Evidential Decision Theory via Partial Markov Categories

Elena Di Lavore and Mario Román

Tallinn University of Technology, Tallinn, Estonia.
elendi@ttu.ee, mroman@ttu.ee

Abstract

We introduce the algebraic structure of partial Markov categories and we use it to encode a synthetic formulation of Evidential Decision Theory. The theory is implemented in the monoidal Kleisli category of the finitary subdistribution monad.

1 Introduction

Evidential Decision Theory is a paradigm in decision theory that focuses on *observational evidence*: given a decision problem, Evidential Decision Theory prescribes the action that we would *observe to have done* in the best possible outcome [Ahm14]. This contrasts with Causal Decision Theory, which prescribes the action that *causes* the best possible outcome [GH78]. In the framework of Evidential Decision Theory, no direct causal connection is required for the action to affect the outcome: it suffices that the observation of the action alters the conditional probability of the outcome via Bayesian update [YS17].

However, formalizing Evidential Decision Theory makes for a subtle art: different decision theories disagree in many well-studied scenarios [GH78], and solutions are sensitive to slight modifications in the problem statement. In order to clarify these disagreements, we need both an intuitive mathematical syntax to model decision problems, and a formal algorithmic procedure to solve them according to the prescriptions of the theory. In this paper, we answer the following question:

What is a minimalistic mathematical framework that can formulate and solve decision problems following Evidential Decision Theory?

Markov categories allow a synthetic approach to probability theory in which we can formulate notions of *conditioning*, *independence*, or *Bayesian network* [Fon13, CJ19, Fri20]. Our main observation is that this framework can be elegantly extended to capture explicit notions of *observation* and *Bayesian update*. The resulting algebraic structure, *partial Markov categories*, proves to be a good theoretical framework for Evidential Decision Theory: partial Markov categories provide both a convenient syntax in terms of string diagrams and a straightforward translation from these diagrams to a simple probabilistic programming language that computes the solution to the given decision problem.

Contributions. Our main contribution is an extension of the framework of Markov categories that can be used to formalize *Evidential Decision Theory*. We introduce the notion of *partial Markov category*, translating the idea of partial Frobenius monoids in *discrete cartesian restriction categories* [CGH12, CO89, DLNS21] to the stochastic setting.

Funding. Elena Di Lavore and Mario Román were supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and the Estonian Research Council grant PRG1210.

2 Partial Markov categories

Symmetric monoidal categories are a framework to reason about processes and, in particular, probabilistic processes. *Markov categories* are an abstract algebra of probabilistic processes proposed by Cho and Jacobs and then unified by Fritz [Fri20, CJ19]. The setting of Markov categories, however, is too restrictive for our aims: it is not possible to encode Bayesian updates *as morphisms* in a Markov category (Remark B.3).

Partial cartesian categories, introduced as *discrete cartesian restriction categories* [CGH12], are an extension of cartesian categories that allows for the encoding of *constraints*: a map may fail if some conditions are not satisfied. Our observation is that a similar extension can be applied to Markov categories to obtain *partial Markov categories*. They provide a setting in which it is possible to (i) constrain, via Bayesian updates; and (ii) reason with stochastic maps.

Definition 2.1. A *partial Markov category* is a symmetric monoidal category $(\mathsf{C}, \otimes, I)$ such that (i) every object $X \in \mathsf{C}_{\mathrm{obj}}$ has a partial Frobenius monoid (Figure 1) structure $(X, \delta_X, \varepsilon_X, \mu_X)$ which is uniform, meaning that $\varepsilon_{X\otimes Y} = \varepsilon_X \otimes \varepsilon_Y$, $\varepsilon_I = \mathrm{id}$, $\delta_{X\otimes Y} = (\delta_X \otimes \delta_Y) ⨟ (\mathsf{id}_X \otimes \sigma_{X,Y} \otimes \mathsf{id}_Y)$, $\delta_I = \mathrm{id}$, $\mu_{X\otimes Y} = (\mathsf{id}_X \otimes \sigma_{X,Y} \otimes \mathsf{id}_Y) ⨟ (\mu_X \otimes \mu_Y)$, and $\mu_I = \mathsf{id}$; and (ii) every morphism $f \colon X \to Y_1 \otimes Y_2$ splits both as $\delta ⨟ (\mathsf{id} \otimes (f_2 ⨟ \delta_{Y_2})) ⨟ (g_1 \otimes \mathsf{id})$ and as $\delta ⨟ ((f_1 ⨟ \delta_{Y_1}) \otimes \mathsf{id}) ⨟ (\mathsf{id} \otimes g_2)$ for some $f_1 \colon X \to Y_1$, some $f_2 \colon X \to Y_2$, some $g_1 \colon X \otimes Y_2 \to Y_1$ and some $g_2 \colon Y_1 \otimes X \to Y_2$.

