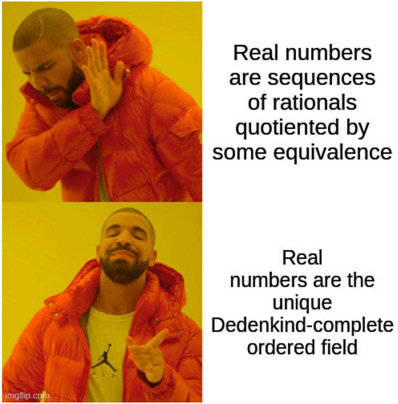
String diagrams for probability

Márk Széles

December 20, 2024

Radboud Universiteit

Mathematics: synthetic vs. analytic



Why categorical probability?

- An axiomatic approach leads to model-agnostic results

Why categorical probability?

- An axiomatic approach leads to model-agnostic results
- Avoid measure theory as much possible

Why categorical probability?

- An axiomatic approach leads to model-agnostic results
- Avoid measure theory as much possible
- Categorical probability comes with the convenient graphical calculus of string diagrams

Why categorical probability?

- An axiomatic approach leads to model-agnostic results
- Avoid measure theory as much possible
- Categorical probability comes with the convenient graphical calculus of string diagrams

I will demonstrate these aspects by an axiomatic study of equality comparison, disintegration, and normalisation of measures.

String diagrams for partial probabilistic computation: a motivating example

I roll a red and a blue die, and see that their sum is 10. What is the probability that the red die shows 5?

String diagrams for partial probabilistic computation: a motivating example

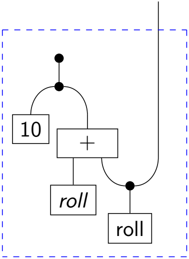
I roll a red and a blue die, and see that their sum is 10. What is the probability that the red die shows 5?

$$\begin{aligned}
&\mathit{red} \sim \mathit{roll}() \\
&\mathit{blue} \sim \mathit{roll}() \\
&\mathit{sum} \leftarrow \mathit{blue} + \mathit{red} \\
&\textbf{observe } \mathit{sum} = 10 \\
&\textbf{return } \mathit{red}
\end{aligned}$$

String diagrams for partial probabilistic computation: a motivating example

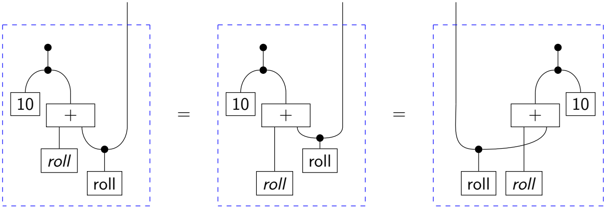
I roll a red and a blue die, and see that their sum is 10. What is the probability that the red die shows 5?

$$
\begin{aligned}
&\mathit{red} \sim \mathit{roll}() \\
&\mathit{blue} \sim \mathit{roll}() \\
&\mathit{sum} \leftarrow \mathit{blue} + \mathit{red} \\
&\textbf{observe } \mathit{sum} = 10 \\
&\textbf{return } \mathit{red}
\end{aligned}
$$



Why string diagrams?

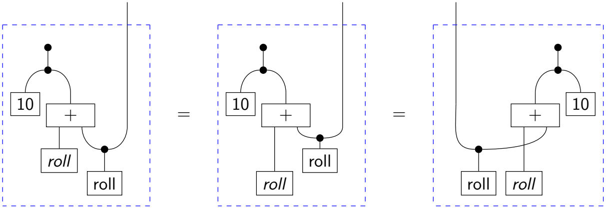
- Topological transformations that 'make sense' are sound.



This saves bookkeeping compared to a term language.

Why string diagrams?

- Topological transformations that 'make sense' are sound.

