The Beta-Bernoulli process and algebraic effects

Sam Staton
Univ. Oxford

Dario Stein
Univ. Oxford

Hongseok Yang
KAIST

Nathanael L. Ackerman
Harvard Univ.

Cameron E. Freer
Borelian

Daniel M. Roy
Univ. Toronto

Abstract

In this paper we use the framework of algebraic effects from programming language theory to analyze the Beta-Bernoulli process, a standard building block in Bayesian models. Our analysis reveals the importance of abstract data types, and two types of program equations, called commutativity and discardability. We develop an equational theory of terms that use the Beta-Bernoulli process, and show that the theory is complete with respect to the measure-theoretic semantics, and also in the syntactic sense of Post. Our analysis has a potential for being generalized to other stochastic processes relevant to Bayesian modelling, yielding new understanding of these processes from the perspective of programming.

2012 ACM Subject Classification Theory of computation → Probabilistic computation

Keywords and phrases Beta-Bernoulli process, Algebraic effects, Probabilistic programming, Exchangeability

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.141

Related Version A full version of the paper is available at [27], `https://arxiv.org/abs/1802.09598`.

Acknowledgements It has been helpful to discuss this work with many people, including Ohad Kammar, Gordon Plotkin, Alex Simpson, and Marcin Szymczak. This work is partly supported by EPSRC grants EP/N509711/1 and EP/N007387/1, a Royal Society University Research Fellowship, and an Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00565, Development of Vulnerability Discovery Technologies for IoT Software Security).

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 141; pp. 141:1–141:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

From the perspective of programming, a family of Boolean random processes is implemented by a module that supports the following interface:

```
module type ProcessFactory = sig type process
                                 val new : H → process
                                 val get : process → bool end
```

where H is some type of hyperparameters. Thus one can initialize a new process, and then get a sequence of Booleans from that process. The type of processes is kept abstract so that any internal state or representation is hidden.

One can analyze a module extensionally in terms of the properties of its interactions with a client program. In this paper, we perform this analysis for the Beta-Bernoulli process, an important building block in Bayesian models. We completely axiomatize its equational properties, using the formal framework of algebraic effects [18].

The following modules are our leading examples. (Here flip (r) tosses a coin with bias r.)

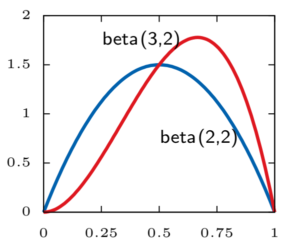
```
module Polya = (struct
  type process = (int * int) ref
  let new(i,j) = ref (i,j)
  let get p = let (i,j) = !p in
    if flip (i/(i+j)) then p := (i+1,j); true
    else p := (i,j+1); false   end : ProcessFactory)
```

```
module BetaBern = (struct
  type process = real
  let new(i,j) = sample_beta(i,j)
  let get(r) = flip(r)

end : ProcessFactory)
```

The left-hand module, Polya, is an implementation of Pólya's urn. An urn in this sense is a hidden state which contains i-many balls marked true and j-many balls marked false. To sample, we draw a ball from the urn at random; before we tell what we drew, we put back the ball we drew as well as an identical copy of it. The contents of the urn changes over time.

The right-hand module, BetaBern, is based on the beta distribution. This is the probability measure on the unit interval $[0, 1]$ that measures the bias of a random source (such as a potentially unfair coin) from which true has been observed $(i - 1)$ times and false has been observed $(j - 1)$ times, as illustrated on the right. For instance beta$(2, 2)$ describes the situation where we only know that neither true nor false are impossible; while in beta$(3, 2)$ we are still ignorant but we believe that true is more likely.



It turns out that these two modules have the same observable behaviour. This essentially follows from de Finetti's theorem (e.g. [24]), but rephrased in programming terms. The equivalence makes essential use of type abstraction: if we could look into the urn, or ask precise questions about the real number, the modules would be distinguishable.