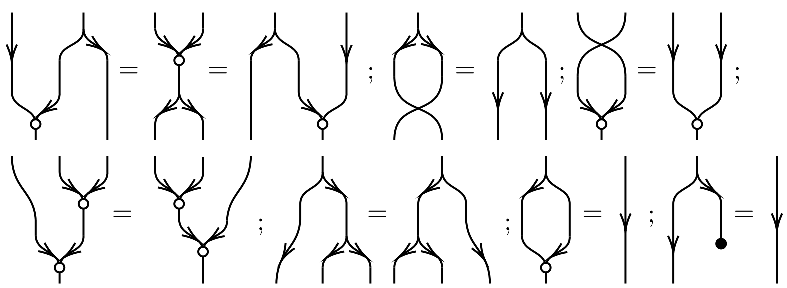

Figure 1: Axioms of a partial Frobenius monoid.

The *Bayesian inversion* of a stochastic channel $g \colon X \to Y$ with respect to a distribution f over X is the stochastic channel $g^{\dagger_f} \colon Y \to X$ classically defined as in Figure 2, left. Bayesian inversions can be defined in any partial Markov category. The *Bayesian inversion* of a morphism $g \colon X \to Y$ with respect to $f \colon I \to X$ is a morphism $g^{\dagger_f} \colon Y \to X$ such that there is a morphism $h \colon I \to Y$ satisfying the equation (i) in Figure 2, right.

$$g^{\dagger_f}(x|y) = \frac{g(y|x)f(x)}{\sum_{x_\bullet \in X} g(y|x_\bullet)f(x_\bullet)};$$

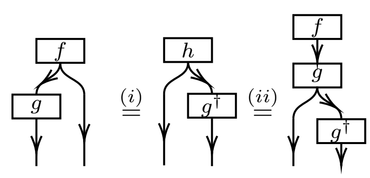

Figure 2: Bayesian inversion.

If the Bayesian inversion, $g^{\dagger_f}$, is moreover *total* (Definition B.5), then the map h is forced to be $f ⨟ g$, as in equation (ii). For instance, in the category of subdistributions, $\mathsf{Kl}(\mathsf{D}_{\leq 1})$ (Appendix B.4), which is the main example of partial Markov categories, there always exists a total Bayesian inversion, even when not all maps are total.

Theorem 2.2 (Bayes' Theorem in a partial Markov category)**.** *An observation induces an update that follows the formula of Bayesian inversion. In other words, observing a deterministic $y \in Y$ from the prior distribution $f : I \to X$ through a channel $g : X \to Y$ is the same, up to scalar, as evaluating the total bayesian inversion of the channel $g^{\dagger_f}$ on y.*

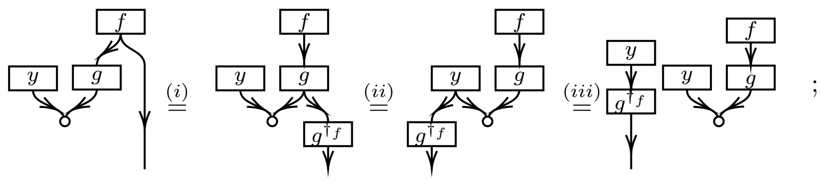

Figure 3: Proof of Bayes' Theorem.

Proof. We employ string diagrams (Figure 3). Equalities follow from: (i) the definition of Bayesian inversion, (ii) the partial frobenius axioms, and (iii) determinism of y. □

3 Evidential Decision Theory

Our formulation of Evidential Decision Theory starts by constructing a model of the given decision problem. The model is described as a morphism in a free partial Markov category and depicted as a string diagram. We give semantics to the model by assigning a subdistribution to each node in the string diagram. We formulate a question about one of these nodes, a special node which we colour in grey. The optimal answer to the problem will be the one that, once observed as the output of this node, maximises the outcome. In this manuscript, we model two decision problems (see also Appendix A) and explicitly compute their solutions in Appendix C.

3.1 Newcomb's Problem

An *agent* confronts two boxes: the box **A** contains 1€; the box **B** contains either 1000€ or 0€. A *predictor* made a prediction and claimed to have left money in **B** if and only if the agent will leave **A** behind. The predictor is perfectly reliable or, in some formulations, reliable with a very high probability [Noz69, YS17] (Figure 4).

Should the agent take both boxes (*"two-boxing"*) or just take the **B** box (*"one-boxing"*)?

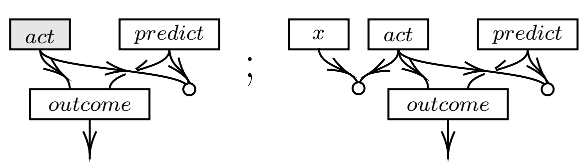

Figure 4: Model for the Newcomb problem and observation.

