otimes I \otimes I) \end{aligned}$$

We can regrade to $\widehat{a}_1 \otimes \widehat{a}_2$ via the coherence isomorphism $(\widehat{a}_1 \otimes \widehat{a}_2) \cong (\widehat{a}_1 \otimes \widehat{a}_2 \otimes I \otimes I \otimes I)$ in \mathbb{G} .

In some situations, \mathbb{G} can be chosen so that these coherence isomorphisms are identity maps so this trivial regrading can be omitted. For example, in many situations in the literature, \mathbb{G} is a partially ordered monoid. In §4.2 we suggest choices for \mathbb{G} that avoid coherence regradings.

2.3.2 Equational Reasoning. We have the following equational reasoning principles. The first is associativity of composition, familiar from the monadic metalanguage [90, 91]. The second and third are commutativity and affine laws (Desiderata 1), and finally the hoisting law (Desiderata 2).

Since we write ' $\Gamma \vdash t : A \& \gamma$ ' for a morphism $t \in \mathbf{C}_\gamma(\Gamma, A)$, we also write ' $\Gamma \vdash t \equiv u : A \& \gamma$ ' to indicate that the morphisms t and u are equal.

PROPOSITION 2.4. *In any graded distributive Markov category, the following equational reasoning principles are valid:*

(1) *(Associativity)*

If $\Gamma \vdash t : A \& \gamma \quad \Gamma, x : A \vdash u : B \& \varepsilon \quad \Gamma, y : B \vdash v : C \& \zeta$
 then $\Gamma \vdash x \leftarrow t ; y \leftarrow u ; v \equiv c_{\alpha^{-1}}[y \leftarrow (x \leftarrow t ; u) ; v] : C \& \gamma \otimes (\varepsilon \otimes \zeta)$
 where $\alpha : (\gamma \otimes \varepsilon) \otimes \zeta \rightarrow \gamma \otimes (\varepsilon \otimes \zeta)$ is the associativity coherence isomorphism in \mathbb{G} .

(2) *(Commutativity)*

If $\Gamma \vdash t : A \& \gamma \quad \Gamma \vdash u : B \& \varepsilon \quad \Gamma, x : A, y : B \vdash u : C \& \zeta$
 then $\Gamma \vdash c_\alpha[x \leftarrow t ; y \leftarrow u ; v] \equiv c_{(\sigma \otimes \zeta) \circ \alpha}[y \leftarrow u ; x \leftarrow t ; v] : C \& (\gamma \otimes \varepsilon) \otimes \zeta$
 where $\sigma : \gamma \otimes \varepsilon \rightarrow \varepsilon \otimes \gamma$ is the symmetry of the monoidal structure in \mathbb{G} .

(3) *(Weakening)*

If $\Gamma \vdash t : A \& \gamma \quad \Gamma \vdash u : B \& \varepsilon$
 then $\Gamma \vdash x \leftarrow t ; u \equiv c_{\pi_2}[u] : B \& \gamma \otimes \varepsilon$
 where $x \notin \Gamma$, and $\pi_2 : \gamma \otimes \varepsilon \rightarrow \varepsilon$ is the canonical projection map from the semicartesian structure of \mathbb{G} .

(4) *(Hoisting)*

If $\Gamma \vdash b : \text{Bool} \& \gamma \quad \Gamma \vdash t : A \& \varepsilon \quad \Gamma, x : A \vdash u : B \& \zeta \quad \Gamma, x : A \vdash v : B \& \zeta$
 then $\Gamma \vdash c_\alpha[\text{if } b \text{ then } x \leftarrow t ; u \text{ else } x \leftarrow t ; v] \equiv c_{(\sigma \otimes \zeta) \circ \alpha}[x \leftarrow t ; \text{if } b \text{ then } u \text{ else } v] : B \& (\gamma \otimes \varepsilon) \otimes \zeta$

PROOF NOTE. By expanding the definitions of the syntactic notation in terms of composition in graded distributive Markov categories. \square

When the coherence isomorphisms c_f are identity morphisms, the regradings can be omitted. We give candidates for \mathbb{G} that allow for this simplification in §4.2. The weakening regrading (π_2) is also typically obvious and so we sometimes elide it. We leave a practical coercion inference algorithm to future work.

Figure 3 illustrates a program equational derivation in this setting, which is similar to the one in Figure 2, but now formally grounded.

These are not the only laws: there are also equational laws such as reflexivity, symmetry, transitivity, congruence, unit laws for the sequencing construction, and beta and eta laws for sum and product types. As we will see in §3, graded Markov categories can be understood in terms of enriched monads, and so we have the full power of an enriched version of the monadic metalanguage [90, 91]. There are no nuances involved in specifying the other laws, beyond the regrading coercions, so for the sake of brevity, we do not write them all out here.

3 Relating Graded Markov Categories and Graded Monads

In §2, we looked at categorical probability and probabilistic programming from the point of view of Markov categories. Monads provide a different view on this. Because this angle will be more familiar to some in the community, we now recall graded and relative monads (§3.1), illustrate their correspondence with Markov categories (Propositions 3.2–3.3), and relate them to notions from enriched category theory (§3.2). For brevity, in this section we focus on definitions and in the next section (§4) we focus on examples, rather than interleaving them.

```

z ← bernoulli ; if z then (x ← knight(a1) ; if x then r else g)
                     else (x ← knight(a2) ; if x then r else b)
= -- Alpha renaming
z ← bernoulli ; if z then (x ← knight(a1) ; if x then r else g)
                     else (y ← knight(a2) ; if y then r else b)
= -- Weakening
z ← bernoulli ; if z then (x ← knight(a1) ; y ← knight(a2) ; if x then r else g)
                     else (x ← knight(a1) ; y ← knight(a2) ; if y then r else b)
= -- Hoisting
z ← bernoulli ; x ← knight(a1) ; y ← knight(a2) ;
if z then (if x then r else g) else (if y then r else b)
= -- Commutativity
x ← knight(a1) ; y ← knight(a2) ; z ← bernoulli ;
if z then (if x then r else g) else (if y then r else b)

```

Fig. 3. A formal equational derivation of two programs with named Knightian choices from §1.2 using the principles of Proposition 2.4.

3.1 Monads and Graded Relative Affine Monads

It is well-established that notions of computation can be modelled by monads [91], including probabilistic and non-deterministic computation (already in [59, 90]). In this section, we recall the flavours of monads relevant to this work.

The emerging view is that *commutative affine monads* form abstract accounts of probability and non-determinism (e.g. [28, 53, 55, 69]). Meanwhile, *relative monads* restrict the domain of first-order computation (e.g. [9]), and *graded monads* classify the side-effects associated with a program (e.g. [61, 92]). We present these concepts in the Kleisli triple setting because this is more conducive to their use in programming languages.

Definition 3.1. A *strong monad* [68, 91] over a cartesian closed category \mathbf{C} is for each $X \in \mathbf{C}$ an object $T(X) \in \mathbf{C}$ and a morphism

$$\eta_X : 1_{\mathbf{C}} \rightarrow [X, T(X)],$$

and a family of morphisms (the Kleisli extensions)

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

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

$$\begin{aligned} f^* \circ \eta_X &= f, \\ (\eta_X)^* &= id_{T(X)}, \\ g^* \circ f^* &= (g \circ f)^*. \end{aligned} \tag{8}$$

Recall that this induces a functor $T : \mathbf{C} \rightarrow \mathbf{C}$ with action $(\eta_Y \circ -)^* : [X, Y] \rightarrow [T(X), T(Y)]$, writing \circ for the internal composition in a cartesian closed category; in fact, T is a \mathbf{C} -enriched functor because the action is on the internal homs. This action moreover induces a left-strength $s : X \times T(Y) \rightarrow T(X \times Y)$ by

$$X \times T(Y) \rightarrow [Y, X \times Y] \times T(Y) \rightarrow [T(Y), T(X \times Y)] \times T(Y) \rightarrow T(X \times Y),$$

where the first and last arrows use the unit and the counit of the closed structure, respectively.

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

$$\begin{array}{ccccc}
 & & T(X) \times T(Y) & & \\
 & \swarrow \hat{s} & & \searrow s & \\
 T(X \times T(Y)) & & & & T(T(X) \times Y) \\
 & \searrow s^* & & \swarrow \hat{s}^* & \\
 & & T(X \times Y) & &
 \end{array}$$

The strong monad is *affine* (e.g. [51, 67]) if the unique map $T(1) \rightarrow 1$ is an isomorphism. A typical example is where $\mathbf{C} = \mathbf{Set}$ and $T = D$ is the finite probability distribution monad (e.g. [55]).