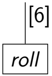


  This saves bookkeeping compared to a term language.
- Sometimes one needs terms with multiple outputs

The semantic universe: subprobabilistic computations

$$\mathit{roll}(0) : [1] \mathrel{\circ\!\!\!\rightarrow} [6]$$

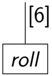
$$\mathit{roll}(0) = \frac{1}{6}\big|1\big\rangle + \frac{1}{6}\big|2\big\rangle + \frac{1}{6}\big|3\big\rangle + \frac{1}{6}\big|4\big\rangle + \frac{1}{6}\big|5\big\rangle + \frac{1}{6}\big|6\big\rangle$$



The semantic universe: subprobabilistic computations

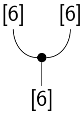
$$roll(0) : [1] \mathrel{\circ\!\!\!\rightarrow} [6]$$

$$roll(0) = \frac{1}{6}\Big|1\Big\rangle + \frac{1}{6}\Big|2\Big\rangle + \frac{1}{6}\Big|3\Big\rangle + \frac{1}{6}\Big|4\Big\rangle + \frac{1}{6}\Big|5\Big\rangle + \frac{1}{6}\Big|6\Big\rangle$$



$$\mathrm{copy} : [6] \mathrel{\circ\!\!\!\rightarrow} [6] \times [6]$$

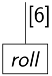
$$\mathrm{copy}(i) = 1|i, i\rangle$$



The semantic universe: subprobabilistic computations

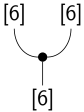
$$\mathit{roll}(0) : [1] \circ\!\!\!\rightarrow [6]$$

$$\mathit{roll}(0) = \frac{1}{6}\Big|1\Big\rangle + \frac{1}{6}\Big|2\Big\rangle + \frac{1}{6}\Big|3\Big\rangle + \frac{1}{6}\Big|4\Big\rangle + \frac{1}{6}\Big|5\Big\rangle + \frac{1}{6}\Big|6\Big\rangle$$



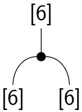
$$\text{copy} : [6] \circ\!\!\!\rightarrow [6] \times [6]$$

$$\text{copy}(i) = 1|i, i\rangle$$



$$\text{compare} : [6] \times [6] \circ\!\!\!\rightarrow [6]$$

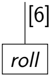
$$\text{compare}(i, j) = \begin{cases} 1|i\rangle & \text{if } i = j \\ \mathbf{0} & \text{otherwise} \end{cases}$$



The semantic universe: subprobabilistic computations

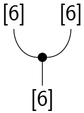
$$\mathit{roll}(0) : [1] \circ\!\!\!\rightarrow [6]$$

$$\mathit{roll}(0) = \frac{1}{6}\Big|1\Big\rangle + \frac{1}{6}\Big|2\Big\rangle + \frac{1}{6}\Big|3\Big\rangle + \frac{1}{6}\Big|4\Big\rangle + \frac{1}{6}\Big|5\Big\rangle + \frac{1}{6}\Big|6\Big\rangle$$



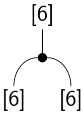
$$\text{copy} : [6] \circ\!\!\!\rightarrow [6] \times [6]$$

$$\text{copy}(i) = 1|i, i\rangle$$



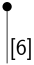
$$\text{compare} : [6] \times [6] \circ\!\!\!\rightarrow [6]$$

$$\text{compare}(i, j) = \begin{cases} 1|i\rangle & \text{if } i = j \\ \mathbf{0} & \text{otherwise} \end{cases}$$



$$\text{discard} : [6] \circ\!\!\!\rightarrow [1]$$

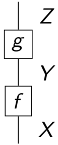
$$\text{discard}(i) = 1|0\rangle$$



Composing subprobability channels

- If $f : X \rightsquigarrow Y$, $g : Y \rightsquigarrow Z$, then

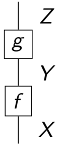
$$g \circ f : X \rightsquigarrow Z$$
$$(g \circ f)(z|x) = \sum_{y \in Y} f(y|x) \cdot g(z|y)$$



Composing subprobability channels

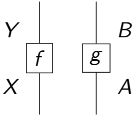
- If $f : X \mathrel{\circ\!\!\rightarrow} Y$, $g : Y \mathrel{\circ\!\!\rightarrow} Z$, then

$$g \circ f : X \mathrel{\circ\!\!\rightarrow} Z$$
$$(g \circ f)(z|x) = \sum_{y \in Y} f(y|x) \cdot g(z|y)$$