Causal Decision Theory prescribes two-boxing [GH78], but a good predictor would anticipate this and leave **B** empty, for a final utility of 1€. Evidential Decision Theory prescribes one-boxing: if we observe one-boxing, by the premise, we must also observe that the predictor filled both boxes. In this case, the agent would get 1000€.

Acknowledgements

We thank Siddharth Bhat, Pim de Haan, Miguel Lopez and Ruben Van Belle for discussion, many helpful suggestions on the first versions of this manuscript and, in particular, for the joint ongoing work on the applications of partial Markov categories to the study of causal networks.

References

[ACU15] Thorsten Altenkirch, James Chapman, and Tarmo Uustalu. Monads need not be endofunctors. *Logical Methods in Computer Science*, 11(1), 2015.

[Ahm14] Arif Ahmed. *Evidence, Decision and Causality.* Cambridge University Press, 2014.

[CG99] Andrea Corradini and Fabio Gadducci. An Algebraic Presentation of Term Graphs, via GS-Monoidal Categories. *Applied Categorical Structures*, 7(4):299–331, 1999.

[CGH12] J. Robin B. Cockett, Xiuzhan Guo, and Pieter Hofstra. Range Categories I: General theory. *Theory and Applications of Categories*, 26(17):412–452, 2012.

[CJ19] Kenta Cho and Bart Jacobs. Disintegration and Bayesian Inversion via String Diagrams. *Mathematical Structures in Computer Science*, pages 1–34, March 2019.

[CO89] Pierre-Louis Curien and Adam Obtulowicz. Partiality, cartesian closedness and toposes. *Inf. Comput.*, 80(1):50–95, 1989.

[DLNS21] Ivan Di Liberti, Fosco Loregiàn, Chad Nester, and Pawel Sobocinski. Functorial semantics for partial theories. *Proc. ACM Program. Lang.*, 5(POPL):1–28, 2021.

[FL22] Tobias Fritz and Wendong Liang. Free GS-monoidal categories and free Markov categories. *CoRR*, abs/2204.02284, 2022.

[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.

[Fri20] Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020.

[GH78] Allan Gibbard and William L. Harper. Counterfactuals and two kinds of expected utility. In *Ifs*, pages 153–190. Springer, 1978.

[HHJW07] Paul Hudak, John Hughes, Simon L. Peyton Jones, and Philip Wadler. A history of Haskell: being lazy with class. In Barbara G. Ryder and Brent Hailpern, editors, *Proceedings of the Third ACM SIGPLAN History of Programming Languages Conference (HOPL-III), San Diego, California, USA, 9-10 June 2007*, pages 1–55. ACM, 2007.

[HKSY17] Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A convenient category for higher-order probability theory. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[Jac19] Bart Jacobs. The mathematics of changing one's mind, via jeffrey's or via pearl's update rule. *J. Artif. Intell. Res.*, 65:783–806, 2019.

[Noz69] Robert Nozick. Newcomb's Problem and Two Principles of Choice. In *Essays in honor of Carl G. Hempel*, pages 114–146. Springer, 1969.

[Pea09] Judea Pearl. *Causality.* Cambridge university press, 2009.

[Ste21] Dario Maximilian Stein. Structural foundations for probabilistic programming languages. *University of Oxford*, 2021.

[SWY+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.

[YS17] Eliezer Yudkowsky and Nate Soares. Functional Decision Theory: a New Theory of Instrumental Rationality. *ArXiv preprint arXiv:1710.05060*, 2017.

A Examples in Evidential Decision Theory

A.1 Death in Damascus

In a completely deterministic world, Death collects people on a designated place on a designated day. If the chosen people is not there to confront Death, they survive (which represents a great utility, say, 1000u).

The legend says that a merchant found Death in Damascus, and Death promised to come for him in the next day. The merchant thought of fleeing to Aleppo, trying to escape death; but that came with a cost (a small negative utility, say, -1u). However, Death is a perfect predictor, so the merchant found Death in Aleppo. Should the merchant have fled to Aleppo?

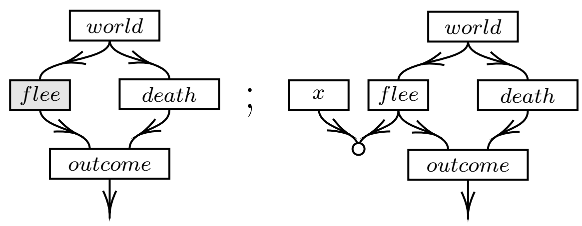

Figure 5: Death in Damascus, deterministic version.

Consider the model in Figure 5. Evidential Decision Theory prescribes just waiting for Death in Damascus. In this model, if Death is really omniscient, it will be impossible to avoid it. It only makes sense to avoid the small negative utility of a last trip to Aleppo, accepting 0u.

