

# The Magmoid of Normalized Stochastic Kernels

Elena Di Lavore  
University of Oxford  
United Kingdom

Mario Román  
University of Oxford  
United Kingdom

Márk Széles  
Radboud University Nijmegen  
The Netherlands

## Abstract

Normalization,  $D(X + 1) \rightarrow D(X) + 1$ , is almost a distributive law; but because one of the distributive law axioms only holds up-to-idempotent, it yields a non-associative composition of normalized kernels. We introduce the Markov magmoid of normalized stochastic kernels: a normalized-by-construction semantics for probabilistic inference, unifying exact Bayesian observations and interventions as two parenthesizations of the same composite. Front-door and back-door criteria follow from the axioms of Markov magmoids; we implement these with non-associative monadic notation.

## 1 Introduction

Normalization is essential in probabilistic inference: Bayes' law needs normalization to rescale the posterior distribution,

$$\text{posterior} \propto \text{likelihood} \cdot \text{prior}.$$

However, probabilistic programming semantics is not necessarily *normalized-by-construction*: normalization mostly appears as an external primitive, and updating usually employs subdistributions [Pan99, BDGS16] or unnormalized distributions [Koz81, Sta17, WCGC18, EPT17]. Let us provide an example.

Imagine solving the famous *Monty Hall problem* [Sel75]: in a game show, you choose from three closed doors for a chance of winning the prize behind one of them; however, after choosing, the host opens one of the empty doors and — with two closed doors standing — invites you to switch your guess. *Should you switch?*

Let us compute. We (i) consider a prior uniform probability that a prize is behind any of three doors ( $L, M, R$ ); then, (ii) after choosing, say, the middle door, the host will randomly and uniformly open a door ( $L, M, R$ ) that cannot be neither the chosen door nor the one with the prize; (iii) we then *observe* that the host opens, e.g., the left door ( $L$ ); and, upon this, (iv) we renormalize the remaining probabilities and conclude we should switch to the right door ( $R$ ): it doubles our chances of getting the prize.

- (i)  $\frac{1}{3} |L\rangle + \frac{1}{3} |M\rangle + \frac{1}{3} |R\rangle$
- (ii)  $\frac{1}{3} |L\rangle|R\rangle + \frac{1}{6} |M\rangle|L\rangle + \frac{1}{6} |M\rangle|R\rangle + \frac{1}{3} |R\rangle|L\rangle$
- (iii)  $\frac{1}{3} |L\rangle|R\rangle + \frac{1}{6} |M\rangle|L\rangle + \frac{1}{6} |M\rangle|R\rangle + \frac{1}{3} |R\rangle|L\rangle$
- (iv)  $\frac{1}{3} |M\rangle|L\rangle + \frac{2}{3} |R\rangle|L\rangle$ .

The phenomenon we seek to study occurs at the last two steps. We (iii) obtain something that is not a full distribution, but only a subdistribution; and then (iv) we multiply by a constant — by 2, in this example — to obtain again a distribution. This interplay between normalized and unnormalized distributions leads one to pick *substochastic kernels* for probabilistic semantics: functions  $X \rightarrow \mathbf{DMY}$ , for  $\mathbf{D}$  the distribution monad and  $\mathbf{M}$  the maybe monad.

However, we may wish to avoid substochasticity: if the last two steps were compressed into one, subdistributions would never appear. Instead, we would work with *normalized kernels*,  $X \rightarrow \mathbf{MDY}$ ,

yielding either nothing or a full distribution. May we compose *normalized kernels*? Is that what we just did?

The quest for a *category* with *normalized kernels* as morphisms has brought us multiple techniques. Let us review some.

**(a) Work up-to-scalar.** Two subdistributions,  $d_1, d_2 \in \mathbf{DMX}$ , share their normalization whenever there exists some positive real number,  $\lambda \in \mathbb{R}^+$ , such that  $d_1(x) = \lambda \cdot d_2(x)$ . Working up-to-scalar means bringing this idea to kernels [SS24, §6.2; PTRSZ25, Definition 2.9]: we identify two substochastic kernels,  $f_1, f_2: X \rightarrow \mathbf{DMY}$ , up to scalar multiplication,  $f_1 \simeq f_2$ , whenever, for some positive real  $\lambda \in \mathbb{R}^+$ , we have

$$f_1(x; y) = \lambda \cdot f_2(x; y).$$

Although compositional, this quotienting does not identify a kernel with its normalization: the normalization constant of a substochastic kernels depends on the input.

**(b) Work up-to-parameterized-scalar.** The obvious solution is to allow the normalization constant to depend on the input. In practice, this identifies two kernels,  $f_1 \simeq f_2$ , whenever there exists a family of positive reals,  $\lambda(x) \in \mathbb{R}^+$  for each  $x \in X$ , satisfying

$$f_1(x; y) = \lambda(x) \cdot f_2(x; y).$$

This quotienting yields *normalized kernels*,  $X \rightarrow \mathbf{MDY}$ , but it stops being preserved by composition. Abstractly, the normalized morphisms of any *partial Markov category* are not necessarily closed under composition [DR23, Definitions 3.1 and 3.19].

**(c) Work up-to-failure.** A more radical solution does yield a category: the so-called *black-hole semantics* [Fri09, SW18]. Black-hole semantics arises from a valid distributive law casting any subdistribution into a distribution,

$$(-)^\perp: \mathbf{DMX} \rightarrow \mathbf{MDX}.$$

From an abstract point of view, the Kleisli category of this distributive law, the category of *partial stochastic kernels*, is the paradigmatic example of a *quasi-Markov category* [FGL<sup>+</sup>25a, Sha25].

$$(d)^\perp = \begin{cases} \perp, & \text{if } d(\perp) > 0; \\ d, & \text{otherwise.} \end{cases}$$

Alas, this distributive law is not helpful for our purposes: it returns failure whenever the input is not a full distribution. Because it equates any probability of failure to failure, we miss the solution to any problem involving subdistributions.

**Accepting Normalization.** None of these solutions constructs a category of normalized kernels. Should we continue this quest? Have we missed further solutions? Fortunately, Sokolova and Woracek classified all possible single-point extensions of distributions [SW18], which allows us to identify the only functorial one: the only candidate category of normalized kernels arises from *working up-to-failure*; in other words, it has *black-hole semantics*.

**Corollary 1.1** (from [SW18, Theorem 5.3]). *Black-hole semantics (Proposition 3.5) determines the only distributive law between the distribution monad and the maybe monad,  $DM \rightarrow MD$ .*

This manuscript argues that we should embrace this result: there is no category of normalized kernels, but a *magmoid* of normalized kernels. Normalized kernel composition is non-associative.

Indeed, the *Monty Hall problem* admits another parenthesization. Imagine we consider everything since the action of the host as a parenthesized subproblem and we (iv) normalize internally before (v) normalizing globally.

- (i)  $\frac{1}{3}|L\rangle + \frac{1}{3}|M\rangle + \frac{1}{3}|R\rangle$
- (ii)  $\frac{1}{3}|L\rangle|R\rangle + \frac{1}{3}|M\rangle(\frac{1}{2}|L\rangle + \frac{1}{2}|R\rangle) + \frac{1}{3}|R\rangle|L\rangle$
- (iii)  $\frac{1}{3}|L\rangle|R\rangle + \frac{1}{3}|M\rangle(\frac{1}{2}|L\rangle + \frac{1}{2}|R\rangle) + \frac{1}{3}|R\rangle|L\rangle$
- (iv)  $\frac{1}{3}|M\rangle|L\rangle + \frac{1}{3}|R\rangle|L\rangle$
- (v)  $\frac{1}{2}|M\rangle|L\rangle + \frac{1}{2}|R\rangle|L\rangle$ .

In this case, (iii) we *intervene* to force the host to open the left door — say, the game show halts otherwise. Because it is forced, the host’s decision stops carrying any inferential information: we are equally likely to see it no matter where the prize is.

The interpretation of the Monty Hall problem has been famously controversial [Sel75, vs]. Arguably, the difference of interpretation is clearer knowing that normalized kernel composition is not associative: *when to normalize* does change the result.

Still, normalized kernels have a rich algebraic structure: both in the discrete and the continuous case, normalized kernels form a monoidal non-associative category with copy-discard maps and conditionals (Theorems 7.9 and 8.7). The category of substochastic kernels acts on the non-associative category of normalized kernels (Corollaries 4.13 and 8.6): *updates act on priors*. Normalization,  $DMX \rightarrow MDX$ , is almost a distributive law, and it interacts with the actual distributive law of subdistributions (Theorems 4.7 and 8.4). Associativity may be unnecessary, after all: associative updating semantics re-emerges from systematic left-association. We abstract all this algebra of normalized kernels, without associativity, into a structure we dub a *Markov magmoid*.

## 1.1 Related work

*Probabilistic programming semantics* employs substochastic kernels, starting with Kozen’s [Koz81] and Panangaden’s substochastic variant of the Giry monad [Gir82, Pan99]. Since these, both operational [PPT08, Par03, LZ12, WCGC18] and denotational semantics [MPYW18, JLMZ21, VKS19] predominantly account for rejection with substochastic [BDGS16, FR19] or unnormalized kernels [EPT17, EPT11, Sta17, DKPS23], with normalization sporadically justifying program equations [SWY<sup>+</sup>16]. *Probabilistic programming languages* either normalize as a program transformation [NCR<sup>+</sup>16], or let their inference algorithms handle it [GMR<sup>+</sup>12, TvDMYW16].

Categorical probability theory, in a line of work starting from Golubtsov, Cho and Jacobs, and Fritz, has abstracted stochastic kernels into *Markov categories* [Gol99, CJ19, Fri20]. Further work has abstracted substochastic kernels and partial stochastic kernels into *partial Markov categories* [DR23] and *quasi-Markov categories* [FGL<sup>+</sup>25b], respectively. In particular, string diagrammatic methods

can model Pearl’s causal interventions [Pea09] by syntactic substitution [Fon13, JKZ21, FK23]. At the same time, multiple string diagrammatic axiomatizations of normalization have been proposed, we highlight those in terms of *normalization boxes* [LT23, JSS25] and *partial Markov categories* [DR23]. Simpson’s probability sheaves constitute another approach to synthetic probability theory [Sim17, Sim24]; for which Stein recently proposed a comparison [Ste25].

Jacobs’ *hypernormalization* [Jac17] — generalized by Garner via *tricocycloids* [Gar18] — is an alternative to normalization: we relate this approach to sesquilaws by proving that every tricocycloid induces a sesquilaw (see Appendix, Theorem J.1), while not all Set-based sesquilaws arise from a tricocycloid.

*Magmoids* and failing distributive laws are relatively infrequent: Munch-Maccagnoni [Mun13] proposed a non-natural monad-comonad distributive law [MMM25] and its magmoid to unify call-by-name and call-by-value. In probability, mass and chance interpretations determine a non-natural distributive law,  $DD \rightarrow DD$  [TPAH25]. Weak distributive laws, instead, [Str09, Böh10, Gar20] appear in imprecise probability [GP20, MSV21, LS24], as distributive laws are insufficient for systems with non-determinism and probability [VW06]. In this non-deterministic case, with applications to probabilistic trace semantics [SS00, SS11, CMN<sup>+</sup>25], Bonchi, Sokolova, and Vignudelli distinguished possibilistic monads for *may*, *must*, and *may-must* semantics [DGHM09, BSV22]; via support (c.f. [FPR21]), we relate these to normalized kernels, partial stochastic kernels, and substochastic kernels, respectively.

## 1.2 Contributions

Normalized kernels do not form a category (Proposition 2.6); we introduce monoidal magmoids (Definition 2.12) — a coherent notion of monoidal non-associative category — and we prove that normalized kernels form a monoidal magmoid with extra structure that we dub a Markov magmoid (Theorem 7.9). Normalization is not a distributive law (Proposition 3.7); we introduce sesquilaws (Definition 4.6); we show that sesquilaws yield the renormalization axiom (Theorem 4.8), a right monad action (Lemma 4.11), and a category action on their magmoid (Theorem 4.12); we prove that normalization forms a sesquilaw (Theorem 4.7). Moreover, we prove that support is a morphism to the possibilistic sesquilaw (Proposition 5.6 and Corollary 5.9). In the continuous case, we prove that normalized kernels form a sesquilaw and a Markov magmoid (Theorems 8.4 and 8.7).

Finally, we introduce commutativity for monoidal magmoids (Definition 6.3) and their string diagrams (Theorem 6.7); we specialize the axioms of Markov magmoids to discrete probability (Definition 9.1) and we derive a synthetic version of the *back-door* and *front-door adjustment formulas* from causality theory (Propositions 9.6 and 9.8). We introduce a left-associative monadic notation with magmoidal semantics (Definition 9.11) to compute problems in causality theory (Example 9.13).

## 1.3 Synopsis

Section 2 introduces the monoidal magmoid of normalized kernels. Section 3 is a background section on distributive laws, while Section 4 introduces sesquilaws. Section 5 compares the probabilistic and possibilistic cases. Section 6 develops commutative magmoids

and their string diagrams. Section 7 introduces Markov magmoids. Section 8 studies normalization in standard Borel spaces. Section 9 presents a specialized application to causality. We provide detailed proofs in the Appendix.

## 2 Normalization

Normalization is difficult to classify categorically. While it is a fundamental operation of probability theory, it is generally regarded as ill-behaved [Jac17]. While it induces a natural transformation that braids the distribution (**D**) and maybe (**M**) monads,  $n_X: \mathbf{DMX} \rightarrow \mathbf{MDX}$  (Definition 2.2), it is not a distributive law. And, while it induces a composition of normalized kernels,  $(\circ)_{X,Y,Z}: \mathbf{Norm}(X; Y) \times \mathbf{Norm}(Y; Z) \rightarrow \mathbf{Norm}(X; Z)$ , it is not associative.

This section explains this missing algebra: we introduce normalization as a monoidal natural transformation in Section 2.1; we recall non-associative categories and monoidal non-associative categories in Sections 2.2 and 2.3; and we finally contribute a coherent notion of monoidal magmoid, that allows us to structure normalized kernels, in Section 2.4.

### 2.1 Normalization

Normalization is inherently partial — it contains a potential division by zero — but explicitly dealing with partiality is tedious. Instead, note that normalized distributions are, equivalently, subdistributions adding up to exactly 0 or 1,

$$\mathbf{MDX} \cong \left\{ d \in \mathbf{DMX} \mid \sum_{x \in X} d(x) = 0 \text{ or } \sum_{x \in X} d(x) = 1 \right\}.$$

Under this interpretation, zero means failure.

**Definition 2.1** (Bracketed division). Let us convene that

$$\left[ \frac{u}{v} \right] = \begin{cases} u/v & \text{when } v \neq 0, \\ 0 & \text{when } v = 0. \end{cases}$$

**Definition 2.2** (Normalization). *Normalization* is the natural transformation,  $n_X: \mathbf{DMX} \rightarrow \mathbf{MDX}$ , defined by

$$n(f)(x) = \left[ \frac{f(x)}{\sum_{x' \in X} f(x')} \right].$$

Normalization is a monoidal natural transformation. Both the finitary distribution monad (**D**) and the maybe monad (**M**) are monoidal monads on sets: both their Kleisli categories, **Stoch** and **Par**, are copy-discard categories. Normalization inherits this structure.

**Proposition 2.3.** *Normalization,  $n_X: \mathbf{DMX} \rightarrow \mathbf{MDX}$ , is a monoidal natural transformation. Normalization of independent distributions is the joint normalization of the distributions,  $n(f \otimes g) = n(f) \otimes n(g)$ .*

$$\left[ \frac{f(x) \cdot g(y)}{\sum_{u \in X, v \in Y} f(u) \cdot g(v)} \right] = \left[ \frac{f(x)}{\sum_{u \in X} f(u)} \right] \cdot \left[ \frac{g(y)}{\sum_{v \in Y} g(v)} \right].$$

Were normalization to form a distributive law, its Kleisli category, **Norm**, would also be monoidal. Perhaps surprisingly, normalization fails to form a distributive law (see Proposition 3.7); and this potential Kleisli category is instead a Kleisli non-associative category.

### 2.2 Normalized kernels are not associative

Defining non-associative “categories” is straightforward: we repeat the definition of category, but without associativity (c.f. [MMM25]). Let us set aside any more technical definition for now and construct the non-associative category of normalized kernels.

**Definition 2.4** (Non-associative category). A *non-associative category*,  $\mathbb{C}$ , consists of collections of objects and morphisms,  $\mathbb{C}(X; Y)$ , for each two objects  $X, Y \in \mathbb{C}_{obj}$ , equipped with a composition operation,  $(\circ): \mathbb{C}(X; Y) \times \mathbb{C}(Y; Z) \rightarrow \mathbb{C}(X; Z)$ , and identities,  $\text{id}_X \in \mathbb{C}(X; X)$ , that are unital, meaning that  $f \circ \text{id}_Y = f = \text{id}_X \circ f$ .

**Proposition 2.5** (Non-associative category of normalized kernels). *Normalized kernels between sets,  $X \rightarrow \mathbf{MDY}$ , form a non-associative category, **Norm**, where composition of two morphisms,  $f: X \rightarrow \mathbf{MDY}$  and  $g: Y \rightarrow \mathbf{MDZ}$ , is defined by*

$$(f \circ g)(x; z) = \left[ \frac{\sum_{v \in Y} f(x; v) \cdot g(v; z)}{\sum_{v \in Y} \sum_{w \in Z} f(x; v) \cdot g(v; w)} \right].$$

*In other words, if we consider the associated substochastic kernels,  $f^\bullet: X \rightarrow \mathbf{DMY}$  and  $g^\bullet: Y \rightarrow \mathbf{DMZ}$ , it is the normalization of their composition as subdistributions,  $f \circ g = n(f^\bullet \circ g^\bullet)$ .*

**Proposition 2.6.** *Normalized kernels are not associative.*

**Remark 2.7.** It may be clarifying to symbolically check for associativity. Arguably, left-associated composition simplifies as expected,

$$((f \circ g) \circ h)(x; w) = \left[ \frac{\sum_{y, z} f(x; y) \cdot g(y; z) \cdot h(z; w)}{\sum_{y, z, w} f(x; y) \cdot g(y; z) \cdot h(z; w)} \right];$$

while right-associated composition may contain different normalization constants on the numerator and the denominator, preventing a similar simplification,

$$(f \circ (g \circ h))(x; w) = \left[ \frac{\sum_y f(x; y) \cdot \left[ \frac{\sum_z g(y; z) \cdot h(z; w)}{\sum_{z, w} g(y; z) \cdot h(z; w)} \right]}{\sum_{y, w} f(x; y) \cdot \left[ \frac{\sum_z g(y; z) \cdot h(z; w)}{\sum_{z, w} g(y; z) \cdot h(z; w)} \right]} \right].$$

Because of this asymmetry, left-associativity will later be our default.

### 2.3 Normalized kernels are monoidal

Monoidal non-associative categories can be defined naïvely (Definition 2.8): functors and natural transformations work as usual. Only coherence requires some care: the pentagon equation [ML71] is ambiguous in the absence of associativity. Let us briefly postpone coherence — and braiding equations — to first construct the monoidal non-associative category of normalized kernels.

**Definition 2.8** (Monoidal non-associative category). A *monoidal non-associative category* is a non-associative category,  $\mathbb{C}$ , endowed with binary and nullary functors,  $(\otimes): \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$  and  $I: 1 \rightarrow \mathbb{C}$ , and natural isomorphisms for associativity,  $\alpha_{X,Y,Z}: (X \otimes Y) \otimes Z \rightarrow X \otimes (Y \otimes Z)$ , and unitality,  $\lambda_X: I \otimes X \rightarrow X$  and  $\rho_X: X \otimes I \rightarrow X$ .

A monoidal non-associative category is *symmetric* when it is moreover endowed with a natural isomorphism representing symmetry,  $\sigma_{X,Y}: X \otimes Y \rightarrow Y \otimes X$ .

**Proposition 2.9.** *Normalized kernels form a symmetric monoidal non-associative category with the cartesian product, where the tensor of  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$  is given by*

$$(f_1 \otimes f_2)(x_1, x_2; y_1, y_2) = f_1(x_1; y_1) \cdot f_2(x_2; y_2).$$



## 2.4 Magmoids

Let us now address coherence: coherence needs structural maps to be associative, but associative morphisms are not even necessarily closed under tensoring.

Still, we may explicitly pick a closed class of associative morphisms:<sup>1</sup> we reserve the name *magmoid*, in this text, for a non-associative category with a chosen subcategory of associative morphisms (Definition 2.11), a concept that we introduce to address coherence (Definition 2.12). Indeed, monoidal magmoids are coherent thanks to this base monoidal category (Remark 2.13).

**Definition 2.10** (Associative morphism). In a non-associative category, a morphism  $f: X \rightarrow Y$  is *associative* whenever, for each  $g_1: X_1 \rightarrow X$ , each  $g_2: X_2 \rightarrow X_1$ , each  $h_1: Y \rightarrow Y_1$  and each  $h_2: Y_1 \rightarrow Y_2$ , we have that  $(g_2 \circ g_1) \circ f = g_2 \circ (g_1 \circ f)$ , that  $(g_1 \circ f) \circ h_1 = g_1 \circ (f \circ h_1)$ , and that  $(f \circ h_1) \circ h_2 = (f \circ h_1) \circ h_2$ .

**Definition 2.11** (Magmoid). A *magmoid* consists of a non-associative category  $\mathbb{M}$ , a category  $\mathbb{A}$ , and an identity-on-objects functor,

$$(-)_\dagger: \mathbb{A} \rightarrow \mathbb{M},$$

whose image is associative.

**Definition 2.12** (Monoidal magmoid). A *monoidal magmoid* consists of a monoidal non-associative category  $(\mathbb{M}, \otimes, I)$ , a monoidal category  $(\mathbb{A}, \otimes, I)$ , and an identity-on-objects functor strictly preserving the monoidal structure,  $(-)_\dagger: \mathbb{A} \rightarrow \mathbb{M}$ , whose image is associative. A monoidal magmoid is *symmetric* when its category, non-associative category, and functor are.

**Remark 2.13** (Coherence for monoidal magmoids). Monoidal magmoids are *strict* when their base monoidal category is. As a consequence, they can be strictified.<sup>2</sup>

Even when normalized kernels do not associate, they do in some useful cases (Proposition 2.14). In particular, functions are associative, and we pick them as the base monoidal category.