The module Polya has a straightforward operational semantics (although we don't formalize that here). By contrast, BetaBern has a straightforward denotational semantics [14]. In Section 2, we provide an axiomatization of equality, which is sound by both accounts. We show completeness of our axiomatization with respect to the denotational semantics of BetaBern (§3, Thm. 9). We use this to show that the axiomatization is in fact syntactically complete (§4, Cor. 13), which means it is complete with respect to *any* semantics.

For the remainder of this section, we give a general introduction to our axioms.

Commutativity and discardability. Commutativity and discardability are important program equations [5] that are closely related, we argue, to exchangeability in statistics.

- *Commutativity* is the requirement that when x is not free in u and y is not free in t,

$$\Big(\textbf{let}\ \ \mathsf{x} = \mathsf{t}\ \textbf{in}\ \ \textbf{let}\ \ \mathsf{y} = \mathsf{u}\ \textbf{in}\ \mathsf{v}\Big) \quad = \quad \Big(\textbf{let}\ \ \mathsf{y} = \mathsf{u}\ \textbf{in}\ \ \textbf{let}\ \ \mathsf{x} = \mathsf{t}\ \textbf{in}\ \mathsf{v}\Big).$$

- *Discardability* is the requirement that when x is not free in u, $\Big(\textbf{let}\ \ \mathsf{x} = \mathsf{t}\ \textbf{in}\ \ \mathsf{u}\Big) = \Big(\mathsf{u}\Big)$.

Together, these properties say that data flow, rather than the control flow, is what matters. For example, in a standard programming language, the purely functional total expressions are commutative and discardable. By contrast, expressions that write to memory are typically not commutative or discardable (a simple example is t=u=a++, v=(x,y)). A simple example of a commutative and discardable operation is a coin toss: we can reorder the outcomes of tossing a single coin, and we can drop some of the results (unconditionally) without changing the overall statistics.

We contend that commutativity and discardability of program expressions is very close to the basic notion of exchangeability of infinite sequences, which is central to Bayesian statistics. Informally, an infinite random process, such as an infinite random sequence, is said to be exchangeable if one can reorder and discard draws without changing the overall statistics. (For more details on exchangeable random processes in probabilistic programming languages, see [1, 28], and the references therein.) A client program for the BetaBern module is clearly exchangeable in this sense: this is roughly Fubini's theorem. For the Polya module, an elementary calculation is needed: it is not trivial because memory is involved.

Conjugacy. Besides exchangeability, the following conjugacy equation is crucial:

$$\Big(\textbf{let}\ \ \mathsf{p{=}M.new(i,j)}\ \textbf{in}\ \ (\mathsf{M.get(p)},\ \mathsf{p})\Big)$$
$$= \Big(\textbf{if}\ \ \mathsf{flip}\ (\mathsf{i/(i{+}j)})\ \textbf{then}\ (\mathsf{true},\ \mathsf{M.new(i{+}1,j)})\ \textbf{else}\ \ (\mathsf{false}\ ,\ \ \mathsf{M.new(i,j{+}1)})\Big).$$

This is essentially the operational semantics of the Polya module, and from the perspective of BetaBern it is the well-known conjugate-prior relationship between the Beta and Bernoulli distributions.

Finite probability. In addition to exchangeability and conjugacy, we include the standard equations of finite, discrete, rational probability theory. To introduce these, suppose that we have a module

```
Bernoulli : sig val get : int * int → bool end
```

which is built so that Bernoulli .get(i,j) samples *with single replacement* from an urn with i-many balls marked true and j-many balls marked false. (In contrast to Pólya's urn, the urn in this simple scheme does not change over time.) So Bernoulli .get(i,j) = flip $(\frac{i}{i+j})$. This satisfies certain laws, first noticed long ago by Stone [29], and recalled in §2.1.

In summary, our main contribution is that these axioms — exchangeability, conjugacy, and finite probability — entirely determine the equational theory of the Beta-Bernoulli process, in the following sense:

- *Model completeness:* Every equation that holds in the measure theoretic interpretation is derivable from our axioms (Thm. 9);
- *Syntactical completeness:* Every equation that is not derivable from our axioms is inconsistent with finite discrete probability (Cor. 13).

We argue that these results open up a new method for analyzing Bayesian models, based on algebraic effects (see §5 and [28][1]).

