Classical Linear Logic in Perfect Banach Lattices

- ² Pedro H. Azevedo de Amorim ⊠©
- ³ Oxford University, UK
- ₄ Leon Witzman ⊠ <a>D
- 5 Nanyang Technological University, Singapore
- 6 Dexter Kozen ⊠©
- 7 Cornell University, USA

⁸ — Abstract

In recent years, researchers have proposed various models of linear logic with strong connections 9 to measure theory, with probabilistic coherence spaces (PCoh) being one of the most prominent. 10 One of the main limitations of the **PCoh** model is that it cannot interpret continuous measures. 11 To overcome this obstacle, Ehrhard has extended **PCoh** to a category of positive cones and linear 12 13 Scott-continuous functions and shown that it is a model of intuitionistic linear logic. In this work we show that the category $PBanLat_1$ of perfect Banach lattices and positive linear functions of 14 norm at most 1 can serve the same purpose, with some added benefits. We show that $PBanLat_1$ is 15 16 a model of classical linear logic (without exponential) and that **PCoh** embeds fully and faithfully in $PBanLat_1$ while preserving the monoidal structure. Finally, we show how $PBanLat_1$ can be used 17 to give semantics to a higher-order probabilistic programming language. 18

¹⁹ 2012 ACM Subject Classification Theory of computation \rightarrow Categorical semantics

20 Keywords and phrases Probabilistic Semantics, Linear Logic, Categorical Semantics

21 Digital Object Identifier 10.4230/LIPIcs...

²² **1** Introduction

Recent work has shown that linear logic has deep connections to the semantics of 23 probabilistic programming languages [7, 13, 10, 12, 11, 34, 9, 22, 6]. By using monoidal 24 closed categories instead of cartesian closed categories, linear logic provides an alternative 25 categorical framework for higher-order functions. This was foreshadowed in early work on 26 probabilistic semantics [23] in which bounded linear operators on Banach lattices were used 27 to interpret a first-order imperative probabilistic programming language. This can be seen as 28 evidence that a linear approach might be a natural alternative to cartesian closed categories. 29 Since then, many probabilistically-flavored models of linear logic have appeared. For 30 instance, the connection between the early work of [23] and linear logic has been recently 31 made precise [6], where the category of regular ordered Banach spaces and regular maps 32 (**RoBan**) was used to extend the semantics of [23] with higher-order functions. It was shown 33 in [6] that **RoBan** is a model of intuitionistic linear logic. 34

An appealing aspect of the **RoBan** model is that ordered Banach spaces are mathematically well-understood objects with a well-developed classical theory, thus providing a plethora of useful theorems to reason about programs. This is illustrated in [6] by using results from ergodic theory to prove the correctness of a Gibbs sampling algorithm implemented in a higher-order language. However, the programming model supported by the semantics is somewhat brittle, in that the soundness of the system depends on a tricky interaction between three different type grammars with several syntactic restrictions.

A different approach was taken in [7], in which a category PCoh was defined and shown
to be a model of classical linear logic. The model was used to interpret a version of PCF
extended with discrete probabilities [13]. Although this category handles discrete probabilities
very nicely, it cannot interpret continuous distributions such as the normal distribution over



© Anonymized;

licensed under Creative Commons License CC-BY 4.0 Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

XX:2 Classical Linear Logic in Perfect Banach Lattices

⁴⁶ \mathbb{R} , a severe limitation for real-world applications. To remedy this, a category of positive ⁴⁷ cones with measurability paths and linear Scott-continuous functions **CLin**_m has recently ⁴⁸ been introduced and shown to be a conservative extension of the intuitionistic fragment of ⁴⁹ **PCoh** [11].

From a programming point of view, the language of [11] is an extension of the simply typed λ -calculus with recursion, making it a simple and expressive programming model. However, the definition of positive cone with measurability paths deviates from standard objects from the probability literature and thus would require a large amount of mathematical effort to rephrase useful theorems that could be used to reason about programs.

Although these previous approaches are valuable contributions to our understanding of higher-order probabilistic programming through linear logic, missing up to now is a comprehensive model that embodies the following desirable aspects:

- ⁵⁸ extends **PCoh** to admit continuous measures;
- is a model of classical (not just intuitionistic) linear logic, thus allowing it to handle other
 computational interpretations of linear logic such as session types;
- ⁶¹ has a simple and expressive programming model that can handle higher-order computation;
- is based on well-understood classical structures from measure theory and functional
 analysis.

In this paper we propose such a model. Our model extends **PCoh** with continuous probabilities and satisfies all of the properties above. Our model is based on complete normed vector lattices, called *Banach lattices*. To accommodate the second point, we work with spaces with an involutive linear negation, the so-called *perfect spaces*.

⁶⁸ Compared to previous models, our model has simpler tensor product, which we believe ⁶⁹ lead to a more perspicuous and theoretically satisfying generalization of **PCoh**. For example, ⁷⁰ we invite a comparison with \mathbf{CLin}_m , where the construction rely on categorical machinery ⁷¹ which, though elegant, are indirect.

Most importantly, Banach lattices can be seen as an abstraction of ordinary measure spaces and are well-studied in functional analysis, with many results from measure theory holding for certain classes of Banach lattices. There is a vast literature on the subject; see [14] for a thorough introduction.

In order to justify the viability of our model, we show that it can be used to interpret a recently introduced higher-order probabilistic calculus [3], and we extend the core calculus of that paper with recursion, the nonlinear modality ! and a new typing rule based on enriched category theory.

⁸⁰ Summary of contributions

- In §3, we define the category PBanLat₁ of perfect Banach lattices and order-continuous
 positive linear operators with norm at most 1 and show that it is a model of classical
 linear logic.
- ⁸⁴ In §4, we show that there is a full and faithful monoidal closed functor $\mathbf{PCoh} \rightarrow \mathbf{PBanLat_1}$. This is a more adequate extension than the model \mathbf{CLin}_m proposed in [11], ⁸⁶ since it also accommodates the classical aspects of the linear structure of \mathbf{PCoh} .
- In §5, we show that $PBanLat_1$ is isomorphic to a category of lattices of positive complete cones.
- ⁸⁹ In §6, we show that **PBanLat**₁ is a model to the recently defined calculus in [3].

Our work contributes both to the study of quantitative models of linear logic as well as to a deeper understanding of higher-order probability theory, shedding light on the importance of linear logic as a vehicle to interpret higher-order programs without cartesian closure.

93 2 Riesz spaces

Our model depends on technical definitions and constructions from the vector lattice literature. This section contains a brief self-contained presentation of the subject. We point the interested reader to the introductory texts [1, 36] for good presentations of much of the material presented in this section.

Although we are primarily interested in Banach lattices—normed vector lattices with a
 completeness property—we start by defining the objects in the general unnormed case.

Definition 1. Let $\mathbb{R}_+ = \{a \in \mathbb{R} \mid a \ge 0\}$. A Riesz space is a partially-ordered vector space (V, \le) over \mathbb{R} such that

- $I02 \quad If x \leq y, then x + w \leq y + w;$
- 103 If $x \leq y$, then $\alpha x \leq \alpha y$ for $\alpha \in \mathbb{R}_+$; and
- 104 it is an upper semilattice with respect to \leq with join operation \lor .

105 It follows that the space is also a lattice with meet operation $x \wedge y = -(-x \vee -y)$.

Many standard vector spaces are Riesz spaces.

- **► Example 2.** The following are Riesz spaces:
- 108 $\square \mathbb{R}^n$ with the pointwise ordering;
- ¹⁰⁹ the set of bounded sequences of real numbers with pointwise ordering;
- the set of signed measures on a measurable space;
- ¹¹¹ the set of bounded measurable functions on a measurable space.

Unlike the real numbers, there are elements that are neither negative nor positive, but a notable characteristic of Riesz spaces is that every element decomposes uniquely into its positive and negative parts.

▶ Definition 3. For v an element of a Riesz space, define $v^+ = v \lor 0$, $v^- = (-v) \lor 0$ and $|v| = v \lor -v = v^+ + v^-$.

Then v^+ and v^- are the unique positive elements such that $v = v^+ - v^-$ and $v^+ \wedge v^- = 0$. Thus Riesz spaces are completely characterized by their positive elements. This often simplifies constructions, as one can often prove a property for the positive elements, then extend to the entire space using this decomposition.

Given a Riesz space V, let V^+ denote the set of positive elements of V. Using the decomposition property mentioned above, it follows that $V = V^+ - V^+$, where – applied to sets denotes elementwise subtraction.

124 2.1 Order convergence

Every topology gives rise to a notion of convergence. For normed spaces, one usually studies convergence in the norm topology. However, ordered spaces also carry an *order topology*.