**Proposition 2.14.** In the normalization magmoid, the associativity equation  $f \circ (g \circ h) = (f \circ g) \circ h$  holds when either

- (1) the kernel  $f$  lifts a partial function,  $X \rightarrow MY$ , or
- (2) the kernel  $g$  lifts a partial function,  $X \rightarrow MY$ , or
- (3) the kernel  $h$  lifts a stochastic kernel,  $X \rightarrow DY$ .

In particular, associativity holds whenever any of the three lifts a function  $X \rightarrow Y$ : functions are associative morphisms.

**Proposition 2.15.** Normalized kernels form a symmetric monoidal magmoid over the category of sets and functions,  $(-)_\dagger: \text{Set} \rightarrow \text{Norm}$ .

However, the monoidal magmoid of normalized kernels is special: it satisfies properties that are not true in arbitrary monoidal magmoids. To study these, we adapt distributive laws.

<sup>1</sup>This technique is common in the categorical semantics of effectful programs [PR97, Jef97, SL13], where one must distinguish a subclass of pure morphisms. Technically, we use non-associative monoidal promonads.

<sup>2</sup>Alternatively, monoidal magmoids are pseudomonoids of the bicategory of non-associative promonads (c.f. [Bén00]), with associative natural transformations. By coherence for pseudomonoids [Ver17], every monoidal magmoid is strictifiable.

## 3 Distributive Laws

Distributive laws [Bec69], their uses and limitations [ZM22], are well-known. Briefly, the composition of two monads is not a monad again — in general, the tensor of two monoids is not a monoid again — but distributive laws endow this composition with monad structure.

**Definition 3.1** (Distributive law [Bec69],  $\blacktriangledown$ ). A *distributive law* between two monads,  $(S, \mu^S, \eta^S)$  and  $(T, \mu^T, \eta^T)$ , on the same category is a natural transformation,  $\psi: TS \rightarrow ST$ , that makes the following four diagrams commute.

$$\begin{array}{ccccc} TTSX & \xrightarrow{\mu^T} & TSX & \xrightarrow{\psi} & STX \\ T\psi \downarrow & & & \nearrow S\mu^T & \\ TSTX & \xrightarrow{\psi^T} & STTX & & \end{array} \quad (1)$$

$$\begin{array}{ccccc} TSSX & \xrightarrow{T\mu^S} & TSX & \xrightarrow{\psi} & STX \\ \psi S \downarrow & & & \nearrow \mu^{ST} & \\ STSX & \xrightarrow{S\psi} & SSTX & & \end{array} \quad (2)$$

$$\begin{array}{ccc} TX & \xrightarrow{\eta^{ST}} & STX \\ T\eta^S \downarrow & \nearrow \psi & \\ TSX & & \end{array} \quad \begin{array}{ccc} SX & \xrightarrow{S\eta^T} & STX \\ \eta^TS \downarrow & \nearrow \psi & \\ TSX & & \end{array} \quad (3, 4)$$

**Definition 3.2** (Monoidal distributive law). A *monoidal distributive law* between two monoidal monads,  $(S, \mu, \eta, u, v)$  and  $(T, \mu', \eta', u', v')$ , is a distributive law whose transformation,  $\psi_X: TSX \rightarrow STX$ , is monoidal.

**Proposition 3.3** ([Bec69]). A distributive law,  $\psi_X: TSX \rightarrow STX$ , between two monads,  $(S, \mu^S, \eta^S)$  and  $(T, \mu^T, \eta^T)$ , induces a monad structure on the composite functor  $ST$ . Given two monoidal monads, a monoidal distributive law between them induces a monoidal monad structure on the composite functor.

### 3.1 Example: substochastic kernels

*Substochastic kernels*, functions of the form  $X \rightarrow \text{DMY}$ , form a monoidal category,  $\text{subStoch}$ , induced by a monoidal distributive law,  $\text{MD} \rightarrow \text{DM}$ . Thanks to this law, normalized kernels can be seen as particular substochastic kernels with a different composition.

**Proposition 3.4** (Subdistributions). Inclusion of normalized distributions into subdistributions,  $(-)^{\bullet}: \text{MDX} \rightarrow \text{DMX}$ , defined on distributions by  $(\perp)^{\bullet} = 1 \mid \perp$  and  $d^{\bullet} = d$  and on morphisms by  $f^{\bullet}(x; y) = f(x; y)$ , induces a monoidal distributive law.

The monoidal Kleisli category of this distributive law is the category of substochastic kernels,  $\text{subStoch}$ .

### 3.2 Example: partial stochastic kernels

*Partial stochastic kernels*, functions  $f: X \rightarrow \text{MDY}$ , correspond to normalized kernels, albeit with a different composition operation. The category of *partial stochastic kernels*,  $\text{parStoch}$ , composes two normalized kernels,  $f: X \rightarrow \text{MDY}$  and  $g: Y \rightarrow \text{MDZ}$ , into

$$(f \circ g)(x; z) = \left[ \sum_{y, z' \in Y} f(x; y) \cdot g(y; z') = 1 \right] \cdot \sum_{y \in Y} f(x; y) \cdot g(y; z),$$

which, using *Iverson brackets*, is zero when not a full distribution.

**Proposition 3.5.** *The natural transformation  $(-)^{\perp}: \mathbf{DM} \rightarrow \mathbf{MD}$  defined by  $f^{\perp}(x) = f(x) \cdot [\sum_x f(x) = 1]$ , induces a monoidal distributive law. Its Kleisli category is the category of partial stochastic kernels,  $\text{parStoch}$ .*

**Remark 3.6.** Partial stochastic kernels are the paradigmatic example of *quasi-Markov category* [FGL<sup>+</sup>25a, Sha25]. While this quasi-Markov category plays a role in black-hole semantics, it does not provide updating semantics nor it addresses the problem of normalization: indeed, it is useful precisely because it marks with failure whenever a normalization problem is encountered.

### 3.3 Counterexample: normalized kernels

Normalization satisfies all of the axioms of a distributive law, except for one: the D-multiplicativity axiom. Next section develops normalization not as a distributive law, but as a sesquilaw.

**Proposition 3.7.** *Normalization is not a distributive law.*

**PROOF.** We show that D-multiplicativity fails. Consider the set  $A = \{x, y\}$  and pick the following distribution over subdistributions:

$$\frac{1}{2} \left| \frac{1}{3} |x\rangle + \frac{2}{3} |\perp\rangle \right\rangle + \frac{1}{2} \left| \frac{2}{3} |y\rangle + \frac{1}{3} |\perp\rangle \right\rangle \in \mathbf{DDMA}.$$

Computing both sides of the diagram in Equation 1 (Definition 3.1) yields different expressions. On the left-hand side, we (i) normalize internally, (ii) normalize externally and (iii) multiply.

$$\begin{array}{ll} (i), \text{ by (Dn)} & \frac{1}{2} \left| \frac{1}{3} |x\rangle + \frac{2}{3} |\perp\rangle \right\rangle + \frac{1}{2} \left| \frac{2}{3} |y\rangle + \frac{1}{3} |\perp\rangle \right\rangle \\ \xrightarrow{\quad} & \frac{1}{2} ||x\rangle + \frac{1}{2} ||y\rangle + 0 |\perp\rangle \\ (ii), \text{ by (nD)} & \frac{1}{2} ||x\rangle + \frac{1}{2} ||y\rangle \\ \xrightarrow{\quad} & \frac{1}{2} ||x\rangle + \frac{1}{2} ||y\rangle \\ (iii), \text{ by } (\mu) & \xrightarrow{\quad} \frac{1}{2} |x\rangle + \frac{1}{2} |y\rangle. \end{array}$$

On the right-hand side, we (i) multiply and (ii) normalize.

$$\begin{array}{ll} (i), \text{ by } (\mu) & \frac{1}{2} \left| \frac{1}{3} |x\rangle + \frac{2}{3} |\perp\rangle \right\rangle + \frac{1}{2} \left| \frac{2}{3} |y\rangle + \frac{1}{3} |\perp\rangle \right\rangle \\ \xrightarrow{\quad} & \frac{1}{6} |x\rangle + \frac{1}{3} |\perp\rangle + \frac{1}{3} |y\rangle + \frac{1}{6} |\perp\rangle \\ (ii), \text{ by (n)} & \xrightarrow{\quad} \frac{1}{3} |x\rangle + \frac{2}{3} |y\rangle. \end{array}$$

However,  $\frac{1}{3} |x\rangle + \frac{2}{3} |y\rangle \neq \frac{1}{2} |x\rangle + \frac{1}{2} |y\rangle$ .  $\square$

## 4 Sesquilaws

Sesquilaws are a particular form of failing distributive law. We define sesquilaws to be sections to an actual distributive law that satisfy all distributive law axioms except for the first one, which is only satisfied up to idempotent. In particular, because they fail one of the axioms, they are almost-distributive laws.

### 4.1 Almost-distributive laws

An *almost-distributive law* could be any candidate distributive law failing one of the axioms. Specifically, we could define non-S-multiplicative, non-S-unital, non-T-multiplicative, and non-T-unital almost-distributive laws, respectively. In this terminology, a *weak distributive law* [Str09, Böh10, Gar20, GP20] would be a non-T-unital almost-distributive law or, sometimes, a non-S-unital almost-distributive law. For the rest of the text, however, let us focus on non-T-multiplicative almost-distributive laws, and simply call them almost-distributive laws.

**Definition 4.1** (Almost-distributive law). An *almost-distributive law* is a candidate distributive law,  $\psi: TS \rightarrow ST$ , failing the T-multiplicativity axiom. A *monoidal almost distributive law* between two monoidal monads is an almost-distributive law whose underlying natural transformation is monoidal.

**Definition 4.2** (Non-associative monad). A *non-associative monad* over a category  $\mathbb{C}$  consists of an endofunctor,  $T: \mathbb{C} \rightarrow \mathbb{C}$ , together with natural transformations  $\eta_X: X \rightarrow TX$  and  $\mu_X: TTX \rightarrow TX$  satisfying the unitality equations,  $T\eta_X \circ \mu_X = \text{id}_X$  and  $\eta_{TX} \circ \mu_X = \text{id}_X$ , but not necessarily the multiplicativity equations. It is *monoidal* whenever these transformations are.

**Proposition 4.3.** *Almost-distributive laws,  $\psi_X: TSX \rightarrow STX$ , induce non-associative monads on their composite functors,  $ST$ . Monoidal almost distributive laws induce monoidal non-associative monads on their composite functors.*

**Proposition 4.4** (Kleisli magmoids). *Any non-associative monad,  $(R, \mu^R, \eta^R)$  over a category  $\mathbb{C}$ , induces a magmoid,  $(\mathbb{C}, \mathbb{K}(R))$ . Any monoidal non-associative monad induces a monoidal magmoid.*

**Proposition 4.5.** *Normalization,  $n_X: \mathbf{DMX} \rightarrow \mathbf{MDX}$ , forms an almost-distributive law.*

Normalization, the monoidal almost-distributive law, induces the Kleisli monoidal magmoid of normalized kernels,  $\text{Norm}$ . Together with the distributive law describing subdistributions, we have “one and a half” distributive laws that interact with each other: a sesquilaw.

### 4.2 Sesquilaws

To recap, normalization,  $n(-): \mathbf{DM} \rightarrow \mathbf{MD}$ , satisfies all distributive law axioms except for the D-multiplicativity axiom. Still, normalization satisfies an equation resembling this missing multiplicativity:  $n(n(f)^{\bullet}; g) = n(f; g)$ , for any two substochastic kernels. Careful inspection reveals that the D-multiplicativity axiom holds *up-to-an-idempotent*: the distributive law of subdistributions,  $(-)^{\bullet}: \mathbf{MD} \rightarrow \mathbf{DM}$ , is the partial inverse inducing this idempotent.

Distributive sesquilaws — or, simply, sesquilaws — abstract this situation with a single extra equation. This equation consists of multiplicativity up to the idempotent determined by a section-retraction pair of two distributive law candidates.

**Definition 4.6** (Sesquilaw,  $\blacktriangledown$ ). A *sesquilaw*,  $(S, T, m, n)$ , between two monads  $(S, \mu^S, \eta^S)$  and  $(T, \mu^T, \eta^T)$ , consists of a distributive law,  $m_X: STX \rightarrow TSX$ , and an almost distributive law,  $n_X: TSX \rightarrow STX$ , forming a section-retraction pair,  $m \circ n = \text{id}$ , and making the following diagram commute.

$$\begin{array}{ccccc} TTSX & \xrightarrow{Tn} & TSTX & \xrightarrow{nT} & STTX & \xrightarrow{S\mu^T} & STX \\ Tn \downarrow & & & & & \nearrow n & \\ TTSX & \xrightarrow{Tm} & TTTSX & \xrightarrow{\mu^T S} & TSX & & \end{array}$$

Alternatively, but equivalently, a sesquilaw must make the following diagram commute.

$$\begin{array}{ccccc} TSTX & \xrightarrow{nT} & STTX & \xrightarrow{S\mu^T} & STX \\ Tm \downarrow & & & & \nearrow n & \\ TTTSX & \xrightarrow{\mu^T S} & TSX & & \end{array}$$

A sesquilaw is monoidal whenever its transformation is.

**Theorem 4.7.** *Normalization and subdistributions form a sesquilaw.*

A sesquilaw is enough to abstractly prove multiple useful facts about normalization. For the rest of this section, we synthetically derive structure from the sesquilaw axioms.

**Theorem 4.8** (Renormalization). *Any sesquilaw,  $(S, T, m, n)$ , induces an idempotent,  $k = (n \circ m): TS \rightarrow TS$ . This idempotent is left-absorptive, meaning the following diagram commutes.*

$$\begin{array}{ccc} TSTSX & \xrightarrow{\mu^{TS}} & TSX \\ kTS \downarrow & & \nearrow k \\ TSTSX & \xrightarrow{\mu^{TS}} & TSX \end{array}$$

**Corollary 4.9** (Renormalization). *The following equation holds for substochastic kernels (c.f. [DRS25, Proposition 3.12; SWY<sup>+</sup>16, §4.1]).*

$$n(f; g) = n(n(f); g).$$

**Remark 4.10.** For instance, sequential Monte Carlo simulations normalize and resample after each observation for efficiency reasons [PW14, Algorithm 1]. Renormalization guarantees the soundness of this transformation [SWY<sup>+</sup>16, §4.1].

**Lemma 4.11.** *Any sesquilaw,  $(S, T, m, n)$ , induces a right action of the monad  $TS$  into the non-associative monad  $ST$ : a natural transformation,  $u_X: STTSX \rightarrow STX$ , making the following two diagrams commute.*

$$\begin{array}{ccc} STX & \xrightarrow{\eta^{TS}} & STTSX \\ \text{id} \searrow & & \downarrow u \\ & & STX \end{array} \quad \begin{array}{ccc} STTSX & \xrightarrow{ST\mu^{TS}} & STTSX \\ uTS \downarrow & & \downarrow u \\ STTSX & \xrightarrow{u} & STX \end{array}$$

This action is defined by either side of the following commutative diagram.

$$\begin{array}{ccccc} STTSX & \xrightarrow{S\mu^{TS}} & STSX & \xrightarrow{mS} & TSSX \\ mTS \downarrow & & & & \downarrow T\mu^S \\ TTSX & \xrightarrow{TmS} & TTSSX & \xrightarrow{\mu^T \mu^S} & TSX \xrightarrow{n} STX \end{array}$$

This general phenomenon for sesquilaws extends automatically to the Kleisli magmoids induced by them: the Kleisli category of the monad acts on the Kleisli magmoid of the non-associative monad.

**Theorem 4.12.** *In the setting of a sesquilaw,  $(S, T, m, n)$ , the Kleisli category of the distributive law,  $\mathbb{K}(m)$ , acts on the Kleisli magmoid of the almost distributive law,  $\mathbb{K}(n)$ .*

**Corollary 4.13.** *Normalized kernels admit an action from substochastic kernels, defined by  $p \triangleleft f = n(p \bullet; f)$ .*

$$(\triangleleft): \text{Norm}(X; Y) \times \text{subStoch}(Y; Z) \rightarrow \text{Norm}(X; Z).$$

That is, satisfying  $p \triangleleft \text{id} = p$  and  $p \triangleleft (f; g) = p \triangleleft f \triangleleft g$ .

## 5 Possibilistic normalization

Probability and possibility are related by supports: the support of a distribution is its subset of possible outcomes, forgetting about the specific probabilities of any of them. So far, we have defined three categories and one magmoid of probabilistic kernels; this section defines their possibilistic analogues and relates them by a support morphism.

Probabilistic	Possibilistic
Stochastic kernel	Affine relation
Substochastic kernel	Subaffine relation
Partial stochastic kernel	Partial affine relation
Normalized kernel	Relation

**Definition 5.1** (Affine powerset monad). The *affine powerset monad* (or, *non-empty finitary powerset monad*),  $R: \text{Set} \rightarrow \text{Set}$ , assigns to a set its non-empty finite subsets,

$$RX = \{U \subseteq X \mid U \text{ finite, and } U \neq \emptyset\}.$$

An *affine relation*,  $f: X \rightarrow RY$ , is a finitary relation such that, for each  $x \in X$ , there exists some  $y \in Y$  related to it,  $f(x; y)$ . Affine relations form a category,  $\text{affRel}$ , the Kleisli category of the affine powerset monad. Note how the *powerset functor* is  $PX \cong MRX$ .

**Definition 5.2.** *Post-selection*,  $p(-): RMX \rightarrow MRX$ , is the natural transformation that removes failure if it can and fails only if it must. Explicitly, it is defined by  $p(A \cup \{\perp\}) = p(A) = A$ , for each  $A \in RX$ , and  $p(\{\perp\}) = \emptyset$ . In other words, it is the join-preserving function satisfying  $p(\{x\}) = \{x\}$  and  $p(\{\perp\}) = \emptyset$ .

**Proposition 5.3.** *Post-selection is a monoidal sesquilaw.*

### 5.1 Support, a sesquilaw homomorphism

**Definition 5.4** (Support). The *support* of a distribution,  $d \in DX$ , is the subset

$$\text{supp}_X(d) = \{x \mid d(x) > 0\},$$

which cannot be empty because  $d$  is a full distribution. Support determines a family of functions,  $\text{supp}_X: DX \rightarrow RX$ , that extends to a natural transformation.

**Definition 5.5** (Sesquilaw homomorphism). A *sesquilaw homomorphism* between two sesquilaws,  $(S, T, \bullet_T, \circ_T)$  and  $(S, R, \bullet_R, \circ_R)$ , sharing their left monad, is a natural transformation  $\alpha: T \rightarrow R$  that is a monad homomorphism – meaning that  $\mu_T \circ \alpha = (\alpha \circ \alpha) \circ \mu_R$  and  $\eta_T \circ \alpha = \eta_R$  – and that, moreover, commutes with the sesquilaw – meaning that the following two diagrams commute.

$$\begin{array}{ccc} ST & \xrightarrow{S\alpha} & SR \\ \bullet_T \downarrow & & \downarrow \bullet_R \\ TS & \xrightarrow{\alpha S} & RS \end{array} \quad \begin{array}{ccc} ST & \xrightarrow{S\alpha} & SR \\ \circ_T \uparrow & & \uparrow \circ_R \\ TS & \xrightarrow{\alpha S} & RS \end{array}$$

**Proposition 5.6.** *Support,  $\text{supp}_X: DX \rightarrow RX$ , is a sesquilaw homomorphism between normalization and post-selection.*

### 5.2 Subaffine relations, partial affine relations

The possibilistic analogues of substochastic kernels ( $\text{subStoch}$ ) and partial stochastic kernels ( $\text{parStoch}$ ) are two lesser-known categories of relations: *may-must relations* [BSV22] (which we call *sub-affine relations*), helpful in trace semantics; and *Dijkstra relations* [CM09] (or, *partial affine relations*), helpful for program logics.

**Definition 5.7** (Subaffine relations). The category  $\text{subRel}$  of *sub-affine relations* (or, *may-must relations*) has, as morphisms, the failure-contemplating relations,  $f: X \rightarrow RMY$ , and composition  $(;)$  is defined by

$$\begin{aligned} (f; g)(x; z) &= \exists_{y \in Y} f(x; y) \wedge g(y; z), \\ (f; g)(x; \perp) &= f(x; \perp) \vee \exists_{y \in Y} f(x; y) \wedge g(y; \perp). \end{aligned}$$

**Definition 5.8** (Partial affine relations, [CM09, 5.14]). The category of *partial affine relations* (or, *relations with Dijkstra composition*),  $\text{parRel}$ , has relations as morphisms,  $f: X \rightarrow \text{MRY}$ , but composition ( $\star$ ) is instead defined by the formula

$$(f \star g)(x; z) = (\exists y \in Y f(x; y) \wedge g(y; z)) \wedge (\nexists y \in Y f(x; y) \wedge g(y; \perp)),$$

where  $g(y; \perp)$  means that  $g(y) = \perp$ . In other words, composition fails if there is a possibility for it to fail.

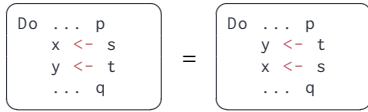
**Corollary 5.9** (Support functors). *Support extends to four strict monoidal identity-on-objects functors.*

- (1)  $\text{supp}: \text{Stoch} \rightarrow \text{affRel}$ ;
- (2)  $\text{supp}: \text{subStoch} \rightarrow \text{subRel}$ ;
- (3)  $\text{supp}: \text{parStoch} \rightarrow \text{parRel}$ ;
- (4)  $\text{supp}: \text{Norm} \rightarrow \text{Rel}$ .

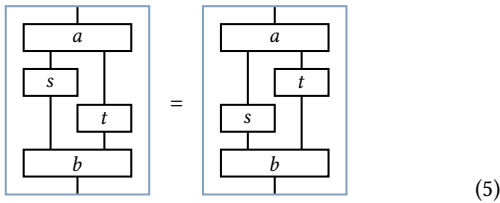
Arguably, then, the stochastic analogue of the category of relations is the *magmoid* of normalized kernels, even if it is only a magmoid. Conversely, from this point of view, what is unexpected is that the magmoid of relations happens to be a category.

## 6 Left and right commutativity

Probabilistic programming with monoidal magmoids raises subtleties. For instance, we expect *commutativity* for probabilistic programs [Sta17, WCGC18, JLMZ21]: reordering the lines of a program must not change its meaning, as long as the lines do not depend on each other. That is, whenever  $x$  is free in  $t$  and  $y$  is free in  $s$ , the following equation holds.