Let \mathcal{J} be a category with finite products and consider a finite product preserving functor $J : \mathcal{J} \rightarrow \mathbf{C}$. A *relative strong monad* [4, 112] T on J is for each $X \in \mathcal{J}$ an object $T(X) \in \mathbf{C}$ and a morphism

$$\eta_X : 1_{\mathbf{C}} \rightarrow [J(X), T(X)],$$

and a family of Kleisli extensions

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

in \mathbf{C} such that (8) 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* (e.g. [61]) is for each $X \in \mathbf{C}$ a functor $T_{(-)}(X) : \mathbb{G}^{\text{op}} \rightarrow \mathbf{C}$ and a morphism

$$\eta_X : 1_{\mathbf{C}} \rightarrow [X, T_I(X)],$$

and a family of Kleisli extensions

$$(-)_{\gamma, \varepsilon}^* : [X, T_{\varepsilon}(Y)] \rightarrow [T_{\gamma}(X), T_{\gamma \otimes \varepsilon}(Y)]$$

in \mathbf{C} , natural in γ and ε , such that for f and g generalised elements of $[X, T_{\gamma}(Y)]$ and $[Y, T_{\varepsilon}(Z)]$, the following equations hold:

$$\begin{aligned}
 (f)_{I, \gamma}^* \circ \eta_X &= T_{\lambda}(Y) \circ f, \\
 (\eta_X)_{\varepsilon, I}^* &= T_{\rho}(X), \\
 (g)_{\zeta \otimes \gamma, \varepsilon}^* \circ (f)_{\zeta, \gamma}^* &= T_{\alpha}(Z) \circ ((g)_{\gamma, \varepsilon}^* \circ f)_{\zeta, \gamma \otimes \varepsilon}^*,
 \end{aligned} \tag{9}$$

where $\lambda : I \otimes \gamma \rightarrow \gamma$, $\rho : \varepsilon \otimes I \rightarrow \varepsilon$, and $\alpha : (\zeta \otimes \gamma) \otimes \varepsilon \rightarrow \zeta \otimes (\gamma \otimes \varepsilon)$ are the left unit, right unit, and associator of \mathbb{G} , respectively.

A *graded relative strong monad* T on J is for each $X \in \mathcal{J}$ a functor $T_{(-)}(X) : \mathbb{G}^{\text{op}} \rightarrow \mathbf{C}$ and a morphism

$$\eta_X : 1_{\mathbf{C}} \rightarrow [J(X), T_I(X)],$$

and a family of Kleisli extensions

$$(-)_{\gamma, \varepsilon}^* : [J(X), T_{\varepsilon}(Y)] \rightarrow [T_{\gamma}(X), T_{\gamma \otimes \varepsilon}(Y)]$$

in \mathbf{C} , natural in γ and ε , such that (9) holds for f and g generalised elements of $[J(X), T_{\gamma}(Y)]$ and $[J(Y), T_{\varepsilon}(Z)]$. The functoriality of T_{γ} and graded left-strength $s_{\gamma} : X \times T_{\gamma}(Y) \rightarrow T_{\gamma}(X \times Y)$ are similarly induced by the action $T_{\rho^{-1}}(Y) \circ (\eta_Y \circ -)_{\gamma, I}^* : [X, Y] \rightarrow [T_{\gamma}(X), T_{\gamma}(Y)]$. The monad is

commutative if \mathbb{G} is symmetric monoidal and the following diagram commutes, where $\sigma : \varepsilon \otimes \gamma \rightarrow \gamma \otimes \varepsilon$ is the symmetric coherence isomorphism of \mathbb{G} .

$$\begin{array}{ccc}
 & T_Y(X) \times T_\varepsilon(Y) & \\
 \hat{s}_Y \swarrow & & \searrow s_\varepsilon \\
 T_Y(X \times T_\varepsilon(Y)) & & T_\varepsilon(T_Y(X) \times Y) \\
 \downarrow (s_\varepsilon)^*_{Y,\varepsilon} & & \downarrow (\hat{s}_Y)^*_{\varepsilon,Y} \\
 T_{Y \otimes \varepsilon}(X \times Y) & \xrightarrow{T_\sigma(X \times Y)} & T_{\varepsilon \otimes Y}(X \times Y)
 \end{array}$$

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

PROPOSITION 3.2. *If T is a graded commutative affine relative monad on a distributive category, then its Kleisli category (e.g. [43, 73]) is a graded Markov category:*

- The objects are the same as \mathcal{J} ;
- The morphisms in $\text{Kl}(T)_Y(X, Y)$ are the morphisms $J(X) \rightarrow T_Y(Y)$ in \mathbf{C} ;
- Composition is via the Kleisli extension.

PROPOSITION 3.3. *Any graded Markov category induces a graded commutative affine relative monad, by*

- $\mathcal{J} = \mathbf{C}_{I,\text{det}}$, the distributive category of I -graded deterministic maps
- The underlying category is $\text{FP}(\mathcal{J}^{\text{op}}, \text{Set})$, the finite product-preserving contravariant presheaves on \mathcal{J} ;
- $J : \mathcal{J} \rightarrow \text{FP}(\mathcal{J}^{\text{op}}, \text{Set})$ is the Yoneda embedding;
- For $Y \in \mathcal{J}$, the functor $T_{(-)}(Y) : \mathbb{G}^{\text{op}} \rightarrow \text{FP}(\mathcal{J}^{\text{op}}, \text{Set})$ is given by

$$T_Y(Y)(X) = \mathbf{C}_Y(X, Y)$$

PROOF NOTE. In both cases, the proof amounts to expanding the definitions. The constructions are similar to [101, §7]. See also [2, Prop. 13] for the non-graded case. \square

(We conjecture that Propositions 3.2–3.3 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.)

3.2 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. [64]) 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 $\mathbf{C}(X, Y) \otimes \mathbf{C}(Y, Z) \rightarrow \mathbf{C}(X, Z)$ in \mathcal{V} . We can also define \mathcal{V} -enriched monoidal categories, by requiring the functor $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ to be \mathcal{V} -enriched (with $\mathbf{C} \times \mathbf{C}$ as in [64, §1.4]). 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* C_0 , which has the same objects but with a hom-set given by $C_0(X, Y) = \mathcal{V}(I, C(X, Y))$. This ordinary category inherits monoidal, limit and colimit structure from C .

Definition 3.4 (e.g. [95]). A \mathcal{V} -enriched Markov category is a \mathcal{V} -enriched symmetric monoidal category such that the monoidal unit is terminal: $C(X, I) \cong 1$, 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 $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$. This extends \mathbb{G} to a good ‘cosmos’ for enrichment since

- $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ embeds \mathbb{G} fully and faithfully (i.e. essentially as a full subcategory), via the Yoneda embedding $y(\gamma) = \mathbb{G}(-, \gamma)$.
- $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ has all limits and colimits, computed pointwise.
- $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ has a semicartesian structure such that the Yoneda embedding is a symmetric monoidal functor. This is given by Day convolution [33], and has the following universal property: for $F, G, H \in [\mathbb{G}^{\text{op}}, \mathbf{Set}]$, to give a natural transformation $F \otimes G \rightarrow H$ is to give a natural family of functions $F(\gamma) \times G(\varepsilon) \rightarrow H(\gamma \otimes \varepsilon)$.
- $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ is moreover monoidal closed.

PROPOSITION 3.5. *To give an $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ -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 $[\mathbb{G}^{\text{op}}, \mathbf{Set}]$ -enriched categories (e.g. [44, 76, 120]), and then translating Definition 3.4 across this correspondence to arrive at Definition 2.3. \square

The correspondence between graded monads and enriched monads is also well understood, including in the relative case (e.g. [8, 83]).

Aside. Recent work by Perrone [95] has considered enriched Markov categories to obtain an abstract view of the distance between probabilities, which allows for an abstract development of entropy. Their enriching category $\mathcal{V} = \text{Div}$ in [95] is indeed semicartesian. The full theory of enriched Markov categories perhaps deserves a more thorough analysis.

4 A Graded Markov Category for Imprecise Probability

We recall ordinary Markov categories for finite probability (§4.1). 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’, §4.3). We conclude this section with a worked example (§4.4). In the subsequent sections (§5–6) we relate this graded Markov category with convex sets of distributions.

4.1 Ordinary Markov Categories for Probability

We recall the Markov category **FinStoch** of finite sets and stochastic maps between them.

In what follows, for notational simplicity, we regard every finite set n with a given enumeration $n = \{a_1, \dots, a_{|n|}\}$. Also, we choose a particular singleton set 1 (empty product), and for every pair of finite sets, we choose a product set $m \times n$ and projections $m \leftarrow m \times n \rightarrow n$. (This will have cardinality $\#(m \times n) = \#m \times \#n$, but there are many isomorphic choices for the set.)

Definition 4.1. Let n be a finite set. A *probability vector* $p \in \mathbb{R}^n$ is an n -indexed sequence of non-negative numbers that sum to 1. We write $D(n)$ for the set of 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 $rp + (1 - r)q$ 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 $i \in n$, 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 $f \cdot p \in D(m)$. In fact, every function $D(n) \rightarrow D(m)$ that preserves convex structure arises from a stochastic matrix in this way. We call such a function a *convex map*.

Definition 4.2 (e.g. [40], Ex. 2.5). The category **FinStoch** of finite sets and stochastic maps has as objects finite sets, and morphisms $m \rightarrow n$ 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 products of finite sets. On morphisms, using the enumeration, we take the Kronecker product of matrices. It is semicartesian where the terminal object is 1 because there is a unique stochastic matrix with one row. This is moreover a Markov category, with $\text{copy}_n : n \rightarrow n \otimes n$ 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 disjoint union, i.e. by adding cardinalities, and with this view the coproduct of morphisms forms block matrices (concatenating the columns).

The monad view on **FinStoch** is as follows. First, we consider the embedding $J : \text{FinSet} \rightarrow \text{Set}$. We then regard D (Definition 4.1) as a J -relative monad $D : \text{FinSet} \rightarrow \text{Set}$, which is affine and commutative. In fact, there is an ordinary monad D' on **Set**, comprising finitely supported probability distributions (e.g. [55, Ch. 2]), and $D = D'J$. The distributive Markov category **FinStoch** can then be regarded as the Kleisli category for this relative monad.