XX:4 Classical Linear Logic in Perfect Banach Lattices

▶ Definition 4. Let D be a directed set and V a Riesz space. A net $\{v_{\alpha}\}_{\alpha\in D}$ is a function 129 $D \to V$. We say that the net is increasing (respectively, decreasing) and write $\{v_{\alpha}\}\uparrow$ 130 (respectively, $\{v_{\alpha}\}\downarrow$) if $\alpha \leq_D \beta$ implies $v_{\alpha} \leq_V v_{\beta}$ (respectively, $v_{\alpha} \geq_V v_{\beta}$).

Definition 5. Given a decreasing net $\{x_{\alpha}\}$, we write $\{x_{\alpha}\} \downarrow 0$ if $\inf\{x_{\alpha}\} = 0$.

▶ **Definition 6** (Order convergence). We say that a net $\{x_{\alpha}\}$ converges in order to x and write $x_{\alpha} \rightarrow x$ if there is a decreasing net $\{y_{\alpha}\} \downarrow 0$ such that for all α , $|x_{\alpha} - x| \leq y_{\alpha}$.

In general, this notion of convergence is neither weaker nor stronger than convergence in norm. However, when a net converges in both order and norm, it converges to the same value in both.

137 2.2 Riesz subspaces, solids, ideals and bands

¹³⁸ In the theory of Riesz spaces, there are classes of subspaces that have many interesting ¹³⁹ properties that will be used in our constructions.

- **Definition 7.** A subset S of a Riesz space is
- solid if $x \in S$ and $|y| \leq |x|$ implies $y \in S$,
- an ideal if it is a solid linear subspace,
- ¹⁴³ a band if it is an ideal and closed under existing suprema.

▶ Definition 8. We say that a Riesz space V is Archimedean if for every $v \in V^+$, $\{v/n\}_{n \in \mathbb{N}} \downarrow$ 145 0. Furthermore, if every bounded subset of V admits a supremum, then we say that V is 146 Dedekind complete.

▶ **Proposition 9.** Every band in a Dedekind complete Riesz space is Dedekind complete.

▶ **Definition 10.** A Riesz subspace $A \subseteq V$ is said to be order dense if for every element 149 $0 < v \in V$ there is an element $a \in A$ such that $0 < a \leq v$.

▶ Theorem 11 ([1], Th. 1.34). A Riesz subspace A is order dense in an Archimedean Riesz space V iff for every $v \in V^+$,

$$\{a \in A \mid 0 \le a \le v\} \uparrow v.$$

150 2.3 Order-continuous functions

As usual when studying vector spaces with extra structure, we care only about linear maps that interact nicely with the extra structure. In our case, the linear functions will have to respect the partial order.

We call a linear function $f: V \to W$ positive if it maps positive elements of V to positive elements of W; that is, it restricts to a function $V^+ \to W^+$. A linear function is *regular* if it can be written as the difference of two positive functions.

Definition 12. A linear function $T: V \to W$ is order-continuous if it is continuous in the order topology. Equivalently, T is order-continuous if $Tv_{\alpha} \to Tv$ whenever $\{v_{\alpha}\}$ is an increasing net with supremum v.

We can also characterize the positive order-continuous functions as those that preserve existing suprema and infima.

Order continuity interacts well with order density. Indeed, it is possible to show using Theorem 11 the following lemma

▶ Lemma 13. If V is an Archimedian Riesz space and $f, g : V \to W$ are two linear order-continuous functions that agree on an order-dense subset of V, then f = g.

This lemma will come in handy when constructing our model. Furthermore, the space of order-continuous linear functions on certain Riesz spaces are well-behaved subsets of the regular linear functions.

▶ Theorem 14 ([1], Th 1.57). If W is Dedekind complete, then the set of order-continuous linear functions $V \rightarrow W$ is a band in the space of regular functions, thus forms a Dedekindcomplete Riesz space.

Proof. The Riesz space structure is given by Th 1.18 in [1].

◀

▶ Definition 15. A Riesz space is separated if for every distinct pair $v_1, v_2 \in V$, there exists an order-continuous linear functional $f: V \to \mathbb{R}$ such that $f(v_1) \neq f(v_2)$.

175 2.4 Normed Riesz spaces

Now we will introduce normed Riesz spaces. In the context of probabilistic semantics, the norm plays an important role, as it can be used to distinguish between arbitrary measures and (sub)-probability distributions, the measures with norm at most 1.

Definition 16. Let V be a real vector space. A norm is a function $\|\cdot\|: V \to \mathbb{R}^+$ such that:

||v|| = 0 iff v = 0

- $\|\alpha v\| = |\alpha| \|v\|$
- $|v + u|| \le ||v|| + ||u||.$

¹⁸⁴ For Riesz spaces, we require the norm to satisfy the additional property

185 $|v| \le |u|$ implies $||v|| \le ||u||$.

If the Riesz space is also complete with respect to the norm, we call it a *Banach lattice*. In vector space models of linear logic, the norm is typically used to distinguish between the product & and the coproduct \oplus , as they both have the same underlying set, but distinct norms. However, in the context of program semantics, the norm also has the extra role of allowing the interpretation of recursive programs.

► **Example 17.** The set $\mathcal{M}(\mathbb{R})$ of signed measures over the Borel σ -algebra on \mathbb{R} is a Riesz space (cf. §2.6). We can equip it with the *total variation* norm

$$\|\mu\| = \mu^+(\mathbb{R}) + \mu^-(\mathbb{R}).$$

Theorem 14 shows that by assuming the right amount of structure on the Riesz space, the set of order-continuous linear functions between Riesz spaces also has a lattice structure. It is not immediately clear whether this result generalizes to the normed case. Luckily, Dedekind completeness is once again enough.

Example 18. Let V and W be normed Riesz spaces with W Dedekind complete. The set of order-continuous linear functions $V \to W$ can be equipped with the *regular norm*

$$||T||_r = \sup_{||x||_V \le 1} ||T|(x)||_W$$

where |T| is given by Theorem 14 and Definition 3.

Definition 19. Let V be a normed Riesz space. The closed unit ball of V is the set $\mathcal{B}(V) = \{v \in V \mid ||v|| \le 1\}.$

XX:6 Classical Linear Logic in Perfect Banach Lattices

198 2.4.1 Banach lattices

Banach lattices are normed Riesz spaces that are also Banach spaces. In the usual categorical study of Banach spaces, the relevant morphisms are the norm-continuous linear functions.

▶ Definition 20. A linear function f between normed Riesz spaces V and W is said to be norm-continuous (or norm-bounded) if $\sup_{v \in \mathcal{B}(V)} ||f(v)||$ is finite.

Since we are interested in spaces with two distinct structures, a partial order and a norm, it is not immediately clear which class of morphisms one should care about. In general, the space of all norm-continuous linear functions between Banach lattices is not a Banach lattice, making them unable to give semantics to linear implication.

Normed Riesz spaces are also problematic, as not every order-continuous function is normcontinuous, making it unclear how one would equip the space of order-continuous functions with a norm. However, if the codomain is a Banach lattice, then every order-continuous linear function is also norm-continuous [1]. This suggests that one should work with Banach lattices but only use order-continuous linear functions.

Definition 21. The category $BanLat_1$ has separated Banach lattices as objects and order-continuous positive linear functions of norm at most one as morphisms.

A subtlety when working with a norm and a partial order is that there are two distinct notions of convergence in play that on the surface appear only tenuously related. However, a useful property has been identified in the literature that brings some harmony between the two.

▶ Definition 22. A normed Riesz space is said to satisfy the (sequential) weak Fatou property if every norm-bounded monotone (sequence) net has a supremum.

In the context of program semantics, the sequential version of this property has been used before to interpret recursive programs [7, 12].

▶ Lemma 23. Let $f : V \to V$ be a positive order-continuous function (not necessarily linear) such that $f(\mathcal{B}(V)) \subseteq \mathcal{B}(V)$. If V satisfies the weak Fatou property, then f admits a fixpoint.

Proof. It can be directly shown that the limit of the ω -chain $\{f^n(0)\}_{n \in \mathbb{N}}$ is a fixpoint of f. Note that when f is linear, the theorem is trivially true, since f(0) = 0.

▶ Lemma 24 ([14] Lem 354B(d)). Every band in a Banach lattice is a Banach lattice.

Theorem 25. If V and W are Banach lattices, then the set of order-continuous linear functions between V and W is a Banach lattice.

Proof. The proof is a direct consequence of Banach lattices being Dedekind complete ([14]
Prop 354E(e)) and the space of order-continuous being a band in the space of regular linear
functions.

233 2.5 Dualities

The category **BanLat**₁ seems to be a good candidate in which to interpret intuitionistic linear logic. However, since the linear negation connective $(-)^{\perp}$ is usually interpreted as the linear dual $V \rightarrow \mathbb{R}$ in models of linear logic based on vector spaces over \mathbb{R} , **BanLat**₁ would

not be able to model *classical* linear logic, since there are examples of Banach lattices that are not isomorphic to their bidual, e.g. finitely supported real sequences.