In terms of string diagrams, boxes may pass each other, as long as no directed path exists between them.

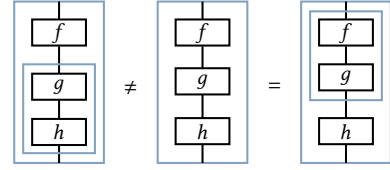


Commutativity is usually identified with monoidality and, specifically, with the interchange equation,  $(s \otimes \text{id}) \circ (\text{id} \otimes t) = (\text{id} \otimes t) \circ (s \otimes \text{id})$ . Perhaps surprisingly, magmoids distinguish monoidality and two analogues of commutativity.

Let us introduce, in Section 6.1, left commutative magmoids and right commutative magmoids, two analogues of commutativity in the left commutative magmoids.

**Remark 6.1** (Intermediate boxes). Before continuing, let us pick the convention — motivated by Remark 2.7 — that string diagrams

are, implicitly, left-associative.<sup>3</sup> This means that, to describe right-associating composition, we must write it explicitly, inside its own box; we depict these intermediate boxes as blue boxes.



This is much in the same way that, when declaring that a binary operation — say, composition ( $\circ$ ) — is left-associative, we must use parentheses only to express right-associating composition.

$$f \circ (g \circ h) \neq f \circ g \circ h = (f \circ g) \circ h.$$

### 6.1 Commutative magmoids

Commutativity does not follow from monoidality. Indeed, reading the string diagrams in Equation (5), we obtain the two following expressions, which are not equated by the axioms of monoidal magmoids.

$$((a \circ (s \otimes \text{id})) \circ (\text{id} \otimes t)) \circ b \neq ((a \circ (\text{id} \otimes t)) \circ (s \otimes \text{id})) \circ b.$$

Thus, commutativity becomes an independent axiom.

**Definition 6.2** (Commutative non-associative monad). A *commutative non-associative monad*,  $(T, \mu, \eta, u, v)$ , is a *monoidal non-associative monad* such that the following two rectangles commute.

$$\begin{array}{ccccc} TT(TX \otimes Y) & \xrightarrow{\mu} & T(TX \otimes Y) & \xrightarrow{T\sigma_L} & TT(X \otimes Y) \\ T\sigma_R \uparrow & & & & \downarrow \mu \\ T(TX \otimes TY) & \xrightarrow{Tu} & TT(X \otimes Y) & \xrightarrow{\mu} & T(X \otimes Y) \\ T\sigma_L \downarrow & & & & \uparrow \mu \\ TT(X \otimes TY) & \xrightarrow{\mu} & T(X \otimes TY) & \xrightarrow{T\sigma_R} & TT(X \otimes Y) \end{array}$$

Here,  $\sigma_L = (\text{id} \otimes \eta) \circ u$  and  $\sigma_R = (\eta \otimes \text{id}) \circ u$  are the left and right strengths of the monad.

**Definition 6.3** (Left commutative magmoid). A *left commutative magmoid* is a monoidal magmoid such that

$$\begin{aligned} (f \circ (g \otimes \text{id})) \circ (\text{id} \otimes h) &= f \circ (g \otimes h); \\ (f \circ (\text{id} \otimes h)) \circ (g \otimes \text{id}) &= f \circ (g \otimes h); \end{aligned}$$

while a *right commutative magmoid* is such that

$$\begin{aligned} (f \otimes \text{id}) \circ ((\text{id} \otimes g) \circ h) &= (f \otimes g) \circ h; \\ (f \circ (\text{id} \otimes h)) \circ (g \otimes \text{id}) &= f \circ (g \otimes h). \end{aligned}$$

**Proposition 6.4** (Monoidal monads are commutative). A *monoidal (associative!) monad* is, in particular, a *commutative non-associative monad*. In other words, in the presence of associativity, monoidality and commutativity coincide.

**Proposition 6.5.** *Monoidal sesquilaws induce commutative non-associative monads. Commutative non-associative monads induce Kleisli left commutative magmoids.*

<sup>3</sup>A similar discussion holds if we were to pick the opposite precedence: we would develop a theory of right commutative diagrams. This theory would be less convenient for our purposes, as we shall see later.



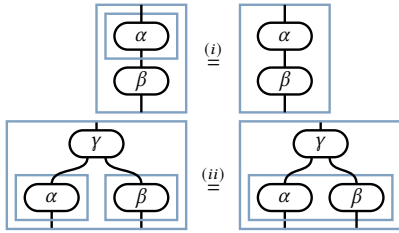
*Remark 6.6.* Commutative non-associative monads do not induce right commutative magmoids: in fact, normalized kernels form a left commutative magmoid that is not a right commutative magmoid.

## 6.2 String diagrams for commutative magmoids

Let us close this section by formalizing string diagrams for commutative magmoids. We prove that commutative magmoids are non-multiplicative algebras of string diagrams. This means that we have an interpretation for each string diagram of morphisms in a commutative magmoid, but this interpretation is not invariant under substitution. As a result, we need to keep the intermediate boxes of Remark 6.1.

**Theorem 6.7.** *Left commutative magmoids are non-multiplicative algebras of the string diagrams monad over tensor schemas, in the sense of Joyal and Street [JS91, Definition 1.4].*

Moreover, these algebras satisfy the following equations for any two diagrams  $\alpha$  and  $\beta$ : we call them (i) left-bias, and (ii) monoidality.



*Remark 6.8* (Reading string diagrams). The reader unfamiliar with string diagrams may prefer reading them directly as normalized kernels. One can read string diagram boxes as kernels by assigning a variable to each one of the wires that appear on it: all *inner wires* (those that do not touch the boundaries of the box) must be bound to a summation; all *nodes* (regarding any inner boxes as nodes) must be multiplied together; and the whole expression must be normalized at the end. As a shortcut, multiple wires connected by a copy or delete node are regarded as a single wire (c.f. [JS91, FL23]).

For instance, the left-hand side of the previous monoidality axiom (ii) reads as follows. It contains five wires: the input  $x$ ; two inner wires,  $u_1$  and  $u_2$ ; and two outputs,  $y$  and  $z$ . At the outer level, it contains three nodes:  $\gamma$ , and two boxes for  $\alpha$  and  $\beta$ . We multiply these three nodes together, we sum over the inner wires, and, finally, we divide by the same expression, now summing also over outputs.

$$\left[ \frac{\sum_{u_1, u_2} \gamma(x; u_1, u_2) \cdot \left[ \frac{\alpha(u_1; y)}{\sum_{y'} \alpha(u_1; y')} \right] \cdot \left[ \frac{\beta(u_2; z)}{\sum_{z'} \beta(u_2; z')} \right]}{\sum_{u_1, u_2, y, z} \gamma(x; u_1, u_2) \cdot \left[ \frac{\alpha(u_1; y)}{\sum_{y'} \alpha(u_1; y')} \right] \cdot \left[ \frac{\beta(u_2; z)}{\sum_{z'} \beta(u_2; z')} \right]} \right]$$

*Remark 6.9* (Normalization boxes). The previous result explains the emergence of *normalization boxes* in categorical probability [LT23, JSS25, DRS25]. String diagrams are insufficient to modulate the non-associativity of normalization, and multiple authors – even without arguing for a magmoid structure – employ these boxes.<sup>4</sup>

<sup>4</sup>Note, however, that the one presented here is different from other axiomatizations of *normalization boxes* [LT23, JSS25]. In particular, normalization is not asked to be the identity on quasitotal morphisms, which allows post-selection to become an example.

## 7 Markov magmoids

Until here, we studied general monoidal magmoids. Let us focus on the structure most relevant to probability. Markov magmoids are those commutative magmoids admitting conditionals, a synthetic analogue of Bayesian disintegration [CJ19, Fri20]. While inspired by *Markov categories*, we need to carefully adapt their definition to account for updates and non-associativity. Let us incrementally build it.

We first introduce the sesquilaw analogue of copy-discard categories: copy-discard sesquilaws, which separate an affine monad for probability and a relevant monad for partiality.

### 7.1 Copy-discard magmoids

In a cartesian monoidal category, morphisms can be freely copied and discarded [Fox76]. Copy-discard sesquilaws are a specialized form of sesquilaw where one of the monad carries discardable effects (affine: total but non-deterministic) and the other carries copyable effects (relevant: partial but deterministic). For these two classes of effects, we have two classes of monads.

**Definition 7.1** (Affine & relevant monoidal monads, [Jac94]). A monoidal monad,  $(T, \mu, \eta, u, v)$ , over a cartesian monoidal category  $(\mathbb{C}, \times, 1, \delta, \epsilon)$ , is *affine* (or, *total*) whenever  $\delta_{DX} \circ u_{X,X} = T\delta_X$ ; and it is *relevant* (or, *deterministic*) whenever  $\epsilon_{DX} \circ v = T\epsilon_X$ .<sup>5</sup>

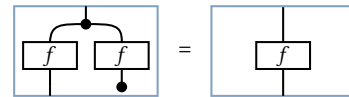
**Definition 7.2** (Copy-discard sesquilaw). A *copy-discard sesquilaw*,  $(S, T, m, n)$ , is a monoidal sesquilaw over a cartesian monoidal category,  $(\mathbb{C}, \times, 1, \delta, \epsilon)$ , between an affine monad,  $(T, \mu^T, \eta^T, u^T, v^T)$ , and a relevant monad,  $(S, \mu^S, \eta^S, u^S, v^S)$ .

**Definition 7.3** (Copy-discard magmoid). A *copy-discard magmoid*,  $(-)_\dagger : \mathbb{C} \rightarrow \mathbb{M}$ , is a symmetric commutative magmoid whose base,  $(\mathbb{C}, \times, 1, \delta, \epsilon)$ , is cartesian monoidal.

### 7.2 Quasitotal magmoids

An important axiom of Markov categories [Fri20] is *totality*: the unused output of a kernel can be discarded without affecting the computation. Totality can be refined to account for updates: the resulting notion is quasitotality [DR23, Sha25].

**Definition 7.4** (Quasitotal magmoid). A *quasitotal magmoid* is a copy-discard magmoid in which every morphism is quasitotal [DR23, Definition 3.1], meaning that it satisfies the following equation stating that it is discardable after copied.



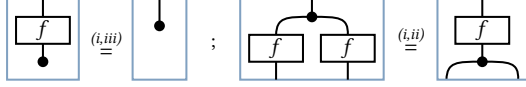
**Proposition 7.5.** *Any copy-discard sesquilaw induces a quasitotal magmoid. As a corollary, normalized kernels are quasitotal: and, exemplifying Remark 6.8, the following equation holds for any normalized kernel  $f : X \rightarrow \text{MDY}$ .*

$$\left[ \frac{\sum_{y'} f(x; y) \cdot f(x; y')}{\sum_{y, y'} f(x; y) \cdot f(x; y')} \right] = f(x; y).$$

<sup>5</sup>Equivalently, a monad is affine exactly when  $\eta_1 : 1 \rightarrow T1$  is an isomorphism. While the usual formulation of affine and relevant monads is in terms of strength [Jac94], we only require their monoidal version.

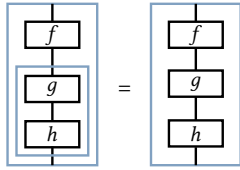


Still, we have four types of morphisms in a quasitotal magmoid: (i) base morphisms which can be copied and discarded; (ii) *total morphisms* (affine), which can be discarded but not copied; (iii) *deterministic morphisms* (relevant), which can be copied but not discarded; and (iv) quasitotal morphisms.



Our next axiom states how these associate; it has no correlate in Markov categories, where associativity is not an issue.

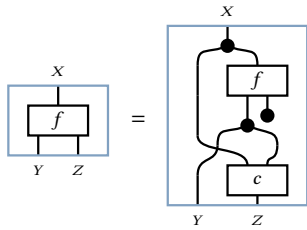
**Definition 7.6** (Left-relevant magmoid). A *left-relevant magmoid* is a copy-discard magmoid that satisfies associativity,  $f \circ (g \circ h) = (f \circ g) \circ h$ , if either  $f$  is deterministic,  $g$  is deterministic, or  $h$  is total.



### 7.3 Markov magmoids

Markov magmoids are well-behaved copy-discard magmoids that admit Bayesian inverses: in synthetic probability theory, the distinguishing axiom allowing Bayesian inverses is the existence of conditionals [CJ17, Fri20].

**Definition 7.7** (Conditional). A copy-discard magmoid admits *conditionals* if, for any morphism  $f: X \rightarrow Y \otimes Z$  there exists some  $c: X \otimes Y \rightarrow Z$  that recovers its second output conditioned on the first, meaning that the following holds.



**Definition 7.8** (Markov magmoid). A *Markov magmoid* is a left-relevant quasitotal magmoid admitting conditionals.

**Theorem 7.9.** *Normalized kernels form a Markov magmoid.*

*Remark 7.10.* Markov magmoid are more expressive and structured than copy-discard sesquialaws; still, most notions of probabilistic kernel – discrete or continuous – admit this extra structure. Let us now develop a continuous example of Markov magmoid: that of standard Borel spaces.

## 8 Normalization in standard Borel spaces

Normalization fails associativity analogously in the continuous case. Let us address measurable spaces next, applying the theory of sesquialaws. The discrete case remains conceptually clearer, though, and we will return to it for the last sections.

This section details the measurable case. We restrict all constructions to standard Borel spaces, which later ensure the existence of conditionals.

**Definition 8.1** (Giry monad, [Law62, Gir82]). The functor taking a measurable space to the set of probability measures over it,  $\mathcal{D}: \mathbf{BorelMes} \rightarrow \mathbf{BorelMes}$ , forms a monad,  $(\mathcal{D}, \eta, \mu)$ , whose unit,  $\eta_X: X \rightarrow \mathcal{D}X$ , is the Dirac delta,  $\eta_X(x) = \delta(x; -)$ , and whose multiplication,  $\mu_X: \mathcal{D}\mathcal{D}X \rightarrow \mathcal{D}X$ , is given by Lebesgue integration,

$$\mu_X(\Omega)(U) = \int_{v \in \mathcal{D}X} v(U) \cdot \Omega(dv).$$

It is moreover an affine monoidal monad over the cartesian monoidal structure of measurable sets, with the following product of measures,

$$(v_1 \otimes v_2)(U) = \int_{y \in Y} \left( \int_{x \in X} \xi_U(x, y) \cdot v_1(dx) \right) \cdot v_2(dy),$$

here,  $\xi$  is the indicator function and the product is commutative by the Fubini theorem. We work with the Kleisli category of the Giry monad restricted to standard Borel spaces, **BorelStoch**.

Normalization is a section to the distributive law that induces Panangaden's variant of the Giry monad.

**Definition 8.2** (Panangaden monad, [Pan99, §3]). Standard Borel spaces admit coproducts and thus a Maybe monad,  $\mathcal{M}: \mathbf{BorelMes} \rightarrow \mathbf{BorelMes}$ . There exists a distributive law,  $(-)_X^\bullet: \mathcal{D}\mathcal{M}X \rightarrow \mathcal{M}\mathcal{D}X$ , yielding a monad structure on  $\mathcal{D}\mathcal{M}$  (c.f. [DR23]); its Kleisli category is **BorelSubstoch**.

Instead of rederiving all of the properties of normalization, we can now use the framework of sesquialaws we just developed.

**Definition 8.3** (Continuous normalization). *Normalization* is the natural transformation  $N_X: \mathcal{D}\mathcal{M}X \rightarrow \mathcal{M}\mathcal{D}X$  defined by

$$N(v)(U) = \left[ \frac{v(U)}{v(X)} \right].$$

**Theorem 8.4.** *Normalization induces a monoidal sesquialaw.*

**Corollary 8.5** (Magmoid of normalized kernels). Normalized kernels between standard Borel spaces,  $X \rightarrow \mathcal{M}\mathcal{D}Y$ , form a monoidal non-associative category, **BorelNorm**, where composition of two kernels,  $f: X \rightarrow \mathcal{M}\mathcal{D}Y$  and  $g: Y \rightarrow \mathcal{M}\mathcal{D}Z$ , is defined by

$$(f \circ g)(x; U) = \left[ \frac{\int_{y \in Y} g(y; U) \cdot f(x; dy)}{\int_{y \in Y} g(y; Z) \cdot f(x; dy)} \right].$$

In other words, if we consider the associated substochastic kernels,  $f^\bullet: X \rightarrow \mathcal{D}\mathcal{M}Y$  and  $g^\bullet: Y \rightarrow \mathcal{D}\mathcal{M}Z$ , it is the normalization of their composition,  $f \circ g = N(f^\bullet \circ g^\bullet)$ . The tensor of normalized kernels coincides with the usual one: we define  $(f_1 \otimes f_2)((x_1, x_2); U)$  as

$$\int_{y_2 \in Y_2} \left( \int_{y_1 \in Y_1} \xi_U(y_1, y_2) \cdot f(x_1; dy_2) \right) \cdot f_2(x_2; dy_2),$$

The monoidal magmoid of normalized kernels, **BorelNorm**, has the cartesian monoidal category of standard Borel measurable spaces at its base, **BorelMes**.

**Corollary 8.6.** *Normalized kernels between standard Borel spaces admit a monoidal category action from substochastic kernels that is compatible with the tensor, defined by*

$$(p \triangleleft f)(x; U) = \frac{\int_{y \in Y} f(y; U) \cdot p(x; dy)}{\int_{y \in Y} f(y; Z) \cdot p(x; dy)}.$$

That is, satisfying  $(p_1 \triangleleft f_1) \otimes (p_2 \triangleleft f_2) = (p_1 \otimes p_2) \triangleleft (f_1 \otimes f_2)$ , with  $p \triangleleft \text{id} = p$  and  $p \triangleleft (f; g) = p \triangleleft f \triangleleft g$ .

**Theorem 8.7.** *BorelNorm is a Markov magmoid.*

**Remark 8.8.** Simplifying left-associating composition requires the *monotone convergence theorem*, exactly as associativity of the Girmonad does [Pan09, Proposition 5.2].

## 9 Discrete Markov magmoids

Markov magmoids can thus be discrete, or continuous. The discrete case allows for some more structure: essentially, it allows us to compare two values. Comparators in the continuous case, while available, collapse: e.g., if we sample two values from a normal distribution,  $X_1 \sim \mathcal{N}(0, 1)$  and  $X_2 \sim \mathcal{N}(0, 1)$ , then  $P(X_1 = X_2) = 0$ .

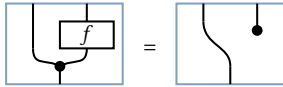
This section introduces discrete Markov magmoids and then, in Section 9.1, it employs their structure to derive results about causal inference: specifically, we prove the *back-door formula* (Proposition 9.6) and the *front-door formula* (Proposition 9.8). We conclude with an implementation in Section 9.2.

**Definition 9.1** (Discrete copy-discard magmoid). A *discrete copy-discard magmoid* is a copy-discard magmoid endowed with *comparators*, idempotent commutative semimonoids  $\mu_X: X \otimes X \rightarrow X$  such that  $\mu_{X \otimes Y} = (\text{id} \otimes \sigma \otimes \text{id}) \circ (\mu_X \otimes \mu_Y)$  and  $\mu_I = \text{id}$ , depicted as a black two-input dot, satisfying the partial Frobenius equations.



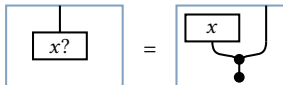
Comparators allow, in the discrete case, to characterize which normalized kernels have full support. That is, when, from each  $x \in X$ , there is a non-zero possibility of sampling each  $y \in Y$ , meaning that  $f(x; y) > 0$  for every  $y \in Y$ .

**Definition 9.2** (Full support). A morphism of a discrete copy-discard magmoid,  $f: X \rightarrow Y$ , has *full support* whenever the following equation holds.

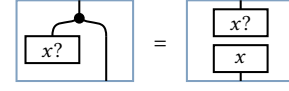


In the following sense, comparators can be used to force the value of a wire: sampling a value from any channel and then comparing with  $x_0 \in X$  outputs  $x_0$ . In the next section, we shall see how this forcing can act both an evidential update or a causal intervention, depending on associativity.

**Proposition 9.3** (Exact observations). *In any discrete Markov magmoid, the exact observation induced by any base morphism,  $x: 1 \rightarrow X$ , is the morphism  $x?: X \rightarrow 1$  defined as follows.*

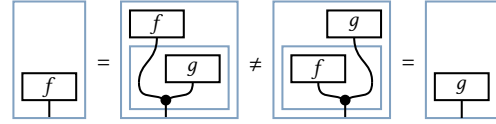


*Exact observations satisfy the following equation.*



**Definition 9.4.** A *discrete Markov magmoid* is a left-relevant quasitotal discrete copy-discard magmoid admitting conditionals.

**Remark 9.5** (Failure of right commutativity). We mentioned in Section 6.1 that normalized kernels do not form a right commutative magmoid. We now have the language to produce a direct counterexample. Consider two coins with different non-trivial biases, e.g.,  $f = 1/2 |a\rangle + 1/2 |b\rangle$  and  $g = 1/3 |a\rangle + 2/3 |b\rangle$ .



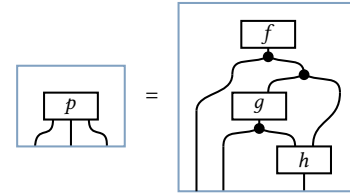
Here, we use that both coins have full support.

### 9.1 Inference in Discrete Markov Magmoids

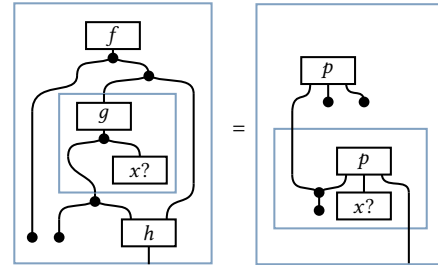
Discrete Markov magmoids allow us to prove basic lemmas for synthetic causal inference (c.f. [Pea09]). As a feature, they distinguish between interventions and observations by reparenthesizing.

Jacobs, Széles, and Stein [JSS25] have recently shown how partial Markov categories [DR23], extended with normalization boxes [LT23], can be applied to synthetic causality (c.f. [YZ22, Pie23]). We now derive synthetic causality from the axioms of discrete Markov magmoids.