A.2 Cheating Death in Damascus with a random oracle

The reader may observe this problem could have been also modelled in a way similar to Newcomb's. The only difference is that, in this case, the predictor is always *adversarial*. How could we cheat against such a predictor? A possible answer is to allow ourselves to use a true random oracle: if we were to decide whether to flee to Damascus or Aleppo based on a random oracle that even Death cannot predict, we would still have a chance of cheating Death.

In this second formulation of the problem (Figure 6), we can use a fair coin that Death cannot predict. What is the strategy the agent should follow?

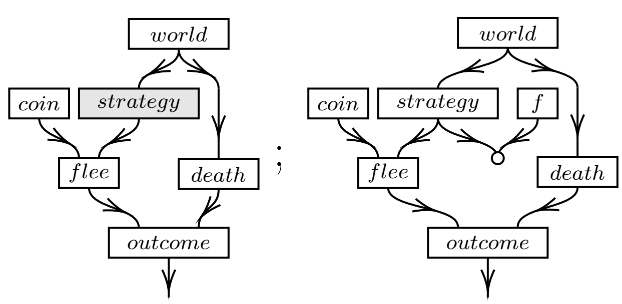

Figure 6: Cheating Death in Damascus with a random coin.

Evidential Decision Theory now prescribes that the agent should use the coin to try to cheat Death. This is no longer a lost cause: the expected utility is now 445.5u, see Appendix C.2.

B Monoidal Categories

B.1 Cartesian monoidal categories

A symmetric monoidal category $(\mathsf{C}, \otimes, I)$ is said to be *cartesian monoidal* whenever the tensor $A \otimes B$ of two objects A and B is their categorical product, and associators and unitors are derived from the universal property of the categorical product.

A well-known characterization of cartesian monoidal categories is *Fox's theorem* [Fox76], which states that a symmetric monoidal category is cartesian if and only if every object has a comonoid structure that is natural and coherent with the monoidal structure. We present a similar characterization: cartesian monoidal categories are monoidal categories where all joint maps split.

Proposition B.1. *A symmetric monoidal category $(\mathsf{C}, \otimes, I)$ is cartesian if and only if (i) every object $X \in \mathsf{C}_{\text{obj}}$ has a uniform comonoid structure $(X, \varepsilon_X, \delta_X)$, meaning that $\varepsilon_{X\otimes Y} = \varepsilon_X \otimes \varepsilon_Y$, $\varepsilon_I = \mathrm{id}$ and that $\delta_{X\otimes Y} = \delta_X \otimes \delta_Y$, $\delta_I = \mathrm{id}$; (ii) every morphism $f \colon X \to I$ is equal to ε_X; and (iii) every morphism $f \colon X \to Y_1 \otimes Y_2$ splits as $\delta_X \mathbin{⨾} (f_1 \otimes f_2)$ for some $f_1 \colon X \to Y_1$ and $f_2 \colon X \to Y_2$.*

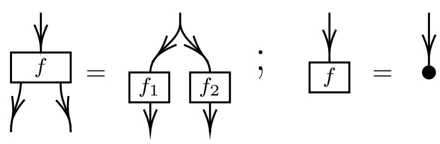

Figure 7: Cartesian category

Proof. We will show that the comonoid structure is natural. Firstly, we know from the premises that the discard is natural: that is, $f \mathbin{⨾} \varepsilon_Y = \varepsilon_X$ for each $f \colon X \to Y$. We now show that every time we split $f \colon X \to Y_1 \otimes Y_2$ into f_1 and f_2, we are forced to admit that $f_1 = f \mathbin{⨾} (\mathrm{id}_{Y_1} \otimes \varepsilon_{Y_2})$ and that $f_2 = f \mathbin{⨾} (\varepsilon_{Y_1} \otimes \mathrm{id}_{Y_2})$. This implies in turn that copying must be a natural transformation: given any $g \colon X \to Y$, we know that $f = g \mathbin{⨾} \delta_Y$ must split into $f = \delta_X \mathbin{⨾} (g \otimes g)$.

By Fox's theorem, that means that the category is cartesian. □

B.2 Markov categories

Markov categories [Fri20] have been defined as categories with a uniform comonoid structure where the counit is moreover natural. The Markov categories better suited for probability theory are those that have *conditionals*. We decide to call *Markov categories* only to those with conditionals. The purpose of this slight change of convention is to make the parallel with cartesian categories more explicit: Markov categories are just cartesian categories with a weaker splitting pattern.

