Commutative Codensity Monads

ZEV SHIRAZI

Department of Computer Science
University of Oxford

MSFP, July 2024

UNIVERSITY OF OXFORD

arXiv > math > arXiv:2405.12917

Search... | All fields | Search

Help | Advanced Search

Mathematics > Category Theory

[Submitted on 21 May 2024]

Commutative codensity monads and probability bimeasures

Zev Shirazi

Several well-studied probability monads have been expressed as codensity monads over small categories of stochastic maps, giving a limit description of spaces of probability measures. In this paper we show how properties of probability monads such as commutativity and affineness can arise from their codensity presentation. First we show that their codensity presentation is closely related to another characterisation of probability monads as terminal endofunctors admitting certain maps into the Giry monad, which allows us to generalise a result by Van Breugel on the Kantorovich monad. We then provide sufficient conditions for a codensity monad to lift to $\mathbf{MonCat}$, and give a characterisation of exactly pointwise monoidal codensity monads; codensity monads that satisfy a strengthening of these conditions. We show that the Radon monad is exactly pointwise monoidal, and hence give a description of the tensor product of free algebras of the Radon monad in terms of Day convolution. Finally we show that the Giry monad is not exactly pointwise monoidal due to the existence of probability bimeasures that do not extend to measures, although its restriction to standard Borel spaces is. We introduce the notion of a $*$-monad and its Kleisli monoidal op-multicategory to describe the categorical structure that organises the spaces of probability polymeasures on measurable spaces.

Subjects: **Category Theory (math.CT)**; Logic in Computer Science (cs.LO); Probability (math.PR)
Cite as: arXiv:2405.12917 **[math.CT]**
(or arXiv:2405.12917v1 **[math.CT]** for this version)
https://doi.org/10.48550/arXiv.2405.12917

Submission history

From: Zev Shirazi [view email]
[v1] Tue, 21 May 2024 16:39:26 UTC (57 KB)

Access Paper:

- View PDF
- HTML (experimental)
- TeX Source
- Other Formats

view license

Current browse context:
math.CT
< prev | next >
new | recent | 2024-05
Change to browse by:
cs
cs.LO
math
math.PR

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

Sections 3,5, 6.2

Overview & Motivation I

1. Commutative monads
   - Monads satisfying a *Fubini* theorem; can handle multivariable effectful functions and dataflow graph is *exchangeable*
   - Linked to current approaches to categorical probability theory and probabilistic programming and its semantics
2. Codensity monads
   - Presentation of a monad which gives us a *formal setting* for semantics, since $\mathit{Kl}(\mathbb{T}) \subseteq [\mathcal{D}, \mathbf{Set}]^{\mathrm{op}}$
   - [Van Belle, 2022] showed that several well-studied *probability monads* have such a presentation with $\mathcal{D} = \mathbf{FinStoch}$, all of which are commutative
   - Can we bake commutativity into the syntax of a probability monad using a codensity presentation?

Overview & Motivation II

3. Exactly pointwise monoidal codensity monads
    - A codensity monad guaranteed to be commutative
    - Kleisli category is monoidal
    - We characterise these as the monads such that this *formal setting* in $[\mathcal{D}, \mathbf{Set}]^{\mathrm{op}}$ still works in the commutative world (we get a formal description of tensor products).
4. $*$-monads (teaser)
    - Monad *of several variables* which is more general than a commutative monad
    - Not always guaranteed commutativity from codensity, but always guaranteed a $*$-monad

Contents

Commutative monads

```
do
  x ← a
  f x
```

$$a \mapsto Tf(a)$$
$$\mapsto \mu_Y Tf(a)$$

$$1 \xrightarrow{a} TX$$

$$X \xrightarrow{f} TY$$

$$f ; a = \mu_Y Tfa$$

Commutative monads: commutative effects

```
do
 x ← a
 y ← b
 f (x,y)
```

$$
\begin{aligned}
a : TX &\mapsto T(\lambda x.(x, b))(a) : T(X \times TY) \\
&\mapsto \mu_{X\times Y} T(\lambda x. T(\lambda y.(x, y))(b))(a) : T(X \times Y) \\
&\mapsto \mu_Z Tf(a \otimes_l b)
\end{aligned}
$$

$$1 \xrightarrow{a} TX \qquad 1 \xrightarrow{b} TY$$

$$f : X \times Y \to TZ$$

Commutative monads: commutative effects

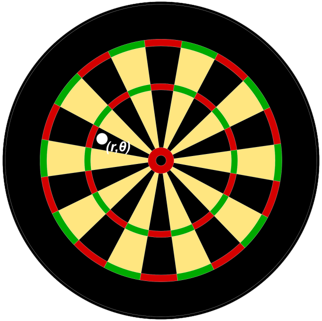
```
do
 y ← b
 x ← a
 f (x,y)
```

$$
\begin{aligned}
b : TY &\mapsto T(\lambda y.(a,y))(b) : T(TX \times Y) \\
&\mapsto \mu_{X\times Y} T(\lambda y.T(\lambda x.(x,y))(a))(b) : T(X \times Y) \\
&\mapsto \mu_Z Tf(a \otimes_r b)
\end{aligned}
$$