2 An algebraic presentation of the Beta-Bernoulli process

In this section, we present syntactic rules for well-formed client programs of the Beta-Bernoulli module, and axioms for deriving equations on those programs.

2.1 An algebraic presentation of finite probability

Recall the module Bernoulli from the introduction which provides a method of sampling with odds $(i : j)$. We will axiomatize its equational properties. Algebraic effects provide a way to axiomatize the specific features of this module while putting aside the general properties of programming languages, such as β/η laws. In this situation the basic idea is that each module induces a binary operation ${}_i?_j$ on programs by

$$t \;{}_i?_j\; u \stackrel{\text{def}}{=} \textbf{if } \mathsf{Bernoulli}.\mathsf{get}(i,j) \textbf{ then } t \textbf{ else } u.$$

Conversely, given a family of binary operations ${}_i?_j$, we can recover $\mathsf{Bernoulli}.\mathsf{get}(i,j) = \mathsf{true}\;{}_i?_j\;\mathsf{false}$. So to give an equational presentation of the Bernoulli module we give a equational presentation of the binary operations ${}_i?_j$. A full programming language will have other constructs and $\beta\eta$-laws but it is routine to combine these with an algebraic theory of effects (e.g. [2, 8, 9, 21]).

▶ **Definition 1.** The *theory of rational convexity* is the first-order algebraic theory with binary operations ${}_i?_j$ for all $i, j \in \mathbb{N}$ such that $i + j > 0$, subject to the axiom schemes

$$w, x, y, z \vdash (w \;{}_i?_j\; x) \;{}_{i+j}?_{k+l}\; (y \;{}_k?_l\; z) = (w \;{}_i?_k\; y) \;{}_{i+k}?_{j+l}\; (x \;{}_j?_l\; z)$$
$$x, y \vdash x \;{}_i?_j\; y = y \;{}_j?_i\; x \qquad x, y \vdash x \;{}_i?_0\; y = x \qquad x \vdash x \;{}_i?_j\; x = x$$

Commutativity $(w \;{}_i?_j\; x) \;{}_k?_l\; (y \;{}_i?_j\; z) = (w \;{}_k?_l\; y) \;{}_i?_j\; (x \;{}_k?_l\; z)$ of operations ${}_k?_l$ and ${}_i?_j$ is a derivable equation, and so is scaling $x \;{}_{ki}?_{kj}\; y = x \;{}_i?_j\; y$ for $k > 0$. Commutativity and discardability $(x \;{}_i?_j\; x = x)$ in this algebraic sense (cf. [15, 22]) precisely correspond to the program equations in Section 1 (see also [9]). The theory first appeared in [29].

2.2 A parameterized algebraic signature for Beta-Bernoulli

In the theory of convex sets, the parameters i, j for get range over the integers. These integers are not a first class concept in our equational presentation: we did not axiomatize integer arithmetic. However, in the Beta-Bernoulli process, or any module M for the ProcessFactory interface, it is helpful to understand the parameters to get as abstract, and new as generating such parameters. To interpret this, we treat these parameters to get as first class. There are still hyperparameters to new, which we do not treat as first class here. (In a more complex hierarchical system with hyperpriors, we might treat them as first class.)

As before, to avoid studying an entire programming language, we look at the constructions

$$\nu_{i,j} p.t \stackrel{\text{def}}{=} \textbf{let } p{=}\mathsf{M}.\mathsf{new}(i,j) \textbf{ in } t \qquad t \;?_p\; u \stackrel{\text{def}}{=} \textbf{if } \mathsf{M}.\mathsf{get}(p) \textbf{ then } t \textbf{ else } u$$

[1] This paper formalizes and proves a conjecture from [28], which is an unpublished abstract.

There is nothing lost by doing this, because we can recover $\mathsf{M.new}(i,j) = \nu_{i,j}p.\,p$ and $\mathsf{M.get}(p) = \mathsf{true} \mathbin{?_p} \mathsf{false}$. In the terminology of [18], these would be called the 'generic effects' of the algebraic operations $\nu_{i,j}$ and $?_p$. Note that $?_p$ is a parameterized binary operation. Formally, our syntax now has two kinds of variables: x, y as before, ranging over continuations, and now also p, q ranging over parameters. We notate this by having contexts with two zones, and write $x : n$ if x expects n parameters.