A recurring challenge in models of linear logic is to make an involutive linear negation typical of finite-dimensional spaces—coexist with !V, which requires infinite-dimensional spaces. Since we are interested in defining a model of classical linear logic, we should only work with Riesz spaces that are isomorphic to their bidual.

▶ Definition 26. Let V^{σ} denote the space of order-continuous functionals $V \multimap \mathbb{R}$. A Riesz space V is said to be perfect if the map $\sigma_V = \lambda x f \cdot f(x) : V \multimap V^{\sigma\sigma}$ is an isomorphism.

We will write σ for σ_V when V is clear from context.

Definition 27. The category $\mathbf{PBanLat}_1$ has perfect Banach lattices as objects and positive order-continuous linear functions of norm at most one as morphisms.

Although the definition of perfect spaces is simple, it is difficult to manipulate in practice. The following theorems provide some alternative characterisations, both in the normed and unnormed cases:

Theorem 28 ([25], v. XIII, Th. 41.4). Let V be a separated normed Riesz space. Then V is perfect and Banach iff V has the weak Fatou property.

- **Theorem 29** ([1], Th 1.71). A Riesz space V is perfect iff
- = it is separated;

whenever $0 \le x_{\alpha} \uparrow$ and $\sup_{\alpha} \{f(x_{\alpha})\} < \infty$ for all positive $f \in V^{\sigma}$, there exists $x \in V$ such that $0 \le x_{\alpha} \uparrow x$.

- **257 Corollary 30.** Bands of perfect Riesz spaces are also perfect.
- **Lemma 31.** Every perfect Riesz space is Dedekind complete.
- ²⁵⁹ **Proof.** The proof follows from the second condition of Theorem 29.

Lemma 32. Every Riesz space of the form V^{σ} is perfect.

Proof. To show the first point of Theorem 29, assume that $f_1 \neq f_2 \in V^{\sigma}$. Then there is $v \in V$ such that $f_1(v) \neq f_2(v)$. Using the fact that $\lambda f.f(v)$ is an element of $V^{\sigma\sigma}$, we can conclude that V^{σ} is separated. For the second point, let us assume that $0 \leq \{f_{\alpha}\}\uparrow$ and that for all $F \in V^{\sigma\sigma}$, if $F \geq 0$, then $\sup_{\alpha} F(f_{\alpha}) < \infty$. From this hypothesis, it follows that for all $v \in V$, if $v \geq 0$, then $\sup_{\alpha} f_{\alpha}(v) = \sup_{\alpha} \sigma(x)(f_{\alpha}) < \infty$. This means that the function $f(x) = \sup_{\alpha} f_{\alpha}(x)$ is well-defined, linear, and order-continuous. By Lemma 1.18 in [1], V^{σ} is Dedekind complete and f bounds f_{α} .

An interesting fact that is not obvious from the definitions is that the bidual of Riesz spaces can be seen as a sort of completion procedure.

It is possible to categorify the theorem above by showing that **PBanLat**₁ is a reflective subcategory of **BanLat**₁, where the reflector is given by the functor $(-)^{\sigma\sigma}$.

▶ **Theorem 33.** The functor $(-)^{\sigma\sigma}$: **BanLat**₁ → **PBanLat**₁ is left adjoint to the forgetful functor U.

Proof. We observe that if $f: V \multimap W$, then $\sigma^{-1} \circ f^{\sigma\sigma} : V^{\sigma\sigma} \multimap W$. In the other direction, if we have a function $f: V^{\sigma\sigma} \multimap W$, we can consider its restriction $f \upharpoonright V : V \multimap W$. To show that these operations are inverses, we use Theorem 11 and Theorem ??, which allow us to show that if two order-continuous functions agree on $\sigma(V)$, then they agree everywhere.

XX:8 Classical Linear Logic in Perfect Banach Lattices

Note that this implies that $PBanLat_1$ is a reflective subcategory of $BanLat_1$, which means that it is closed under the same (co)limits that exists in $BanLat_1$.

Lemma 34. If V is a separated Riesz space, then the function $\sigma: V \multimap V^{\sigma\sigma}$ is injective.

The theorem and lemma above are useful because they imply that if V is a separated Riesz space and W is a perfect Riesz space, then every order-continuous linear function $f: V \to W$ extends uniquely to a function $V^{\sigma\sigma} \to W$.

284 2.6 Signed measures as Riesz spaces

Measures are usually defined as countably additive, nonnegative real-valued functions on a σ -algebra. Signed measures provide a slight generalization by dropping the requirement of nonnegativity.

▶ Definition 35. Let (X, Σ) be a measurable space. A signed measure is a function $\mu : \Sigma \to \mathbb{R}$ such that $\mu(\emptyset) = 0$ and $\mu(\bigcup_{i \in \mathbb{N}} A_i) = \sum_{i \in \mathbb{N}} \mu(A_i)$ for disjoint sets $(A_i)_{i \in \mathbb{N}}$.

An important difference between ordinary measures and signed measures is that signed measures come equipped with a natural vector space structure. Indeed, it can be shown that signed measures are perfect Riesz spaces.

▶ Lemma 36. Let (X, Σ) be a measurable space. The space $\mathcal{M}(X, \Sigma)$ of signed measures is a normed Riesz space.

Proof. The vector space structure is defined pointwise with lattice structure defined by $\mu \lor \nu = (\mu - \nu)^+ + \nu$ using the Hahn-Jordan decomposition and the norm is the total-variation norm.

When a measure μ is positive, its total variation norm is its total mass $\mu(X)$.

Theorem 37. Let (X, Σ) be a measurable space. The space $\mathcal{M}(X, \Sigma)$ of signed measures with the total variation norm is a perfect Banach lattice.

Proof. The proof follows by applying Theorem 28, the lemma above and observing that since
 the order of measures is given pointwise, you can define their suprema pointwise as well.

303 3 Models of linear logic

The categorical semantics of linear logic is very well understood; see Mellies [27] for an overview. In this section, we show that **PBanLat**₁ is a model of classical linear logic.

306 3.1 Symmetric Monoidal Closed Structure

In order for **PBanLat₁** to interpret the multiplicative fragment of linear logic, i.e. give semantics to a linear λ -calculus with tensors, it must be a symmetric monoidal closed category. Concretely, it needs a *monoidal product* \otimes such that for every object A, the functor $A \otimes$ has a right adjoint $A \rightarrow -$, known as *linear implication*.

For models based on vector spaces, the monoidal product is typically given by the *tensor* product. For such models, linear implication has a natural interpretation in terms of linear functions. Furthermore, since our spaces are perfect, we have an involutive *linear negation* A^{\perp} defined as the space $A \rightarrow \mathbb{R}$, and, in models of classical linear logic, the equation

 $A \otimes B = (A \multimap B^{\perp})^{\perp}$ holds. Thus the tensor product \otimes can be defined in terms of linear implication \multimap and negation $^{\perp}$ in such models.

Note that this circumvents one of the main complications with the model of [11], where the existence of a suitable monoidal product is established non-constructively using a categorical density argument.

320 3.1.1 Internal Homs

Since the category **PBanLat**₁ has order-continuous linear functions with norm at most 1 as morphisms, it makes sense to define the internal hom object $V \multimap W$ as the space of order-continuous linear functions between perfect Banach lattices V and W. This definition is justified by the following theorem.

Lemma 38 (c.f. Appendix B). If V and W are perfect Riesz spaces, then the set of order continuous linear functions $V \multimap W$ is a perfect Riesz space.

From Lemma 25 and the theorem above, it follows that if V and W are perfect Banach lattices, then so is $V \multimap W$. By using standard techniques from the literature on vector models of linear logic, we have

Theorem 39. The operation \multimap : **PBanLat**₁^{op} × **PBanLat**₁ \rightarrow **PBanLat**₁ *is functorial.*

331 3.1.2 Monoidal structure

As mentioned above, the monoidal structure on vector space models of linear logic is usually defined as a tensor product, and monoidal closure is obtained from the universal property of tensor products. The usual recipe for defining tensor products is to use a free construction modulo the tensor product equations. When working with infinite-dimensional spaces, a completion procedure may be required as well.

Indeed, this is the approach taken in [15], in which a tensor product is defined for perfect Riesz spaces via a more traditional construction using the completion of the algebraic tensor product. It is also shown in [15] that $V \otimes W \cong (V \multimap W^{\perp})^{\perp}$, meaning that their construction is isomorphic to ours.

In contrast, our construction starts with the definition $V \otimes W \triangleq (V \multimap W^{\sigma})^{\sigma}$, as required by the laws of linear logic. We then show that it satisfies the expected universal property of tensor products: for every biliear function $f: V \times W \to Y$, there is a unique linear function $\hat{f}: V \otimes W \to Y$ such that $\hat{f} \circ \iota = f$, where $\iota: V \times W \to Y$ is the bilinear inclusion function. We show this using the fact that the internal hom can be used to classify bilinear functions using $V \multimap (W \multimap Y)$, then showing that this space is isomorphic to $V \otimes W \multimap Y$.