$$1 \xrightarrow{a} TX \qquad 1 \xrightarrow{b} TY$$

$$f : X \times Y \to TZ$$

Commutative effects: Example



if coin $=$ head

r $+ \theta$

else r $- \theta$

Commutative monads: Connection to probability

- Distribution monad $DX = \{p\colon X \to [0,1] \ : \sum_{x \in X} p(x) = 1\}$, whose Kleisli morphisms are random functions, is commutative.
- More generally the Giry monad of *continuous distributions* (actually measures) on **Meas** is commutative.
- Theory of commutative monads tied to current approaches to categorical probability theory.
- For semantics of probabilistic programming we look for commutative *probability* monads on cartesian closed categories.

Commutative monads: Categorically

- In functional programming commutativity is a **property**
- On general monoidal categories, we can define commutative monads, but we need **structure**:

Theorem (A. Kock, 1972)

For $\mathcal{C}$ a monoidal category, there is a 1-1 correspondence between commutative (strong) monads , and lax monoidal monads, i.e. monads $\mathbb{T}$ with natural maps

$$\kappa_{X,Y} \colon TX \otimes TY \to T(X \otimes Y)$$

which are coherent with μ and η in addition to the monoidal structure.

Contents

Pointwise codensity monads

Setting

$\mathcal{D}$ small
Objects $X, Y, ..$
Morphisms m, n

$\xrightarrow{F}$

$\mathcal{C}$ complete
Objects $A, B, ..$
Morphisms $f, g, ..$

Slogan

If F is any functor it has a codensity monad $\mathbb{T}$ on $\mathcal{C}$.

- Idea: F is a *model* of theory of algebraic effects, $\mathbb{T}$ is its largest extension

Pointwise codensity monads

$\mathcal{D}$ small		$\mathcal{C}$ complete
Objects $X, Y, ..$	$\xrightarrow{F}$	Objects $A, B, ..$
Morphisms m, n		Morphisms $f, g, ..$

- $TA \in \operatorname{ob} \mathcal{C}$
- $h \colon A \to FX \implies \mathrm{ev}_h \colon TA \to FX$
- $Fm\ \mathrm{ev}_h = \mathrm{ev}_{Fm\ h}$
- (TA, ev_h) is a limit over $(A \downarrow F)$

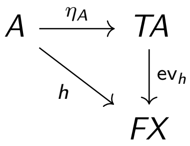
Pointwise codensity monads

Monad structure

Unit:

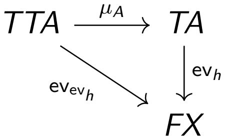
$$\eta_A \colon A \to TA$$

unique map satisfying

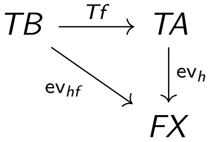


Monad structure

Multiplication:



On morphisms $f : B \to A$:



Pointwise codensity monads

- If you're familiar with ends:

$$TA = \int_{X \in \mathrm{ob}\,\mathcal{D}} FX^{\mathcal{C}(A,FX)}$$

- In Haskell

newtype Codensity f a = $\forall \beta$.a $\rightarrow$ f β $\rightarrow$ f β
instance Monad Codensity f where
    return a = λh.h a
    m $\gg\!=$ k = λh. m (λx. k x y)

- f is a type constructor, interpreted as a functor $|\mathcal{C}| \rightarrow \mathcal{C}$

Codensity monads: A type theory

$$A ::= \quad c \mid TA$$
$$\underline{X} ::= \quad \underline{d} \mid A \to \underline{X} \mid \underline{X} \implies \underline{Y}$$

$$\frac{\Gamma \vdash h : A \to \underline{X}}{\Gamma \vdash \mathrm{ev}\ h : TA \to \underline{X}}$$

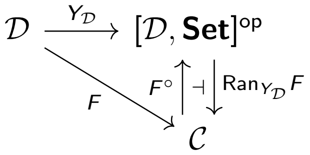
$$\frac{\Gamma \vdash t : C \quad \Gamma, x : A \to \underline{X} \vdash k : C \to \underline{X}}{\Gamma \vdash \nu x.k\ t : TA}$$