**Proposition 9.6** (Back-door adjustment formula). *In a discrete Markov magmoid, let a joint state,  $p: 1 \rightarrow U \otimes X \otimes Y$ , admit the following factorization into total morphisms where, moreover,  $f$  has full support.*



*Then, the following equation holds.*

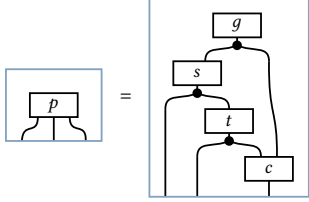


**Corollary 9.7.** *Let any distribution,  $p$ , with full support, on three visible variables,  $U$ ,  $X$ , and  $Y$ , such that  $U$  influences  $X$  and  $Y$ , and that  $X$  influences  $Y$ . Then, an intervention on  $X$  can be rewritten as*

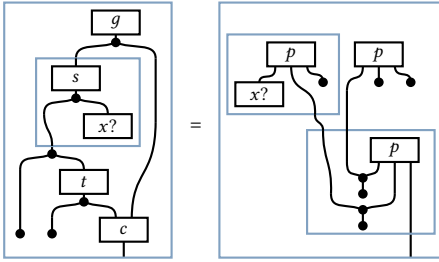
$$p_{\text{do}(x)}(y) = \sum_u \sum_{x', y'} p(u, x', y') \cdot \frac{p(u, x, y)}{\sum_{y' \in Y} p(u, x, y')}.$$

Pearl's front-door formula [Pea09] is slightly more complicated, and its proof uses all the axioms of Markov magmoids.

**Proposition 9.8** (Front-door adjustment formula). *In a discrete Markov magmoid, let a joint state,  $p: 1 \rightarrow X \otimes Z \otimes Y$ , admit the following factorization into total morphisms where, moreover, both  $t$  and  $(g \circ s)$  below have full support.*



Then, the following equation holds.



In other words, an intervention on the variable  $X$  can be rewritten as a composition, in the Markov magmoid, of the observational data.

**Corollary 9.9.** *Given any normalized distribution,  $p$ , on three visible variables,  $X$ ,  $Z$  and  $Y$ , and one hidden variable  $U$ , assume that  $U$  only influences  $X$  and  $Y$ , and that the influence of  $X$  on  $Y$  is entirely mediated by  $Z$ .*

$$p_{do x}(y) = \sum_z \frac{\sum_{y'} p(x, z, y')}{\sum_{y', z'} p(x, z', y')} \cdot \sum_{x', y', z'} \frac{p(x', z, y)}{\sum_{y'} p(x', z, y')} \cdot p(x', z', y')$$

## 9.2 Magmoidal programming

Inference problems may also be described in a programming language; this section extends the monadic metalanguage to the magmoidal case. The monadic metalanguage is implicitly right-associative,<sup>6</sup> but we will also define — and mostly use — a left-associative version. This novelty means that the language admits a *normalized by construction* semantics.

For the rest of the paper, definitions and computations are also implemented in the attached Racket v9.0 code (🔗, [DRS26]).

**Definition 9.10** (Right do-notation 🔗). Almost monadic right-associating do-notation, for a  $\text{Set}$ -based almost monad,  $(T, \mu, \eta)$ , is inductively defined by the following two clauses, for any element  $x \in X$  and any computation  $f \in TX$ , with  $x$  any variable potentially

bound in the do-notation expression  $p$ .

$$\begin{aligned} \text{rDo } T \text{ return } x &\equiv \eta(x); \\ \text{rDo } T \begin{array}{l} x \leftarrow f \\ p \dots \end{array} &\equiv (\lambda x. \text{rDo } T \begin{array}{l} p \dots \end{array})^\dagger(f). \end{aligned}$$

**Definition 9.11** (Left do-notation 🔗). Almost monadic left associating do-notation, for an almost monad  $T: \text{Set} \rightarrow \text{Set}$ , is inductively defined by strengthening the induction hypothesis accumulating an expression  $a$  and a variable  $x$ ,

$$\begin{aligned} \text{accDo } t \ x \ a \text{ return } y &\equiv \text{rDo } t \begin{array}{l} x \leftarrow a \\ \text{return } y \end{array} \\ \text{accDo } t \ x \ a \begin{array}{l} y \leftarrow f \\ p \dots \end{array} &\equiv \text{accDo } t \ (y \cdot x) \begin{array}{l} \text{rDo } t \begin{array}{l} x \leftarrow a \\ y \leftarrow f \\ \text{return } (y \cdot x) \end{array} \\ \text{return } y \end{array} \end{aligned}$$

and finally evaluating on an empty accumulated expression.

$$\text{lDo } t \begin{array}{l} p \dots \end{array} \equiv \text{accDo } m \ '() \begin{array}{l} \text{rDo } m \text{ return } '() \\ p \dots \end{array}$$

In other words, we *left-fold* the statements.

**Example 9.12** (Monty Hall problem, 🔗). As we saw in the introduction, the Monty Hall problem has two solutions, depending on associativity. Consider the following left-associated program: each of the non-zero numbered lines corresponds exactly to each computation step we took in the Introduction (Section 1).

```
0 lDo Norm
1   car <- (uniform 'left 'middle 'right)
2   opened <- (host car 'middle)
3   '() <- (observe opened 'left)
4   return car
```

Left-associating do-notation gives the usual answer to the Monty Hall problem: that changing doors is twice as likely to get the prize. Evaluating the previous expression, we obtain  $1/3 |L\rangle + 2/3 |R\rangle$ . Expanding the program according to Definition 9.11 yields the following nested program.

```
rDo Norm
  '(opened car) <- (rDo Norm
    '(opened car) <- (rDo Norm
      '(opened car) <- (rDo Norm
        '(car) <- (rDo Norm
          '() <- (rDo Norm
            return '()
          '(car) <- (uniform 'left 'middle 'right)
          return '(car))
        opened <- (host car 'middle)
        return '(opened car))
      (observe opened 'left)
      return '(opened car))
    return '(opened car))
  return car
```

<sup>6</sup>Indeed, only right-associative do-notation is implemented in functional programming languages, e.g. Haskell. The rationale is that only lawful monads may be considered.

Because of the failure of associativity, this is different from the program we would obtain just by reading the original Monty Hall problem in a right-associative fashion. Instead, right-associating do-notation answers that we are equally likely to win on any of the two doors:  $1/2 |L\rangle + 1/2 |R\rangle$ .

```
rDo Norm
car <- (uniform 'left 'middle 'right)
opened <- (host car 'middle)
'() <- (observe opened 'left)
return car
```

The reader will notice that, conversely, we could simulate this right-associating do-notation block in terms of nested left-associating do-notation.

```
lDo Norm
car <- (uniform 'left 'middle 'right)
opened <- (lDo Norm
  opened <- (host car 'middle)
  '() <- (observe opened 'left)
  return opened)
return car
```

*Example 9.13* (Smoking causes cancer, 🚬). Let us take a classical example from Pearl’s work on causality. The following (unrealistic) observational data could perhaps suggest that smoking has an innocuous or protective effect on cancer.

SURVEY		no-cancer	cancer
smoker	tar	323	57
smoker	no-tar	18	2
nonsmoker	tar	1	19
nonsmoker	no-tar	38	342

To actually quantify the effect of smoking on cancer, in principle, we would need data extracted from an interventional study forcing patients to smoke — which, previsibly, we want to avoid. That is, we would need to compute the left-hand side of the concluding equation of Proposition 9.8. Instead, by Proposition 9.8, we may compute the right-hand side, which gets translated as follows.

```
lDo Norm
z <- (lDo Norm
  (list xi z yi) <- p
  '() <- (observe xq xi)
  return z)
x <- (lDo Norm
  (list x zi yi) <- p
  return x)
y <- (lDo Norm
  (list xi zi y) <- p
  '() <- (observe x xi)
  '() <- (observe z zi)
  return y)
return y
```

Using only survey data, we compute that — even with unrealistic observational data — the incidence of cancer for a smoker ( $219/400 |C\rangle + 181/400 |NC\rangle$ ) is larger than that for a non-smoker ( $201/400 |C\rangle + 199/400 |NC\rangle$ ). We shall conclude that smoking causes cancer.

From this perspective, causal inference is the resolution of equations in discrete Markov magmoids. Alternatively to most probabilistic programming literature, we do not use a normalization operator, but the ability to modulate associativity of a Markov magmoid.

## 10 Conclusions

Normalization has been under-explored in categorical probability theory and denotational probabilistic semantics; most inference semantics either ignore normalization — preferring substochastic kernels or unnormalized kernels instead — or treat it as an ad-hoc operator, a feature of the semantics.

As a result, most semantic universes for probabilistic programming are not *normalized by construction*. We have introduced the Markov magmoid of normalized stochastic kernels as a first example of *normalized by construction* denotational semantics for probabilistic inference.

Our proposed explanation for this gap is that normalization arises from a failure of associativity, and failures of associativity are counterintuitive. For instance, we could naïvely say that, to solve an inference problem, one must (1) set up a prior distribution, (2) compute a stochastic kernel, and (3) update the prior with the observation. This description misses an essential point: how to associate these instructions or, in other words, *when to normalize*. Any normalized probabilistic semantics needs to clearly address this point, and a revision of existing categorical probabilistic semantics to understand their potential non-associative behaviour is warranted.

While non-associativity could seem too high of a price to pay, being explicit about associativity has conceptual advantages: it makes it possible to derive multiple properties of normalization from a few axioms — rendering the continuous case much simpler than it would be otherwise — and it allows us to distinguish between observations and interventions as two different parenthesizations of the same expression.

We abstracted this algebra by introducing *sesquilaws*: almost-distributive laws that are multiplicative only up to idempotent. That normalization is not a distributive law may be folklore, but it seems absent from the literature. We went one step further and classified this particular failure of normalization.

Monoidal sesquilaws induce commutative magmoids, another newly introduced notion refining the non-associative case. For commutative magmoids, we provided both a graphical calculus and a programming notation. Remarkably, these seem to mostly coincide — although not exactly — with the existing string diagrammatic calculi for normalization over *partial Markov categories* [DR23, LT23, JSS25]; we ground these axiomatizations in formal category theory via sesquilaws and commutative magmoids.

Finally, we proposed a different approach to categorical causality: we saw how solving causality problems corresponds to solving equations on a discrete Markov magmoid. While Pearl’s *do-calculus*, more than a synthetic axiomatization, is a collection of rules on top of probabilistic reasoning, we propose an algebraic presentation of the rules needed to discuss causality problems. Further work on automating these solutions — e.g. via rewriting, showing completeness for *identifiability* — is warranted and not covered by this manuscript; it is a promising avenue for a categorical semantics of causality and causal programming.

More generally, we reassert — as weak distributive laws, duploids, or mass-chance interpretations do — that failing distributive laws contain, in many cases, mathematical structure worth studying on its own.



# Acknowledgements

We thank Soichiro Fujii, Bart Jacobs, Jack Liell-Cock, Paolo Perrone, Seo Jin Park, Zev Shirazi, Ana Sokolova, Sam Staton, Yun Chen Tsai, and Ruben van Belle for discussion on earlier drafts of this article.

# References

- [BDGS16] Johannes Borgström, Ugo Dal Lago, Andrew D. Gordon, and Marcin Szymczak. A lambda-calculus foundation for universal probabilistic programming. In Jacques Garrigue, Gabriele Keller, and Eijiro Sumii, editors, *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 33–46. ACM, 2016. doi:10.1145/2951913.2951942.
- [Bec69] Jon Beck. Distributive laws. In *Seminar on triples and categorical homology theory*, pages 119–140. Springer, 1969.
- [Bén00] Jean Bénabou. Distributors at work. *Lecture notes written by Thomas Streicher*, 11, 2000.
- [Böh10] Gabriella Böhm. The weak theory of monads. *Advances in Mathematics*, 225(1):1–32, 2010.
- [BSV22] Filippo Bonchi, Ana Sokolova, and Valeria Vignudelli. The theory of traces for systems with nondeterminism, probability, and termination. *Logical Methods in Computer Science*, 18(2), 2022. doi:10.46298/LMCS-18(2:21)2022.
- [CJ17] Kenta Cho and Bart Jacobs. Kleisli semantics for conditioning in probabilistic programming. 2017.
- [CJ19] Kenta Cho and Bart Jacobs. Disintegration and Bayesian Inversion via String Diagrams. *Mathematical Structures in Computer Science*, pages 1–34, March 2019, 1709.00322. doi:10.1017/S0960129518000488.
- [CM09] Robin Cockett and Ernie Manes. Boolean and classical restriction categories. *Mathematical Structures in Computer Science*, 19(2):357–416, 2009. doi:10.1017/S0960129509007543.
- [CMN<sup>+</sup>25] Corina Cirstea, Lawrence S. Moss, Victoria Noquez, Todd Schmid, Alexandra Silva, and Ana Sokolova. A complete inference system for probabilistic infinite trace equivalence. In *EACSL Annual Conference on Computer Science Logic (CSL)*, 2025. doi:10.4230/LIPICCS.CSL.2025.30.
- [DGHM09] Yuxin Deng, Rob J. van Glabbeek, Matthew Hennessy, and Carroll Morgan. Testing finitary probabilistic processes. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, volume 5710 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2009. doi:10.1007/978-3-642-04081-8\_19.
- [DKPS23] Swaraj Dash, Younesse Kaddar, Hugo Paquet, and Sam Staton. Affine monads and lazy structures for bayesian programming. *Proc. ACM Program. Lang.*, 7(POPL):1338–1368, 2023. doi:10.1145/3571239.
- [DR23] Elena Di Lore and Mario Román. Evidential decision theory via partial markov categories. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–14. IEEE, 2023. doi:10.1109/LICS56636.2023.10175776.
- [DRS25] Elena Di Lore, Mario Román, and Paweł Sobociński. Partial Markov categories. *arXiv preprint*, 2025. doi:10.48550/arXiv.2502.03477.
- [DRS26] Elena Di Lore, Mario Román, and Márk Széles. Racket v9.0 Implementation of the Markov Magmoid of Normalized Stochastic Kernels, 2026. doi:10.5281/zenodo.18348005.
- [EPT11] Thomas Ehrhard, Michele Pagani, and Christine Tasson. The computational meaning of probabilistic coherence spaces. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 87–96. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.29.
- [EPT17] Thomas Ehrhard, Michele Pagani, and Christine Tasson. Measurable cones and stable, measurable functions: a model for probabilistic higher-order programming. *Proceedings of the ACM on Programming Languages*, 2(POPL):1–28, 2017. doi:10.1145/3158147.
- [FGL<sup>+</sup>25a] Tobias Fritz, Tomás Gonda, Antonio Lorenzin, Paolo Perrone, and Areeb Shah Mohammed. Empirical measures and strong laws of large numbers in categorical probability. *arXiv preprint arXiv:2503.21576*, 2025.
- [FGL<sup>+</sup>25b] Tobias Fritz, Tomás Gonda, Antonio Lorenzin, Paolo Perrone, and Areeb Shah-Mohammed. Empirical measures and strong laws of large numbers in categorical probability. *CoRR*, abs/2503.21576, 2025, 2503.21576. doi:10.48550/ARXIV.2503.21576.
- [FK23] Tobias Fritz and Andreas Klingler. The d-separation criterion in categorical probability. *J. Mach. Learn. Res.*, 24:46:1–46:49, 2023. URL <https://jmlr.org/papers/v24/22-0916.html>.
- [FL23] Tobias Fritz and Wendong Liang. Free GS-Monoidal categories and free Markov categories. *Applied Categorical Structures*, 31(2):21, 2023. doi:10.1007/S10485-023-09717-0.
- [Fon13] Brendan Fong. Causal Theories: A Categorical Perspective on Bayesian Networks. *Master’s Thesis, University of Oxford. ArXiv preprint arXiv:1301.6201*, 2013.
- [Fox76] Thomas Fox. Coalgebras and Cartesian Categories. *Communications in Algebra*, 4(7):665–667, 1976.
- [FPR21] Tobias Fritz, Paolo Perrone, and Sharwin Rezagholi. Probability, valuations, hyperspace: Three monads on top and the support as a morphism. *Mathematical Structures in Computer Science*, 31(8):850–897, 2021.
- [FR19] Claudia Faggian and Simona Ronchi Della Rocca. Lambda calculus and probabilistic computation. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785699.
- [FR20] Tobias Fritz and Eigil Fjeldgren Rischel. Infinite products and zero-one laws in categorical probability. *Compositionality*, 2:3, 2020.
- [Fri09] Tobias Fritz. Convex spaces I: Definition and examples. *arXiv preprint arXiv:0903.5522*, 2009.
- [Fri20] Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020, 1908.07021.
- [Gar18] Richard Garner. Abstract hypernormalisation, and normalisation-by-trace-evaluation for generative systems. *arXiv preprint arXiv:1811.02710*, 2018, 1811.02710.
- [Gar20] Richard Garner. The Vietoris monad and weak distributive laws. *Applied Categorical Structures*, 28(2):339–354, 2020.
- [Gir82] Michèle Giry. A categorical approach to probability theory. In *Categorical aspects of topology and analysis*, pages 68–85. Springer, 1982.
- [GMR<sup>+</sup>12] Noah D. Goodman, Vikash Mansinghka, Daniel M. Roy, Kallista A. Bonawitz, and Joshua B. Tenenbaum. Church: a language for generative models. *CoRR*, abs/1206.3255, 2012, 1206.3255. URL <http://arxiv.org/abs/1206.3255>.
- [Gol99] Petr Viktorovich Golubtsov. Axiomatic description of categories of information transformers. *Problemy Peredachi Informatsii*, 35(3):80–98, 1999.
- [GP20] Alexandre Goy and Daniela Petrisan. Combining weak distributive laws: Application to up-to techniques. *CoRR*, abs/2010.00811, 2020, 2010.00811.
- [Jac94] Bart Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69(1):73–106, 1994. doi:https://doi.org/10.1016/0168-0072(94)90020-5.
- [Jac17] Bart Jacobs. Hyper normalisation and conditioning for discrete probability distributions. *Log. Methods Comput. Sci.*, 13(3), 2017. doi:10.23638/LMCS-13(3:17)2017.
- [Jef97] Alan Jeffrey. Premonoidal categories and flow graphs. *Electron. Notes Theor. Comput. Sci.*, 10:51, 1997. doi:10.1016/S1571-0661(05)80688-7.
- [JKZ21] Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference via string diagram surgery: A diagrammatic approach to interventions and counterfactuals. *Mathematical Structures in Computer Science*, 31(5):553–574, 2021.
- [JLMZ21] Xiaodong Jia, Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhev. Commutative monads for probabilistic programming languages. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–14. IEEE, 2021. doi:10.1109/LICS52264.2021.9470611.
- [JS91] André Joyal and Ross Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88(1):55–112, 1991. doi:10.1016/0001-8708(91)90003-P.
- [JSS25] Bart Jacobs, Márk Széles, and Dario Stein. Compositional inference for Bayesian networks and causality. In *Mathematical Foundations of Programming Semantics*, 2025.
- [Koz81] Dexter Kozen. Semantics of probabilistic programs. *J. Comput. Syst. Sci.*, 22(3):328–350, 1981. doi:10.1016/0022-0000(81)90036-2.
- [Law62] F William Lawvere. The category of probabilistic mappings. *Unpublished preprint*, 1962. Preprint available online, at <https://ncatlab.org/nlab/files/lawvereprobability1962.pdf>.
- [LS24] Jack Liell-Cock and Sam Staton. Compositional imprecise probability. *CoRR*, abs/2405.09391, 2024, 2405.09391. doi:10.48550/ARXIV.2405.09391.
- [LT23] Robin Lorenz and Sean Tull. Causal models in string diagrams, 2023, 2304.07638.
- [LZ12] Ugo Dal Lago and Margherita Zorzi. Probabilistic operational semantics for the lambda calculus. *RAIRO Theor. Informatics Appl.*, 46(3):413–450, 2012. doi:10.1051/ITA/2012012.
- [ML71] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer Verlag, 1971. doi:10.1007/978-1-4757-4721-8.