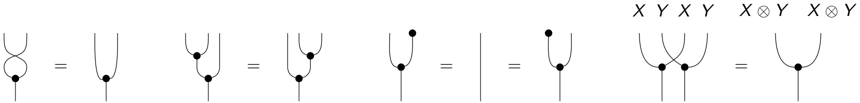
- If $f : X \mathrel{\circ\!\!\rightarrow} Y$, $g : A \mathrel{\circ\!\!\rightarrow} B$, then

$$(f \otimes g) : X \times A \mathrel{\circ\!\!\rightarrow} Y \times B$$
$$(f \otimes g)(y, b|x, a) = f(y|x) \cdot g(b|a)$$



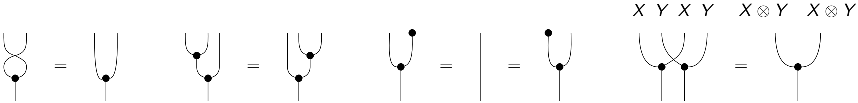
The axioms of copy and compare (CDC-categories)

- Copy and discard:

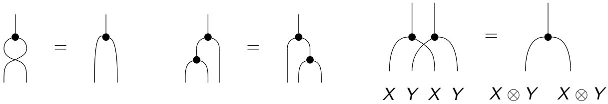


The axioms of copy and compare (CDC-categories)

- Copy and discard:

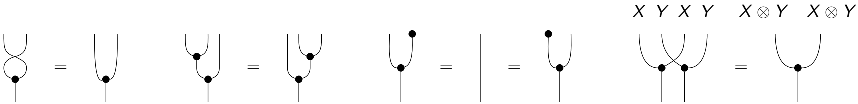


- Compare:

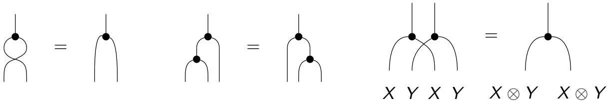


The axioms of copy and compare (CDC-categories)

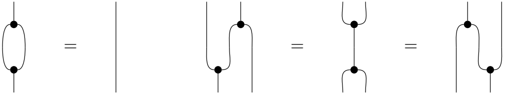
- Copy and discard:



- Compare:



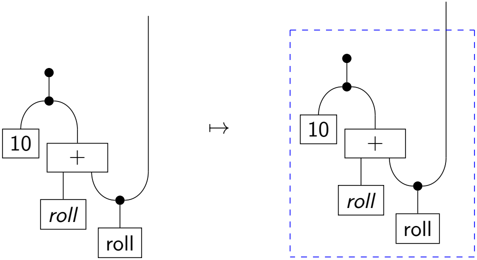
- Copy-compare interaction:



Examples of CDC-categories

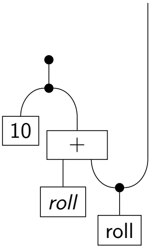
- Finite sets and subprobability channels: **FinSubStoch**
- Finite dimensional vector spaces and linear maps: **FinVect**
- Sets and relations: **Rel**
- Standard Borel spaces and subprobability kernels: $\mathbf{BorelStoch}_{\leq}$
- ...

Normalisation

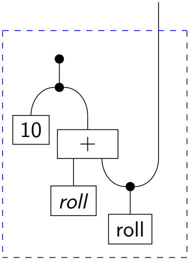


$$\tfrac{1}{36}\left|4\right\rangle + \tfrac{1}{36}\left|5\right\rangle + \tfrac{1}{36}\left|6\right\rangle \quad \mapsto \quad \tfrac{1}{3}\left|4\right\rangle + \tfrac{1}{3}\left|5\right\rangle + \tfrac{1}{3}\left|6\right\rangle$$

Normalisation



$\mapsto$



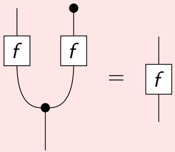
$$\tfrac{1}{36}\big|4\big\rangle + \tfrac{1}{36}\big|5\big\rangle + \tfrac{1}{36}\big|6\big\rangle \qquad \mapsto \qquad \tfrac{1}{3}\big|4\big\rangle + \tfrac{1}{3}\big|5\big\rangle + \tfrac{1}{3}\big|6\big\rangle$$

But how do we normalise the zero subdistribution?