Definition B.2. A *Markov category* is a symmetric monoidal category $(\mathsf{C}, \otimes, I)$ such that (i) every object $X \in \mathsf{C}_{\text{obj}}$ has a uniform comonoid structure $(X, \delta_X, \varepsilon_X)$, meaning that $\varepsilon_{X\otimes Y} = \varepsilon_X \otimes \varepsilon_Y$, $\varepsilon_I = \mathrm{id}$ and that $\delta_{X\otimes Y} = (\delta_X \otimes \delta_Y) \mathbin{⨾} (\mathsf{id}_X \otimes \sigma_{X,Y} \otimes \mathsf{id}_Y)$, $\delta_I = \mathrm{id}$; (ii) every morphism $f \colon X \to I$ is equal to ε_X; and (iii) every morphism $f \colon X \to Y_1 \otimes Y_2$ splits both as $\delta \mathbin{⨾} (\mathsf{id} \otimes (f_2 \mathbin{⨾} \delta_{Y_2})) \mathbin{⨾} (g_1 \otimes \mathsf{id})$ and as $\delta \mathbin{⨾} ((f_1 \mathbin{⨾} \delta_{Y_1}) \otimes \mathsf{id}) \mathbin{⨾} (\mathsf{id} \otimes g_2)$ for some $f_1 \colon X \to Y_1$, some $f_2 \colon X \to Y_2$, some $g_1 \colon X \otimes Y_2 \to Y_1$ and some $g_2 \colon Y_1 \otimes X \to Y_2$.

In this situation, $g_2 \colon Y_1 \otimes X \to Y_2$ is said to be the *conditional* of f with respect to Y_1 and $g_1 \colon X \otimes Y_2 \to Y_1$ is said to be the *conditional* of f with respect to Y_2.

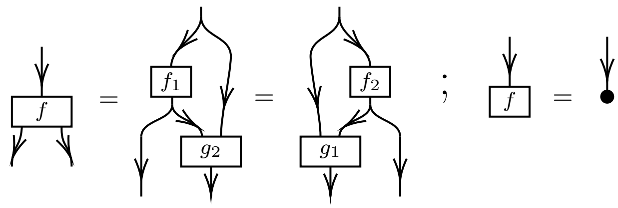

Figure 8: Markov category with conditionals.

Remark B.3. Performing a Bayesian update on a model means comparing an observation with a prediction in order to update the model that produced the prediction. In a monoidal category, a process that simply checks whether two inputs are equal must be of type $A \otimes A \to I$. The only possible map of this type in a Markov category is the trivial one and thus this comparison is not possible. This is what motivates the passage to partial Markov categories.

B.3 Copy-Discard categories and partial Markov categories

Copy-Discard categories is the name we give to any category where each object has a uniform comonoid structure. The comultiplication is what we call the "copy" and the counit is what we call the "discard": they have the same operations as cartesian monoidal categories, but neither is assumed to be natural. Copy-Discard categories have been called GS-monoidal categories when they have been applied to graph rewriting [CG99] (see [FL22, Remark 2.2] for a history of the term).

Definition B.4. A *Copy-Discard category* is a symmetric monoidal category C where every object $X \in \mathsf{C}$ has a cocommutative comonoid structure $(X, \delta_X, \varepsilon_X)$ and this structure is uniform: $\delta_{X\otimes Y} = (\delta_X \otimes \delta_Y) ⨟ (\mathsf{id}_X \otimes \sigma_{X,Y} \otimes \mathsf{id}_Y)$, $\delta_I = \mathsf{id}$, $\varepsilon_{X\otimes Y} = \varepsilon_X \otimes \varepsilon_Y$, and $\varepsilon_I = \mathsf{id}$.

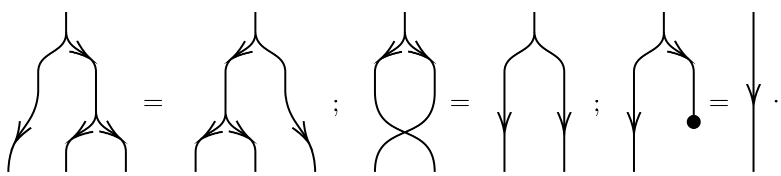

Figure 9: Axioms of a cocommutative comonoid.

Copying an discarding are not required to be natural: this means that only some of the morphisms can be copied or discarded. We call these *deterministic* and *total*, respectively.

Definition B.5. A morphism $f \colon X \to Y$ in a Copy-Discard category is called *deterministic* if $f ⨟ \delta_Y = \delta_X ⨟ (f \otimes f)$; and *total* if $f ⨟ \varepsilon_Y = \varepsilon_X$.

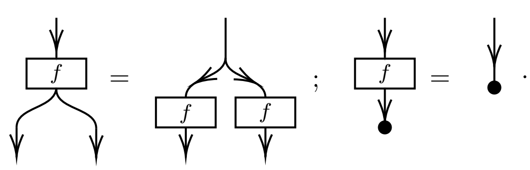

Figure 10: Deterministic and total morphisms.

Probability theory requires more structure than the one given by Copy-Discard monoidal categories. Explicitly, it is usually assumed that a category that encodes a theory of probability will have a notion of *conditional* [Fri20, CJ19].