Codensity monads: Examples

- If $F\colon 1 \to \mathcal{C}$ is the inclusion of the subobject X

$$TX = X^{\mathcal{C}(A,X)}$$

  In **Set**, or in enriched settings, just the continuation on X.
- If F is a right adjoint to L, then $\mathbb{T}$ is the monad induced by the adjunction $L \dashv F$
- If $\mathbb{T}$ is the pointwise codensity monad of a functor F in our setting, then there's an adjunction inducing the monad, where $F^{\circ}A = \mathcal{C}(A, F-)$:



Contents

Commutative codensity monads

- In functional programming
  1. Commutativity is a property
  2. F is from a discrete category. Adding structure to F (e.g. in functor typeclass) can only add structure to $\mathbb{T}$, not properties.
- In category theory, lax monoidal monad is **structure**
- We can ask if F has structure, when does $\mathbb{T}$
- In particular if F is lax monoidal, i.e. has natural, coherent maps

$$\kappa_{X,Y} \colon FX \otimes FY \to F(X \otimes Y)$$

$$\iota \colon I \to FI$$

then is $\mathbb{T}$?

Exactly pointwise monoidal codensity monads

$$h\colon A \to FX \qquad\qquad k\colon B \to FY$$

$$\kappa_{X,Y}(h \otimes k)\colon A \otimes B \to F(X \otimes Y)$$

$$\mathrm{ev}_{h,k} = \mathrm{ev}_{\kappa_{X,Y}(h\otimes k)}\colon T(A \otimes B) \to F(X \otimes Y)$$

- $F(m \otimes n)\mathrm{ev}_{h,k} = \mathrm{ev}_{Fmh,Fnk}$
- $(T(A \otimes B), \mathrm{ev}_{h,k})$ is cone over $(A \downarrow F) \times (B \downarrow F)$

$\mathbb{T}$ is *exactly pointwise monoidal* in degree 2 if this is a limit cone

Exactly pointwise monoidal codensity monads

$$h_i \colon A_i \to FX_i \qquad i = 1, .., n$$

$$\bar{F}_{[X_1,..,X_n]} \bigotimes_{i=1}^{n} h_i \colon \bigotimes_{i=1}^{n} A_i \to F(\bigotimes_{i=1}^{n} X_i)$$

$$\mathrm{ev}_{[h_1,..,h_n]} \colon T(\bigotimes_{i=1}^{n} A_i) \to F(\bigotimes_{i=1}^{n} X_i)$$

- $F(\bigotimes_{i=1}^{n} m_i)\mathrm{ev}_{[h_1,..,h_n]} = \mathrm{ev}_{[Fm_1 h_1,..,Fm_n h_n]}$
- $(T(\bigotimes_{i=1}^{n} A_i), \mathrm{ev}_{[h_1,..,h_n]})$ is cone over $\prod_{i=1}^{n}(A_i \downarrow F)$

$\mathbb{T}$ is *exactly pointwise monoidal* in degree $k \geq 1$ if this is a limit cone.

Exactly pointwise monoidal codensity monads

$$\iota \colon I \to FI \implies \mathrm{ev}_\iota \colon TI \to FI$$

$\mathbb{T}$ is *exactly pointwise monoidal* in degree 0 if this is an isomorphism, and is *exactly pointwise monoidal* if its exactly pointwise in every degree.

Theorem (Koudenberg 2015, Weber 2016)

If $\mathbb{T}$ is exactly pointwise monoidal, it is commutative.

- In fact we can significantly weaken these conditions

Exactly pointwise monoidal codensity monads

Lax monoidal structure:

$$\chi_{A,B} \colon TA \otimes TB \to T(A \otimes B)$$

is unique map satisfying

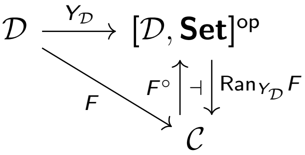
$$\begin{array}{ccc} TA \otimes TB & \xrightarrow{\chi_{A,B}} & T(A \otimes B) \\ {\scriptstyle \mathrm{ev}_h \otimes \mathrm{ev}_k} \downarrow & & \downarrow {\scriptstyle \mathrm{ev}_{h,k}} \\ FX \otimes FY & \xrightarrow[\kappa_{X,Y}]{} & F(X \otimes Y) \end{array}$$

These behave a lot like pointwise codensity monads in **Cat**:

1. $T(\bigotimes_{i=1}^{k} X_i)$ has limit structure allowing us to define all the structure of $\mathbb{T}$.
2. Monoidal Kleisli category can be seen to *live in* $[\mathcal{D}, \mathbf{Set}]^{\mathrm{op}}$
3. In a formal sense (future work)

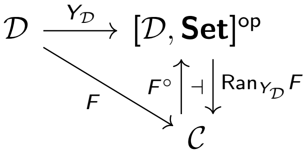
Commutativity: Characterising exactly pointwise

- When $\mathbb{T}$ is the pointwise codensity monad of F



- $\mathcal{C}_{\mathbb{T}}$ is equivalent to the full subcategory $\mathcal{A} \subseteq [\mathcal{D}, \mathbf{Set}]^{\mathrm{op}}$ of functors of the form $\mathcal{D}(A, F-)$, $A \in \mathrm{ob}\, \mathcal{C}$
- Can this picture be lifted to the commutative setting?