▶ **Definition 2.** The term formation rules for the theory of Beta-Bernoulli are:

$$\frac{-}{\Gamma \,|\, \Delta, x : m, \Delta' \vdash x(p_1 \dots p_m)}\;(p_1 \dots p_m \in \Gamma) \qquad \frac{\Gamma, p \,|\, \Delta \vdash t}{\Gamma \,|\, \Delta \vdash \nu_{i,j}p.t}\;(i, j > 0)$$

$$\frac{\Gamma \,|\, \Delta \vdash t \quad \Gamma \,|\, \Delta \vdash u}{\Gamma \,|\, \Delta \vdash t \mathbin{?_p} u}\;(p \in \Gamma) \qquad \frac{\Gamma \,|\, \Delta \vdash t \quad \Gamma \,|\, \Delta \vdash u}{\Gamma \,|\, \Delta \vdash t \mathbin{{}_i?_j} u}\;(i + j > 0)$$

where Γ is a parameter context of the form $\Gamma = (p_1, \dots, p_\ell)$ and Δ is a context of the form $\Delta = (x_1 \colon m_1, \dots, x_k \colon m_k)$. Where $x\colon 0$, we often write x for $x()$. For the sake of a well-defined notion of dimension in 3.2.4, we disallow the formation of $\nu_{i,0}$ and $\nu_{0,i}$.

We work up-to α-conversion and substitution of terms for variables must avoid unintended capture of free parameters. For example, substituting $x \mathbin{?_p} y$ for w in $\nu_{1,1}p.w$ yields $\nu_{1,1}q.(x \mathbin{?_p} y)$, while substituting $x \mathbin{?_p} y$ for $z(p)$ in $\nu_{1,1}p.z(p)$ yields $\nu_{1,1}p.(x \mathbin{?_p} y)$.

2.3 Axioms for Beta-Bernoulli

The axioms for the Beta-Bernoulli theory comprise the axioms for rational convexity (Def. 1) together with the following axiom schemes.

Commutativity. All the operations commute with each other:

$$p, q \,|\, w, x, y, z : 0 \vdash (w \mathbin{?_q} x) \mathbin{?_p} (y \mathbin{?_q} z) = (w \mathbin{?_p} y) \mathbin{?_q} (x \mathbin{?_p} z) \tag{C1}$$

$$- \,|\, x \colon 2 \vdash \nu_{i,j}p.(\nu_{k,l}q.x(p,q)) = \nu_{k,l}q.(\nu_{i,j}p.x(p,q)) \tag{C2}$$

$$q \,|\, x, y \colon 1 \vdash \nu_{i,j}p.(x(p) \mathbin{?_q} y(p)) = (\nu_{i,j}p.x(p)) \mathbin{?_q} (\nu_{i,j}p.y(p)) \tag{C3}$$

$$- \,|\, x, y : 1 \vdash \nu_{i,j}p.(x(p) \mathbin{{}_k?_l} y(p)) = (\nu_{i,j}p.x(p)) \mathbin{{}_k?_l} (\nu_{i,j}p.y(p)) \tag{C4}$$

$$p \,|\, w, x, y, z : 0 \vdash (w \mathbin{{}_i?_j} x) \mathbin{?_p} (y \mathbin{{}_i?_j} z) = (w \mathbin{?_p} y) \mathbin{{}_i?_j} (x \mathbin{?_p} z) \tag{C5}$$

Discardability. All operations are idempotent:

$$- \,|\, x \colon 0 \vdash (\nu_{i,j}p.x) = x \qquad\qquad p \,|\, x \colon 0 \vdash x \mathbin{?_p} x = x \tag{D1–2}$$

Conjugacy.

$$- \,|\, x, y : 1 \vdash \nu_{i,j}p.(x(p) \mathbin{?_p} y(p)) = (\nu_{i+1,j}p.x(p)) \mathbin{{}_i?_j} (\nu_{i,j+1}p.y(p)) \tag{Conj}$$

A theory of equality for terms in context is built, as usual, by closing the axioms under substitution, congruence, reflexivity, symmetry and transitivity. It immediately follows from conjugacy and discardability that $x \mathbin{{}_i?_j} y$ is definable as $\nu_{i,j}p.(x \mathbin{?_p} y)$ for $i, j > 0$.