Definition B.6. A Copy-Discard category C has *conditionals* if, for every morphism $f\colon X \to Y_1 \otimes Y_2$, there are $g_1\colon X \otimes Y_2 \to Y_1$, $f_1\colon X \to Y_1$, $g_2\colon Y_1 \otimes X \to Y_2$ and $f_2\colon X \to Y_2$ such that $f = \delta \,⨟\, (\mathsf{id} \otimes (f_2 \,⨟\, \delta_{Y_2})) \,⨟\, (g_1 \otimes \mathsf{id}) = \delta \,⨟\, ((f_1 \,⨟\, \delta_{Y_1}) \otimes \mathsf{id}) \,⨟\, (\mathsf{id} \otimes g_2)$, i.e. they satisfy the equations on the left in Figure 8.

Evidential probability theory can be encoded using *comparators*: a comparator declares that some constraint – usually an observation, on which we condition – needs to be satisfied in a probabilistic process.

Definition B.7. A Copy-Discard category C has *comparators* if every object X has a morphism $\mu_X\colon X \otimes X \to X$ that is uniform, commutative, associative and satisfies the Frobenius axioms with the Copy-Discard structure, as in Figure 1.

Definition B.8. A *Markov category* is a Copy-Discard category with conditionals, in which the counit is natural. A *partial Markov category* is a Copy-Discard category with conditionals and comparators.

Proposition B.9. *Bayesian inversions are just a particular case of conditionals. In a Copy-Discard category with conditionals, all Bayesian inversions exist.*

Proof. This can be easily checked by applying the axiom of conditionals to the morphism defined by $f \,⨟\, \delta_X \,⨟\, (g \otimes \mathsf{id})$. □

B.4 Subdistributions

A *subdistribution* σ over X is a distribution whose total probability is allowed to be less than 1. In other words, it is a distribution over $X + 1$. This means that a morphism $f\colon X \to Y$ in $\mathsf{Kl}(\mathsf{D}_{\leq 1})$ represents a stochastic channel that has some probability of failure. We interpret the value of $f(x)$ in $y \in Y$ as the probability of y given x according to the channel f, and we indicate it as $f(y \mid x)$.

The symmetric monoidal Kleisli category of the finitary subdistribution monoidal monad, $\mathsf{D}_{\leq 1}$, is the main example for partial Markov categories. It is the semantic universe where we compute the solutions to the decision problems in Section 3 and Appendix A.

Definition B.10. The *finitary subdistribution monad* $\mathsf{D}_{\leq 1}\colon \mathsf{Set} \to \mathsf{Set}$ is defined on a sets by

$$\mathsf{D}_{\leq 1}(X) = \left\{ \sigma\colon X \to [0,1] \;\middle|\; \{x \in X \mid \sigma(x) > 0\} \text{ is finite, and } \sum_{x \in X} \sigma(x) \leq 1 \right\};$$

and on functions, $f\colon X \to Y$, by

$$\mathsf{D}_{\leq 1}(f)(y \mid \sigma) = \sum_{f(x) = y} \sigma(x),$$

for any subdistribution $\sigma \in \mathsf{D}_{\leq 1}(X)$ and any element $y \in Y$.

The monad multiplication $\mu_X\colon \mathsf{D}_{\leq 1}(\mathsf{D}_{\leq 1}(X)) \to \mathsf{D}_{\leq 1}(X)$ is defined by

$$\mu_X(x \mid p) = \sum_{\sigma \in \mathsf{D}_{\leq 1}(X)} p(\sigma) \cdot \sigma(x);$$

the monad unit $\eta_X \colon X \to \mathsf{D}_{\leq 1}(X)$ is defined by $\eta_X(x \mid x') = \mathsf{d}_{x'}(x)$, where $\mathsf{d}_x \in \mathsf{D}(X)$ is the Dirac distribution that assigns probability 1 to x and 0 to everything else.

Remark B.11. The fact that $\mathsf{D}_{\leq 1}$ is a functor and a monad can be seen by the fact that there is a distributive law between the *Maybe* monad $(- + 1)$ with the finitary *Distribution* monad D. Their composition is the finitary subdistribution monad $\mathsf{D}_{\leq 1} = \mathsf{D}(- + 1)$. The distributive law $\lambda \colon \mathsf{D}(-) + 1 \to \mathsf{D}(- + 1)$ is defined by $\lambda_X(\sigma) = \sigma^*$ and $\lambda_X(*) = \mathsf{d}_*$, where $\sigma \in \mathsf{D}(X)$ and σ^* is σ extended to $X + 1$ by $\sigma(*) = 0$.

We show that $\mathsf{Kl}(\mathsf{D}_{\leq 1})$ is a partial Markov category.