Normalised maps

Definition

We call a map $f : X \multimap Y$ *normalised* if



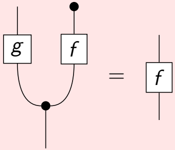
This translates to

$$\forall x \in X. \|f(x)\| \in \{0, 1\}$$

The ‘normalised by’ relation

Definition

A map $g : X \rightsquigarrow Y$ *normalises* $f : X \rightsquigarrow Y$ if



In this case, we write $f \preceq g$.

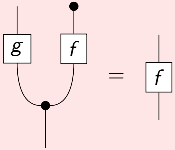
This translates to the following condition for all $x \in X$.

$$f(x) \neq \mathbf{0} \Longrightarrow \forall y \in Y . g(y|x) = \frac{f(y|x)}{\|f(x)\|}$$

The ‘normalised by’ relation

Definition

A map $g : X \mathrel{\circ\!\!\rightarrow} Y$ *normalises* $f : X \mathrel{\circ\!\!\rightarrow} Y$ if



In this case, we write $f \preceq g$.

This translates to the following condition for all $x \in X$.

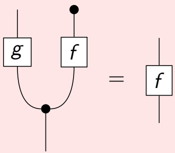
$$f(x) \neq \mathbf{0} \Longrightarrow \forall y \in Y. g(y|x) = \frac{f(y|x)}{\|f(x)\|}$$

- The relation $\preceq$ is a partial order on normalised maps

The 'normalised by' relation

Definition

A map $g : X \multimap Y$ *normalises* $f : X \multimap Y$ if



In this case, we write $f \preceq g$.

This translates to the following condition for all $x \in X$.

$$f(x) \neq \mathbf{0} \Longrightarrow \forall y \in Y . g(y|x) = \frac{f(y|x)}{\|f(x)\|}$$

- The relation $\preceq$ is a partial order on normalised maps
- The dashed box should select the least normalisation

The axioms of the dashed box

Definition

A *normalisation structure* assigns to every $f : X \multimap Y$ a normalised $\mathrm{nrm}(f) : X \multimap Y$ such that

1. $f \preceq \mathrm{nrm}(f)$

The axioms of the dashed box

Definition

A *normalisation structure* assigns to every $f : X \multimap Y$ a normalised $\mathrm{nrm}(f) : X \multimap Y$ such that

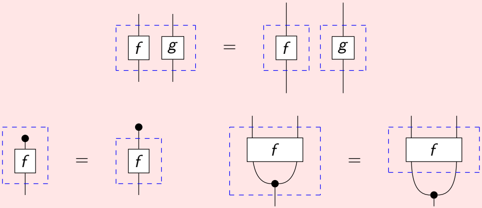
1. $f \preceq \mathrm{nrm}(f)$
2. If $f \preceq f$, then $\mathrm{nrm}(f) = f$

The axioms of the dashed box

Definition

A *normalisation structure* assigns to every $f : X \rightsquigarrow Y$ a normalised $\mathrm{nrm}(f) : X \rightsquigarrow Y$ such that

1. $f \preceq \mathrm{nrm}(f)$
2. If $f \preceq f$, then $\mathrm{nrm}(f) = f$
3. The following hold



The axioms of the dashed box

Proposition

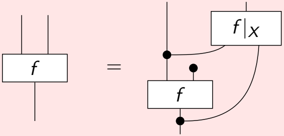
The dashed box assigns a least normalisation to each morphism.
Therefore, a CD-category can admit at most one normalisation structure.

Disintegration

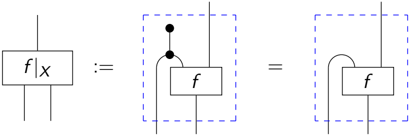
'How to compute $P(Y|X, Z)$ from $P(X, Y|Z)$?'.

Definition

If $f : Z \mathrel{\circ\!\!\!\to} X \times Y$, then a *disintegration* of f is a map $f_X : X \times Z \mathrel{\circ\!\!\!\to} Y$ that satisifes

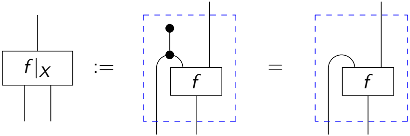


Can we use comparator and normalisation to compute a disintegration?



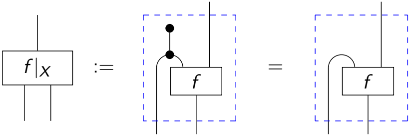
Deriving disintegration

Can we use comparator and normalisation to compute a disintegration?