Lemma 40. $V \otimes W \multimap Y \cong V \multimap W \multimap Y$.

Proof. Recall that if V and W are perfect Riesz spaces, then $V \multimap W \cong W^{\sigma} \multimap V^{\sigma}$. Then

 $V \otimes W \multimap Y = (V \multimap W^{\sigma})^{\sigma} \multimap Y$ $\cong Y^{\sigma} \multimap (V \multimap W^{\sigma}) \cong V \multimap Y^{\sigma} \multimap W^{\sigma}$ $\cong V \multimap W \multimap Y.$

Theorem 41. $V \otimes W$, defined as $(V \multimap W^{\sigma})^{\sigma}$, satisfies the universal property of tensor products.

◀

XX:10 Classical Linear Logic in Perfect Banach Lattices

Proof. Observe that the set of (norm bounded) bilinear order-continuous functions $V \times W \rightarrow$ Y is (isometrically, in the normed case) isomorphic to $V \multimap W \multimap Y$. We must not show $V \otimes W \multimap Y \cong V \multimap W \multimap Y$. This is exactly Lemma 40.

Using the universal property (??) and the (easy to prove) facts that $V \otimes (W \otimes Y) \cong$ ($V \otimes W$) $\otimes Y$ and $V \otimes W \cong W \otimes V$, we can conclude:

559 • Theorem 42. PBanLat₁ *is a symmetric monoidal closed category.*

It is difficult in general to give an intuitive characterization of the elements of a tensor 360 product. This is also the case with our construction. Nevertheless, in the context of measures, 361 we can give some intuition for the elements of $\mathcal{M}(A) \otimes \mathcal{M}(B)$. Let μ_A and μ_B be probability 362 distributions on measurable spaces A and B, respectively. The product distribution $\mu_A \otimes \mu_B$ 363 is the joint probability distribution on $A \times B$ with marginals μ_A and μ_B obtained by sampling 364 μ_A and μ_B independently. This is an element of $\mathcal{M}(A) \otimes \mathcal{M}(B)$, but there are also other joint 365 distributions in $\mathcal{M}(A) \otimes \mathcal{M}(B)$ that do not represent independent samples. For example, 366 let $A = B = \{0, 1\}$ and consider the joint distribution $\frac{1}{2}(\delta_0 \otimes \delta_0 + \delta_1 \otimes \delta_1)$. Sampling this 367 distribution returns (0,0) or (1,1), each with probability 1/2, so the two components are 368 clearly not independent. 369

In general, not every joint distribution is an element of the tensor product, as explained in [6]. From a programming point of view, the universal property of tensor products says that the behavior of a program taking inputs of type $\mathcal{M}(A) \otimes \mathcal{M}(B)$ is fully characterized by its behavior on inputs that are independent distributions over A and B.

374 3.2 *-autonomous categories

Classical linear logic differs from its intuitionistic variant by requiring that linear negation 375 be involutive, that is, $A^{\perp \perp} = A$ for every formula A. Categorically, this is modeled by 376 *-autonomous categories, symmetric monoidal closed categories C with a functor $(-)^*$: 377 $\mathbf{C}^{\mathrm{op}} \to \mathbf{C}$ such that every object A is naturally isomorphic to A^{**} and for every three objects 378 A, B, C, there is a natural bijection $\operatorname{Hom}(A \otimes B, C^*) \cong \operatorname{Hom}(A, (B \otimes C)^*)$. Equivalently, a 379 *-autonomous category is a symmetric monoidal closed category C equipped with a *dualizing* 380 object \perp such that for every object A, the unit $\partial_A : A \to (A \multimap \bot) \multimap \bot$ is an isomorphism. 381 In our case, the dualizing object is \mathbb{R} , the unit is the linear function $\sigma_V: V \to V^{\sigma\sigma}$, and 382 the isomorphism holds by assumption. 383

Theorem 43. $PBanLat_1$ is a *-autonomous category.

385 3.3 Cartesian and co-Cartesian structure

Cartesian and co-Cartesian structure are useful in the formation of product and sum types. In models of linear logic, these are represented by linear conjunction & and disjunction \oplus , respectively. In **PBanLat**₁, both operations have $V \times W$ as their underlying set with lattice operations defined componentwise. In the normed case, we can distinguish them by choosing different norms.

Definition 44. Let V and W be normed Riesz spaces. We define

392 The product $V \& W = (V \times W, \|-\|_{sum})$, where $\|(v, w)\|_{sum} = \|v\| + \|w\|$.

set the coproduct $V \oplus W = (V \times W, \|-\|_{\max})$, where $\|(v, w)\|_{\max} = \max(\|v\|, \|w\|)$.

398

Since convergence for both is defined componentwise, by using Theorem 28 we can show that if V and W are perfect and Banach, then V & W and $V \oplus W$ are as well. The unit \top for the product and 0 for the coproduct are both the trivial Riesz space $\{0\}$.

597 • Theorem 45. $PBanLat_1$ is (co-)Cartesian.

4 Probabilistic coherence spaces and Banach lattices

Probabilistic coherence spaces (PCS) [7] are a model of linear logic with a vector space flavor. It has been shown by Ehrhard [11] that its intuitionistic fragment can be fully and faithfully embedded in a category of positive cones. In this section, we show that Banach lattices can handle not only the intuitionistic fragment of PCS, but the classical one as well. We make use of the vector space construction presented in the original paper [7].

▶ **Definition 46.** A Probabilistic Coherence Space (PCS) is a pair $(X, \mathcal{P}(X))$, where X is a countable set and $\mathcal{P}(X) \subseteq X \to \mathbb{R}^+$ called the web such that:

- 406 $\forall a \in X \exists \varepsilon_a > 0 \ \varepsilon_a \cdot \delta_a \in \mathcal{P}(X), \text{ where } \delta_a(a') = 1 \text{ iff } a = a' \text{ and } 0 \text{ otherwise};$
- $407 \qquad \blacksquare \ \forall a \in X \ \exists \lambda_a \ \forall x \in \mathcal{P}(X) \ x_a \leq \lambda_a;$

 $= \mathcal{P}(X)^{\perp \perp} = \mathcal{P}(X), \text{ where } \mathcal{P}(X)^{\perp} = \{ x \in X \to \mathbb{R}^+ \, | \, \forall v \in \mathcal{P}(X) \, \sum_{a \in X} x_a v_a \leq 1 \}.$

▶ **Definition 47.** Let $(X, \mathcal{P}(X))$ be a PCS. Its linear negation is the PCS $(X, \mathcal{P}(X)^{\perp})$.

▶ Definition 48. Let $(X, \mathcal{P}(X))$ and $(Y, \mathcal{P}(Y))$ be PCSs. The PCS $X \multimap Y$ is the pair ($X \times Y, \mathcal{P}(X \multimap Y)$), where $P(X \multimap Y)$) = { $M : X \times Y \to \mathbb{R}^+ | \forall v \in \mathcal{P}(X) \ M \cdot v \in \mathcal{P}(Y)$ }.

The intuition behind Definition 46 is that the web of every PCS corresponds to the positive unit ball of a partially-ordered vector space. This idea is used by Ehrhard and Danos [7] to define a functor that maps every PCS to a Banach space. It is possible to show that this vector space can be equipped with a Riesz space structure, where the order is defined pointwise.

⁴¹⁷ ► Definition 49. Given a PCS (|X|, PX), we define BX = { $u \in \mathbb{R}^{|X|} | |u| \in \mathcal{P}X$ } and ⁴¹⁸ $eX = \bigcup_{\lambda>0} \lambda BX$. The pair ($eX, u \mapsto \sup_{u' \in \mathcal{P}X^{\perp}} \langle |u|, u' \rangle$) is the normed Riesz space associated ⁴¹⁹ with the PCS (|X|, PX).

420 It is shown in [7] that eX is a Banach space. Furthermore, the lattice structure can be 421 defined pointwise, making eX a Banach lattice. Later in this section we will show that e can 422 be made into a functor.

423 4.1 PCoh and duality

Ehrhard and Danos have shown that the partial order plays an important role in understanding how to generalize **PCoh** [7]. In their attempt to obtain an intrinsic representation for probabilistic coherence spaces, the authors use exclusively the norm and cannot prove the equation $e(X^{\perp}) = e(X)^{\perp}$, where $e(X)^{\perp}$ is the norm dual. That said, we can show that this functor preserves order-duality. Note that the proof uses the fact that 2^X is a directed set.

▶ **Theorem 50** (c.f. Appendix C). For every probabilistic coherence space X, there is a natural isomorphism $e(X^{\perp}) \cong e(X)^{\sigma}$.

431 Corollary 51. For every PCS $(X, \mathcal{P}(X))$ the vector space eX is a perfect Banach lattice.

XX:12 Classical Linear Logic in Perfect Banach Lattices

432 Since convergence for PCS is defined componentwise, it is possible to use a similar proof
 433 technique to show

- **434** ► **Theorem 52.** The operation *e* is monoidal closed and functorial.
- **Proof.** The functoriality of e has been proven in Section 5.1 of [7].