4.2 Choices of Products

As noted above, the monoidal structure and copy maps of **FinStoch** depend on the choice of products of sets. Categorically this does not matter, because it is all the same up to canonical isomorphism. But since the grades enter the syntax of the language (§2.2), the choice of product can affect the syntax because identity maps can be omitted while isomorphisms cannot. (Subtleties about the choice of structure are not unusual in categorical logic, e.g. [25].) We consider two choices:

Numerals: Say that a *numeral* is a finite set of the form $\{1, \dots, n\} \subseteq \mathbb{N}$, for some natural number $n \in \mathbb{N}$.

- We choose the empty product $1 = \{1\}$ to be the unit numeral.
- For numerals M and N we choose the product to be the numeral for the integer multiple $|M| \times |N|$, with the projections coming from quotient and remainder. For all other finite sets, it does not matter which product structure we choose.

Since numerals are closed under this choice of product structure, we could cut down to the skeletal full subcategory of **FinStoch** whose objects are numerals. This has the property that the unitors and associator isomorphisms are identity maps, but the symmetry isomorphisms are not.

This choice of product is reminiscent of de Bruijn notation, where the indexing is derived from syntactic positions and reindexing is explicit, but weakening can be clumsy.

Name-respecting choices: Let \mathbb{A} be an infinite set of names. If $m \subseteq \mathbb{A}$ is a finite set of names then the set of functions 2^m describe true/false assignments to those names, as in Knightian choices.

- We let the empty product be the set of functions 2^\emptyset .
- If $m, n \subseteq \mathbb{A}$ are disjoint finite sets of names then we choose the product $2^m \times 2^n$ to be the set $2^{m \cup n}$. The projections $2^m \leftarrow 2^{m \cup n} \rightarrow 2^n$ are given by restricting the functions to subsets. For all other finite sets, it does not matter which product structure we choose.

This kind of choice of product structure has the property that for sets of the form 2^n , the unitor isomorphisms are identities; and the symmetry and associativity isomorphisms

$$2^m \times 2^n \cong 2^n \times 2^m \quad 2^m \times (2^n \times 2^p) \cong (2^m \times 2^n) \times 2^p$$

are identity maps for disjoint m, n and p .

This choice of product is reminiscent of nominal techniques (e.g. [98]). It is convenient for the language of §2.3 because under the disjointness assumption, all the coherence regradings in Proposition 2.4 are identity functions, and so can be omitted. Moreover, the regrading in the affine law can often be elided because it can be inferred from the grades. (Again, formal grade inference algorithms are beyond the scope of this paper.)

4.3 The Graded Markov Category \mathbf{Imp}

We now introduce our graded Markov category for imprecise probability, \mathbf{Imp} (Definition 4.5). We first introduce a general construction for graded Markov categories (Proposition 4.3). This is a variation on the ‘para’ construction [39], also called monoidal indeterminates [48]. Via the connections between Markov categories and commutative affine relative monads (§3.1), it is equivalently a graded version of the reader monad transformer [80] of the finite distributions monad.

$$T_\gamma(X) = [\gamma \Rightarrow D(X)] \quad (10)$$

PROPOSITION 4.3. *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*

$$\mathbf{C}_\gamma(X, Y) = \mathbf{C}(\gamma \otimes X, Y) \quad (\gamma \in \mathbb{G}).$$

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

$$f^\star(g) = g \circ (f \otimes X) \in \mathbf{C}_\varepsilon(X, Y).$$

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

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

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

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

Definition 4.4. A stochastic map $f \in \mathbf{FinStoch}(m, n)$ is *surjective* if for every $j \in \{1 \dots n\}$ there exists $i \in \{1 \dots m\}$ such that f_i is the Dirac distribution at j . In other words, the induced convex map $D(m) \rightarrow D(n)$ is surjective. Let $\mathbf{FinStoch}_{\text{Surj}}$ be the category of finite sets and surjective stochastic maps. This is a semicartesian monoidal subcategory of $\mathbf{FinStoch}$.

Definition 4.5. The graded distributive Markov category \mathbf{Imp} is the $\mathbf{FinStoch}_{\text{Surj}}$ -graded version of $\mathbf{FinStoch}$, according to Proposition 4.3.

Imp is equivalently the graded Kleisli category of the monad (10) over **FinSet** by the following isomorphisms, where regrading is done via the Kleisli extension of the finite distributions monad.

$$\mathbf{FinSet}(X, [\gamma \Rightarrow D(Y)]) \cong \mathbf{FinSet}(\gamma \otimes X, D(Y)) \cong \mathbf{FinStoch}(\gamma \otimes X, Y) = \mathbf{Imp}_\gamma(X, Y).$$

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

- For binary probabilistic choice with bias 0.5, we consider the morphism **bernoulli** $\in \mathbf{Imp}_1(1, 2)$ given by the column vector $\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$.
- For a Knightian choice, we consider the morphism **knight** $\in \mathbf{Imp}_2(1, 2)$ given by the unit diagonal matrix. Formally this depends on the choice of product \otimes (§4.2). For the name-respecting choice of product, for any $a \in \mathbb{A}$ we let $\widehat{a} = 2^{\{a\}} (\cong 2)$ and we have **knight**(a) $\in \mathbf{Imp}_{\widehat{a}}(1, 2)$.

Thus **Imp** is a model of the language in §2.3.

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 $\mathbf{FinStoch}_{\text{Surj}}$ 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 $2^B \rightarrow 2^A$ induced by injections $A \rightarrow B$ (connecting even closer with nominal sets [98]; in this case, the semicartesian monoidal structure amounts to the 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.

4.4 Example Calculation with Imp

In this example, to keep calculations brief, we use the numeral choice of product (§4.2).

Example 4.6. Consider the programs (1) and (2), the named variants of the programs 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. Program (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, g and h are the conditionals on the Knightian choices in each branch, and the sum of morphisms is the inverse of the bijection (7) that commutes products with the hom-functor.

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

On the other hand, program (2) is the morphism

$$(\widehat{\pi}_1^\star(g) + \widehat{\pi}_2^\star(h)) \circ f = \begin{pmatrix} 1 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix} \in \mathbf{Imp}_4(1, 3),$$

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 to the Dirac distributions $\widehat{\pi}_1, \widehat{\pi}_2 \in \mathbf{FinStoch}_{\mathbf{Surj}}(2 \otimes 2, 2)$, to account for the decoupling of their Knightian uncertainties.

5 First Theorem: Relationship with the Convex Powerset of Distributions Monad

In this section, we recall the properties of convex powersets of distributions. These form a monad \mathbf{CP} (Definition 5.4) that has been proposed as a model for imprecise probability (see also [15, 52, 63, 87, 88] and elsewhere). In contrast to our model \mathbf{Imp} (§4), it is not graded but also not commutative. We connect our category \mathbf{Imp} with convex powersets of distributions via the Kan extension method of [41] (§5.2) and show that this yields an op-lax functor (§5.3). The interpretation is that composition in our category gives tighter uncertainty bounds (Theorem 5.6), which we illustrate in §5.4.

We begin by recalling some basic properties of convex sets of distributions.

Definition 5.1. 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 5.2. *For any convex map $f : D(m) \rightarrow D(n)$ 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 $q, q' \in \text{image}(f)$, 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 $q +_r q' \in \text{image}(f)$.

The set $\text{image}(f)$ 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 $f : D(m) \rightarrow D(n)$ such that $\text{image}(f) = S$. \square

5.1 Convex Powersets of Distributions

We write $\mathbf{CP}(n)$ for the finitely generated convex subsets of $D(n)$. It supports convex combinations: if $r \in [0, 1]$ and $S, T \in \mathbf{CP}(n)$ then

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

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

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

PROPOSITION 5.3. *There is a family of functions $\phi_{m,n} : \mathbf{Imp}_m(1, n) \rightarrow \mathbf{CP}(n)$, that takes $f \in \mathbf{Imp}_m(1, n)$ to its image $\text{image}(f) \in \mathbf{CP}(n)$, and the family is natural in $m \in \mathbf{FinStoch}_{\mathbf{Surj}}$ and $n \in \mathbf{Set}$.*

PROOF. First, the fact that the image of f is convex is Lemma 5.2. For naturality in m , suppose $g \in \mathbf{FinStoch}_{\mathbf{Surj}}(m', m)$. Then naturality in m amounts to the fact that

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

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

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

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 . \square

5.2 Connection to Kan Extensions

Definition 5.4. The construction CP extends to a relative monad [52]. The unit morphism $\eta_n : n \rightarrow \text{CP}(n)$ picks out the singleton set containing the Dirac vector, $\eta_n(i) = \{\delta_i\}$. The Kleisli extension takes a function $f : m \rightarrow \text{CP}(n)$ to a function $f^* : \text{CP}(m) \rightarrow \text{CP}(n)$ given by

$$f^*(X) = \bigoplus_{x \in \text{ext}(X)} \sum_{i \in m} x(i)f(i),$$

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

Fritz and Perrone [41, 42] propose a method to extract a canonical 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 $\text{CP}(2)$ (considering sets up to their convex closure):

- **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 $\{\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)\} \oplus \{\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)\} = \{\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)\}$.

Therefore, we can follow through the derivation of Figure 2 to see that the CP monad *cannot* be commutative (apparently contradicting [52, Lemma 5.2]) since the convex sets in Figure 1b–1c are different. (For another argument, note that CP contains two binary idempotent symmetric operations, \oplus and $+_{0.5}$, and see our discussion on the Eckmann-Hilton-like obstacle (17) in §7.2.)

By contrast, Imp (§4.3) *does* satisfy our desiderata (§1.1). So there cannot be a monad morphism between Imp and CP . Nonetheless, the Kan extension of our graded Markov category Imp , regarded as a graded monad via Proposition 3.3, gives the finitely-generated convex powerset of distributions monad CP as a functor, just not as a monad.