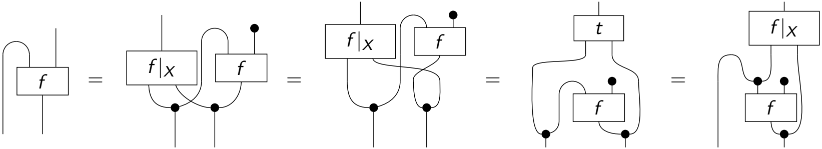


Deriving disintegration

Can we use comparator and normalisation to compute a disintegration?

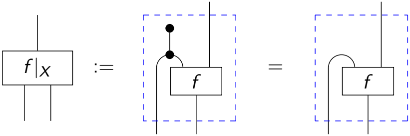


The answer is ‘sometimes’:

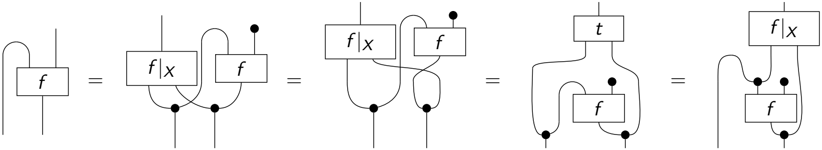


Deriving disintegration

Can we use comparator and normalisation to compute a disintegration?



The answer is ‘sometimes’:



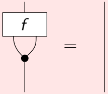
Remark

This does not recover disintegration in $\mathbf{BorelStoch}_{\leq}$

Comparators from disintegration

Proposition

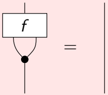
A map $f : X \times X \circ\!\!\rightarrow X$ is a disintegration of the copier iff



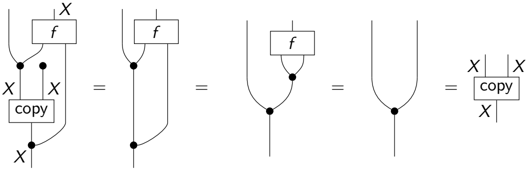
Comparators from disintegration

Proposition

A map $f : X \times X \rightsquigarrow X$ is a disintegration of the copier iff



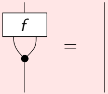
Proof:



Comparators from disintegration

Proposition

A map $f : X \times X \rightsquigarrow X$ is a disintegration of the copier iff



Proposition (by me)

A CD-category has comparators if and only if the copiers have minimal disintegrations.

Question to the room

Question

How to integrate string diagrammatic reasoning with proof assistants?

Question to the room

Question

How to integrate string diagrammatic reasoning with proof assistants?

- There is a good attempt: ‘chyp’ by Aleks Kissinger. (https://github.com/akissinger/chyp)

Question to the room

Question

How to integrate string diagrammatic reasoning with proof assistants?

- There is a good attempt: 'chyp' by Aleks Kissinger. (https://github.com/akissinger/chyp)
- The rewrite theory exists (see references)

Question to the room

Question

How to integrate string diagrammatic reasoning with proof assistants?

- There is a good attempt: ‘chyp’ by Aleks Kissinger. (https://github.com/akissinger/chyp)
- The rewrite theory exists (see references)
- It is not easy to integrate with existing proof assistant infrastructure (e.g. Coq)

References

- K. Cho and B. Jacobs, “Disintegration and Bayesian inversion via string diagrams,” Mathematical Structures in Computer Science, vol. 29, no. 7, pp. 938–971, 2019. doi:10.1017/S0960129518000488
- E. Di Lavore and M. Román, “Evidential Decision Theory via Partial Markov Categories,” 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), pp. 1-14, 2023. doi: 10.1109/LICS56636.2023.10175776
- R. Lorenz and S. Tull, “Causal models in string diagrams,” arXiv preprint, 2023. doi:10.48550/arXiv.2304.07638
- F. Bonchi, F. Gadducci, A. Kissinger, P. Sobocinski, and F. Zanasi, “String Diagram Rewrite Theory I: Rewriting with Frobenius Structure,” Journal of the ACM (JACM), 69.2, pp. 1 - 58, 2020. doi: 10.1145/3502719