Characterising exactly pointwise



- $[\mathcal{D}, \mathbf{Set}]^{\mathrm{op}}$ has a monoidal structure given by Day convolution:

$$H \otimes_{\mathrm{Day}} G\,(-) = \int^{(X_1,X_2)\in \mathrm{ob}\,\mathcal{D}\times\mathcal{D}} \mathcal{D}(X_1 \otimes X_2, -) \times HX_1 \times GX_2$$

- $\bar{\mathcal{A}}$ is full monoidal closure of $\mathcal{A}$ (so generated by $\mathcal{D}(A, F-)$, for $A \in \mathrm{ob}\,\mathcal{C}$),

Characterising exactly pointwise

Theorem (S. 2024)

$\mathbb{T}$ is exactly pointwise monoidal iff $\mathcal{A}$ is a coreflective subcategory of $\bar{\mathcal{A}}$ such that

1. *The coreflection $i \dashv L$ is oplax monoidal*
2. *The monoidal structure defined on $\mathcal{A}$ by*

$$\mathcal{D}(A, F-) \otimes \mathcal{D}(B, F-) = L(\mathcal{D}(A, F-) \otimes_{Day} \mathcal{D}(B, F-)$$

makes $\mathcal{A}$ monoidally equivalent to $\mathcal{C}_{\mathbb{T}}$

$$\mathcal{C} \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \mathcal{A} \underset{\longleftarrow}{\overset{\hookrightarrow}{\perp}} \bar{\mathcal{A}} \subseteq [\mathcal{D}, \mathbf{Set}]^{op}$$

Contents

$*$-monads

Slogan

If F is any functor in our setting, it has a codensity monad $\mathbb{T}$

- This doesn't lift to the lax monoidal world if

$$T(\bigotimes_{i=1}^{n} A_i)$$

isn't a limit cone over $\prod_{i=1}^{n}(A_i \downarrow F)$

- But in our setting we still have limits

$$(T[A_1, .., A_n], \mathrm{ev}_{[h_1,..h_n]})$$

which have interesting categorical structure.

$*$-monads

Slogan

If F is any lax monoidal functor in our setting, it has a codensity $*$-monad

- $*$-monads act like monads of multiple variables/arbitrary arity
- Instead an assignment on objects

$$X \mapsto TX$$

  we get an assignment of lists of objects

$$[X_1, .., X_n] \mapsto T[X_1, .., X_n]$$

- We get *multiplication maps* and *unit maps* like in the case of monads.

$*$-monads

- $*$-monads have a version of a Kleisli category. Maps of type

$$X \to T[Y_1, .., Y_n]$$

$$Y_i \to T[Z_{i1}, .., Z_{ik_i}] \qquad i = 1, .., n$$

can be composed to a map of type

$$X \to T[Z_{11}, .., Z_{nk_n}]$$

- Furthermore this Kleisli *op-multicategory* is monoidal. Maps of type

$$X \to T[Z_1, .., Z_n] \qquad Y \to [W_1, .., W_m]$$

can be tensored to a map

$$X \otimes Y \to [Z_1, .., Z_n, W_1, .., W_m]$$

Contents

Future work

$*$-monads

- Much to be understood about $*$-monads and their generalisations from theoretical point of view
    1. $*$-monads can be generalised for symmetric monoidal categories, or any 2-monad on **Cat**. Similar structures give *enriched multicategories*.
    2. Connection to the theory of generalised multicategories by Hyland, Shulman & Crutwell
- How can $*$-monads help model effects in functional programming?
- Implementation in Haskell

Codensity type theory

- Work out a formal syntax for codensity
- Can codensity type theory be combined with type theories for Markov categories to provide a new kind of probabilistic programming language?

References I

Hermida, C. (2000).
Representable Multicategories.
Advances in Mathematics, 151(2):164–225.

Kock, A. (1972).
Strong functors and monoidal monads.
Archiv der Mathematik, 23(1):113–120.

Leinster, T. (2013).
Codensity and the ultrafilter monad.
Theory and Applications of Categories, 28(1):332–370.

References II

Van Belle, R. (2022).
Probability monads as codensity monads.
Theory and Applications of Categories, 38(21):811–842.

Voigtländer, J.
Asymptotic Improvement of Computations over Free Monads.
pages 388–403, Berlin, Heidelberg. Springer Berlin Heidelberg.

Weber, M. (2016).
Algebraic Kan extensions along morphisms of internal algebra classifiers.
Tbilisi Mathematical Journal, 9(1):65–142.