PROPOSITION 5.5. *The family $\phi_{m,n} : \text{Imp}_m(1, n) \rightarrow \text{CP}(n)$ exhibits $\text{CP} : \text{FinSet} \rightarrow \text{Set}$ as the Kan extension of*

$$\text{Imp}_{(-)}(1, =) : \text{FinStoch}_{\text{Surj}}^{\text{op}} \rightarrow [\text{FinSet}, \text{Set}]$$

along the unique functor $\text{FinStoch}_{\text{Surj}}^{\text{op}} \rightarrow 1$.

$$\begin{array}{ccc} \text{FinStoch}_{\text{Surj}}^{\text{op}} & \xrightarrow{!} & 1 \\ & \searrow \text{Imp}_{(-)}(1, =) & \downarrow \text{CP} \\ & & [\text{FinSet}, \text{Set}] \end{array}$$

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

$$\Phi : \underset{m \in \text{FinStoch}_{\text{Surj}}^{\text{op}}}{\text{colim}} \text{Imp}_m(1, n) \rightarrow \text{CP}(n)$$

(induced by ϕ) is a bijection. This function Φ is given by $\Phi[m, f \in \text{FinStoch}(m, n)] = \text{image}(f)$. To see that it is surjective we recall that every finitely generated convex set is the image of

some convex function $D(m) \rightarrow D(n)$ (Lemma 5.2). To see that it is injective we suppose that $\text{image}(f) = \text{image}(f')$, for $f \in \text{FinStoch}(m, n)$ and $f' \in \text{FinStoch}(m', n)$. We must show that $[m, f] = [m', f']$ in the colimit. It suffices to find m'' with $h \in \text{FinStoch}(m, n)$ and surjections $g \in \text{FinStoch}_{\text{Surj}}(m, m'')$ and $g' \in \text{FinStoch}_{\text{Surj}}(m', m'')$ such that the following diagram commutes:

$$\begin{array}{ccc}
 m & & m' \\
 \searrow g & & \swarrow g' \\
 & m'' & \\
 \downarrow f & \downarrow h & \downarrow f' \\
 n & &
 \end{array}$$

The finitely generated convex set $\text{image}(f) = \text{image}(f')$ 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, 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. \square

5.3 An Op-lax Functor and Tighter Uncertainty Bounds

From the CP monad, we can build a Kleisli category as usual.

- The objects of $\text{Kl}(\text{CP})$ are natural numbers.
- The morphisms $m \rightarrow n$ are functions $m \rightarrow \text{CP}(n)$.
- 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 $\text{Kl}(\text{CP})$ 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 5.3 to an identity-on-objects op-lax functor $\text{Imp} \rightarrow \text{Kl}(\text{CP})$.

THEOREM 5.6. *Consider the assignment of a morphism $f \in \text{Imp}_\gamma(m, n)$ to $R(f) : m \rightarrow \text{CP}(n)$ given by $R(f)(i) = \text{image}(f(-, i))$. This defines an op-lax functor*

$$\text{Imp} \rightarrow \text{Kl}(\text{CP})$$

PROOF NOTES. It is straightforward that $R(\text{id}) = \text{id}$. 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 suppose that the domain of f is 1. So consider $f \in \text{Imp}_\gamma(1, m)$ and $g \in \text{Imp}_\epsilon(m, n)$. So $(g \circ f) \in \text{FinStoch}(\gamma \times \epsilon, n)$. We must show that for all $(i, j) \in (\gamma \times \epsilon)$, the probability vector $(g \circ f)(i, j)$ is in

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

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

$$(g * f) = \left(\gamma \times \epsilon^m \xrightarrow{f} m \times \epsilon^m \xrightarrow{(\text{eval}, \pi_1)} \epsilon \times m \xrightarrow{g} n \right);$$

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

$$(g \circ f) = \left(\gamma \times \epsilon \xrightarrow{f} m \times \epsilon \xrightarrow{\text{swp}} \epsilon \times m \xrightarrow{g} n \right).$$

The function $(\gamma \times \varepsilon) \rightarrow (\gamma \times \varepsilon^m)$ that copies ε is an injection and exhibits

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

Moreover, we have that

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

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

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

(Here, we are regarding **Imp** with discrete order enrichment but non-trivial local grading, and **Kl(CP)** with non-trivial order enrichment but trivial local grading. There may be interesting ways to unify the two different enrichments.)

5.4 Discussion and Example of Tighter Uncertainty Bounds

Example 5.7. We again revisit programs (1) and (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 4.6),

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

and a morphism that employs Knightian uncertainty on each of its inputs ($g + h$ from Example 4.6),

$$g = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \text{Imp}_2(2, 3).$$

(notating this as a 3-dimensional matrix for clarity, rather than as a vector in $\mathbb{R}^{(2 \times 2) \times 3}$). Then, considering sets up to their convex closure, $R(f) : 1 \rightarrow \text{CP}(2)$ maps the singleton set to

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

$R(g) : 2 \rightarrow \text{CP}(3)$ 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 4.6) $R(g \circ f) : 1 \rightarrow \text{CP}(3)$ 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 1b 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 **Kl(CP)**, we lose the ability to distinguish which outcomes were related via the same Knightian choices. So the morphism $R(g) \circ R(f) : 1 \rightarrow \text{CP}(3)$ 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 1c. 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 \mathbf{ImP} obtains tighter bounds on the imprecise probabilities.

6 Second Theorem: Maximality as a Compositional Theory

In Proposition 5.3 we gave a family of maps ϕ that convert our compositional imprecise probability into convex sets of probability distributions. These maps are not injective, and in this sense the model of \mathbf{ImP} is intensional. This raises a question of whether we could have made a less intensional model than \mathbf{ImP} while still maintaining Desiderata 1 and 2 and the connection to convex sets of distributions. In Theorem 6.2 we answer this question negatively in the following sense: we cannot quotient the hom-sets of \mathbf{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). In this way, \mathbf{ImP} is maximal.

Definition 6.1. 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* $F : \mathbf{C} \rightarrow \mathbf{D}$ is given by a mapping from the objects of \mathbf{C} to the objects of \mathbf{D} and a family of mappings from $\mathbf{C}_Y(X, Y) \rightarrow \mathbf{D}_Y(F(X), F(Y))$, strictly preserving the composition, monoidal and coproduct structure, and the copy maps.

Aside. 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. [64]), together with the requirement that the copy maps are preserved, which is in common with the ordinary Markov category literature [40]. We could also formulate this in terms of monad morphisms, following §3.1.

THEOREM 6.2. *Let \mathbf{C} be $\mathbf{FinStoch}_{\mathbf{Surj}}$ -graded distributive Markov category with a graded distributive Markov functor $F : \mathbf{ImP} \rightarrow \mathbf{C}$ and a natural family of functions*

$$\psi_{m,n} : \mathbf{C}_m(1, n) \rightarrow \mathbf{CP}(n)$$

such that

$$\phi_{m,n} : \mathbf{ImP}_m(1, n) \rightarrow \mathbf{CP}(n)$$

(Proposition 5.3) factors through ψ . Then F is faithful: if $F(f) = F(g)$ in \mathbf{C} then also $f = g$ in \mathbf{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

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

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 $m \rightarrow m+n \leftarrow n$ via postcomposition with the unit of D . Since F is a graded distributive Markov functor and $F_{m,n}(f) = F_{m,n}(g)$,

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

where we define $f \boxplus g = (f + g) \circ \mathbf{beroulli}$ for f and g with domain 1. Applying ψ gives

$$\phi_{m,m+n}(\jmath \circ f \boxplus \iota \circ d) = \phi_{m,m+n}(\jmath \circ g \boxplus \iota \circ d). \quad (11)$$

Now, for all $i \in m$, $(\jmath \circ f \boxplus \iota \circ d)(i)$ are independent because they each use a different dimension. They are all the extremal vertices on the convex hull $\phi_{n,m+n}(\jmath \circ f \boxplus \iota \circ d)$. Moreover, they must be the same vertices as $(\jmath \circ g \boxplus \iota \circ d)(i)$ for respective $i \in m$ because the convex hulls are the same (11). Therefore,

$$\jmath \circ f \boxplus \iota \circ d = \jmath \circ g \boxplus \iota \circ d.$$

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

$$\begin{aligned} f(i)(j) &= 2 \times (j \circ f \boxplus \iota \circ d)(i)(j), \\ g(i)(j) &= 2 \times (j \circ g \boxplus \iota \circ d)(i)(j). \end{aligned}$$

So $f = g$. □

6.1 Remarks on Different Approaches to Quotients

We briefly remark on a different approach to quotients in a locally graded category. This is a general method but connects to 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, depending on how we order the quantifiers:

($\forall\exists$) **The approach leading to the CP monad (Theorems 5.6 and 6.2):** We could equate two open terms if, for every valuation of the free variables, there is a regrading that equates them. This violates Desideratum 1 but gives rise to an op-lax functor (Theorem 5.6).

($\exists\forall$) **Alternative quotient:** We could equate two open terms if there is a regrading such that for every valuation of the free variables they are made equal. This violates Desideratum 2, as we discuss below.

For closed terms with no free variables, the two approaches are the same and give rise to a convex set of distributions (Proposition 5.3).