- [MMM25] Éléonore Mangel, Paul-André Mellès, and Guillaume Munch-Maccagnoni. Classical notions of computation and the Hasegawa-Thielecke theorem. *arXiv preprint arXiv:2502.13033*, 2025.
- [MPYW18] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An introduction to probabilistic programming. *CoRR*, abs/1809.10756, 2018. doi:10.10756.
- [MSV21] Matteo Mio, Ralph Sarkis, and Valeria Vignudelli. Combining nondeterminism, probability, and termination: Equational and metric reasoning. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–14. IEEE, 2021. doi:10.1109/LICS52264.2021.9470717.
- [Mun13] Guillaume Munch-Maccagnoni. *Syntax and Models of a non-Associative Composition of Programs and Proofs. (Syntaxe et modèles d'une composition non-associative des programmes et des preuves)*. PhD thesis, Paris Diderot University, France, 2013. URL <https://tel.archives-ouvertes.fr/tel-00918642>.
- [NCR<sup>+</sup>16] Praveen Narayanan, Jacques Carette, Wren Romano, Chung-chieh Shan, and Robert Zinkov. Probabilistic inference by program transformation in Hakaru (system description). In *International Symposium on Functional and Logic Programming - 13th International Symposium, FLOPS 2016, Kochi, Japan, March 4-6, 2016, Proceedings*, pages 62–79. Springer, 2016. doi:10.1007/978-3-319-29604-3\_5.
- [Pan99] Prakash Panangaden. The Category of Markov Kernels. *Electronic Notes in Theoretical Computer Science*, 22:171–187, January 1999. doi:10.1016/S1571-0661(05)80602-4.
- [Pan09] Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009.
- [Par03] Sungwoo Park. A calculus for probabilistic languages. In Zhong Shao and Peter Lee, editors, *Proceedings of TLDI'03: 2003 ACM SIGPLAN International Workshop on Types in Languages Design and Implementation, New Orleans, Louisiana, USA, January 18, 2003*, pages 38–49. ACM, 2003. doi:10.1145/604174.604180.
- [Pea09] Judea Pearl. *Causality*. Cambridge University Press, 2009.
- [Pie23] Robin Piedeleu. Basic causal inference via string diagrams. *Blog post*, 2023. Available at <https://piedeleu.com/posts/diagrammatic-causal-inference/>. Accessed on January 2026. Archived at <https://web.archive.org/web/20241116235642/https://piedeleu.com/posts/diagrammatic-causal-inference/>.
- [PPT08] Sungwoo Park, Frank Pfenning, and Sebastian Thrun. A probabilistic language based on sampling functions. *ACM Trans. Program. Lang. Syst.*, 31(1):4:1–4:46, 2008. doi:10.1145/1452044.1452048.
- [PR97] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Math. Struct. Comput. Sci.*, 7(5):453–468, 1997. doi:10.1017/S0960129597002375.
- [PTRSZ25] Robin Piedeleu, Mateo Torres-Ruiz, Alexandra Silva, and Fabio Zanasi. A complete axiomatisation of equivalence for discrete probabilistic programming. In *European Symposium on Programming*, pages 202–229. Springer, 2025.
- [PW14] Brooks Paige and Frank D. Wood. A compilation target for probabilistic programming languages. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1935–1943. JMLR.org, 2014. URL <http://proceedings.mlr.press/v32/paige14.html>.
- [Sel75] Steve Selvin. Letters to the editor. *The American Statistician*, 29(1):67–71, 1975. doi:10.1080/00031305.1975.10479121.
- [Sha25] Areeb Shah Mohammed. Partializations of Markov categories, 2025. 2509.05094. URL <https://arxiv.org/abs/2509.05094>.
- [Sim17] Alex Simpson. Probability sheaves and the Giry monad. In Filippo Bonchi and Barbara König, editors, *7th Conference on Algebra and Coalgebra in Computer Science, CALCO 2017, Ljubljana, Slovenia, June 12-16, 2017*, volume 72 of *LIPICs*, pages 1:1–1:6. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.CALCO.2017.1.
- [Sim24] Alex Simpson. Equivalence and conditional independence in atomic sheaf logic. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 70:1–70:14. ACM, 2024. doi:10.1145/3661814.3662132.
- [SL13] Sam Staton and Paul Blain Levy. Universal properties of impure programming languages. In Roberto Giacobazzi and Radhia Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, pages 179–192. ACM, 2013. doi:10.1145/2429069.2429091.
- [SS00] Eugene W Stark and Scott A Smolka. A complete axiom system for finite-state probabilistic processes. In *Proof, language, and interaction: essays in honour of Robin Milner*, pages 571–595. 2000.
- [SS11] Alexandra Silva and Ana Sokolova. Sound and complete axiomatization of trace semantics for probabilistic systems. In Michael W. Mislove and Joël Ouaknine, editors, *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25-28, 2011*, volume 276 of *Electronic Notes in Theoretical Computer Science*, pages 291–311. Elsevier, 2011. doi:10.1016/j.entcs.2011.09.027.
- [SS24] Dario Stein and Sam Staton. Probabilistic programming with exact conditions. *J. ACM*, 71(1):2:1–2:53, 2024. doi:10.1145/3632170.
- [Sta17] Sam Staton. Commutative semantics for probabilistic programming. In *European Symposium on Programming*, pages 855–879. Springer, 2017.
- [Ste25] Dario Stein. Random variables, conditional independence and categories of abstract sample spaces. In *40th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2025, Singapore, June 23-26, 2025*, pages 472–484. IEEE, 2025. doi:10.1109/LICS65433.2025.00042.
- [Str09] Ross Street. Weak distributive laws. *Theory and Applications of Categories*, 22:313–320, 2009.
- [SW18] Ana Sokolova and Harald Woracek. Termination in convex sets of distributions. *Log. Methods Comput. Sci.*, 14(4), 2018. doi:10.23638/LMCS-14(4:17)2018.
- [SWY<sup>+</sup>16] Sam Staton, Frank Wood, Hongseok Yang, Chris Heunen, and Ohad Kammar. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In *2016 31st annual ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 1–10. IEEE, 2016.
- [TPAH25] Yun Chen Tsai, Kittiphon Phalakarn, S. Akshay, and Ichiro Hasuo. Chance and mass interpretations of probabilities in markov decision processes. In Patricia Bouyer and Jaco van de Pol, editors, *36th International Conference on Concurrency Theory, CONCUR 2025, Aarhus, Denmark, August 26-29, 2025*, volume 348 of *LIPICs*, pages 33:1–33:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.CONCUR.2025.33.
- [TvdMYW16] David Tolpin, Jan-Willem van de Meent, Hongseok Yang, and Frank D. Wood. Design and implementation of probabilistic programming language anglican. In Tom Schrijvers, editor, *Proceedings of the 28th Symposium on the Implementation and Application of Functional Programming Languages, IFL 2016, Leuven, Belgium, August 31 - September 2, 2016*, pages 6:1–6:12. ACM, 2016. doi:10.1145/3064899.3064910.
- [Ver17] Dominic Verdon. Coherence for braided and symmetric pseudomonoids. *CoRR*, abs/1705.09354, 2017. 1705.09354.
- [VKS19] Matthijs Vákár, Ohad Kammar, and Sam Staton. A domain theory for statistical probabilistic programming. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.
- [vS] Marilyn vos Savant. Parade 16: Ask Marilyn (Archived). <https://web.archive.org/web/20130121183432/http://marilynvossavant.com/game-show-problem/>. Accessed: 2013-01-21.
- [VW06] Daniele Varacca and Glynn Winskel. Distributing probability over non-determinism. *Math. Struct. Comput. Sci.*, 16(1):87–113, 2006. doi:10.1017/S0960129505005074.
- [WCGC18] Mitchell Wand, Ryan Culpepper, Theophilos Giannakopoulos, and Andrew Cobb. Contextual equivalence for a probabilistic language with continuous random variables and recursion. *Proc. ACM Program. Lang.*, 2(ICFP):87:1–87:30, 2018. doi:10.1145/3236782.
- [YZ22] Yimu Yin and Jiji Zhang. Markov categories, causal theories, and the do-calculus, 2022. 2204.04821.
- [ZM22] Maaike Zwart and Dan Marsden. No-go theorems for distributive laws. *Logical Methods in Computer Science*, Volume 18, Issue 1, Jan 2022. doi:10.46298/lmcs-18(1:13)2022.

## A Proofs for Section 1 (Introduction)

**Definition A.1** (Extension). Given two monads,  $(S, \mu^S, \eta^S)$  and  $(T, \mu^T, \eta^T)$ , an  $S$ -extension of the  $T$ -algebra  $(A, \alpha)$  is a  $T$ -algebra  $(SA, \alpha^\#)$  such that  $\alpha \circ \eta_A^S = T(\eta_A^S) \circ \alpha^\#$ .

$$\begin{array}{ccc} TA & \xrightarrow{T\eta^S} & TSA \\ \alpha \downarrow & & \downarrow \alpha^\# \\ A & \xrightarrow{\eta^S} & SA \end{array}$$

*Remark A.2.* A *convex algebra* is an algebra for the finitely-supported distribution monad  $(D)$ . A *one-point extension* of a convex algebra  $(A, \alpha)$ , in the sense of Sokolova and Woracek [SW18] is precisely an  $M$ -extension, an extension over the maybe monad.

**Theorem A.3** (Black-hole is unique [SW18, Theorem 5.3]). *The black-hole one-point extension, defined by*

$$\alpha^\#(d) = \begin{cases} d & \text{if } d(\perp) = 0 \\ \perp & \text{if } d(\perp) > 0 \end{cases}$$

*is the only functorial one-point extension of convex algebras.*

It is well-known that a distributive law  $\psi: TS \rightarrow ST$  of a monad  $S$  over another monad  $T$  gives a lifting of the monad  $S$  to the category of algebras of  $T$  [Bec69]. The lifted monad acts on algebras by extending them.

**Proposition A.4** (Extensions from distributive laws). *Any distributive law,  $\psi: TS \rightarrow ST$ , determines a functorial  $S$ -extension of  $T$ -algebras.*

*PROOF.* Given a distributive law,  $\psi: TS \rightarrow ST$  and a  $T$ -algebra  $(A, \alpha)$ , consider the morphism  $\alpha^\# = \psi_A \circ S(\alpha)$ . The lifted monad,  $\hat{S}$ , following Beck's work [Bec69], acts on objects  $(A, \alpha)$  by  $\hat{S}(A, \alpha) = (S(A), \alpha^\#)$ .

Let us check that  $(SA, \alpha^\#)$  is an  $S$ -extension of  $(A, \alpha)$ . We use (i) Beck's construction, (ii)  $S$ -unitality of the distributive law, and (iii) naturality.

$$T(\eta_A^S) \circ \alpha^\# \stackrel{(i)}{=} T(\eta_A^S) \circ \psi_A \circ S(\alpha) \stackrel{(ii)}{=} \eta_{TA}^S \circ S(\alpha) \stackrel{(iii)}{=} \alpha \circ \eta_A^S$$

This concludes the proof.  $\square$

**Corollary 1.1** (from [SW18, Theorem 5.3]). *Black-hole semantics (Proposition 3.5) determines the only distributive law between the distribution monad and the maybe monad,  $DM \rightarrow MD$ .*

*PROOF.* We combine Sokolova and Woracek's result (Proposition A.3) with the previous Proposition A.4.  $\square$

## B Proofs for Section 2 (Normalization)

**Definition B.1** (Finitary distribution monad). The *finitary distribution monad*,  $D: \text{Set} \rightarrow \text{Set}$ , assigns, to each set, the set of finitely-supported distributions on it,

$$D(X) = \left\{ d: X \rightarrow [0, 1] \mid \#\text{supp}(d) < \infty, \sum_{x \in X} d(x) = 1 \right\}.$$

**Definition B.2** (Maybe monad). The *maybe monad* (sometimes called the *option monad*),  $M: \text{Set} \rightarrow \text{Set}$ , assigns to each set  $X$ , the same set with an extra disjoint element usually denoted by  $\perp \in MX$ . That is,  $MX = X + \{\perp\}$ .

**Definition B.3** (Monoidal natural transformation). A *monoidal natural transformation* between two lax monoidal functors,

$$\begin{aligned} (F, u^F, v^F): (A, \otimes_a, I_a) &\rightarrow (B, \otimes_b, I_b) & \text{and} \\ (G, u^G, v^G): (A, \otimes_a, I_a) &\rightarrow (B, \otimes_b, I_b), \end{aligned}$$

is a *natural transformation* between the underlying functors,  $\alpha_X: FX \rightarrow GX$ , additionally making the following diagrams commute.

$$\begin{array}{ccc} FX \otimes_b FY & \xrightarrow{u^F} & F(X \otimes_a Y) & I_b & \xrightarrow{v^G} & GI_a \\ \alpha_X \otimes_b \alpha_Y \downarrow & & \downarrow \alpha_{X \otimes_a Y} & v^F \downarrow & \nearrow \alpha_I & \\ GX \otimes_b GY & \xrightarrow{u^G} & G(X \otimes_a Y) & FI_a & & \end{array}$$

**Proposition 2.5** (Non-associative category of normalized kernels). *Normalized kernels between sets,  $X \rightarrow MDY$ , form a non-associative category,  $\text{Norm}$ , where composition of two morphisms,  $f: X \rightarrow MDY$  and  $g: Y \rightarrow MDZ$ , is defined by*

$$(f \circ g)(x; z) = \left[ \frac{\sum_{v \in Y} f(x; v) \cdot g(v; z)}{\sum_{v \in Y} \sum_{w \in Z} f(x; v) \cdot g(v; w)} \right].$$

*In other words, if we consider the associated substochastic kernels,  $f^\bullet: X \rightarrow DMZ$  and  $g^\bullet: Y \rightarrow DMZ$ , it is the normalization of their composition as subdistributions,  $f \circ g = n(f^\bullet; g^\bullet)$ .*

*PROOF.* We define  $\text{id}(x; x') = [x = x']$ . Let us check that it is right unital. Let  $f: X \rightarrow MDY$  be a normalized kernel.

$$\begin{aligned} (f \circ \text{id})(x; y) &= \left[ \frac{\sum_{y'} f(x; y') \cdot [y' = y]}{\sum_{y', y} f(x; y') \cdot [y' = y]} \right] \\ &= \left[ \frac{f(x; y)}{\sum_y f(x; y)} \right] = f(x; y). \end{aligned}$$

On the last step, either  $\sum_{y'} f(x; y') = 0$ , which implies  $f(x; y) = 0$ , or  $\sum_{y'} f(x; y') = 1$ , which simplifies the division. Left unitality is analogous.  $\square$

**Proposition 2.6.** *Normalized kernels are not associative.*

*PROOF.* Let us provide a specific counterexample. Consider a coin flip,  $f = 1/2|a\rangle + 1/2|b\rangle$ , followed by a channel marking it with different failure probabilities  $g(a) = 1/3|x\rangle + 2/3|z\rangle$  and  $g(b) = 1/2|y\rangle + 1/2|z\rangle$ , and then followed by a channel that fails,  $h(x) = |x\rangle$  and  $h(y) = |y\rangle$ , but  $h(z) = 0$ .

In this case, let us check that  $(f \circ g) \circ h \neq f \circ (g \circ h)$ . We have the following computation for the left-hand side,

$$\begin{array}{l} \xrightarrow{f} \\ \xrightarrow{g} \\ \xrightarrow{h} \end{array} \quad \begin{array}{l} 1/2|a\rangle + 1/2|b\rangle \\ 1/6|x\rangle + 2/6|z\rangle + 1/4|y\rangle + 1/4|z\rangle \\ 2/5|x\rangle + 3/5|y\rangle, \end{array}$$

while the right-hand side composition amounts to  $(g \circ h)(a) = 1|x\rangle$  and  $(g \circ h)(b) = 1|y\rangle$ , and thus the result is  $1/2|x\rangle + 1/2|y\rangle$ .

$$\begin{array}{l} \xrightarrow{f} \\ \xrightarrow{g} \\ \xrightarrow{h} \end{array} \quad \begin{array}{l} 1/2|a\rangle + 1/2|b\rangle, \\ 1/2|1/3|x\rangle + 2/3|z\rangle + 1/2|1/2|y\rangle + 1/2|z\rangle, \\ 1/2|x\rangle + 1/2|x\rangle. \end{array}$$

This contradicts associativity.  $\square$

**Proposition 2.9.** *Normalized kernels form a symmetric monoidal non-associative category with the cartesian product, where the tensor of  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$  is given by*

$$(f_1 \otimes f_2)(x_1, x_2; y_1, y_2) = f_1(x_1; y_1) \cdot f_2(x_2; y_2).$$

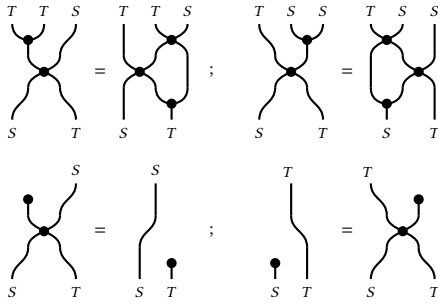
PROOF. Most of the structure is direct. The interchange law corresponds to the following equation,

$$\left[ \frac{\sum_{y_1, y_2} f_1(x_1; y_1) \cdot f_2(x_2; y_2) \cdot g_1(y_1; z_1) \cdot g_2(y_2; z_2)}{\sum_{y_1, y_2, z_1, z_2} f_1(x_1; y_1) \cdot f_2(x_2; y_2) \cdot g_1(y_1; z_1) \cdot g_2(y_2; z_2)} \right] = \left[ \frac{\sum_{y_1} f_1(x_1; y_1) \cdot g_1(y_1; z_1)}{\sum_{y_1, z_1} f_1(x_1; y_1) \cdot g_1(y_1; z_1)} \right] \cdot \left[ \frac{\sum_{y_2} f_2(x_2; y_2) \cdot g_2(y_2; z_2)}{\sum_{y_2, z_2} f_2(x_2; y_2) \cdot g_2(y_2; z_2)} \right]$$

which uses distributivity of products over sums.  $\square$

### C Proofs for Section 3 (Distributive Laws)

**Definition C.1** (Distributive law [Bec69], [23]). A distributive law between two monads,  $(S, \mu, \nu)$  and  $(T, \mu, \nu)$ , on the same category is a natural transformation  $\psi_X: TSX \rightarrow STX$  that moreover makes the following diagrams commute.



**Definition C.2** (Monoidal monad). A *monoidal monad*, given by a tuple  $(T, \mu, \eta, u, v)$ , on a strict monoidal category  $(\mathbb{C}, \otimes, I)$  consists of a monad  $(T, \mu, \eta)$ , whose underlying functor is lax monoidal: there exist structural natural transformations,

$$u_{X,Y}: TX \otimes TY \rightarrow T(X \otimes Y) \text{ and } v: I \rightarrow TI,$$

satisfying associativity,  $(u_{X,Y} \otimes \text{id}_Z) \circ u_{X \otimes Y, Z} = (\text{id}_X \otimes u_{Y,Z}) \circ u_{X, Y \otimes Z}$ , and unitality,  $(\text{id}_{TX} \otimes v) \circ u = \text{id}_{TX}$  and  $(v \otimes \text{id}_{TX}) \circ u = \text{id}_{TX}$ .

$$\begin{array}{ccc} TX \otimes TY \otimes TZ & \xrightarrow{\text{id}_{TX} \otimes u_{Y,Z}} & TX \otimes T(Y \otimes Z) \\ u_{X,Y} \otimes \text{id}_{TZ} \downarrow & & \downarrow u_{X, Y \otimes Z} \\ T(X \otimes Y) \otimes TZ & \xrightarrow{u_{X \otimes Y, Z}} & T(X \otimes Y \otimes Z) \end{array}$$

$$\begin{array}{ccc} TX & \xrightarrow{\text{id} \otimes v} & TX \otimes TI \\ \downarrow v \otimes \text{id} & & \downarrow u_{X, I} \\ TI \otimes TX & \xrightarrow{u_{I, X}} & TX \end{array}$$

Moreover, the unit and multiplication of the monad are monoidal natural transformations, meaning that the following diagrams commute.

$$\begin{array}{ccc} X \otimes Y & \xrightarrow{\eta_{X \otimes Y}} & T(X \otimes Y) \\ \eta_X \otimes \eta_Y \downarrow & \nearrow u_{X,Y} & \\ TX \otimes TY & & \end{array}$$

$$\begin{array}{ccc} I & \xrightarrow{\eta_I} & TI \\ \text{id} \downarrow & \nearrow v & \\ I & & \end{array}$$

$$\begin{array}{ccc} I & \xrightarrow{v} & TI \\ v \downarrow & & \uparrow \mu_I \\ TI & \xrightarrow{T\eta} & TTI \end{array}$$

$$\begin{array}{ccc} TTX \otimes TTY & \xrightarrow{\mu \otimes \mu} & TX \otimes TY \xrightarrow{u} T(X \otimes Y) \\ \downarrow u & & \nearrow \mu \\ T(TX \otimes TY) & \xrightarrow{Tu} & TT(X \otimes Y) \end{array}$$

**Definition C.3** (Monoidal distributive law). A monoidal distributive law between monoidal monads,  $(S, \mu, \eta, u, v)$  and  $(T, \mu, \eta, u, v)$ , is a distributive law,  $\psi_X: TSX \rightarrow STX$ , whose transformation is monoidal

$$\begin{array}{ccc} TSX \otimes TSY & \xrightarrow{u_{SX, SY}} T(SX \otimes SY) & \xrightarrow{T u_{X, Y}} TS(X \otimes Y) \\ \psi_X \otimes \psi_Y \downarrow & & \downarrow \psi_{X \otimes Y} \\ STX \otimes STY & \xrightarrow{u_S} S(TX \otimes TY) & \xrightarrow{S u_{X, Y}} ST(X \otimes Y) \end{array}$$

$$\begin{array}{ccc} I & \xrightarrow{v} & SI \xrightarrow{Sv} STI \\ v \downarrow & & \nearrow \psi_I \\ TI & \xrightarrow{Tv} & STI \end{array}$$

### D Proofs for Section 4 (Sesquilaws)

**Proposition 4.3.** *Almost-distributive laws,  $\psi_X: TSX \rightarrow STX$ , induce non-associative monads on their composite functors,  $ST$ . Monoidal almost distributive laws induce monoidal non-associative monads on their composite functors.*

PROOF. The unit of the composite non-associative monad is  $\eta = \eta^S \eta^T$ ; the multiplication is  $\mu = S\psi/T; \mu^S \mu^T$ . From unitality of the almost distributive law, unitality of the non-associative monad follows.

Whenever the almost distributive law is moreover monoidal, the unit and multiplication of the monad become monoidal by construction.  $\square$

**Proposition 4.4** (Kleisli magmoids). *Any non-associative monad,  $(R, \mu^R, \eta^R)$  over a category  $\mathbb{C}$ , induces a magmoid,  $(\mathbb{C}, \mathbb{K}(R))$ . Any monoidal non-associative monad induces a monoidal magmoid.*

PROOF. Let us first construct a non-associative category,  $\mathbb{K}(R)$ : the construction is similar to that of the Kleisli category; the only difference is that associativity is never used. Its objects are those of  $\mathbb{C}$ , the base category; its morphisms from  $X$  to  $Y$  correspond to morphisms  $X \rightarrow RY$  of the base category.

Composition of two morphisms,  $f: X \rightarrow RY$  and  $g: Y \rightarrow RZ$ , uses the multiplication  $\mu_X: RRX \rightarrow RX$ . It is defined, in terms of the base category, by  $f \circ g = f \circ Rg; \mu_Z$ . Identities are defined, in terms of the base category, by  $\eta^R$ . From the unitality axioms of the almost distributive law, it follows that composition is unital with identities.