- 436 Another important theorem which is direct to show is.
- **437 • Theorem 53.** The functor $e : \mathbf{PCoh} \to \mathbf{PBanLat}_1$ is full and faithful.

5 Categories of Cones and PBanLat₁

Even though **PBanLat**₁ is a mathematically natural model of linear logic, it relies on tools from functional analysis not usually familiar to computer scientists. On the other hand, in recent years, cones have found numerous applications in semantics of programming languages and logics. In this section we show that **PBanLat**₁ is isomorphic to a category cones, meaning that computer scientists can translate their intuitions about cones to this novel setting without having to learn functional analysis.

As it was frequently mentioned throughout this paper, every Banach lattice gives rise to a positive cone. Furthermore, since every **PBanLat**₁ morphism $f: V \to W$ is positive and has norm at most 1, it restricts to a linear function $\mathcal{B}(V)^+ \to \mathcal{B}(W)^+$. With this observations we state a few definitions from [5, 11], which assume that the cones are separated.

Definition 54. A cone C is a \mathbb{R}^+ -semimodule with a norm $\|\cdot\|: C \to \mathbb{R}^+$.

Every cone can be equipped with the partial order $x \le y$ if and only if there is a z such that x + z = y, meaning that it is possible to define a partial subtraction operation whenever $x \le y$, calling y - x the element such that x + (y - x) = y.

A function $f: C_1 \to C_2$ between cones is linear if it commutes with addition and scalar multiplication, it is monotonic if it preserves the order relation, and it is Scott-continuous if for every directed set x_{α} with supremum x, $\sup_{\alpha} f(x_{\alpha}) = f(x)$. As is the case with partially-ordered vector spaces, there are different classes of cones where the order and the norm have particular properties:

- **458 Definition 55.** A cone C is said to be:
- 459 Sequentially complete if every norm-bounded sequence has a least upper bound.
- 460 Directed complete if every norm-bounded directed set has a least upper bound.
- \blacksquare A lattice cone if the poset structure is a lattice.

Using this notation, it seems appropriate to imagine that there should be a functor **PBanLat₁** \rightarrow **CLat**, where **CLat** is the category of directed complete cone lattices. It is unclear, however, if there is a mapping on morphisms. Luckily, the lemma below guarantees that the mapping is well-defined.

Lemma 56. Let V and W be two perfect Banach lattices and $f : V \to W$ a linear, positive function of norm at most 1. The function f is order-continuous if and only if sup_{x∈A} f(x) = f(v) whenever $A \subseteq V^+$ is a non-empty upwards-directed set with supremum v.

⁴⁷⁰ **Proof.** This result is a direct consequence of the weak Fatou property.

Since the mapping on morphisms is basically the identity, the functorial laws hold, which allows us to conclude that there is a functor $PBanLat_1 \rightarrow CLat$.

Next, we would like to map every positive cone to a vector space. Let C be a positive cone and define

475 $C - C = \{(c_1, c_2) \mid c_1, c_2 \in C\} / \sim,$

where \sim is the binary relation $(c_1, c_2) \sim (c_3, c_4)$ iff $c_1 + c_4 = c_2 + c_3$. Intuitively, C - Ccorresponds to the vector space of formal differences $c_1 - c_2$ of elements in C. The equivalence relation is used to capture the fact that, for instance, (3, 2) and (4, 3) should represent the same real number, since 3 - 2 = 1 = 4 - 3.

Theorem 57. Let C be a directed complete cone lattice. Then C - C is a perfect Banach lattice.

⁴⁸² **Proof.** The proof can be found in the Appendix D.

◄

By linearity, Scott-continuous functions $f : C \to D$ with norm at most 1 extend to order-continuous functions $f : (C - C) \to (D - D)$ with norm at most 1 and we can prove that there is a functor **CLatLin** \to **PBanLat**₁. With this functor and the positive cone restriction functor defined, it is a direct calculation to show:

487 • Theorem 58. The categories $PBanLat_1$ and CLatLin are isomorphic.

Variables x, y, zReals \mathbb{R} \in rMK Expressions M $x \mid r \mid$ uniform $\mid (M_1, M_2) \mid \pi_1 M \mid \pi_2 M \mid$ let x = M in N ::= $x \mid \lambda x.t \mid t \mid u \mid t \otimes u \mid$ let $x \otimes y = t$ in $u \mid$ sample t_i as x_i in MLL Expressions t, u::=Types MK $\mathbb{R} \mid \tau \times \tau$ τ ::=Types LL $1 \mid \mathcal{M}\tau \mid \underline{\tau} \multimap \underline{\tau} \mid \underline{\tau} \otimes \underline{\tau}$::= $\underline{\tau}$ Linear Contexts Γ $x_1: \underline{\tau}_1, \ldots, x_n: \underline{\tau}_n$::=Г MK Contexts $::= x_1:\tau_1,\ldots,x_n:\tau_n$

Figure 1 Terms and Types of λ_{MK}^{LL}

488 6 A Probabilistic Calculus

Though it is theoretically interesting understanding how **PBanLat₁** relates to existing 489 models of linear logic, we are also interested in using it as a semantic basis for a language 490 with probabilistic primitives. Being symmetric monoidal closed, it can give semantics to 491 the linear λ -calculus. This, however, is insufficient from a programming point of view. The 492 linearity restrictions are severely limiting in terms of which programs one can define in this 493 language. A frequently used solution to this lack of expressivity is to use the exponential 494 modality, where the coKleisli category is Cartesian closed, meaning that it can interpret the 495 λ -calculus. 496

However, even though we have not defined a linear logic exponential in **PBanLat**₁, we can still get non-linear programming by using recent work [3] that proposes a new syntax for programming with linear operators and Markov kernels. The proposed two-level calculus allows for non-linear programs to be defined by using a lax-monoidal modality.

XX:14 Classical Linear Logic in Perfect Banach Lattices

501 The λ_{MK}^{LL} metalanguage

The semantic structure used to interpret the calculus of [3] is given by a triple $(\mathbf{C}, \mathbf{L}, \mathcal{M})$, where **C** is roughly a category of Markov kernels¹, **L** is a symmetric monoidal closed category and $\mathcal{M}: \mathbf{C} \to \mathbf{L}$ is a lax monoidal functor.

This two-level structure manifests itself at the syntactic level by having a two-level syntax: 505 the first level is used to program kernels while the second one serves as a kind of metalanguage 506 that has access to higher-order functions, both of which are depicted in Figure 1, The linear 507 language has linear function types, which allows for higher-order programming and, unlike 508 most languages based on linear logic, it has a modality \mathcal{M} , which corresponds to the types 509 that may be sampled from. The variables bound by the linear context are, roughly speaking, 510 computations. In the language for kernels there are no linearity restrictions and, therefore, 511 variables, i.e. samples from distributions, can be freely duplicated and discarded. Under this 512 perspective, the variables in MK programs should be thought of as values. The intuition 513 behind this language is that linearity forbids distributions to be sampled more than once, 514 but once you have the sample in hands, it can be used as many times as you want. 515

Each layer has its own typing judgement relations \vdash_{LL} and \vdash_{MK} , which we go over in more detail in Appendix A. We highlight one of the most interesting rules; it is the rule that allows programs to be transported between layers:

SAMPLE

519

$$\frac{x_1:\tau_1\cdots x_n:\tau_n\vdash_{MK}M:\tau\quad \Delta;\Gamma_i\vdash_{LL}t_i:\mathcal{M}\tau_i\qquad 0< i\leq n}{\Delta;\Gamma_1,\cdots,\Gamma_n\vdash_{LL}\mathsf{sample}\ t_i\ \mathsf{as}\ x_i\ \mathsf{in}\ M:\mathcal{M}\tau}$$

Operationally, it samples from n LL programs $\{t_i\}_i$, each sample is bound to the corresponding variable in $\{x_i\}_i$ and finally the continuation M is executed.

We want to model λ_{MK}^{LL} with **PBanLat**₁. For that we still need a CD category and a lax monoidal functor. For the CD category we will use the category of measurable spaces and sub-Markov kernels.

▶ Definition 59. The category sStoch has measurable spaces as objects and sub-Markov kernels as morphisms, i.e. measurable functions between a measurable space and the space of subprobability distributions over a measurable space.

sStoch is a CD category, which means that it is symmetric monoidal, with the monoidal
 product being the product measurable space.

Theorem 60 (c.f. Appendix E). There is a lax monoidal functor $\mathcal{M} : \mathbf{sStoch} \rightarrow \mathbf{PBanLat_1}$.

This means that the triple (sStoch, PBanLat₁, \mathcal{M}) is a λ_{MK}^{LL} model.