Proposition B.12. *The monoidal Kleisli category of the finitary subdistribution commutative monad,* $\mathsf{Kl}(\mathsf{D}_{\leq 1})$*, is a partial Markov category.*

Proof. Every partial function can be lifted to a subdistribution. In fact, there is an inclusion monoidal functor $\iota \colon \mathsf{Kl}(- + 1) \to \mathsf{Kl}(\mathsf{D}_{\leq 1})$. Every set can be endowed with a partial Frobenius monoid structure, $(X, \varepsilon_X, \delta_X, \mu_X)$, given by the partial functions $\varepsilon(x) = *$, $\delta_X(x) = (x, x)$ and

$$\mu(x, x') = \begin{cases} x \text{ if } x = x' \\ * \text{ otherwise.} \end{cases}$$

This structure can then be lifted to the category of subdistributions.

We now check that $\mathsf{Kl}(\mathsf{D}_{\leq 1})$ satisfies the existence of conditionals. Let $f \colon X \to Y_1 \otimes Y_2$ be any morphism. We will split it as $f = \delta_X \mathbin{⨾} (\mathsf{id} \otimes (f_2 \mathbin{⨾} \delta_{Y_2})) \mathbin{⨾} (g_1 \otimes \mathsf{id})$ for some g_1 and f_2. We pick $f_2 \colon X \to Y_2$ to be

$$f_2(y_2 \mid x) = \sum_{y \in Y_1} f(y, y_2 \mid x), \text{ and } f_2(* \mid x) = f(* \mid x),$$

and $g_1 \colon X \otimes Y_2 \to Y_1$ to be

$$g_1(y_1 \mid x, y_2) = \begin{cases} \dfrac{f(y_1, y_2 \mid x)}{\sum_{y \in Y_1} f(y, y_2 \mid x)} & \text{if } \sum_{y \in Y_1} f(y, y_2 \mid x) \neq 0, \\ c & \text{if } \sum_{y \in Y_1} f(y, y_2 \mid x) = 0. \end{cases}$$

This defines a valid conditional split; note that

$$f(y_1, y_2 \mid x) = \frac{f(y_1, y_2 \mid x)}{\sum\limits_{y \in Y_1} f(y, y_2 \mid x)} \sum_{y \in Y_1} f(y, y_2 \mid x) = f_2(y_2 \mid x) \cdot g_1(y_1 \mid x, y_2), \text{ and}$$

$$f(* \mid x) = f_2(* \mid x) + \sum_{y \in Y_2} f_2(y \mid x) \cdot g_1(* \mid x, y),$$

where we used that $\sum_{y \in Y_1} f(y, y_2 \mid x) = 0$ implies $f(y_1, y_2 \mid x) = 0$ for each $y_1 \in Y_1$. The second conditional split follows a symmetric reasoning. □

C Implementation

C.1 Newcomb's Problem

The following is the model for Newcomb's Problem. An agent will take action `a` with an uninformative prior. A predictor will try to predict it with `p`, again using an uninformative prior. We observe that the prediction is correct. Which is the action `x` that we would like to observe we have chosen?

```
newcomb :: Action → Distribution Value
newcomb x = do
    a ← act
    p ← predict
    observe (a == p)
    observe (a == x)
    return $ outcome a p

act :: Distribution Action
act = Distribution λcase
    OneBox → 1 / 2
    TwoBox → 1 / 2

predict :: Distribution Action
predict = Distribution λcase
    OneBox → 1 / 2
    TwoBox → 1 / 2

outcome :: Action → Action → Value
outcome OneBox OneBox = Hundred
outcome OneBox TwoBox = Zero
outcome TwoBox OneBox = HundredOne
outcome TwoBox TwoBox = One
```

Our program will evaluate `argmax newcomb` to the answer `OneBox`.

C.2 Death in Damascus

The following is the model for the "Death in Damascus" problem. We sample a `merchant` from the population of the world, and this information is also known by `death`, who uses it to decide which city to go to. The merchant throws a `coin` and chooses whether to flee or to stay following some strategy. Which is the strategy `f` that we would like to observe the merchant to have chosen?

```
deathInDamascus :: Strategy → Distribution Outcome
deathInDamascus f = do
    merchant ← world
    cityDeath ← death merchant
    coin ← throwCoin
    cityMerchant ← return (flee (strategy merchant) coin)
    observe (f == strategy merchant)
    return $ outcome cityMerchant cityDeath

world :: Distribution Merchant
world = Distribution λcase
    FleeingMerchant → 1 / 3
    StayingMerchant → 1 / 3
    RandomMerchant  → 1 / 3

strategy :: Merchant → Strategy
strategy FleeingMerchant = Fleeing
strategy StayingMerchant = Staying
strategy RandomMerchant  = Random
```

```
throwCoin :: Distribution Coin
throwCoin = Distribution λcase
    Heads → 1 / 2
    Tails → 1 / 2

flee :: Strategy → Coin → City
flee Fleeing _ = Aleppo
flee Staying _ = Damascus
flee Random Heads = Aleppo
flee Random Tails = Damascus

death :: Merchant → Distribution City
death FleeingMerchant = return Aleppo
death StayingMerchant = return Damascus
death RandomMerchant = Distribution λcase
    Damascus → 1 / 2
    Aleppo   → 1 / 2

outcome :: City → City → Outcome
outcome Aleppo Aleppo     = MerchantTravelsAndMeetsDeath
outcome Aleppo Damascus   = MerchantTravelsAndEscapes
outcome Damascus Aleppo   = MerchantEscapes
outcome Damascus Damascus = MerchantMeetsDeath
```