Let us now construct the identity-on-objects functor  $(-)_\dagger: \mathbb{C} \rightarrow \mathbb{K}(R)$ . It acts as  $(f)_\dagger = f \circ \eta^R$ , and it is again direct to check first that it defines a functor, and then, by cases, that the image of this functor is associative.  $\square$

**Proposition 4.5.** *Normalization,  $n_X: DMX \rightarrow MDX$ , forms an almost-distributive law.*

PROOF. Let us check the axioms of an almost-distributive law. The proof is parallel to that of the discrete case.

$$\begin{array}{ccc} DMMX & \xrightarrow{D\mu^M} DMX & \xrightarrow{\psi} MDX \\ \psi_M \downarrow & & \nearrow \mu^{MD} \\ MDMX & \xrightarrow{M\psi} MMDX & \end{array}$$



$$\begin{array}{ccc}
 DX & \xrightarrow{\eta^{MD}} & MDX \\
 D\eta^M \downarrow & \nearrow \psi & \\
 DMX & & 
 \end{array}
 \quad
 \begin{array}{ccc}
 MX & \xrightarrow{M\eta^D} & MDX \\
 \eta^D M \downarrow & \nearrow \psi & \\
 DMX & & 
 \end{array}$$

For  $M$ -multiplicativity, we start by noticing that an element  $d \in DMMX$  means a distribution over  $X + \{\perp\} + \{\perp'\}$ . Let us reason by cases. If  $d(\perp) + d(\perp') = 1$ , then  $n(\mu^M(d)) = \perp = \mu^M(n(n(d)))$ , either because  $d(\perp') = 1$ , or because  $d(\perp') \neq 1$  but then  $d(\perp) > 0$  and  $n(d)(\perp) = 1$ . Assume, thus, that  $d(\perp) + d(\perp') \neq 1$ . We then need to prove that normalizing both separately is the same as normalizing after identifying both.

$$\begin{aligned}
 \mu^M(n(n(d)))(x) &= \left[ \frac{n(d)(x)}{\sum_{x_1 \in X} n(d)(x_1)} \right] \\
 &= \left[ \frac{\left[ \frac{d(x)}{\sum_{x_2 \in X + \{\perp\}} d(x_2)} \right]}{\sum_{x_1 \in X} \left[ \frac{d(x_1)}{\sum_{x_3 \in X + \{\perp\}} d(x_3)} \right]} \right] \\
 &= \left[ \frac{d(x)}{\sum_{x_1 \in X} d(x_1)} \right] = n(\mu^M(d)).
 \end{aligned}$$

For  $M$ -unitality, we must check that the normalization of a normalized distribution is itself: by  $\eta^M(d)(x) = d(x)$ , we conclude that  $\sum_{x \in X} \eta^M(d)(x) = \sum_{x \in X} d(x) = 1$ , and thus  $n(\eta^M(d)) = \eta^M(d)$ .

For  $D$ -unitality, we reason by cases on  $MX$ . We directly check that  $n(\eta^D(\perp)) = n(\perp) = \perp$  and  $n(\eta^D(x)) = n(|x\rangle) = x$ .  $\square$

**Definition D.1** (Sesquilaw, [21]). A sesquilaw,  $(S, T, \bowtie, \bowtie')$ , between two monads, consists of a distributive law  $(\bowtie): ST \rightarrow TS$  and an almost distributive law  $(\bowtie'): TS \rightarrow ST$ , forming a section-retraction pair,  $(\bowtie) \circ (\bowtie') = \text{id}$ , and satisfying any of the following two equivalent equations.

A sesquilaw is *monoidal* whenever its transformation is.

**Theorem 4.7.** Normalization and subdistributions form a sesquilaw.

**PROOF.** We already know that normalization and the inclusion into subdistributions are mutual inverses, and that normalization forms an almost distributive law (Proposition 4.5).

Let us prove that the second formulation of the axiom of distributive sesquilaws holds.

$$\begin{array}{ccc}
 DMDX & \xrightarrow{nD} & MDDX \xrightarrow{M\mu} MDX \\
 Dm \downarrow & \nearrow n & \\
 DDMX & \xrightarrow{\mu M} & DMX
 \end{array}$$

Let  $d \in DMDX$  be, equivalently, a subdistribution of distributions. We must prove that normalizing and flattening the distributions is the same as, while regarding the distributions as subdistributions, flattening and then normalizing. In other words, we seek to prove

$$M\mu(n(d)) = n(\mu^D(d(-\bullet))).$$

On the left hand side, we use the monad multiplication and normalization.

$$\begin{aligned}
 M\mu(n(d))(x) &= \sum_{\alpha \in DX} n(d)(\alpha) \cdot \alpha(x) \\
 &= \sum_{\alpha \in DX} \left[ \frac{d(\alpha)}{\sum_{\beta \in DX} d(\beta)} \right] \cdot \alpha(x).
 \end{aligned}$$

On the right hand side, we use the normalization and monad multiplication.

$$\begin{aligned}
 n(\mu(d(-\bullet)))(x) &= \left[ \frac{\mu(d(-\bullet))(x)}{\sum_{y \in X} \mu(d(-\bullet))(y)} \right] \\
 &= \left[ \frac{\sum_{\alpha \in DX} d(\alpha^*) \cdot \alpha(x)}{\sum_{y \in X} \sum_{\alpha \in DX} d(\alpha^*) \cdot \alpha(y)} \right] \\
 &= \left[ \frac{\sum_{\alpha \in DX} d(\alpha^*) \cdot \alpha(x)}{\sum_{\alpha \in DX} d(\alpha^*) \cdot \sum_{y \in X} \alpha(y)} \right] \\
 &= \left[ \frac{\sum_{\alpha \in DX} d(\alpha^*) \cdot \alpha(x)}{\sum_{\alpha \in DX} d(\alpha^*)} \right].
 \end{aligned}$$

The last step uses that all  $\alpha \in DX$  are full distributions:  $\sum_{y \in X} \alpha(y)$  must be exactly 1. This concludes the proof.  $\square$

**Theorem 4.8 (Renormalization).** Any sesquilaw,  $(S, T, m, n)$ , induces an idempotent,  $k = (n \circ m): TS \rightarrow TS$ . This idempotent is left-absorptive, meaning the following diagram commutes.

$$\begin{array}{ccccc}
 TSTSX & \xrightarrow{\mu^{TS}} & TSX & \xrightarrow{k} & TSX \\
 kTS \downarrow & & & \nearrow k & \\
 TSTSX & \xrightarrow{\mu^{TS}} & TSX & & 
 \end{array}$$

**PROOF.** We employ string diagrams for this proof. We need to prove that any sesquilaw,  $(\bowtie, \bowtie', S, T)$ , induces an idempotent,  $(\bowtie \circ \bowtie'): TS \rightarrow TS$ , and that this idempotent is left-absorptive, meaning that the following equation holds.

Let us prove a slightly stronger equation where we omit composition with the distributive law  $(\bowtie)$ . We use (i) the multiplicativity axiom, (ii) the sesquilaw equation, (iii) that sesquilaws are inverses, (iv) the sesquilaw equation, (v) the multiplicativity axiom. This concludes the proof.  $\square$

**Lemma 4.11.** Any sesquilaw,  $(S, T, m, n)$ , induces a right action of the monad  $TS$  into the non-associative monad  $ST$ : a natural transformation,  $u_X: STTSX \rightarrow STX$ , making the following two diagrams commute.

$$\begin{array}{ccc}
 STX & \xrightarrow{\eta^{TS}} & STTSX \\
 \text{id} \searrow & & \downarrow u \\
 & & STX
 \end{array}
 \quad
 \begin{array}{ccc}
 STTSX & \xrightarrow{ST\mu^{TS}} & STTSX \\
 uTS \downarrow & & \downarrow u \\
 STTSX & \xrightarrow{u} & STX
 \end{array}$$

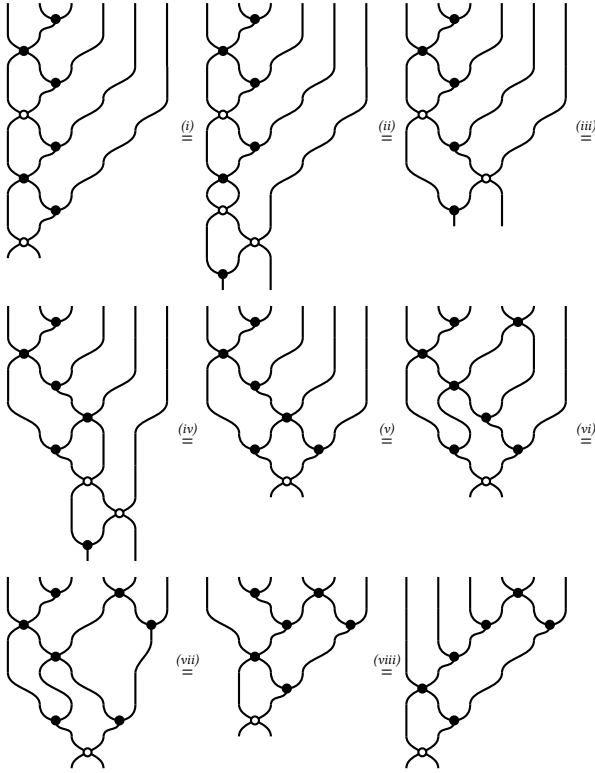


Figure 2: Multiplicativity for the right action of a sesquilaw.

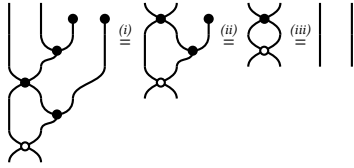


Figure 3: Unitality for the right action of a sesquilaw.

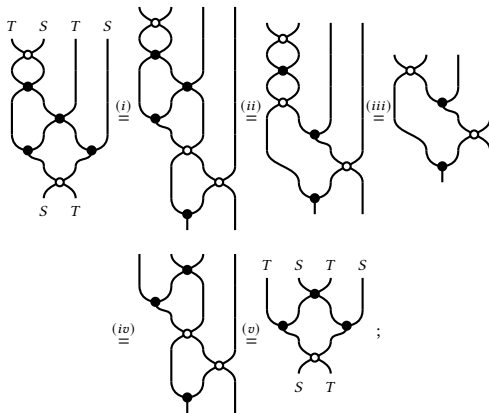


Figure 1: Proof of the renormalization equation.

This action is defined by either side of the following commutative diagram.

$$\begin{array}{ccccc}
 STTSX & \xrightarrow{S\mu^TS} & STSX & \xrightarrow{mS} & TSSX \\
 mTS \downarrow & & & & \downarrow T\mu^S \\
 TSTSX & \xrightarrow{TmS} & TTSSX & \xrightarrow{\mu^T\mu^S} & TSX \xrightarrow{n} STX
 \end{array}$$

PROOF. Let us use string diagrams for this proof. We will show that any sesquilaw,  $(\bowtie, \bowtie^*, S, T)$ , induces an action of the monad  $TS$  into the almost monad  $ST$ . We define the action,  $u: STTS \rightarrow ST$ , by string diagrams, as any of the two following equivalent definitions.

$$\begin{array}{c} \text{box } u \\ \hline \end{array} = \begin{array}{c} \text{string diagram 1} \end{array} = \begin{array}{c} \text{string diagram 2} \end{array}$$

We reason by string diagrams (in Figure 2). We use (i) the multiplicativity axiom, (ii) that sesquilaws are inverses, (iii) the sesquilaw equation, (iv) the multiplicativity axiom, (v) the multiplicativity of the distributive law, (vi, viii) associativity of the monad, and (vii) the multiplicativity of the distributive law.

The proof of unitality is more direct (Figure 3): it uses (i, ii) unitality of both monads and (iii) that sesquilaws are inverses.

With these two axioms, we have built an action. In the monoidal case, we repeat the exact same proof just considering that all natural transformations are monoidal natural transformations.  $\square$

**Theorem 4.12.** *In the setting of a sesquilaw,  $(S, T, m, n)$ , the Kleisli category of the distributive law,  $\mathbb{K}(m)$ , acts on the Kleisli magmoid of the almost distributive law,  $\mathbb{K}(n)$ .*

PROOF. We define the action,  $(\triangleleft): \mathbb{K}(n)(X; Y) \times \mathbb{K}(m)(Y; Z) \rightarrow \mathbb{K}(n)(X; Z)$ , as  $p \triangleleft f = p; STf; u_Z$ , with the natural transformation  $u_X: STTSX \rightarrow STX$  defined as in Lemma 4.11. Multiplicativity and unitality follow from Lemma 4.11.  $\square$

**Corollary 4.13.** *Normalized kernels admit an action from sub-stochastic kernels, defined by  $p \triangleleft f = n(p^\bullet; f)$ .*

$$(\triangleleft): \text{Norm}(X; Y) \times \text{subStoch}(Y; Z) \rightarrow \text{Norm}(X; Z).$$

That is, satisfying  $p \triangleleft \text{id} = p$  and  $p \triangleleft (f; g) = p \triangleleft f \triangleleft g$ .

PROOF. The result follows from Proposition 4.9.

$$\begin{aligned}
 p \triangleleft (f; g) &= n(p^\bullet; f; g) = n(n(p^\bullet; f); g) \\
 &= n(p^\bullet; f) \triangleleft g = p \triangleleft f \triangleleft g.
 \end{aligned}$$

By section-retraction,  $n(p^\bullet; \text{id}) = n(p^\bullet) = p$ .  $\square$

## E Proofs for Section 5 (Possibilistic normalization)

**Proposition 5.3.** *Post-selection is a monoidal sesquilaw.*

PROOF. A better description of post-selection start by regarding non-empty subsets, the elements of  $RX$ , as *non-null predicates* on  $X$ : predicates that are true on at least one element. We take this point of view for this proof, as we shall see it simplifies it considerably.<sup>7</sup>

<sup>7</sup>Alternatively, the reader familiar with *tricocycloids* may prefer to regard this as a consequence of Theorem J.1: tricocycloid homomorphisms also induce sesquilaw homomorphisms; and, in this case, the possibilistic tricocycloid is the terminal one. We choose not to develop this theory here.

Post-selection of a *predicate*,  $\alpha: X \rightarrow \{0, 1\}$  or  $\alpha \in \mathbf{RMX}$ , is defined by  $\alpha(x)$  except when it is null,

$$p(\alpha) = \left[ \frac{\alpha(x)}{\bigvee_{x \in X} \alpha(x)} \right].$$

From here on, we repeat proofs analogous to those of the probabilistic case. For instance, let us check that post-selection is a monoidal natural transformation. It suffices to note that the following formula holds.

$$\left[ \frac{\alpha(x)}{\bigvee_{x \in X} \alpha(x)} \right] \wedge \left[ \frac{\beta(x)}{\bigvee_{x \in X} \beta(x)} \right] = \left[ \frac{\alpha(x) \wedge \beta(x)}{\bigvee_{x \in X} \alpha(x) \wedge \bigvee_{x \in X} \beta(x)} \right].$$

Let us also check the sesquilaw axiom. It is well-known that there exists a distributive law of the maybe monad over any other Set-monad. Let  $\psi \in \mathbf{RMRX}$  be, equivalently, a predicate on non-null predicates. On the lower path, we use the monad multiplication and normalization.

$$\begin{aligned} M\mu(p(\psi))(x) &= \bigvee_{\alpha \in \mathbf{RX}} p(\psi)(\alpha) \wedge \alpha(x) \\ &= \bigvee_{\alpha \in \mathbf{RX}} \left[ \frac{\psi(\alpha)}{\bigvee_{\beta \in \mathbf{RX}} \psi(\beta)} \right] \wedge \alpha(x). \end{aligned}$$

On the right hand side, we use the normalization and monad multiplication.

$$\begin{aligned} p(\mu(\psi(-\bullet)))(x) &= \left[ \frac{\mu(\psi(-\bullet))(x)}{\bigvee_{y \in X} \mu(\psi(-\bullet))(y)} \right] \\ &= \left[ \frac{\bigvee_{\alpha \in \mathbf{DX}} \psi(\alpha^\bullet) \wedge \alpha(x)}{\bigvee_{y \in X} \bigvee_{\alpha \in \mathbf{RX}} \psi(\alpha^\bullet) \wedge \alpha(y)} \right] \\ &= \left[ \frac{\bigvee_{\alpha \in \mathbf{DX}} \psi(\alpha^\bullet) \wedge \alpha(x)}{\bigvee_{\alpha \in \mathbf{DX}} \psi(\alpha^\bullet) \wedge \bigvee_{y \in X} \psi(y)} \right] \\ &= \left[ \frac{\bigvee_{\alpha \in \mathbf{DX}} \psi(\alpha^\bullet) \wedge \alpha(x)}{\bigvee_{\alpha \in \mathbf{DX}} \psi(\alpha^\bullet)} \right]. \end{aligned}$$

The last step uses that all  $\alpha \in \mathbf{RX}$  are non-null predicates:  $\bigvee_{y \in X} \alpha(y)$  must be exactly 1. The rest of the proof is analogous.  $\square$

**Proposition 5.6.** *Support,  $\text{supp}_X: \mathbf{DX} \rightarrow \mathbf{RX}$ , is a sesquilaw homomorphism between normalization and post-selection.*

**PROOF.** We must prove that the following two diagrams commute.

$$\begin{array}{ccc} \mathbf{MDX} & \xrightarrow{\text{Msupp}} & \mathbf{MRX} \\ \bullet_D \downarrow & & \downarrow \bullet_R \\ \mathbf{DMX} & \xrightarrow{\text{suppM}} & \mathbf{RMX} \end{array} \quad \begin{array}{ccc} \mathbf{MD} & \xrightarrow{\text{Msupp}} & \mathbf{MR} \\ \circ_D \uparrow & & \uparrow \circ_R \\ \mathbf{DM} & \xrightarrow{\text{suppM}} & \mathbf{RM} \end{array}$$

The first diagram is direct. Let us discuss the second one. On the left-hand side, we have

$$\begin{aligned} \text{supp}(n(d))(x) &= (n(d) > 0) \\ &= \left( \left[ \frac{d(x)}{\sum_{x'} d(x')} \right] > 0 \right) \\ &= (d(x) > 0). \end{aligned}$$

Meanwhile, on the right-hand side, we get to the same result.

$$\begin{aligned} p(\text{supp}(d))(x) &= \left[ \frac{\text{supp}(d)(x)}{\bigvee_{x' \in X} \text{supp}(x')} \right] \\ &= \left[ \frac{(d(x) > 0)}{\bigvee_{x' \in X} (d(x') > 0)} \right] \end{aligned}$$

$$= (d(x') > 0).$$

This concludes the proof.  $\square$

## F Proofs for Section 6 (Left and right commutativity)

**Definition F.1** (Tensor schema, [JS91, Definition 1.4]). A *tensor schema*  $\mathcal{D}$  – also known as a *polyquiver* or *polygraph* – consists of a set of objects,  $\mathcal{D}_{obj}$ , and, for each two lists of objects,  $X_1, \dots, X_n \in \mathcal{D}_{obj}$  and  $Y_1, \dots, Y_m \in \mathcal{D}_{obj}$ , a set of morphisms

$$\mathcal{D}(X_1, \dots, X_n; Y_1, \dots, Y_m).$$

**Theorem F.2** (String diagrams, [JS91, Definition 1.2]). String diagrams over a tensor schema  $\mathcal{D}$  – deformation classes of boxed progressive plane diagrams, in the original [JS91, Definition 1.2] – form a strict monoidal category,  $\text{String}(\mathcal{D})$ ; moreover, this is the free strict monoidal category over a tensor schema.

As a consequence, there is an adjunction between tensor schemas and strict monoidal categories,

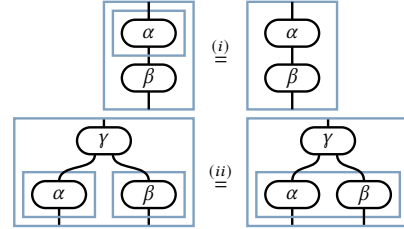
$$\text{TensorSch}(\mathcal{D}, \text{Forget}(\mathbb{C})) \cong \text{MonCat}(\text{String}(\mathcal{D}), \mathbb{C});$$

which, in turn, induces a monad,  $\mathbf{S} = \text{Forget} \circ \text{String}$ , in the category of tensor schemas.

**Remark F.3.** String diagrams form a monad on the category of polyquivers,  $\mathbf{S}: \text{PolyQuiver} \rightarrow \text{PolyQuiver}$ , and its category of algebras is precisely the category of strict monoidal categories and strict functors [JS91, Theorem 2.3].

**Theorem 6.7.** Left commutative magmoids are non-multiplicative algebras of the string diagrams monad over tensor schemas, in the sense of Joyal and Street [JS91, Definition 1.4].

Moreover, these algebras satisfy the following equations for any two diagrams  $\alpha$  and  $\beta$ : we call them (i) left-bias, and (ii) monoidality.



**PROOF.** We start by constructing, for each commutative magmoid,  $\mathbb{A}$ , an algebra  $\varphi: \mathbf{S}(\mathbb{A}) \rightarrow \mathbb{A}$ . Given any string diagram,  $s \in \mathbf{S}(\mathbb{A})$ , we may use that it is acyclic – progressive in the original [JS91, §2] – to deduce the existence of a topological ordering of its nodes,  $[f_1, \dots, f_n]$ .

Let us fix a representative string diagram, up to deformation, where the nodes appear progressively in the given topological order (this we obtain from Joyal and Street's theorem, and contains the core of the proof [JS91, §2]). For each node  $f_i$  in the representative string diagram, define  $\phi(f_i) = \text{id} \otimes \dots \otimes f_i \otimes \dots \otimes \text{id}$ , to be the given morphism tensored with as many identities as wires it has on each side. Note that this is not well-defined under the choice of topological ordering, we will need to show our final interpretation of the whole diagram is.

Now, define the interpretation of the string diagram as the left-associated composition of node interpretations,

$$\varphi(s) = (((\phi(f_1) \circ \phi(f_2)) \circ \phi(f_3)) \circ \dots) \circ \phi(f_n).$$

We must show that this is well-defined under the choice of topological ordering. Any two topological orderings can be reached from each other by swapping two independent adjacent nodes:  $f_i$  and  $f_{i+1}$  for some index  $i$ . Let us call  $\hat{s}$  to the string diagram after swapping these two nodes. We have that,

$$\begin{aligned} \varphi(s) &= (((\phi(f_1) \circ \phi(f_2)) \circ \dots) \circ \phi(f_i)) \circ \phi(f_{i+1}) \dots) \circ \phi(f_n); \\ \varphi(\hat{s}) &= (((\phi(f_1) \circ \phi(f_2)) \circ \dots) \circ \phi(f_{i+1})) \circ \phi(f_i) \dots) \circ \phi(f_n). \end{aligned}$$

Here, however, we may use the axiom of commutative magmoids to prove that both terms are indeed equal. Explicitly, if two nodes –  $i$  and  $j$  – were independent in the original graph, then  $\phi(f_i)$  and  $\phi(f_j)$  interchange by the axioms of monoidal magmoids.

We may now check that the algebra is unital: if  $s_f$  is a string diagram consisting of a single node  $f$ , we have that  $\varphi(s_f) = f$ , by definition.