533 7 Related work

There have been a number of semantics of linear logic based on vector space-like objects. Two important families of such semantics are the ones based on probabilistic coherence spaces and the ones based on Banach spaces. As we will explain below, we see our model as a nice synthesis of these two approaches.

¹ a CD category, to be more precise

538 Positive Cone Semantics of Linear Logic

To overcome the limitation that **PCoh** cannot represent continuous distributions, Ehrhard et al. define a cartesian closed category $\mathbf{CStab_m}$ [12], which uses normed \mathbb{R}^+ -semimodule which are in correspondence with positive cones of partially ordered vector spaces—to interpret a probabilistic variant of PCF with continuous distributions. In a follow-up paper, Ehrhard [11] has defined a category $\mathbf{CLin_m}$ of sequentially complete positive cones with measurability paths and linear Scott continuous maps in which **PCoh** embeds fully and faithfully.

A similar approach was taken in [32], which defined a category **CCones** of so-called coherent cones and linear contractive functions and showed that it is a model of classical linear logic. These cones come equipped with a different notion of completeness that is stronger than sequential completeness but weaker than ours.

From a mathematical point of view, the objects of both **CCones** and **CStab_m** are not as well understood as Banach lattices, making them not ideal semantic frameworks to reason about probabilistic programs. Besides, our model provides a clear mathematical justification for having Fatou-like properties in the semantics: it is forced upon it by Theorem 28 instead of being there for denotational reasons, as is the case of **CStab_m**, or in enabling the exponential construction, as is the case of **CCones**, showing a kind of canonicity of our model.

556 Vector Space Semantics of Linear Logic

⁵⁵⁷ Dahlqvist and Kozen [6] have defined a category of partially ordered Banach spaces ⁵⁵⁸ **RoBan**, shown that it is a model of intuitionistic linear logic, and used it to interpret a ⁵⁵⁹ higher-order imperative probabilistic language with while loops and soft-conditioning.

Their model also uses a mathematically well-understood class of vector spaces. That being said, by using a more general class of vector spaces than we do, their model has less structure than ours. A practical consequence of this lack of structure is that in order to guarantee the soundness of their semantics, they define 6 type grammars that are used for different program constructs. As an example, in order to interpret conditionals and while loops the context may only have Dedekind complete types.

Another relevant vector space model is the one based on complex coherent Banach spaces [18]. However, since they are complex vector spaces, it is unclear if it would be possible to embed **PCoh** into them.

 $_{569}$ Neither RoBan nor $CStab_m$ are models of classical linear logic.

570 8 Conclusion

In this paper we have shown that **PBanLat**₁ is a model of classical linear logic that conservatively extends **PCoh** and can be used to give semantics to a recursive probabilistic calculus. Our model differs from existing extensions of **PCoh** that only extends **PCoh**'s intuitionistic fragment, meaning that they do not have an involutive negation. We believe that our model is a good fit for formal verification purposes because Riesz spaces have decades of research and have been extensively used in the formalization of stochastic processes.

For future work, we are interested in showing that **PBanLat**₁ can accommodate exponentials and use this category for reasoning about correctness properties of probabilistic programs such as inference algorithms.

580 Acknowledgements

[suppressed for double-blind reviewing]

582		References
583	1	Charalambos D Aliprantis and Owen Burkinshaw. Positive operators. Springer, 2006.
584	2	Robert J Aumann. Borel structures for function spaces. Illinois Journal of Mathematics, 1961.
585	3	Pedro H. Azevedo de Amorim. A higher-order language for markov kernels and linear operators.
586		In Foundations of Software Science and Computation Structures (FoSSaCS), 2023.
587	4	Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael
588		Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic
589		programming language. Journal of statistical software, 2017.
590	5	Raphaëlle Crubillé. Probabilistic stable functions on discrete cones are power series. In <i>Logic</i>
591		in Computer Science (LICS), 2018.
592	6	Fredrik Dahlqvist and Dexter Kozen. Semantics of higher-order probabilistic programs with
593		conditioning. In Principles of Programming Languages (POPL), 2019.
594	7	Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order
595		probabilistic computation. Information and Computation, 209(6):966–991, 2011.
596	8	Swaraj Dash, Younesse Kaddar, Hugo Paquet, and Sam Staton. Affine monads and lazy
597		structures for bayesian programming. In Principles of Programming Languages (POPL), 2023.
598	9	Thomas Ehrhard. On Köthe sequence spaces and linear logic. Mathematical Structures in
599		Computer Science, $12(5):579-623$, 2002.
600	10	Thomas Ehrhard. Differentials and distances in probabilistic coherence spaces. arXiv preprint
601		arXiv:1902.04836, 2019.
602	11	Thomas Ehrhard. On the linear structure of cones. In Logic in Computer Science (LICS),
603		2020.
604	12	Thomas Ehrhard, Michele Pagani, and Christine Tasson. Measurable cones and stable,
605		measurable functions: a model for probabilistic higher-order programming. In <i>Principles of</i>
606	10	Programming Languages (POPL), 2017.
607	13	Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces are
608	14	Tully abstract for probabilistic PCF. In Principles of Programming Languages (POPL), 2014.
609	14	David H Fremin. Measure theory. forres Fremin, 2000.
610	15	Philosophical Society, pages 45–52, Combridge University Proce 1968
611	16	Tablas Fritz A synthetic approach to markov kernels, conditional independence and theorems
612	10	on sufficient statistics Advances in Mathematics 370:107239 2020
614	17	Timon Gehr, Sasa Misailovic, and Martin Vechev. Psi: Exact symbolic inference for probabil-
615		istic programs In Computer Aided Verification (CAV) 2016
616	18	Jean-Yves Girard. Coherent banach spaces: a continuous denotational semantics. <i>Theoretical</i>
617		Computer Science, 227(1-2):275–297, 1999.
618	19	Noah D Goodman, Vikash K Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua B
619		Tenenbaum. Church: a language for generative models. In Proceedings of the Twenty-Fourth
620		Conference on Uncertainty in Artificial Intelligence, 2008.
621	20	Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A convenient category for
622		higher-order probability theory. In Logic in Computer Science (LICS), 2017.
623	21	Steven Holtzen, Guy Van den Broeck, and Todd Millstein. Scaling exact inference for discrete
624		probabilistic programs, 2020.
625	22	Marie Kerjean and Christine Tasson. Mackey-complete spaces and power series–a topological
626		model of differential linear logic. Mathematical Structures in Computer Science, 28(4):472–507,
627		2018.
628	23	Dexter Kozen. Semantics of probabilistic programs. In Symposium on Foundations of Computer
629		Science (SFCS), 1979.

- Alexander K Lew, Marco F Cusumano-Towner, Benjamin Sherman, Michael Carbin, and
 Vikash K Mansinghka. Trace types and denotational semantics for sound programmable
 inference in probabilistic languages. In *Principles of Programming Languages (POPL)*, 2019.
- WAJ Luxemberg and AC Zaanen. Notes on Banach function spaces VI-XIII. Proceedings of
 the Koninklijke Nederlandse Akademie van Wetenschappen, Series A, 66:251–263, 1963.
- ⁶³⁵ 26 Vikash K Mansinghka, Ulrich Schaechtle, Shivam Handa, Alexey Radul, Yutian Chen, and
 ⁶³⁶ Martin Rinard. Probabilistic programming with programmable inference. In *Programming* ⁶³⁷ Language Design and Implementation (PLDI), 2018.
- Paul-André Mellies. Categorical semantics of linear logic. Panoramas et syntheses, 27:15–215,
 2009.
- Praveen Narayanan, Jacques Carette, Wren Romano, Chung-chieh Shan, and Robert Zinkov.
 Probabilistic inference by program transformation in hakaru (system description). In *Functional and Logic Programming (FLOPS)*, 2016.
- Praveen Narayanan and Chung-chieh Shan. Symbolic disintegration with a variety of base
 measures. ACM Transactions on Programming Languages and Systems (TOPLAS), 2020.
- G45 30 Feras A Saad, Martin C Rinard, and Vikash K Mansinghka. Sppl: probabilistic programming
 with fast exact symbolic inference. In *Programming Language Design and Implementation*(*PLDI*), 2021.
- G48 31 Chung-chieh Shan and Norman Ramsey. Exact bayesian inference by symbolic disintegration.
 G49 In Principles of Programming Languages (POPL), 2017.
- Sergey Slavnov. Linear logic in normed cones: probabilistic coherence spaces and beyond.
 Mathematical Structures in Computer Science, 31(5):495–534, 2021.
- ⁶⁵² 33 Dario Stein and Sam Staton. Compositional semantics for probabilistic programs with exact
 ⁶⁵³ conditioning. In *Logic in Computer Science (LICS)*, 2021.
- G54 34 Christine Tasson and Thomas Ehrhard. Probabilistic call by push value. Logical Methods in
 G55 Computer Science, 15, 2019.
- Bavid Tolpin, Jan-Willem van de Meent, Hongseok Yang, and Frank Wood. Design and
 implementation of probabilistic programming language anglican. In Implementation and
 Application of Functional programming Languages (IFL), 2016.
- 659 36 Adriaan C Zaanen. Introduction to operator theory in Riesz spaces. Springer, 2012.