The ‘alternative quotient’ approach ($\exists\forall$) amounts to the ‘connectedness’ quotient in [48, Def. 2.1]. This does not satisfy Desideratum 2 (commuting if-then-else). Informally, it would allow us to work up to a different regrading on the ‘then’ branch versus the ‘else’ branch, which leads to the inconsistency. More formally, the construction of [48] does not yield a category with coproducts in general. For this reason, this alternative quotient approach is not a counterexample to Theorem 6.2. Nonetheless, it could be a useful approach in a metalanguage for combining models that do not need a general if-then-else construction.

7 Context and Related Work

Our focus here is on imprecise probability. This can be thought of as a kind of non-determinism. Specifically, we are combining the effects of probabilistic choices (**bernoulli**) with non-deterministic choices arising from unknown probability distributions (**knight**). There is a broader general interest in non-determinism and its combination with probability. Non-determinism arises in many semantic situations beyond the motivation from unknown distributions in imprecise probability.

Comparison with abstraction/refinement. One arguably different motivation is in program abstraction and refinement. There, one describes a problem by writing a non-deterministic program that solves it in a non-determined way; the problem is then solved by refining that non-deterministic program. When the refined program is still probabilistic, the mathematical analysis is similar to imprecise probability. For instance, illustrations essentially the same as Figure 1 appear in work on the refinement of probabilistic programs [84, Fig. 6.4.2, Fig. 6.5.1]. However, the motivation is arguably different and the desiderata (§1.1) may be less relevant in program refinement.

Contrast with random sets and random bags. An arguably different kind of non-determinism appears where there are many appropriate results that we want to collect. In this sense, for instance, database queries are non-deterministic if they return multiple results, and Prolog is non-deterministic. When combined with probability, this leads more naturally to random sets or random bags, which contrast the sets of distributions shown in Figure 1. Random bags do arise in

probabilistic databases and point process theory. These applications have been considered from a monad perspective [29, 30], and the monads have long been discussed (e.g. [26, 27, 54, 63, 71, 116]).

Broadly, there is a contrast between *sets of distributions*, which arise in imprecise probability and program refinement, and *distributions over sets*, which appear in probabilistic databases and point processes.

7.1 Relationship with Work on Distributive Laws of Monads

There is a large literature exploring distributive laws of monads to find elegant approaches to combining existing monads for probability and non-determinism (e.g. [1, 16, 23, 26, 47, 71, 96, 116]). In fact, the reader monad transformer that we use here is a distributive law of monads. However, our emphasis and motivation are from the commutativity desiderata (§1.1) rather than distributivity issues. We note that commutativity and affinity imply some distributive behaviour, allowing us to reorder computations into non-determinism followed by probability or vice versa. In (14) below, we derive a typical distributivity equation from just the basic desiderata. But that is not quite the same as a distributive law of monads, which is a specific equational requirement [97, Def. 3, Thm. 5]; in some situations, it amounts to a weak distributive law of monads [47].

Even when there is a distributive law between commutative monads, the resulting composite monad need not be commutative. Indeed both the random bags monad [27] and the powerdomain of indexed valuations monad [116] arise from distributive laws between commutative monads, but neither composite monads are commutative. This suggests that given our desiderata (§1.1), distributive laws of monads are not necessarily the best starting point.

7.2 Algebraic Perspective on Probability and Non-determinism

Some previous work on combining probability and non-determinism takes the perspective of algebraic theories. Our desiderata (§1.1) can be viewed from the point of view of algebraic theories, via algebraic effects (e.g. [100]), which we now briefly explore. We define two binary operations:

$$(t +_{0.5} u) \stackrel{\text{def}}{=} \text{if } \mathbf{bernoulli} \text{ then } t \text{ else } u \quad (t \oplus u) \stackrel{\text{def}}{=} \text{if } \mathbf{knight} \text{ then } t \text{ else } u$$

Regarding Desideratum 1, commutativity means that each operation is a homomorphism for the other:

$$\begin{aligned} (s \oplus t) \oplus (u \oplus v) &= (s \oplus u) \oplus (t \oplus v) \\ (s +_{0.5} t) +_{0.5} (u +_{0.5} v) &= (s +_{0.5} u) +_{0.5} (t +_{0.5} v) \\ (s \oplus t) +_{0.5} (u \oplus v) &= (s +_{0.5} u) \oplus (t +_{0.5} v) \end{aligned} \tag{12}$$

and affinity says that

$$t \oplus t = t \quad \text{and} \quad t +_{0.5} t = t. \tag{13}$$

In general, an algebraic structure satisfying commutativity and affinity is recognized as an abstract theory of probability, being the algebraic counterpart to Markov categories, and is sometimes called a ‘mode’ [102].

Desideratum 2 is always assumed in algebraic effects.

From these five axioms (12)–(13), we can derive distributivity equations:

$$\begin{aligned} t \oplus (u +_{0.5} v) &\stackrel{\text{affinity}}{=} (t +_{0.5} t) \oplus (u +_{0.5} v) \stackrel{\text{commutativity}}{=} (t \oplus u) +_{0.5} (t \oplus v) \\ t +_{0.5} (u \oplus v) &\stackrel{\text{affinity}}{=} (t +_{0.5} t) +_{0.5} (u \oplus v) \stackrel{\text{commutativity}}{=} (t +_{0.5} u) \oplus (t +_{0.5} v) \end{aligned} \tag{14}$$

The first equation in (14) instantiates to

$$r \oplus (g +_{0.5} b) = (r +_{0.5} r) \oplus (g +_{0.5} b) = (r \oplus g) +_{0.5} (r \oplus b)$$

which is the problematic derivation of Figure 2 written in algebraic form. If we have an intuition of \oplus as a Minkowski sum (see §5.1) then the left-hand side appears to be Figure 1b and the right-hand side is Figure 1c, especially since the latter can be further rearranged using the commutativity and affinity laws to

$$(r \oplus (g +_{0.5} r)) \oplus ((r +_{0.5} b) \oplus (g +_{0.5} b))$$

which enumerates the four extreme points of Figure 1c.

The symmetry laws

$$t +_{0.5} u = u +_{0.5} t \quad (15)$$

$$t \oplus u = u \oplus t \quad (16)$$

are also desirable, but we note that it is already known that Desiderata 1 ((12)–(13)) are incompatible with them, since together they imply that the two binary operations $(\oplus, +_{0.5})$ are equal! (e.g. [36, 37]):

$$t \oplus u = (t \oplus u) +_{0.5} (t \oplus u) = (t \oplus u) +_{0.5} (u \oplus t) = (t +_{0.5} u) \oplus (u +_{0.5} t) = (t +_{0.5} u) \oplus (t +_{0.5} u) = t +_{0.5} u \quad (17)$$

We can regard the various previous algebraic works as different ways to avoid this Eckmann-Hilton-like obstacle (17), by omitting various equations:

- The point process monad [27, 54, 63, 71] omits the last commutativity law (12) and the affine law for \oplus (13); it keeps the first distributivity law of (14).
- The powerdomain of indexed valuations [116] also omits the last commutativity law (12) but omits the affine law (13) for $+_{0.5}$ instead of for \oplus ; it keeps the first distributivity law of (14).
- The convex powerset of distributions monad (e.g. [16]) also drops the last commutativity law but it keeps the *second* distributivity law of (14).

As we now explain, our approach can be viewed as a graded version of the algebraic perspective, in which we have *all the equational laws*.

7.3 Graded Algebraic Perspective

We can look at our graded category **Imp** from this algebraic perspective, using the graded algebraic theories of Kura [73, §3.1]. We now summarize this angle. From the algebraic perspective, the proposal in this paper is to name the Knightian branching, which amounts to having named binary operators $\oplus_{a_1}, \oplus_{a_2}, \dots$. To be precise, in the formalism of [73, §3.1], we have a graded signature

$$+_{0.5} \in \Sigma_{2,0} \quad \oplus_a \in \Sigma_{2,2^{\{a\}}}$$

where $a \in \mathbb{A}$. That is, we have a binary operation $+_{0.5}$ with empty grade and binary operations \oplus_a with grade $2^{\{a\}}$. From this, we define a grammar for terms $T_g(X)$ with variables in X at grade g [73, Def. 10]:

$$\frac{x \in X}{x \in T_{2^0}(X)} \quad \frac{t, u \in T_g(X)}{t +_{0.5} u \in T_g(X)} \quad \frac{t, u \in T_g(X)}{t \oplus_a u \in T_{2^{\{a\}} \otimes g}(X)} \quad \frac{t \in T_g(X)}{c_f(t) \in T_h(X)} f: h \rightarrow g$$

Here ‘ $c_f(t)$ ’ indicates regrading along $f: h \rightarrow g$. Henceforth we elide c_f where f is a ‘weakening’ regrading, i.e. a canonical projection $2^S \rightarrow 2^{S'}$ for $S' \subseteq S \subset \mathbb{A}$.

We can consider analogues of all the laws (12)–(16) above, in this graded setting:

$$\begin{aligned} (u \oplus_b v) \oplus_a (x \oplus_b y) &= (u \oplus_a x) \oplus_b (v \oplus_a y) \\ (u +_{0.5} v) +_{0.5} (x +_{0.5} y) &= (u +_{0.5} x) +_{0.5} (v +_{0.5} y) \\ (u \oplus_a v) +_{0.5} (x \oplus_a y) &= (u +_{0.5} x) \oplus_a (v +_{0.5} y) \end{aligned} \quad (18)$$

$$x +_{0.5} x = x \quad (19)$$

$$x \oplus_a x = x \quad (20)$$

$$x +_{0.5} y = y +_{0.5} x \quad (21)$$

$$x \oplus_a y = c_{\neg}(y \oplus_a x) \quad (22)$$

where $\neg : 2^{\{a\}} \rightarrow 2^{\{a\}}$ is the non-trivial bijection.