Finally, let us prove the three equations of the statement. Let  $\alpha, \beta, \gamma \in S(\mathcal{D})$  be three string diagrams. Let  $[f_1, \dots, f_n]$  and  $[g_1, \dots, g_m]$  be two orderings of the nodes of  $\alpha$  and  $\beta$ . We have

- (1)  $\varphi([\varphi(\alpha)]) = \varphi(\alpha)$ , by unitality;
- (2)  $\varphi([\varphi(\alpha)] \otimes [\varphi(\beta)]) = (\varphi(\alpha) \otimes \text{id}) \circ (\text{id} \otimes \varphi(\beta)) = \varphi(\alpha) \otimes \varphi(\beta)$ , using that, because the nodes in  $\alpha$  and  $\beta$  are independent, we may pick a topological ordering that places all nodes in  $\alpha$  before all nodes in  $\beta$ ; from that topological ordering, we have by induction applying the interchange law,
 
$$\begin{aligned} &(((\phi(f_1) \circ \phi(f_2)) \circ \phi(f_3)) \circ \dots) \circ \phi(g_m) = \\ &(((\phi(\alpha) \circ \phi(g_1)) \circ \phi(g_2)) \circ \dots) \circ \phi(g_m) = \\ &(((\phi(g_1) \circ \phi(g_2)) \circ \dots) \circ \phi(g_m)) \circ \phi(\alpha) = \\ &\phi(\beta) \circ \phi(\alpha). \end{aligned}$$
- (3)  $\varphi([\varphi(\alpha)] \circ \beta) = ((\varphi(\alpha) \circ g_1) \circ \dots) \circ g_m$ , using that, because all nodes in  $\alpha$  appear before nodes in  $\beta$ , a topological ordering can be constructed from the topological ordering in  $\alpha$  followed by the topological ordering in  $\beta$ .

These close the proof.  $\square$

**Remark F.4 (Non-multiplicative algebras of string diagrams).** Monoidal magmoids are not algebras for this string diagrams monad, but there are two canonical algebra structures on any monoidal magmoid: right-associative and left-associative evaluation,  $\text{ev}_R: S(\mathbb{A}) \rightarrow \mathbb{A}$  and  $\text{ev}_L: S(\mathbb{A}) \rightarrow \mathbb{A}$ . Both of these evaluations satisfy the unitality axiom of algebras, but both fail the multiplicativity axiom.

$$\begin{array}{ccc} \mathbb{A} & \xrightarrow{\quad} & S(\mathbb{A}) \\ & \searrow & \downarrow \mu_{\mathbb{A}} \\ & & \mathbb{A} \end{array} \quad \begin{array}{ccc} S(S(\mathbb{A})) & \xrightarrow{S(\text{ev}_\bullet)} & S(\mathbb{A}) \\ \downarrow \mu_{S(\mathbb{A})} & \neq & \downarrow \text{ev}_\bullet \\ S(\mathbb{A}) & \xrightarrow{\text{ev}_\bullet} & \mathbb{A} \end{array}$$

As a result, a string diagram composed of string diagrams has a different semantics from its flattening.

**Remark F.5 (Normalization boxes).** This explains the emergence of normalization boxes [LT23, JSS25]. Considering elements of the monad of string diagrams,  $S(\mathbb{A})$ , is insufficient to modulate the non-associativity of normalization. Instead, authors use boxes that

themselves contain diagrams with boxes; that is, elements of the free monad

$$S^*(\mathbb{A}) = \mathbb{A} + S(\mathbb{A}) + S(S(\mathbb{A})) + \dots$$

## G Proofs for Section 7 (Markov magmoids)

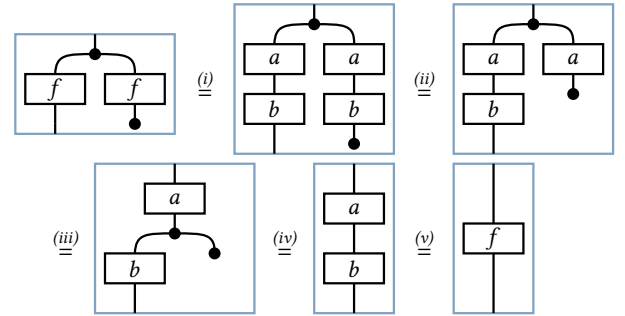
**Proposition G.1** (Normalization copy-discard sesquialaw). *Normalized kernels form a copy-discard magmoid. Normalization induces a copy-discard sesquialaw.*

**Proposition 7.5.** *Any copy-discard sesquialaw induces a quasitotal magmoid. As a corollary, normalized kernels are quasitotal: and, exemplifying Remark 6.8, the following equation holds for any normalized kernel  $f: X \rightarrow \text{MDY}$ .*

$$\left[ \frac{\sum_{y'} f(x; y) \cdot f(x; y')}{\sum_{y, y'} f(x; y) \cdot f(x; y')} \right] = f(x; y).$$

**PROOF.** More generally, let us prove that any morphism that factors into a deterministic morphism followed by a total morphism is quasitotal. Let  $f: X \rightarrow Z$  be a morphism that factors as  $f = a \circ b$  for a deterministic morphism  $a: X \rightarrow Y$  and a total morphism  $b: Y \rightarrow Z$ .

We reason that the morphism is quasitotal using (i) the factorization, (ii) that  $b$  is total, (iii) that  $a$  is deterministic, (iv) the comonoid axioms, and (v) the factorization.



Finally, we are left to prove that, in the Kleisli category of a copy-discard sesquialaw,  $(T, S, n, m)$ , every morphism factors into a deterministic morphism followed by a total morphism. Indeed, morphisms of the Kleisli category are of the form  $f: X \rightarrow STY$ , and they compose as  $f \circ g = f; STg; S\eta T; \mu^S \mu^T$ . Because the monad  $S$  is relevant, any morphism of the form  $g; S\eta^T$  is deterministic; because the monad  $T$  is affine, any morphism of the form  $h; \eta^S$  is total. It suffices to note that  $f = (f; S\eta^T) \circ \eta^S$  by unitality of the sesquialaw.  $\square$

**Lemma G.2.** *Any copy-discard sesquialaw,  $(S, T, m, n)$ , induces a quasitotal magmoid that satisfies associativity,  $f \circ (g \circ h) = (f \circ g) \circ h$ , if either*

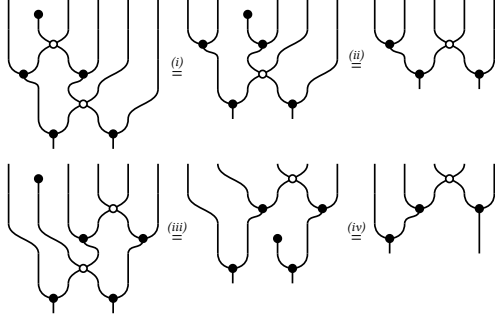
- (1)  $f = u; S\eta^T$  for some  $u: X \rightarrow SY$ ;
- (2)  $g = v; S\eta^T$  for some  $v: Y \rightarrow SZ$ ; or
- (3)  $h = w; \eta^S T$  for some  $w: Z \rightarrow TW$ .

**PROOF.** We employ string diagrams. The two sides of the associativity equation can be regarded as natural transformations,  $STSTST \rightarrow ST$ ; we equate these.

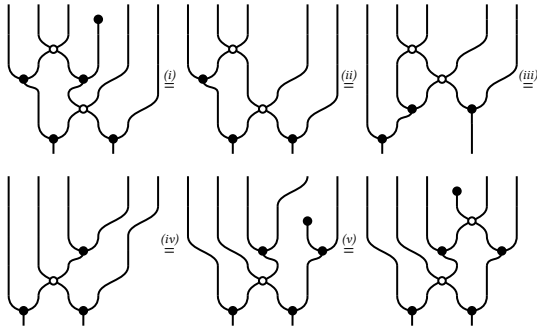
Let us consider the first case. On the left hand side, we use (i) unitality of the almost distributive law, and (ii) unitality of the



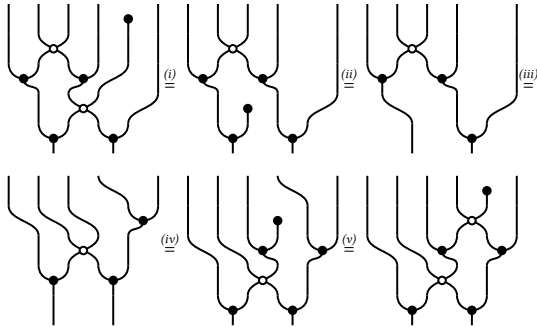
monad  $T$ . On the right hand side, we also use (iii) unitality of the almost distributive law, and (iv) unitality of the monad  $T$ . Both sides coincide by associativity of the monad  $S$ .



Let us consider the second case. We use (i) unitality of the monad  $T$ , (ii) associativity of the monad  $S$ , (iii) multiplicativity of the almost distributive law, and (iv) unitality of the almost distributive law.



Let us consider the third case. We use (i) unitality of the almost distributive law; (ii) unitality of the monad  $S$ ; (iii) associativity of the monad  $T$ ; (iv) unitality of the monad  $T$ ; (v) unitality of the almost distributive law.



We have thus shown associativity in any of the three cases.  $\square$

**Lemma G.3.** *Normalized kernels form a left-relevant magmoid.*

**PROOF.** Our proof strategy is to apply Lemma G.2: we will prove that, in the normalization magmoid, deterministic morphisms are precisely those of the form  $u \circ M\eta^D$ ; while total morphisms are precisely those of the form  $w \circ D\eta^M$ .

By definition, a total morphism must satisfy  $\sum_{y \in Y} f(x; y) = 1$ . This means that there exists a stochastic kernel,  $u(x; y) = f(x; y)$ , such that  $f = u \circ M\eta^D$ .

By definition, a deterministic channel must satisfy  $f(x; y) = f(x; y) \cdot f(x; y)$ , meaning that, either  $f(x; y) = 0$  or  $f(x; y) = 1$ . In particular, because probabilities must sum to less or equal than 1, for each  $x$ , there must exist at most a single  $y_x$  such that  $f(x; y_x) = 1$ . This means there exists a partial function, defined by  $v(x) = y_x$  whenever  $y_x$  exists and undefined otherwise, such that  $f = w \circ D\eta^M$ .

Applying Lemma G.2, we conclude that normalized kernels form a left-relevant magmoid.  $\square$

**Theorem 7.9.** *Normalized kernels form a Markov magmoid.*

**PROOF.** We separately prove that normalized kernels form a quasitotal magmoid (Proposition 7.5) and that it is a left-relevant magmoid (Lemma G.3). We only need to show it admits conditionals.

By definition, a conditional must satisfy

$$f(x; y, z) = \left[ \frac{\sum_{z_1 \in Z} f(x; y, z_1) \cdot c(x, y; z)}{\sum_{z_1, z \in Z} f(x; y, z_1) \cdot c(x, y; z)} \right].$$

Let us define the following normalized kernel,

$$c(x, y, z) = \left[ \frac{f(x; y, z)}{\sum_{z \in Z} f(x; y, z)} \right],$$

and let us proceed by cases: for each  $x$  and  $y$ , either

$$\sum_{z \in Z} f(x; y, z) = 0,$$

and then the axiom of conditionals vacuously holds; or

$$\sum_{z \in Z} f(x; y, z) \neq 0,$$

and we may simplify both coefficients to prove the axiom of conditionals. We have proven that  $c$  is a conditional of  $f$ .

An alternative proof that we do not develop here starts by realizing that the category of substochastic kernels has quasitotal conditionals [DR23], and that, thanks to the sesquialaw, these are inherited by the quasitotal magmoid of normalized kernels.  $\square$

## H Proofs for Section 8 (Normalization in standard Borel spaces)

**Theorem 8.4.** *Normalization induces a monoidal sesquialaw.*

**PROOF.** Let us first prove that normalization is monoidal. Consider two measures,  $\nu_1 \in \mathcal{D}X$  and  $\nu_2 \in \mathcal{D}Y$ . We use (i) the definition of normalization and the tensor of measures, (ii) the definition of the indicator function, (iii) linearity of the Lebesgue integral, (iv) linearity of the Lebesgue integral, and (v) the definition of the tensor of measures.

$$\begin{aligned} & N(\nu_1 \otimes \nu_2)(U) \\ & \stackrel{(i)}{=} \left[ \frac{\int_{y \in Y} \left( \int_{x \in X} \xi_U(x, y) \cdot \nu_1(dx) \right) \cdot \nu_2(dy)}{\int_{y \in Y} \left( \int_{x \in X} \xi_{X \times Y}(x, y) \cdot \nu_1(dx) \right) \cdot \nu_2(dy)} \right] \\ & \stackrel{(ii)}{=} \left[ \frac{\int_{y \in Y} \left( \int_{x \in X} \xi_U(x, y) \cdot \nu_1(dx) \right) \cdot \nu_2(dy)}{\int_{y \in Y} \left( \int_{x \in X} 1 \cdot \nu_1(dx) \right) \cdot \nu_2(dy)} \right] \\ & \stackrel{(iii)}{=} \left[ \frac{\int_{y \in Y} \left( \int_{x \in X} \xi_U(x, y) \cdot \nu_1(dx) \right) \cdot \nu_2(dy)}{\nu_1(X) \cdot \nu_2(Y)} \right] \end{aligned}$$

$$\begin{aligned}
&\stackrel{(iv)}{=} \int_{y \in Y} \left( \int_{x \in X} \xi_U(x, y) \cdot \left[ \frac{v_1(dx)}{v_1(X)} \right] \right) \cdot \left[ \frac{v_2(dy)}{v_2(Y)} \right] \\
&\stackrel{(v)}{=} (N(v_1) \otimes N(v_2))(U).
\end{aligned}$$

Let us now prove that normalization determines an almost distributive law.

$$\begin{array}{ccc}
\mathcal{D}MMX & \xrightarrow{\mathcal{D}\mu^M} & \mathcal{D}MX \xrightarrow{N} M\mathcal{D}X \\
\downarrow N\mathcal{M} & & \nearrow \mu^M \mathcal{D} \\
M\mathcal{D}MX & \xrightarrow{MN} & MM\mathcal{D}X
\end{array}$$
  

$$\begin{array}{ccc}
\mathcal{D}X & \xrightarrow{\eta^M \mathcal{D}} & M\mathcal{D}X \\
\downarrow \mathcal{D}\eta^M & \nearrow N & \\
\mathcal{D}MX & & 
\end{array}
\quad
\begin{array}{ccc}
MX & \xrightarrow{M\eta^D} & M\mathcal{D}X \\
\downarrow \eta^D M & \nearrow N & \\
\mathcal{D}MX & & 
\end{array}$$

For  $\mathcal{M}$ -multiplicativity, note that an element  $d \in \mathcal{D}MMX$  is a measure over  $X + \{\perp\} + \{\perp'\}$ . Let us reason by cases. If  $d(\{\perp, \perp'\}) = 1$ , then  $N(\mu^M(d)) = \perp = \mu^M(N(N(d)))$ , either because  $d(\perp') = 1$ , or because  $d(\perp') \neq 1$  but then  $d(\perp) > 0$  and  $N(d)(\perp) = 1$ . Assume, thus, that  $d(\{\perp, \perp'\}) \neq 1$ . We then need to prove that normalizing both separately is the same as normalizing after identifying both.

$$\begin{aligned}
\mu^M(N(N(d)))(U) &= \left[ \frac{N(d)(U)}{N(d)(X)} \right] = \left[ \frac{\left[ \frac{d(U)}{d(X + \{\perp\})} \right]}{\left[ \frac{d(X)}{d(X + \{\perp\})} \right]} \right] \\
&= \left[ \frac{d(U)}{d(X)} \right] = n(\mu^M(d)).
\end{aligned}$$

For  $\mathcal{M}$ -unitality, we must check that the normalization of a normalized distribution is itself: by  $\eta^M(d)(U) = d(U)$ , we conclude that  $\eta^M(d)(X) = d(X) = 1$ , and thus  $N(\eta^M(d)) = \eta^M(d)$ . For  $\mathcal{D}$ -unitality, we reason by cases on  $MX$ . We check that  $N(\eta^D(\perp)) = N(\perp) = \perp$  and that  $N(\eta^D(x)) = N(\perp x) = x$ .

Let us now prove that normalization and subdistributions form a sesquilaw.

We first need to prove that normalization and the inclusion into subdistributions are a section-retraction pair. Given any normalized distribution,  $v \in M\mathcal{D}X$ , the measure of the whole set must either be zero or one,  $v(X) = 1$  or  $v(X) = 0$ . If it is zero, it follows that the measure of any subset,  $U \subseteq X$ , must also be zero,  $v(U) = 0$ ; as a consequence, in any of the two cases,

$$N(v)(U) = \left[ \frac{v(U)}{v(X)} \right] = v(U).$$

Let us prove that the second formulation of the axiom of distributive sesquilaws holds.

$$\begin{array}{ccc}
\mathcal{D}M\mathcal{D}X & \xrightarrow{N\mathcal{D}} & M\mathcal{D}X \xrightarrow{M\mu} M\mathcal{D}X \\
\downarrow \mathcal{D}m & & \nearrow N \\
\mathcal{D}\mathcal{D}MX & \xrightarrow{\mu M} & \mathcal{D}MX
\end{array}$$

Let  $v \in \mathcal{D}M\mathcal{D}X$  be a subdistribution of distributions. We must prove that normalizing and flattening the distributions is the same as, while regarding the distributions as subdistributions, flattening and then normalizing. In other words, we seek to prove

$$M\mu(N(v)) = N(\mu(v(-\bullet))).$$

On the left hand side, we use (i) the definition of monad multiplication for the Girmonad, and (ii) the definition of normalization.

$$\begin{aligned}
M\mu(N(v(-\bullet)))(U) &\stackrel{(i)}{=} \int_{\alpha \in \mathcal{D}X} \alpha(U) \cdot N(v(-\bullet))(\alpha) \\
&\stackrel{(ii)}{=} \int_{\alpha \in \mathcal{D}X} \alpha(U) \cdot \left[ \frac{v(d\alpha)}{v(\mathcal{D}X)} \right].
\end{aligned}$$

On the right hand side, we use (i) the definition of normalization, (ii) the definition of monad multiplication for the Girmonad and, finally, (iii) that the  $\alpha \in \mathcal{D}X$  elements are full distributions, meaning that  $\alpha(X) = 1$ .

$$\begin{aligned}
N(\mu(v))(U) &\stackrel{(i)}{=} \left[ \frac{\mu(v)(U)}{\mu(v)(X)} \right] \\
&\stackrel{(ii)}{=} \left[ \frac{\int_{\alpha \in \mathcal{D}X} \alpha(U) \cdot v(d\alpha)}{\int_{\alpha \in \mathcal{D}X} \alpha(X) \cdot v(d\alpha)} \right] \\
&\stackrel{(iii)}{=} \left[ \frac{\int_{\alpha \in \mathcal{D}X} \alpha(U) \cdot v(d\alpha)}{v(\mathcal{D}X)} \right]
\end{aligned}$$

Lastly, we may divide by cases: whether  $v(\mathcal{D}X) = 0$  or  $v(\mathcal{D}X) \neq 0$ , we may use linearity of the integral to equate the two sides.  $\square$

**Lemma H.1.** *BorelNorm is a left-relevant magmoid.*

**PROOF.** Our proof strategy is to apply Lemma G.2: we will prove that, in **BorelNorm**, deterministic morphisms are precisely those of the form  $u \circ M\eta^D$ ; while total morphisms are precisely those of the form  $w \circ D\eta^M$ . Let us highlight that this is not automatic nor does it follow from the discrete case: indeed, if we were to allow normalized kernels between arbitrary measurable spaces, Lemma G.2 would not apply: deterministic maps would not coincide with measurable maps [Fri20, Example 10.4].

Any normalized kernel between standard Borel spaces,  $f: X \rightarrow M\mathcal{D}Y$ , can be regarded as a stochastic kernel to the standard Borel space  $MY$ . Any such deterministic kernel must satisfy, by definition, that  $f(x; S)^2 = f(x; S)$ . Thus, it must create measures yielding either 0 or 1; that is, taking values in the set  $\{0, 1\}$ . In standard Borel spaces,  $\{0, 1\}$ -valued measures correspond to delta measures [Fri20, Example 10.5, Example 10.4]. Finally, a delta measure over  $MY$  is either a delta measure over  $Y$  or a zero measure. In other words, deterministic kernels correspond to partial measurable functions.

Any total kernel between standard Borel spaces,  $f: X \rightarrow M\mathcal{D}Y$ , must be such that  $f(x; Y) = 1$ ; that is, it must yield a full measure. This means that there exists a stochastic kernel,  $u(x; S) = f(u; S)$ , such that  $f = u \circ M\eta^D$ .

Applying Lemma G.2, we conclude that normalized kernels between standard Borel spaces form a left-relevant magmoid.  $\square$

**Theorem 8.7.** *BorelNorm is a Markov magmoid.*

**PROOF.** We separately prove that normalized kernels between standard Borel spaces form a quasitotal magmoid (Proposition 7.5 and Theorem 8.4), and that they form a left-relevant magmoid (Lemma H.1). We only need to show that it admits conditionals.

We will use that **BorelSubstoch** has quasitotal conditionals. This is not a direct result, but has already been shown in the context of partial Markov categories [DR23, Theorem 3.12].

Thanks to the sesquilaw, we know that there exists a faithful functor  $(-)^{\bullet}: \mathbf{BorelNorm} \rightarrow \mathbf{BorelSubstoch}$ . Quasitotal maps in **BorelSubstoch** must satisfy  $f(x; Y)^2 = f(x; Y)$ , and thus they must

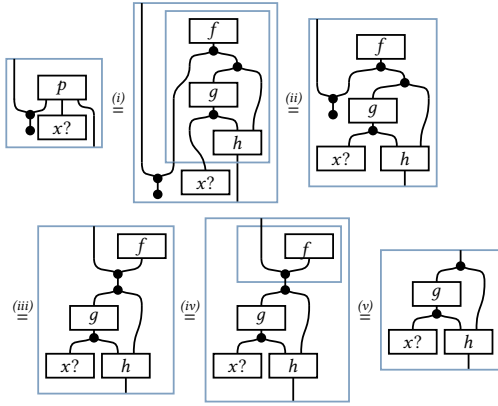


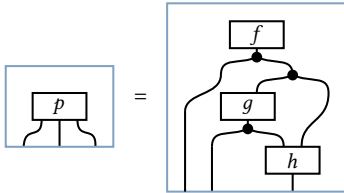
Figure 5: Back-door formula, second part of the proof.

yield normalized measures: in other words, they must be of the form  $f = u^*$  for some normalized channel  $u$ .