A A Metalanguage for Linear Operators and Markov Kernels

In this section we further explain the two-level language λ_{MK}^{LL} and its semantics. The language MK corresponds to an effectful language with probabilistic primitives and where free variables are assumed to be values, as opposed to computations. For instance, the program $x : \mathbb{N}, y : \mathbb{N} \vdash_{MK} x + y : \mathbb{N}$ is interpreted as a deterministic program. This language is interpreted in a CD category, which can be seen as an abstraction for programming with commutative effects [16].

Definition 61. CD categories are symmetric monoidal categories such that every object A has a commutative comonoid structure $copy_A : A \to A \otimes A$ and $delete_A : A \to 1$ satisfying certain structural properties.

In the context of probabilistic programming, there are many CD categories to choose from. In particular, for any subprobability monad, its Kleisli category is a CD category. This is the case for the **sStoch** category, since it can be characterized as the category of measurable sets and measurable functions $A \to \mathcal{G}(B)$, where \mathcal{G} is the subprobability monad over **Meas**.

The language LL is basically a linear λ -calculus. By itself, linearity limits the expressivity of the language quite a bit. In the original paper, the author argues that for probabilistic programming, the linear usage of variables is, semantically, too restrictive, since many probabilistic which are linear, in the semantic sense, may use variables for than once [3]. This observation led to the introduction of the \mathcal{M} modality in the LL language which allows MK programs to be called from an LL program. Semantically, this is interpreted as a lax monoidal functor.

▶ Definition 62. Let C and D be monoidal categories. A (lax) monoidal functor is a functor $F: \mathbf{C} \to \mathbf{D}$ equipped with a natural transformation $\varepsilon_{A,B}: FA \otimes_{\mathbf{D}} FB \to F(A \otimes_{\mathbf{C}} B)$ and a morphism $I_{\mathbf{D}} \to F(I_{\mathbf{C}})$ making certain coherence diagrams commute.

From a programming point of view, types $\mathcal{M}\tau$ should be thought of as types from which one can sample from. Supposing that the language has a primitive uniform for the uniform distribution over the unit interval the Sample construct can be used to write the program

sample uniform as x in (x + x)

The program above samples from a uniform distribution and adds the result to itself. This program illustrates why this syntax increases the expressivity of the linear λ -calculus. By allowing the continuation x + x to be an MK program, variables may be freely reused or discards without worrying about syntactic restriction imposed by linearity.

However, once inside the MK language, there is no way of going back to the higher-order language, meaning that the program sample uniform as x in (sample uniform as y in (x+y)) is not well-typed. This is mitigated by lax monoidality, which makes it possible to simultaneously sample from distributions: sample (uniform, uniform) as (x, y) in (x + y).

Definition 63. A model of λ_{MK}^{LL} is a triple $(\mathbf{C}, \mathbf{L}, \mathcal{M})$, where \mathbf{C} , a symmetric monoidal closed category \mathbf{L} and $\mathcal{M} : \mathbf{M} \to \mathbf{C}$ is a lax monoidal functor.

⁶⁹⁹ The typing rules are depicted in Figure 2. They are basically the amalgamation of the ⁷⁰⁰ rules for programming with CD categories, i.e. a first-order expression language with pairs, ⁷⁰¹ with symmetric monoidal closed categories, i.e. the linear λ -calculus with tensor types. The

Const VAR Uniform $r \in \mathbb{R}$ $\Gamma, x : \tau \vdash_{MK} x : \tau$ $\Gamma \vdash_{MK} r : \mathbb{R}$ $\Gamma \vdash_{MK}$ uniform : \mathbb{R} Pair Let $\Gamma \vdash_{MK} t : \tau_1 \qquad \Gamma \vdash_{MK} u : \tau_2$ $\Gamma \vdash_{MK} t : \tau_1 \qquad \Gamma, x : \tau_1 \vdash_{MK} u : \tau$ $\Gamma \vdash_{MK} \mathsf{let} \ x = t \ \mathsf{in} \ u : \tau$ $\Gamma \vdash_{MK} (t, u) : \tau_1 \times \tau_2$ Proj2 Proj1 AXIOM $\Gamma \vdash_{MK} t : \tau_1 \times \tau_2$ $\Gamma \vdash_{MK} t : \tau_1 \times \tau_2$ $\boxed{\Gamma \vdash_{MK} \pi_2 t : \tau_2} \qquad \qquad \boxed{x : \tau \vdash_{LL} x : \tau}$ $\Gamma \vdash_{MK} \pi_1 t : \tau_1$ Abstraction Unit $\frac{\Gamma, x: \tau_1 \vdash_{LL} t: \tau_2}{\Gamma \vdash_{LL} \lambda x.t: \tau_1 \multimap \tau_2}$ $\overline{\cdot \vdash_{LL}}$ unit : 1 Tensor Application $\Gamma_1 \vdash_{LL} t: \tau_1 \multimap \tau_2 \qquad \Gamma_2 \vdash_{LL} u: \tau_1$ $\Gamma_1 \vdash_{LL} t : \tau_1 \qquad \Gamma_2 \vdash_{LL} u : \tau_2$ $\Gamma_1, \Gamma_2 \vdash_{LL} t \otimes u : \tau_1 \otimes \tau_2$ $\Gamma_1, \Gamma_2 \vdash_{LL} t \ u : \tau_2$ LetTensor $\Gamma_1 \vdash_{LL} t: \tau_1 \otimes \tau_2 \qquad \Gamma_2, x: \tau_1, y: \tau_2 \vdash_{LL} u: \tau$ $\Gamma_1, \Gamma_2 \vdash_{LL} \mathsf{let} \ x \otimes y = t \ \mathsf{in} \ u : \tau$ SAMPLE $\frac{x_1:\tau_1 \cdots x_n:\tau_n \vdash_{MK} M:\tau \quad \Delta; \Gamma_i \vdash_{LL} t_i:\mathcal{M}\tau_i \qquad 0 \leq i < n}{\Delta; \Gamma_1, \cdots, \Gamma_n \vdash_{LL} \mathsf{sample} \ t_i \ \mathsf{as} \ x_i \ \mathsf{in} \ M:\mathcal{M}\tau}$

Figure 2 Typing rules for λ_{MK}^{LL}

XX:20 Classical Linear Logic in Perfect Banach Lattices

$$\frac{V_{\text{AR}}}{\tau \times \Gamma \xrightarrow{id_{\tau} \times del_{\Gamma}} \tau} \qquad \begin{array}{c} \overset{\text{LET}}{\Gamma \xrightarrow{M} \tau_{1} \quad \Gamma \times \tau_{1} \xrightarrow{N} \tau_{2}} \\ & \Gamma \xrightarrow{copy;(id \times M);N} \tau_{2} \end{array} \qquad \begin{array}{c} \xrightarrow{\Gamma \xrightarrow{M} \tau_{1} \quad \Gamma \xrightarrow{N} \tau_{2}} \\ & \Gamma \xrightarrow{copy;(id \times M);N} \tau_{2} \end{array} \qquad \begin{array}{c} \xrightarrow{\Gamma \xrightarrow{M} \tau_{1} \quad \Gamma \xrightarrow{N} \tau_{2}} \\ & \Gamma \xrightarrow{copy;(M \times N)} \tau_{1} \times \tau_{2} \end{array} \\ & \times \overset{\text{ELIM}_{i}}{\Gamma \xrightarrow{M} \tau_{1} \times \tau_{2}} \\ & \frac{\Gamma \xrightarrow{M} \tau_{1} \times \tau_{2}}{\Gamma \xrightarrow{M;(id_{\tau_{i}} \times del)} \tau_{i}} \end{array}$$

$$\begin{array}{cccc} V_{\text{AR}} & & \text{ABSTRACTION} & & \text{APPLICATION} \\ \hline \underline{\tau} & \underline{\tau} & & \overline{\tau} & & \overline{\tau} & \underline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & & \overline{\tau} & & \overline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & & \overline{\tau} & & \overline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & & & \overline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & & & \overline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} & \underline{\tau} & & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \\ \hline \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} & \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} & \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} & \underline{\tau} \\ \underline{\tau} & \underline{\tau} &$$

SAMPLE

$$\frac{\tau_1 \times \cdots \times \tau_n \xrightarrow{M} \tau \qquad \Gamma_i \xrightarrow{t_i} \mathcal{M}\tau_i}{\Gamma_1 \otimes \cdots \otimes \Gamma_n \xrightarrow{t_1 \otimes \cdots \otimes t_n} \mathcal{M}\tau_1 \otimes \cdots \otimes \mathcal{M}\tau_n \xrightarrow{\mu} \mathcal{M}(\tau_1 \times \cdots \times \tau_n) \xrightarrow{\mathcal{M}M} \mathcal{M}\tau_i}$$

Figure 3 Categorical Semantics of λ_{MK}^{LL}

main novelty is the introduction of the lax monoidal modality \mathcal{M} and its accompanying typing rule Sample which connects the MK and LL languages.