Revisiting the example of Figure 2, we can use the graded deductive system of [73, Thm. 1] to derive:

$$r \oplus_{a_1} (g +_{0.5} b) = (r +_{0.5} r) \oplus_{a_1} (g +_{0.5} b) = (r \oplus_{a_1} g) +_{0.5} (r \oplus_{a_1} b).$$

yet it is consistent to assume

$$(r \oplus_{a_1} g) +_{0.5} (r \oplus_{a_1} b) \neq (r \oplus_{a_1} g) +_{0.5} (r \oplus_{a_2} b).$$

with different names on the right-hand side. These two terms correspond to the difference between Figures 1b and 1c.

Our locally graded category **Imp** (Definition 4.5) is such that its dual $(\mathbf{Imp})^{\text{op}}$ can be regarded as a model of this graded algebraic theory, in the sense of [73, Def. 11] (mildly generalized to use locally graded categories instead of actions). We formulate the model by interpreting $(+_{0.5})$ and (\oplus_a) using **bernoulli** and **knight** from §4.3. All the equations (18)–(22) are satisfied in this model.

One view is that we have side-stepped the Eckmann-Hilton-like obstacle (17) by requiring the explicit regrading in the symmetry law $x \oplus_a y = c_{\neg}(y \oplus_a x)$.

Aside. We note that a named binary choice already appears in the probabilistic setting in [107], where it has a different intuitive meaning: there, a stands for an urn but not a specific draw, and $?_a$ denotes sampling from urn a according to Polya's scheme: replace with two copies of what was drawn. This is different because each $?_a$ can be used multiple times, gathering statistics about an urn, whereas a Knightian draw can only be used once.

8 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, satisfying the Desiderata of §1.1. The account gives a refined bound on the uncertainty (Theorem 5.6) and is maximal among the compositional accounts (Theorem 6.2).

There are several future directions. An initial question is how to accommodate iteration. The convex sets considered in this article are all finitely generated. 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. [46, 62, 63, 116]) and metric structures (e.g. [87, 88]).

It would be interesting to compare to another recent compositional framework combining unknowns with probability by Stein and Samuelson, currently focusing on Gaussians [108].

Our approach is based on random elements, and so is the quasi-Borel-space probability monad (e.g. [49, 113]), so this might be a good approach to accommodating function spaces. On the other hand, we are enriching in $[\mathbf{FinStoch}_{\text{Surj}}^{\text{op}}, \mathbf{Set}]$, which seems closely related to the toposes recently used in [79, 105] for program logic and independence of random variables; in particular, [79] considers sheaves on $\mathbf{FinSet}_{\text{Surj}}$.

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, there may be 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 [75]).

Acknowledgements

The 2023/2024 Aria workshops emphasised aspects of imprecise probability as discussion topics, and there it was helpful to discuss with David, Alexander Appel, Dylan Braithwaite, Vanessa Kosoy, Elena Di Lavoro, Alex Lew, Owen Lynch, Sean Moss, Zenna Tavares, and others. It has been helpful to discuss with colleagues in Oxford (particularly Jeremy Gibbons and Paolo Perrone, and some time ago Kwok Ho Cheung regarding his thesis work [23]), and at the 2024 Bellairs workshop. We also received helpful feedback from Hugo Paquet, Dario Stein, Ralph Sarkis, and MFPS 2024 reviewers, where this material was presented as an ‘early announcement’.

Research supported by the Clarendon Scholarship, ERC Consolidator Grant BLAST, AFOSR Project FA9550-21-1-0038, and the ARIA programme on Safeguarded AI.

References

- [1] F. Abou-Saleh, K.-H. Cheung, and J. Gibbons. 2016. Diagrammatic Reasoning about Probability and Nondeterminism. In *Proc. PPS 2016*.
- [2] N. Ackerman, C. Freer, Y. Kaddar, J. Karwowski, S. Moss, D. Roy, S. Staton, and H. Yang. 2024. Probabilistic programming interfaces for random graphs: Markov categories, graphons, and nominal sets.. In *Proc. POPL 2024*. <https://doi.org/10.1145/3632903>
- [3] A. Adjé, O. Bouissou, J. Goubault-Larrecq, E. Goubault, and S. Putot. 2013. Static Analysis of Programs with Imprecise Probabilistic Inputs. In *Proc. VSTTE 2013*. 22–47. https://doi.org/10.1007/978-3-642-54108-7_2
- [4] T. Altenkirch, J. Chapman, and T. Uustalu. 2015. Monads need not be endofunctors. *Logical Methods in Computer Science* 11, 3 (2015). [https://doi.org/10.2168/LMCS-11\(3\)2015](https://doi.org/10.2168/LMCS-11(3)2015)
- [5] M. Amarante and F. Maccheroni. 2007. *When an Event Makes a Difference*. Springer, Berlin, Heidelberg, 27–32. https://doi.org/10.1007/978-3-540-48935-1_3
- [6] A. Appel and V. Kosoy. 2020. Basic inframeasure theory. LessWrong. <https://www.lesswrong.com/posts/YAa4qcMyoucRS2Ykr/basic-inframeasure-theory>.
- [7] A. Appel and V. Kosoy. 2021. Inframeasures and domain theory. LessWrong. <https://www.lesswrong.com/posts/vrbidMicaoHBhZGp/inframeasures-and-domain-theory>.
- [8] N. Arkor and D. McDermott. 2024. The formal theory of relative monads. *J. Pure Appl. Algebra* 228, 9 (2024). <https://doi.org/10.1016/j.jpaa.2024.107676>
- [9] K. Asada. 2010. Arrows are Strong Monads. In *Proc. MSFP 2010*. <https://doi.org/10.1145/1863597.1863607>
- [10] L. Aslett, F. Coolen, and J. Bock. 2022. *Uncertainty in Engineering: Introduction to Methods and Applications*. <https://doi.org/10.1007/978-3-030-83640-5>
- [11] T. Barker. 2016. A Monad for Randomized Algorithms. In *Proc. MFPS 2016*. <https://doi.org/10.1016/j.entcs.2016.09.031>
- [12] P. Battigalli, S. Cerreia-Vioglio, F. Maccheroni, and M. Marinacci. 2015. Self-Confirming Equilibrium and Model Uncertainty. *American Economic Review* 105, 2 (February 2015), 646–77. <https://doi.org/10.1257/aer.20130930>
- [13] P. Battigalli, R. Corrao, and M. Dufwenberg. 2019. Incorporating belief-dependent motivation in games. *Journal of Economic Behavior & Organization* 167 (2019), 185–218. <https://doi.org/10.1016/j.jebo.2019.04.009>
- [14] J. Bernoulli. 1713. *Ars conjectandi, opus posthumum*. Thurneyesen.
- [15] F. Bonchi, A. Sokolova, and V. Vignudelli. 2019. The Theory of Traces for Systems with Nondeterminism and Probability. In *Proc. LICS 2019*. <https://doi.org/10.1109/LICS.2019.8785673>
- [16] F. Bonchi, A. Sokolova, and V. Vignudelli. 2021. Presenting Convex Sets of Probability Distributions by Convex Semilattices and Unique Bases. In *Proc. CALCO 2021*. <https://doi.org/10.4230/LIPIcs.CALCO.2021.11>
- [17] G. Boudol and I. Castellani. 1994. Flow models of distributed computations: three equivalent semantics for CCS. *Inform. Comput.* 114 (1994), 247–314. <https://doi.org/10.1006/inco.1994.1088>
- [18] D. Braithwaite, J. Hedges, and T. St Clere Smithe. 2023. The Compositional Structure of Bayesian Inference. In *Proc. MFCS 2023*. <https://doi.org/10.4230/LIPIcs.MFCS.2023.24>
- [19] M. Caprio, S. Dutta, K. J. Jang, V. Lin, R. Ivanov, O. Sokolsky, and I. Lee. 2024. Credal Bayesian Deep Learning. [arXiv:2302.09656 \[cs.LG\]](https://arxiv.org/abs/2302.09656)