C.3 Partial Markov Category of Subdistributions

The following is the library for Evidential reasoning using the partial Markov category of subdistributions. The subdistribution monad is better modelled here as a relative monad [ACU15] from `Finitary` types to arbitrary types. We employ rebindable syntax in order to be able to use do-notation [HHJW07] for the Kleisli category of this relative monad.

```
-- Evidential Decision Theory via Partial Markov Categories.

-- An experimental implementation of a probabilistic programming
-- language with the primitives of a Partial Markov Category.

{-# LANGUAGE GADTs #-}
{-# LANGUAGE RebindableSyntax #-}
{-# LANGUAGE DataKinds #-}
{-# LANGUAGE DeriveAnyClass #-}
{-# LANGUAGE DeriveGeneric #-}
{-# LANGUAGE DerivingStrategies #-}
{-# LANGUAGE LambdaCase #-}
{-# LANGUAGE BlockArguments #-}


module Bayes where

import Prelude hiding ((>>=), (>>), return)
import Data.Finitary
import Data.Ord
```

```
import Data.List
import GHC.Generics (Generic)

-- Finitary subdistribution monad.
data Distribution a where
  Distribution :: (Finitary a) ⇒ (a → Rational) → Distribution a

-- Properties of a distribution.
total :: Distribution a → Rational
total (Distribution d) = sum [ d x | x ← inhabitants ]

weight :: Distribution a → a → Rational
weight (Distribution f) a = f a

-- Normalization here is just a pretty printing thing.
normalize :: Distribution a → Distribution a
normalize (Distribution f) = Distribution $ λa →
  f a / (total (Distribution f))
instance (Finitary a, Show a) ⇒ Show (Distribution a) where
  show d =
    let (Distribution f) = normalize d in
      unlines [ show (f a, a) | a ← inhabitants ]


-- Rebindable do notation.
(>>=) :: (Finitary a , Finitary b) ⇒
  Distribution a → (a → Distribution b) → Distribution b
(>>=) (Distribution d) f = Distribution $ λb →
  sum $ [ (d a) ∗ (weight (f a) b) | a ← inhabitants]

(>>) :: (Finitary a, Finitary b) ⇒
  Distribution a → Distribution b → Distribution b
(>>) d (Distribution f) = Distribution $ λb → (total d) ∗ (f b)

fail :: Distribution a → String
fail = undefined

ifThenElse :: Bool → a → a → a
ifThenElse True x _ = x
ifThenElse False _ y = y

-- Distribution combinators.
return :: (Finitary a) ⇒ a → Distribution a
return x = Distribution (λy →
  case (x == y) of
    True → 1
    False → 0)

absurd :: (Finitary a) ⇒ Distribution a
absurd = Distribution (λa → 0)

observe :: Bool → Distribution ()
observe True = return ()
```

```
observe False = absurd


class (Finitary a) ⇒ Valuable a where
  value :: a → Rational

expectedValue :: (Valuable a) ⇒ Distribution a → Rational
expectedValue u =
  let (Distribution d) = normalize u in
    sum $ fmap (λx → value x ∗ d x) inhabitants

argmax :: (Finitary a, Valuable b) ⇒ (a → Distribution b) → a
argmax f = maximumBy (comparing (expectedValue ◦ f)) inhabitants
```

D Conclusions

Related work. There exists a vast literature on categorical semantics for probabilistic programming languages (see, e.g. [SWY+16, Ste21]). However, while the internal language of Markov categories has been studied, the notion of partial Markov category and its diagrammatic syntax have remained unexplored. In the same way that the structure of partial Markov categories informs the toy probabilistic programming language that we implement in this manuscript, we expect that real, higher-order and continuous probabilistic programming languages can inform the precise categorical structure needed for the study of decision theory.

Further work. Evidential Decision Theory needs careful modelling to solve problems such as the “Smoking Lesion Problem” [YS17]. It is sometimes claimed that these problems are better solved by Causal Decision Theory [GH78], which makes use of “interventions” to apply an action to a node of a causal graph [Pea09, Jac19].

The present manuscript deals only with the case of discrete probability theory. It seems difficult to find well-behaved partial Markov categories accounting for the case of continuous probability theory. We conjecture that this can be solved, and that freely generated partial Markov categories might be enough: by applying Bayes’ theorem, one could rewrite a morphism in a partial Markov category to only use Bayesian inversions. We could then reuse well-known Markov categories with conditionals that do allow for continuous distributions [HKSY17, Fri20].