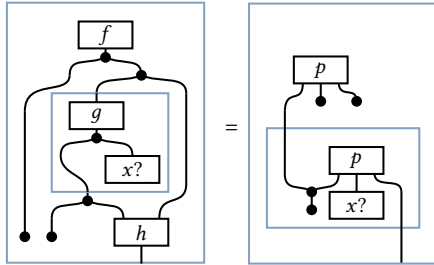
Finally, because **BorelSubstoch** admits quasitotal conditionals,  $f^*$  must have a conditional  $c$  that, being quasitotal, must be of the form  $c = v^*$ . Then, by faithfulness, we conclude that  $v$  must be a conditional of  $f$ .  $\square$

## I Proofs for Section 9 (Discrete Markov magmoids)

**Proposition 9.6 (Back-door adjustment formula).** *In a discrete Markov magmoid, let a joint state,  $p: 1 \rightarrow U \otimes X \otimes Y$ , admit the following factorization into total morphisms where, moreover,  $f$  has full support.*



Then, the following equation holds.



**PROOF.** Let us first simplify the uppermost part of the diagram (Figure 4). We use (i) the factorization assumption, (ii) the left-bias axiom of string diagrams for commutative magmoids, together with associativity of the comultiplication, and (iii) the assumption

that the morphisms  $g$  and  $h$  are total, together with the comonoid axioms.

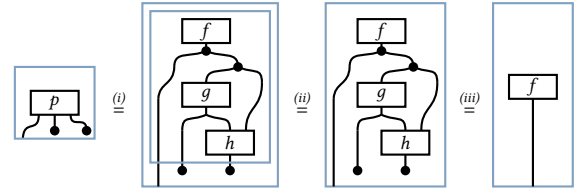


Figure 4: Back-door formula, first part of the proof.

Let us simplify the second part of the diagram (Figure 5). We now use (i) the factorization assumption, (ii) the left-bias axiom of string diagrams for commutative magmoids, (iii) the Frobenius equation, (iv) the left-bias axiom of string diagrams for commutative magmoids, and (v) the assumption that  $f$  has full support.

Let us conclude the proof (Figure 6). We (i) substitute both simplifications on the original statement, (ii) we apply the left-bias axiom of string diagrams for commutative magmoids, (iii) we use that we are in a left-relevant magmoid, and (iv) we apply the monoidality axiom of string diagrams for commutative magmoids.  $\square$

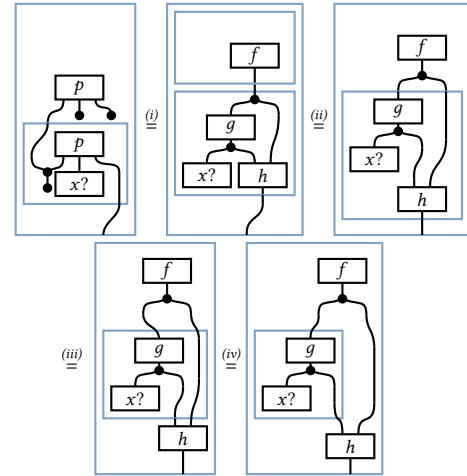
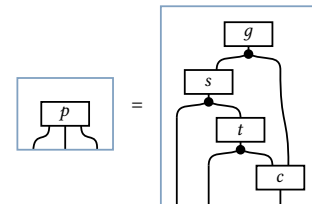


Figure 6: Back-door formula, final part of the proof.

**Proposition 9.8 (Front-door adjustment formula).** *In a discrete Markov magmoid, let a joint state,  $p: 1 \rightarrow X \otimes Z \otimes Y$ , admit the following factorization into total morphisms where, moreover, both  $t$  and  $(g \circ s)$  below have full support.*



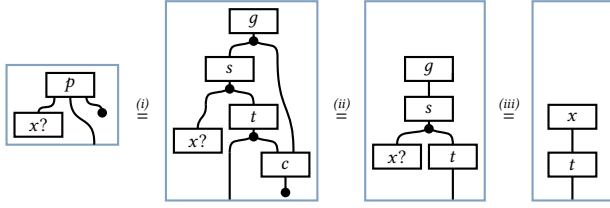


Figure 9: Front-door formula, third part of the proof.

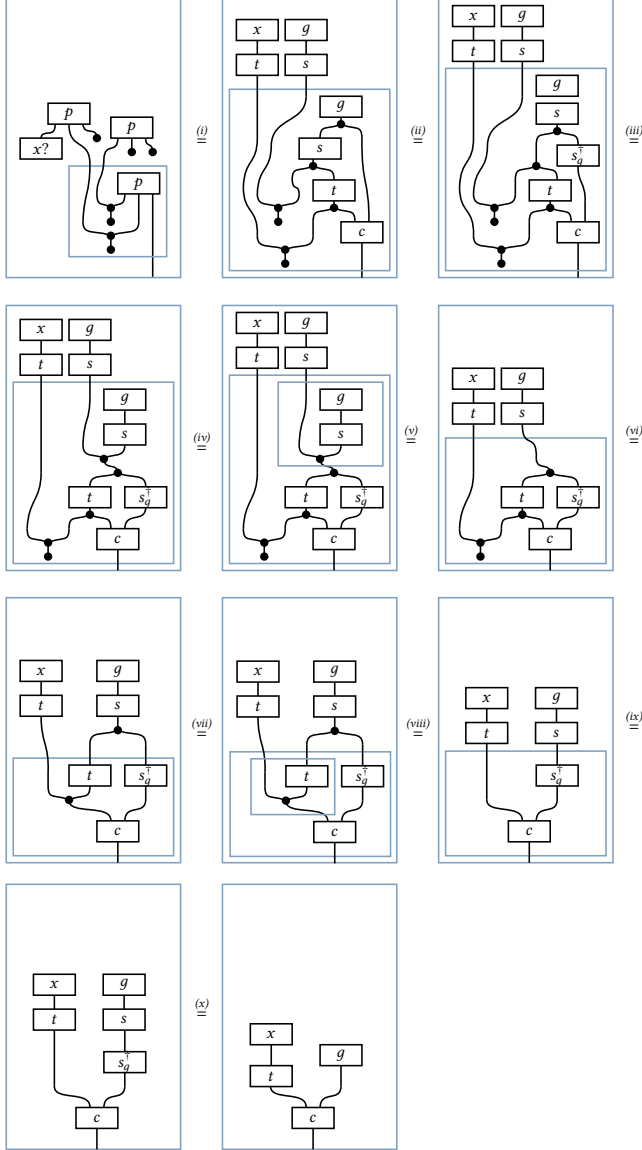
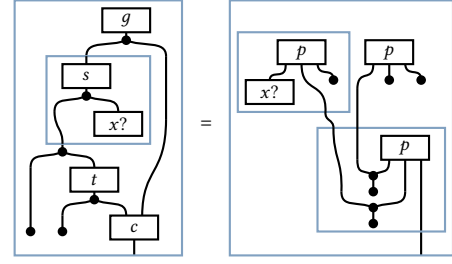


Figure 10: Proof of the synthetic front-door criterion.

Then, the following equation holds.



In other words, an intervention on the variable  $X$  can be rewritten as a composition, in the Markov magmoid, of the observational data.

PROOF. Let us start by simplifying the left-hand side (Figure 7). We use (i) simplification of an exact observation and full support of  $s$ , and (ii) the comonoid axioms.

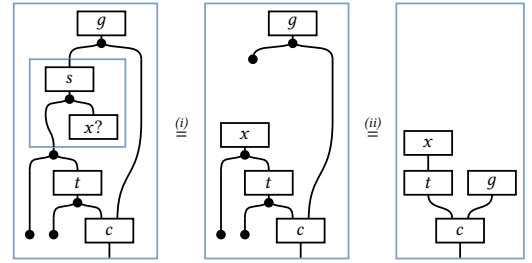


Figure 7: Front-door formula, first part of the proof.

Let us now simplify each one of the two factors. The first factor simply evaluates part of the distribution (Figure 8). We use (i) the axiom of left-relevant magmoids, after substitution, and (ii) the comonoid axioms.

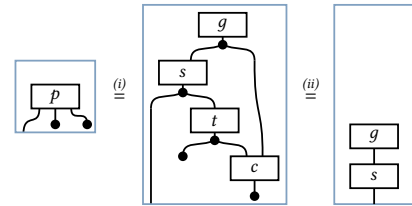


Figure 8: Front-door formula, second part of the proof.

The second factor computes the marginal of a conditional distribution (Figure 9). We use (i) the left-bias axiom of string diagrams for commutative magmoids, after substitution; (ii) totality of  $c$  and the comonoid axioms; and (iii) the simplification of an exact observation together with the full support assumption of  $(g \circ s)$ .

Finally, let us address the main claim (Figure 10). We use (i,iv,vii) the left-bias axiom of string diagrams for commutative magmoids, after substitution; (ii) the existence of Bayesian inverses, from conditionals (see any text on Markov categories, e.g. [FR20]), together with the left-bias axiom of string diagrams for commutative magmoids; (iii) the Frobenius equation; (iv) the left-bias axiom of string diagrams for commutative magmoids; (v) the full support assumption of  $(g \circ s)$ ; (vi) the Frobenius equation, together with associativity of the comultiplication; (viii) the full support assumption on  $t$ ; (ix) the axiom of left-relevant magmoids; and (x) the definition of Bayesian inverses.  $\square$



## J Proofs for Section 10 (Conclusions)

**Theorem J.1** (Tricocycloid to sesquilaw). *Every Set-based tricocycloid induces a Set-based sesquilaw.*

PROOF SKETCH. Let us first recall that every tricocycloid,  $H$ , induces a linear exponential monad  $T$ , satisfying  $T(X + Y) \cong TX + H \times TX \times TY + TY$ .

From this property, let us construct a distributive law and an almost distributive law. The first is the distributive law over the Maybe monad, described by inclusion

$$(-)^{\bullet}: TX + 1 \rightarrow TX + H \times TX + 1;$$

and the second uses projections to extract an element of the monad from the first and the second summand,

$$(-)^{\circ}: TX + H \times TX + 1 \rightarrow TX + 1.$$

Additionally, note that we could construct another distributive law that sends the second summand to the failure element,

$$(-)^{*}: TX + H \times TX + 1 \rightarrow TX + 1;$$

this would yield a *black-hole semantics*.

It remains to show that these induce a sesquilaw. Let us write  $X \otimes Y = X + H \times X \times Y + Y$  and let us check that the axiom for sesquilaws holds.

$$\begin{array}{ccccc} TTX \otimes 1 & \xrightarrow{\circ} & TTX + 1 & \xrightarrow{\mu+1} & TX + 1 \\ \downarrow i & & & \nearrow \circ & \\ TTX \otimes T(H \times TX) \otimes 1 & \xrightarrow{\mu \otimes 1} & TX \otimes 1 & & \end{array}$$

This diagram can be split by an arrow,  $\mu \otimes 1: TTX \otimes 1 \rightarrow TX \otimes 1$ , the upper quadrangle then commutes because of naturality; the left triangle can be checked to commute by cases on the coproduct.  $\square$

## K Complete Racket implementation

The Racket v9.0 code is also available as an anonymized repository [DRS26].

### DISTRIBUTIONS.RKT

```
1 #lang racket
2
3 ;; NORMALIZED DISTRIBUTIONS.
4 ;;
5 ;; This file implements normalized distributions and
6 ;; subdistributions using a standard encoding as
7 ;; lists of pairs element/probability.
8
9 (require racket/struct)
10 (require sesquilaw/left-do)
11
12
13 (define (pair x y) (list x y))
14
15 (define (validity xs)
16   (apply + (map second xs)))
17
18 (define/match (dist-map f xs)
19   [(f '()) null]
20   [(f (cons (list x v) ys))
21    (cons (list (f x) v) (dist-map f ys)) ])
22
23 (define/match (dist-map-values f xs)
24   [(f '()) null]
25   [(f (cons (list x v) ys))
26    (cons (list x (f v)) (dist-map-values f ys)) ])
27
28 (define/match (remove-zeroes xs)
```

```
29   [( '()) null]
30   [( (cons (list x 0) ys)) (remove-zeroes ys) ]
31   [( (cons (list x v) ys))
32    (cons (list x v) (remove-zeroes ys)) ])
33
34 (define/match (weight-of-point x xs)
35   [(x '()) 0]
36   [(x (cons (list x v) ys))
37    (+ v (weight-of-point x ys)) ]
38   [(x (cons (list y v) ys))
39    (weight-of-point x ys) ])
40
41 (define/match (dist-remove x xs)
42   [(x '()) '()]
43   [(x (cons (list x v) xs)) (dist-remove x xs) ]
44   [(x (cons (list y v) xs))
45    (cons (list y v) (dist-remove x xs)) ])
46
47 (define/match (reweight xs)
48   [( '()) '()]
49   [( (cons (list x v) ys))
50    (let ([w (+ v (weight-of-point x ys))])
51      (cons (list x w) (reweight (dist-remove x ys)))) )])
52
53 (define (condense xs) (reweight (remove-zeroes xs)))
54
55 (define/match (rescale xss)
56   [( (list xs v))
57    (dist-map-values (lambda (x) (* v x)) xs) ])
58
59 ;; Monad combinators.
60
61 (define (dist-join xss)
62   (condense (apply append (map rescale xss))))
63
64 (define (dist-normalize xs)
65   (condense
66    (dist-map-values
67     (lambda (v) (/ v (validity xs))) xs)))
68
69 (define (dist-bind xs f)
70   (dist-join (dist-map f xs)))
71
72 (define (dist-return x)
73   (list (pair x #e1)))
74
75 ;; Basic distributions.
76
77 (define (dist-uniform ls)
78   (map (lambda (x) (pair x (/ #e1 (length ls)))) ls))
79
80 (define dist-void
81   (list))
82
83 (define-syntax distribution
84   (syntax-rules ()
85     [( _ [x v] p ... )
86      (cons (pair x v) (distribution p ...)) ]
87     [( _ ) (list)]))
88
89 (define-syntax uniform
90   (syntax-rules ()
91     [( _ x ... ) (dist-uniform (list x ...))]))
92
93 (provide
94   dist-bind dist-return dist-uniform dist-void
95   dist-normalize distribution uniform pair)
```

### NORM.RKT

```
1 #lang racket
2
3 ;; NORM.
4 ;; An implementation of the normalization almost monad,
```

```

5 ;; inheriting the return from distributions and using a
6 ;; modified bind that renormalizes. We also implement
7 ;; an observe statement.
8
9 (require sesquialaw/left-do)
10 (require sesquialaw/distributions)
11
12
13 (define norm-return
14   dist-return)
15
16 (define (norm-bind xs f)
17   (dist-normalize (dist-bind xs f)))
18
19 (define Norm
20   (monad norm-return norm-bind))
21
22 (define (observe x y)
23   (if (equal? x y)
24       (uniform '())
25       (uniform)))
26
27
28 (provide Norm norm-return norm-bind observe)

```

**RIGHT-DO.RKT**

```

1 #lang racket
2
3 ;; DO-NOTATION.
4 ;; This file implements do-notation for magmoids: both
5 ;; right-associating do-notation (the usual one in
6 ;; Haskell) and left-associating do-notation (which is
7 ;; novel in magmoids).
8
9
10 ;; ALMOST MONADS.
11 ;; The following is a common interface to monad-like
12 ;; structures that do not necessarily satisfy any of the
13 ;; monad axioms.
14
15 (struct monad (return bind))
16
17
18 ;; RIGHT DO-NOTATION.
19 ;; Given a monad m, do notation is implemented
20 ;; inductively by the following two rewriting rules.
21 (define-syntax rDo
22   (syntax-rules (<- return)
23
24     ;; (1) A statement (x <- f), followed by the rest of
25     ;; the program (p ...) is translated into a Kleisli
26     ;; extension,  $(\lambda x. \{ p \})^\dagger (f)$ .
27     [ (rDo m
28       x <- f
29       p ...)
30
31       ((monad-bind m) f
32        (match-lambda [x (rDo m p ...)]))] ]
33
34     ;; (2) A return (return x) is translated to the unit
35     ;; of the monad,  $\eta(x)$ .
36     [ (rDo m
37       return x)
38
39       ((monad-return m) x) ]])
40
41 (provide rDo (struct-out monad))

```

**LEFT-DO.RKT**

```

1 #lang racket
2
3 (require sesquialaw/right-do)
4
5 ;; LEFT DO-NOTATION.
6

```

```

7 ;; Let us now implement left do-notation in terms of
8 ;; right do-notation. It will be inductively implemented
9 ;; as a left-fold with an accumulator that contains the
10 ;; first part of the program. Let us first describe the
11 ;; more general accumulator version: the left-associating
12 ;; version is the particular case that leaves the
13 ;; accumulator empty.
14
15 (define-syntax accDo
16   (syntax-rules (<- return)
17
18     ;; (1) A statement (x <- f), followed by the rest of
19     ;; the program (p ...) is added to the accumulator
20     ;; (acc) and the variable is added to the accumulated
21     ;; variables (accVar).
22     [ (accDo m accVar acc
23       x <- f
24       p ...)
25
26       (accDo m (list x accVar) (rDo m
27         accVar <- acc
28         x <- f
29         return (list x accVar))
30       p ...) ]
31
32     ;; (2) A return statement (return x) is translated to
33     ;; a right do-notation block evaluating the
34     ;; accumulator.
35     [ (accDo m accVar acc
36       return x)
37
38       (rDo m
39         accVar <- acc
40         return x) ]])
41
42
43 ;; Finally, we declare that a left-associating do
44 ;; notation block is the same as an accumulating
45 ;; do-notation block with an empty accumulator.
46 (define-syntax lDo
47   (syntax-rules (<- return)
48
49     ;; (3) An arbitrary left-associating block is the
50     ;; same as an accumulator block with an empty
51     ;; accumulator.
52     [ (lDo m
53       p ...)
54
55       (accDo m '()
56         (rDo m
57           return '())
58         p ...) ]])
59
60
61 (provide rDo lDo (struct-out monad))

```

**MONTY-HALL.RKT**

```

1 #lang racket
2
3 ;; MONTY HALL.
4 ;; This file implements the Monty Hall problem, a famous
5 ;; probability puzzle originally posed by Steve Selvin in
6 ;; a letter to American Statistician.
7 ;;
8 ;; Reference:
9 ;; Letters to the Editor. American Statistician.
10 ;; Steve Selvin, 1975.
11
12 (require sesquialaw/left-do)
13 (require sesquialaw/distributions)
14 (require sesquialaw/norm)
15
16
17 ;; DESCRIPTION.
18
19 ;; We are in a game show, and a prize (a car, in the

```

## The Magmoid of Normalized Stochastic Kernels

```

20 ;; original is hidden behind one of three doors (left,
21 ;; middle, and right). We constestant picks a door (say,
22 ;; the middle one). For dramatic effect, the host opens
23 ;; one of the non-chosen doors (say, the left one). The
24 ;; host does so avoiding the door that does contain the
25 ;; car (for it would spoil the show) and otherwise
26 ;; randomly and uniformly. Finally, the host offers us to
27 ;; change doors and pick the other one that remains
28 ;; closed. Should we change doors?
29
30 ;; HOST.
31 ;; Let us first formalize the behaviour of the host: it
32 ;; picks randomly and uniformly among the doors that have
33 ;; not been chosen and that, moreover, do not contain the
34 ;; car.
35 (define (host car choice)
36   (match (cons car choice)
37     [(cons 'left 'left) (uniform 'middle 'right)]
38     [(cons 'left 'middle) (uniform 'right)]
39     [(cons 'left 'right) (uniform 'left)]
40     [(cons 'middle 'left) (uniform 'right)]
41     [(cons 'middle 'middle) (uniform 'left 'right)]
42     [(cons 'middle 'right) (uniform 'left)]
43     [(cons 'right 'left) (uniform 'middle)]
44     [(cons 'right 'middle) (uniform 'left)]
45     [(cons 'right 'right) (uniform 'left 'middle)]))
46
47 ;; FORMULATION.
48
49 ;; Let us formalize the Monty Hall problem using
50 ;; do-notation. We repeat the exact same formalization
51 ;; twice: once using right-associating do-notation and
52 ;; once using left-associating do-notation. The result
53 ;; will be different in both cases.
54 ;;
55 ;; The program lines mean that
56 ;; (1) the car is distributed uniformly;
57 ;; (2) the host (knowing our choice) opens a door;
58 ;; (3) we observe the host opened the left door.
59 ;;
60 ;; What is the probability distribution of the car?
61
62 (define l-monty-hall
63   (lDo Norm
64     car <- (uniform 'left 'middle 'right)
65     opened <- (host car 'middle)
66     '() <- (observe opened 'left)
67     return car))

```

```

68
69 (define r-monty-hall
70   (rDo Norm
71     car <- (uniform 'left 'middle 'right)
72     opened <- (host car 'middle)
73     '() <- (observe opened 'left)
74     return car))

```

### SMOKING.RKT

```

1 #lang racket
2
3 (require sesquialaw/left-do)
4 (require sesquialaw/distributions)
5 (require sesquialaw/norm)
6
7
8 (define survey
9   (distribution
10    [(list 'smoker 'tar 'nocancer) 323/800]
11    [(list 'smoker 'tar 'cancer) 57/800]
12    [(list 'nonsmoker 'tar 'nocancer) 1/800]
13    [(list 'nonsmoker 'tar 'cancer) 19/800]
14    [(list 'smoker 'notar 'nocancer) 18/800]
15    [(list 'smoker 'notar 'cancer) 2/800]
16    [(list 'nonsmoker 'notar 'nocancer) 38/800]
17    [(list 'nonsmoker 'notar 'cancer) 342/800]))
18
19 (define (front-door data i)
20   (lDo Norm
21     z <- (lDo Norm
22       (list xp z yp) <- data
23       '() <- (observe i xp)
24       return z)
25     x <- (lDo Norm
26       (list x zp yp) <- data
27       return x)
28     y <- (lDo Norm
29       (list xp zp y) <- data
30       '() <- (observe x xp)
31       '() <- (observe z zp)
32       return y)
33     return y))
34
35 (front-door survey 'smoker)
36 (front-door survey 'nonsmoker)

```

## **Rights statement**

*For the purpose of Open Access the Author has applied a Creative Commons Attribution-ShareAlike 4.0 International public copyright license to any Author Accepted Manuscript version arising from this submission.*