- [20] A. Carboni, S. Lack, and R. F. C. Walters. 1993. Introduction to extensive and distributive categories. *J. Pure Appl. Algebra* 84, 2 (1993), 145–158. [https://doi.org/10.1016/0022-4049\(93\)90035-R](https://doi.org/10.1016/0022-4049(93)90035-R)
- [21] G. L. Cattani, M. P. Fiore, and G. Winskel. 1998. A Theory of Recursive Domains with Applications to Concurrency. In *Proc. LICS 1998*. <https://doi.org/10.1109/LICS.1998.705658>
- [22] S. Cerreia-Vioglio, R. Corrao, and G. Lanzani. 2023. Dynamic Opinion Aggregation: Long-run Stability and Disagreement. *The Review of Economic Studies* 91, 3 (July 2023), 1406–1447. <https://doi.org/10.1093/restud/rdad072>
- [23] K.-H. Cheung. 2017. *Distributive Interaction of Algebraic Effects*. Ph.D. Dissertation. University of Oxford. <https://ora.ox.ac.uk/objects/uuid:66106628-0a71-4564-bc34-c398db766818>
- [24] J. R. B. Cockett. 1993. Introduction to distributive categories. *Math. Struct. Comput. Sci.* 3, 3 (1993), 277–307. <https://doi.org/10.1017/S0960129500000232>
- [25] P.-L. Curien, R. Garner, and M. Hofmann. 2014. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science* 546 (August 2014), 99–119. <https://doi.org/10.1016/j.tcs.2014.03.003>
- [26] F. Dahlqvist, L. Parment, and A. Silva. 2018. Layer by layer – combining monads. In *Proc. ICTAC 2018*. https://doi.org/10.1007/978-3-030-02508-3_9
- [27] S. Dash. 2024. *A Monadic Theory of Point Processes*. Ph.D. Dissertation. University of Oxford.
- [28] S. Dash, Y. Kaddar, H. Paquet, and S. Staton. 2023. Affine monads and lazy structures for Bayesian programming. In *Proc. POPL 2023*. <https://doi.org/10.1145/3571239>
- [29] S. Dash and S. Staton. 2020. A monad for probabilistic point processes. In *Proc. ACT 2020*. <https://doi.org/10.4204/EPTCS.333.2>
- [30] S. Dash and S. Staton. 2021. Monads for Measurable Queries in Probabilistic Databases. In *Proc. MFPS 2021*. <https://doi.org/10.4204/EPTCS.351.3>
- [31] D. ‘davidad’ Dalrymple. 2024. Safeguarded AI: constructing safety by design. ARIA Programme Thesis. <https://www.aria.org.uk/wp-content/uploads/2024/01/ARIA-Safeguarded-AI-Programme-Thesis-V1.pdf>
- [32] D. ‘davidad’ Dalrymple, J. Skalse, Y. Bengio, S. Russell, M. Tegmark, S. Seshia, S. Omohundro, C. Szegedy, B. Goldhaber, N. Ammann, A. Abate, J. Halpern, C. Barrett, D. Zhao, T. Zhi-Xuan, J. Wing, and J. Tenenbaum. 2024. Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems. [arXiv:2405.06624](https://arxiv.org/abs/2405.06624) [cs.AI]
- [33] B. Day. 1969. On closed categories of functors. In *Proc. Midwest Category Seminar IV (Lect. Notes Math., Vol. 137)*. Springer. <https://doi.org/10.1007/BFb0060438>
- [34] P. Di Gianantonio and A. Edalat. 2024. A Cartesian Closed Category for Random Variables. In *Proc. LICS 2024*. <https://doi.org/10.1145/3661814.3662126>
- [35] E. Di Lavoro and M. Román. 2023. Evidential Decision Theory via Partial Markov Categories. In *Proc. LICS 2023*. <https://doi.org/10.1109/LICS56636.2023.10175776>
- [36] B. Eckman and P. J. Hilton. 1961. Structure maps in group theory. *Fundamenta Mathematicae* 50 (1961), 207–221. <https://doi.org/10.4064/fm-50-2-207-221>
- [37] B. Eckmann and P. J. Hilton. 1962. Group-like Structures in General Categories I: Multiplications and Comultiplications. *Math. Ann.* 145, 3 (June 1962), 227–255. <https://doi.org/10.1007/BF01451367>
- [38] S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. 2003. *Constructing Probability Boxes and Dempster-Shafer Structures*. Technical Report SAND2002-4015. Sandia National Laboratories. <https://doi.org/10.2172/809606>
- [39] B. Fong, D. Spivak, and R. Tuyéras. 2021. Backprop as Functor: A Compositional Perspective on Supervised Learning. In *Proc. LICS 2019*. <https://doi.org/10.1109/LICS.2019.8785665>
- [40] T. Fritz. 2020. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Adv. Math.* 370 (2020). <https://doi.org/10.1016/j.aim.2020.107239>
- [41] T. Fritz and P. Perrone. 2018. A Criterion for Kan Extensions of Lax Monoidal Functors. (2018). arxiv:1809.10481.
- [42] T. Fritz and P. Perrone. 2019. A Probability Monad as the Colimit of Spaces of Finite Samples. *Theory and Applications of Categories* 34 (2019). <https://doi.org/10.48550/arXiv.1712.05363>
- [43] M. Gaboardi, S. Katsumata, D. Orchard, and T. Sato. 2021. Graded Hoare logic and its categorical semantics. In *Proc. ESOP 2021*. https://doi.org/10.1007/978-3-030-72019-3_9
- [44] B. Gavranovic. 2024. *Fundamental Components of Deep Learning: A category-theoretic approach*. Ph.D. Dissertation. Strathclyde University. <https://doi.org/10.48550/arXiv.2403.13001>
- [45] J. Goubault-Larrecq. 2007. Continuous previsions. In *Proc. CSL 2007*. https://doi.org/10.1007/978-3-540-74915-8_40
- [46] J. Goubault-Larrecq. 2008. Prevision Domains and Convex Powercones. In *Proc. FOSSACS 2008*. https://doi.org/10.1007/978-3-540-78499-9_23
- [47] A. Goy and D. Petrisan. 2020. Combining probabilistic and non-deterministic choice via weak distributive laws. In *LICS 2020*. <https://doi.org/10.1145/3373718.3394795>
- [48] C. Hermida and R. Tennent. 2012. Monoidal indeterminates and categories of possible worlds. *Theoretical Computer Science* 430 (2012). <https://doi.org/10.1016/j.tcs.2009.07.083>

[49] C. Heunen, O. Kammar, S. Staton, and H. Yang. 2017. A convenient category for higher-order probability theory. In *Proc. LICS 2017*. <https://doi.org/10.1109/LICS.2017.8005137>

[50] P. J. Huber. 1981. *Robust statistics*. Wiley. <https://doi.org/10.1002/0471725250>

[51] B. Jacobs. 1994. Semantics of weakening and contraction. *Ann. Pure & Appl. Logic* 69, 1 (1994), 73–106. [https://doi.org/10.1016/0168-0072\(94\)90020-5](https://doi.org/10.1016/0168-0072(94)90020-5)

[52] B. Jacobs. 2008. Coalgebraic Trace Semantics for Combined Possibilistic and Probabilistic Systems. In *Proc. CMCS 2008*. <https://doi.org/10.1016/j.entcs.2008.05.023>

[53] B. Jacobs. 2018. From probability monads to commutative effectuses. *J. Log. Algebr. Methods Program.* 94 (2018), 200–237. <https://doi.org/10.1016/j.jlamp.2016.11.006>

[54] B. Jacobs. 2021. From multisets over distributions to distributions over multisets. In *Proc. LICS 2021*. <https://doi.org/10.1109/LICS52264.2021.9470678>

[55] B. Jacobs. 2023. Structured Probabilistic Reasoning. Available from the author’s homepage. Draft book.

[56] B. Jacobs, A. Kissinger, and F. Zanasi. 2019. Causal Inference by String Diagram Surgery. In *Proc. FOSSACS 2019*. https://doi.org/10.1007/978-3-030-17127-8_18

[57] A. Javanmardi, D. Stutz, and E. Hüllermeier. 2024. Conformalized Credal Set Predictors. arXiv:2402.10723 [stat.ML]

[58] X. Jia, B. Lindenholvius, M. W. Mislove, and V. Zamdzhev. 2021. Commutative Monads for Probabilistic Programming Languages. In *Proc. LICS 2021*. 1–14. <https://doi.org/10.1109/LICS52264.2021.9470611>

[59] C. Jones and G. D. Plotkin. 1989. A probabilistic powerdomain of evaluations. In *Proc. LICS 1989*. 186–195. <https://doi.org/10.1109/LICS.1989.39173>

[60] O. Kammar and G. D. Plotkin. 2012. Algebraic foundations for effect-dependent optimisations. In *Proc. POPL 2012*. 349–360. <https://doi.org/10.1145/2103656.2103698>

[61] S. Katsumata. 2014. Parametric effect monads and semantics of effect systems. In *Proc. POPL 2014*. <https://doi.org/10.1145/2535838.2535846>

[62] K. Keimel. 2005. Topological Cones: Foundations for a Domain Theoretical Semantics Combining Probability and Nondeterminism. In *MFPS 2005*. <https://doi.org/10.1016/j.entcs.2005.11.066>

[63] K. Keimel and G. D. Plotkin. 2017. Mixed powerdomains for probability and nondeterminism. *Log. Methods Comput. Sci.* 13 (2017). [https://doi.org/10.23638/LMCS-13\(1:2\)2017](https://doi.org/10.23638/LMCS-13(1:2)2017)

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

[65] F. H. Knight. 1921. *Risk, uncertainty and profit*. Houghton Mifflin. <https://doi.org/10.1017/CBO9780511817410.005>

[66] A. Kock. 1970. Monads on symmetric monoidal closed categories. *Arch. Math.* 21 (1970), 1–10. <https://doi.org/10.1007/BF01220868>

[67] A. Kock. 1971. Bilinearity and cartesian closed monads. *Math. Scand.* 29, 2 (1971). <https://doi.org/10.7146/math.scand.a-11042>

[68] A. Kock. 1972. Strong Functors and Monoidal Monads. *Arch. Math* 23, 1 (December 1972), 113–120. <https://doi.org/10.1007/BF01304852>

[69] A. Kock. 2012. Commutative monads as a theory of distributions. *Theory Appl. Categ.* 26, 4 (2012), 97–131. <https://doi.org/10.48550/arXiv.1108.5952>

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

[71] D. Kozen and A. Silva. 2024. *Multisets and Distributions*. Springer, Cham, 168–187. https://doi.org/10.1007/978-3-031-61716-4_11

[72] D. Krpelík and T. Basu. 2021. *Introduction to Imprecise Probabilities*. Chapter 13, 35–79. https://doi.org/10.1007/978-3-030-60166-9_2

[73] S. Kura. 2020. Graded algebraic theories. In *Proc. FOSSACS 2020*. https://doi.org/10.1007/978-3-03-45231-5_21

[74] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. 2013. Weighted Relational Models of Typed Lambda-Calculi. In *Proc. LICS 2013*. <https://doi.org/10.1109/LICS.2013.36>

[75] P. B. Levy. 2007. Amb Breaks Well-Pointedness, Ground Amb Doesn’t. In *Proc. MFPS 2007*. <https://doi.org/10.1016/j.entcs.2007.02.036>

[76] P. B. Levy. 2019. Locally graded categories. Slides available from the author’s webpage.