Much like the typing rules, the categorical semantics of λ_{MK}^{LL} is the combination of the categorical semantics of the internal languages of CD categories and the linear λ -calculus with the exception of the Sample rule that makes use of the functor \mathcal{M} . The full semantics is depicted in Figure 3.

B Proof of Lemma 38

By Theorem 14, $V \multimap W$ is a Riesz space. Applying Theorem 29, we can also show that it is perfect. To show separability, let $f_1, f_2 : V \multimap W$ be distinct functions. Then there is a point $v \in V$ such that $f_1(v) \neq f_2(v)$. Since W is perfect, it is separated, therefore there exists $g : W \multimap \mathbb{R}$ such that $g(f_1(v)) \neq g(f_2(v))$. Then the order-continuous function $\lambda f.g(f(v))$ separates the points f_1 and f_2 , therefore $V \multimap W$ is separated.

Now let $0 \leq \{f_{\alpha}\}$ be an increasing net such that $\sup_{\alpha} F(f_{\alpha}) < \infty$ for all positive $F: (V \multimap W) \multimap \mathbb{R}$. We can define an f such that $f_{\alpha} \uparrow f$ pointwise. Let $v \in V^+$ and let $F: W \multimap \mathbb{R}$ be a positive functional. Consider the functional $\lambda f.F(f(v)): (V \multimap W) \multimap \mathbb{R}$. By hypothesis, $\sup_{\alpha} (F(f_{\alpha}(v))) < \infty$, and since W is perfect and $\{f_{\alpha}(v)\}$ is a positive net in W, there exists $f(v) \in W$ such that $f_{\alpha}(v) \uparrow f(v)$. This defines f on elements of V^+ , and for arbitrary $v \in V$ we take $f(v) = f(v^+) - f(v^-)$. Then $\sup_{\alpha} f_{\alpha} = f$.

720 C Proof of Theorem 50

If $u \in e(X^{\perp})$, consider the element $f_u = \lambda x \cdot \langle u^+, x \rangle - \langle u^-, x \rangle$. It is possible to show that the function $\lambda x \cdot \langle u, x \rangle$ is positive and Scott-continuous, therefore order-continuous for every $u \in \mathcal{P}(X)$. Using this result, it is not hard to show that $f_u \in e(X)^{\sigma}$.

⁷²⁴ Conversely, consider an element $f \in e(X)^{\sigma}$. Without loss of generality, we can assume ⁷²⁵ that f is positive. We want to associate to f an element in $e(X^{\perp})$. As is shown by [7], we ⁷²⁶ can alternatively characterize the space e(X) as

$$\{u \in \mathbb{R}^{|X|} \mid \exists \lambda > 0 \ \forall u' \in \mathcal{P}(X^{\perp}) \ \langle |u|, u' \rangle \le \lambda \}.$$

⁷²⁸ Consider the function $f_{\delta} = \lambda x. f(\delta_x)$. Let us show that $f_{\delta} \in e(X^{\perp})$. To do this, we show ⁷²⁹ that for every $u \in \mathcal{P}(X)$, $\langle |f'|, u \rangle$ is uniformly bounded. Let $(u_{\alpha})_{\alpha \in \mathbb{P}_{\text{fin}}(X)}$ be the ascending ⁷³⁰ net $u_{\alpha,a} = u_a$ if $a \in \alpha$ and 0 otherwise. By expanding the definition, we get the equality

731
$$\langle |f_{\delta}|, u_{\alpha} \rangle = \sum_{a \in |X|} |f(\delta_a)| u_{\alpha,a}$$

732
$$\sum_{a \in |X|} |f(\delta_a u_{\alpha,a})| = \sum_{a \in |X|} f(\delta_a u_{\alpha,a}).$$

We get the last equality from f being a positive function. Since every u_{α} has finite support, the expression above is well defined.

$$\sum_{a \in |X|} f(\delta_a u_{\alpha,a}) = f\left(\sum_{a \in |X|} \delta_a u_{\alpha,a}\right) = f(u_\alpha)$$

Since f is order-continuous and monotone and $\{u_{\alpha}\}$ is an increasing net, we can conclude that $\langle |f_{\delta}|, u \rangle \leq f(u)$, therefore for every $u \in \mathcal{P}(X)$, $\langle |f_{\delta}|, u \rangle \leq ||f||$ and $f_{\delta} \in e(X^{\perp})$. If f is not positive, we decompose it as the difference of two positive maps $f = f^+ - f^-$ and define $f_{\delta} = f_{\delta}^+ - f_{\delta}^-$.

⁷³⁷ A direct calculation shows that this is indeed an isomorphism.

738 **D** Proof of Theorem 57

⁷³⁹ Let C be a directed complete lattice cone. In order to define functions over it we use ⁷⁴⁰ the universal property of quotients: it suffices to define it over every pair (c_1, c_2) while ⁷⁴¹ guaranteeing that the function acts the same over every equivalence class.

For instance, the vector space structure can be simply defined componentwise. Let $(c_1, c_2), (c_3, c_4) \in C - C$ then we define

744
$$(c_1, c_2) + (c_3, c_4) = (c_1 + c_3, c_2 + c_4)$$

745
$$\alpha(c_1, c_2) = (\alpha c_1, \alpha c_2) \text{ for } \alpha \ge 0$$

$$\alpha(c_1, c_2) = (-\alpha c_2, -\alpha c_1)$$
 otherwise

The lattice operations require a bit more ingenuity, and we first observe the equation $u \lor v = u + (v - u)^+$ which holds in every Riesz space, reducing the lowest upper bound operation to addition and the positive part. By doing some algebraic manipulations we get that if $(c_1, c_2), (c_3, c_4) \in C - C$ then we define $(c_1, c_2) \lor (c_3, c_4) = (c_1, c_2) - ((c_3, c_4) - (c_1, c_2))^+ =$ $(c_1, c_2) + (c_3 + c_2 - (c_1 + c_4) \land (c_2 + c_3), 0) = (c_1 + c_3 + c_2 - (c_1 + c_4) \land (c_2 + c_3), c_2)$. The lattice equations such as commutativity and idempotency follow by unfolding the definitions and from C being a lattice.

Before defining a norm over C - C we first need the following lemma

XX:22 Classical Linear Logic in Perfect Banach Lattices

Lemma 64. $(C - C)^+ \cong \{(c, 0) | c \in C\} \cong C$.

Proof. The mapping $\{(c,0) | c \in C\} \rightarrow (C-C)^+$ is the injection through the equivalence class function and the mapping in the other direction can be constructed by observing that whenever $(c_1, c_2) \ge (0, 0)$ it can be shown that $c_1 \ge c_2$ and, therefore, $(c_1 - c_2, 0) = (c_1, c_2)$ and this decomposition is unique, since (c, 0) = (d, 0) implies, by definition of ~ that c = d. The second isomorphism is trivial.

Given a norm over C it is possible to extend it to a norm over C - C. This follows from the property of normed Riesz spaces, where |||v||| = ||v|| which forces us to define $||(c_1, c_2)|| = |||(c_1, c_2)|||_C$. Note that since $|(c_1, c_2)|$ is a positive element of C - C, by the lemma above it can be mapped back to an element of C which, in turn, has a norm.

Therefore, we have shown that C - C is a normed Riesz space. Since C has the directed completeness property it follows that C - C has the weak Fatou property and, therefore, it is Banach and perfect.

E Proof of Theorem 60

There is a standard functor \mathcal{M} that maps measurable sets to the vector space of signed measures and sub-Markov kernels $f: A \to MB$ to the linear function $\mathcal{M}f(\mu) = \int f d\mu$. The proof of linearity is standard, but order-continuity requires a few words. Let $\{\mu_{\alpha}\} \downarrow 0$ be a descending arrow, $\mathcal{M}f(\mu_{\alpha}) = \int f d\mu_{\alpha} \leq \int 1 d\mu_{\alpha} = \mu_{\alpha}(A)$ which, as μ_{α} goes to zero, so does $\mu_{\alpha}(A)$, making \tilde{f} order-continuous. The functorial laws also follows from standard proofs from the literature.

To show that \mathcal{M} is lax monoidal, we need to define a natural transformation $\mu_{X,Y}$: $\mathcal{M}(X) \otimes \mathcal{M}(Y) \to \mathcal{M}(X \times Y)$ which is easily defined by the universal property of the tensor product and a morphism $\varepsilon : \mathbb{R} \multimap \mathcal{M}(1)$ which maps a real number r to the measure $r\delta_{\{*\}}$, where * is the only member of the singleton set 1. Showing that the necessary diagrams commute follows from the universal property of the tensor product.