[77] P. B. Levy, J. Power, and H. Thielecke. 2003. Modelling environments in call-by-value programming languages. *Inform. Comput.* 185 (2003). [https://doi.org/10.1016/S0890-5401\(03\)00088-9](https://doi.org/10.1016/S0890-5401(03)00088-9)

[78] A. K. Lew, M. F. Cusumano-Towner, B. Sherman, M. Carbin, and V. K. Maninghka. 2020. Trace Types and Denotational Semantics for Sound Programmable Inference in Probabilistic Languages. In *Proc. POPL 2020*. <https://doi.org/10.1145/3371087>

[79] J. Li, J. Aytac, P. Johnson-Freyd, A. Ahmed, and S. Holtzen. 2024. A Nominal Approach to Probabilistic Separation Logic. In *Proc. LICS 2024*. <https://doi.org/10.1145/3661814.3662135>

[80] S. Liang, P. Hudak, and M. Jones. 1995. Monad transformers and modular interpreters. In *Proc. POPL 1995*. <https://doi.org/10.1145/199448.199528>

[81] S. Mac Lane. 1998. *Categories for the Working Mathematician*. Springer. <https://doi.org/10.1007/978-1-4757-4721-8>

[82] D. McDermott. 2019. *Reasoning about effectful programs and evaluation order*. Ph.D. Dissertation. University of Cambridge.

[83] D. McDermott and T. Uustalu. 2022. Flexibly Graded Monads and Graded Algebras. In *Proc. MPC 2022*. https://doi.org/10.1007/978-3-031-16912-0_4

[84] A. McIver and C. Morgan. 2005. *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer. <https://doi.org/10.1007/b138392>

[85] A. K. McIver and C. Morgan. 2001. Partial correctness for probabilistic demonic programs. *Theoret. Comput. Sci.* 266 (2001), 513–541. [https://doi.org/10.1016/S0304-3975\(00\)00208-5](https://doi.org/10.1016/S0304-3975(00)00208-5)

[86] M. Mio. 2014. Upper-Expectation Bisimilarity and Łukasiewicz mu-Calculus. In *Proc. FOSSACS 2014*. https://doi.org/10.1007/978-3-642-54830-7_22

[87] M. Mio, R. Sarkis, and V. Vignudelli. 2021. Combining Nondeterminism, Probability, and Termination: Equational and Metric Reasoning. In *Proc. LICS 2021*. <https://doi.org/10.1109/LICS52264.2021.9470717>

[88] M. Mio and V. Vignudelli. 2020. Monads and Quantitative Equational Theories for Nondeterminism and Probability. In *Proc. CONCUR 2020*. <https://doi.org/10.4230/LIPIcs.CONCUR.2020.28>

[89] M. W. Mislove, J. Ouaknine, and J. Worrell. 2003. Axioms for Probability and Nondeterminism. In *Proc. EXPRESS 2003*. <https://doi.org/10.1016/j.entcs.2004.04.019>

[90] E. Moggi. 1989. Computational Lambda-Calculus and Monads. In *Proc. LICS 1989*. 14–23. <https://doi.org/10.1109/LICS1989.39155>

[91] E. Moggi. 1991. Notions of computation and monads. *Information and Computation* (1991). [https://doi.org/10.1016/0890-5401\(91\)90052-4](https://doi.org/10.1016/0890-5401(91)90052-4)

[92] D. Orchard, P. Wadler, and H. Eades III. 2020. Unifying graded and parameterised monads. In *Proc. MSFP 2020*. <https://doi.org/10.4204/EPTCS.317.2>

[93] Y. Oren, V. Vadocz, M. T. J. Spaan, and W. Böhmer. 2024. Epistemic Monte Carlo Tree Search. arXiv:2210.13455 [cs.LG]

[94] E. Pavlidis, I. Paya, D. Peel, and C. Sirlidopoulos. 2011. Nonlinear dynamics in economics and finance and unit root testing. *The European Journal of Finance* (August 2011). <https://doi.org/10.1080/1351847X.2011.607006>

[95] P. Perrone. 2023. Markov Categories and Entropy. *IEEE Transactions on Information Theory* (2023). <https://doi.org/10.1109/TIT.2023.3328825>

[96] D. Petrisan and R. Sarkis. 2021. Semialgebras and Weak Distributive Laws. In *Proc. MFPS 2021*. <https://doi.org/10.4204/EPTCS.351.14>

[97] M. Piróg and S. Staton. 2017. Backtracking with cut via a distributive law and left-zero monoids. *J. Functional Programming* 27 (2017). <https://doi.org/10.1017/S0956796817000077>

[98] A. M. Pitts. 2013. *Nominal Sets: names and symmetry in computer science*. CUP. <https://doi.org/10.1017/CBO9781139084673>

[99] G. Plotkin and J. Power. 2003. Algebraic operations and generic effects. *Appl. Categ. Structures* 11, 1 (2003), 69–94. <https://doi.org/10.1023/A:1023064908962>

[100] G. D. Plotkin and J. Power. 2002. Notions of Computation Determine Monads. In *Proc. FOSSACS 2002*. https://doi.org/10.1007/3-540-45931-6_24

[101] J. Power. 2006. Generic models for computational effects. *Theor. Comput. Sci.* 364, 2 (2006), 254–269. <https://doi.org/10.1016/j.tcs.2006.08.006>

[102] A. B. Romanowska and J. D. H. Smith. 2002. *Modes*. World Scientific. <https://doi.org/10.1142/4953>

[103] D. Shiebler. 2021. Categorical Stochastic Processes and Likelihood. *Compositionality* 3 (April 2021). Issue 1. <https://doi.org/10.32408/compositionality-3-1>

[104] A. Simpson. 2017. Probability Sheaves and the Giry Monad. In *Proc. CALCO 2017*. <https://doi.org/10.4230/LIPIcs.CALCO.2017.1>

[105] A. Simpson. 2024. Equivalence and Conditional Independence in Atomic Sheaf Logic. In *Proc. LICS 2024*. <https://doi.org/10.1145/3661814.3662132>

[106] S. Staton. 2017. Commutative Semantics for Probabilistic Programming. In *Proc. ESOP 2017*. 855–879. https://doi.org/10.1007/978-3-662-54434-1_32

[107] S. Staton, D. Stein, H. Yang, N. L. Ackerman, C. E. Freer, and D. M. Roy. 2018. The Beta-Bernoulli process and algebraic effects. In *Proc. ICALP 2018*. <https://doi.org/10.4230/LIPIcs.ICALP.2018.141>

[108] D. Stein and R. Samuelson. 2024. A Categorical Treatment of Open Linear Systems. arXiv:2403.03934 [cs.LO]

[109] D. Stein and S. Staton. 2021. Compositional Semantics for Probabilistic Programs with Exact Conditioning. In *Proc. LICS 2021*. <https://doi.org/10.1109/LICS52264.2021.9470552>

- [110] D. Stutz, A. G. Roy, T. Matejovicova, P. Strachan, A. T. Cemgil, and A. Doucet. 2023. Conformal prediction under ambiguous ground truth. arXiv:2307.09302 [cs.LG]
- [111] Z. Tavares, J. Koppel, X. Zhang, R. Das, and A. Solar-Lezama. 2021. A language for counterfactual generative models. In *Proc. ICML 2021*. 10173–10182.
- [112] T. Uustalu. 2010. Strong relative monads. In *Proc. CMCS 2010*.
- [113] M. Vákár, O. Kammar, and S. Staton. 2019. A domain theory for statistical probabilistic programming. In *Proc. POPL 2019*, Vol. 3. 36:1–36:29. <https://doi.org/10.1145/3290349>
- [114] J.-W. van de Meent, B. Paige, H. Yang, and F. Wood. 2018. An Introduction to Probabilistic Programming. (2018). arXiv:1809.10756 [stat.ML]
- [115] G. van Heerdt, J. Hsu, J. Ouaknine, and A. Silva. 2018. Convex Language Semantics for Nondeterministic Probabilistic Automata. In *Proc. ICTAC 2018*. https://doi.org/10.1007/978-3-030-02508-3_25
- [116] D. Varacca and G. Winskel. 2006. Distributing probability over non-determinism. *Math. Structures Comput. Sci.* (2006), 87–113. <https://doi.org/10.1017/S0960129505005074>
- [117] M. Vasile. 2021. *Optimization Under Uncertainty with Applications to Aerospace Engineering*. Springer. <https://doi.org/10.1007/978-3-030-60166-9>
- [118] P. Vicig. 2008. Financial Risk Measurement with Imprecise Probabilities. *Int. J. Approx. Reasoning* 49, 1 (September 2008), 159–174. <https://doi.org/10.1016/j.ijar.2007.06.009>
- [119] P. Walley. 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall. <https://doi.org/10.1007/978-1-4899-3472-7>
- [120] R. Wood. 1976. *Indical methods for relative categories*. Ph. D. Dissertation. Dalhousie University.
- [121] M. A. Zanger, W. Böhmer, and M. T. J. Spaan. 2023. Diverse Projection Ensembles for Distributional Reinforcement Learning. arXiv:2306.07124 [cs.LG]

Received 2024-07-11; accepted 2024-11-07