As an example, consider $t(r) = (r \mathbin{?_p} x) \mathbin{?_p} (y \mathbin{?_p} r)$ that represents tossing a coin with bias p twice, continuing with x or y if the results are different, or with r otherwise. One can show that $x \mathbin{{}_1?_1} y$ is a unique fixed point of t, i.e. $x \mathbin{{}_1?_1} y = t(x \mathbin{{}_1?_1} y)$; see the full paper [27] for detail. This is exactly von Neumann's trick [31] to simulate a fair coin toss with a biased one.

(For more details on the general axiomatic framework with parameters, see [25, 26], where it is applied to predicate logic, π-calculus, and other effects.)

3 A complete interpretation in measure theory

In this section we give an interpretation of terms using measures and integration operators, the standard formalism for probability theory (e.g. [19, 24]), and we show that this interpretation is complete (Thm. 9). Even if the reader is not interested in measure theory, they may still find value in the syntactical results of §4 which we prove using this completeness result.

By the Riesz–Markov–Kakutani representation theorem, there are two equivalent ways to view probabilistic programs: as probability kernels and as linear functionals. Both are useful.

Programs as probability kernels.

Forgetting about abstract types for a moment, terms in the BetaBern module are first-order probabilistic programs. So we have a standard denotational semantics due to [14] where terms are interpreted as probability kernels and ν as integration. Let $I = [0, 1]$ denote the unit interval. We write $\beta_{i,j}$ for the Beta(i, j)-distribution on I, which is given by the density function $p \mapsto \frac{1}{B(i,j)}p^{i-1}(1-p)^{j-1}$, where $B(i, j) = \frac{(i-1)!(j-1)!}{(i+j-1)!}$ is a normalizing constant.

For contexts of the form $\Gamma = (p_1, \dots, p_\ell)$ and $\Delta = (x_1 : m_1, \dots, x_k : m_k)$, we let $[\![\Delta]\!] \stackrel{\text{def}}{=} \sum_{i=1}^{k} I^{m_i}$ consist of a copy of I^{m_i} for every variable $x_i : m_i$. This has a σ-algebra $\Sigma([\![\Delta]\!])$ generated by the Borel sets. We interpret terms $\Gamma \,|\, \Delta \vdash t$ as probability kernels $[\![t]\!] : I^\ell \times \Sigma([\![\Delta]\!]) \to [0, 1]$ inductively, for $\vec{p} \in I^\ell$ and $U \in \Sigma([\![\Delta]\!])$:

$$[\![x_i(p_{j_1}, \dots, p_{j_m})]\!](\vec{p}, U) = 1 \text{ if } (i, p_{j_1} \dots p_{j_m}) \in U, \text{ 0 otherwise}$$

$$[\![u \;{}_{i}?_{j}\; v]\!](\vec{p}, U) = \tfrac{1}{i+j}\Big(i([\![u]\!](\vec{p}, U)) + j([\![v]\!](\vec{p}, U))\Big)$$

$$[\![u \;?_{p_j}\; v]\!](\vec{p}, U) = p_j([\![u]\!](\vec{p}, U)) + (1 - p_j)([\![v]\!](\vec{p}, U))$$

$$[\![\nu_{i,j} q.t]\!](\vec{p}, U) = \int_0^1 [\![t]\!]((\vec{p}, q), U)\, \beta_{i,j}(\mathrm{d}q) \qquad \Big[= \int_0^1 [\![t]\!]((\vec{p}, q), U)\ \tfrac{1}{B(i,j)} q^{i-1}(1-q)^{j-1}\ \mathrm{d}q \Big]$$

▶ **Proposition 3.** *The interpretation is sound: if $\Gamma \,|\, \Delta \vdash t = u$ is derivable then $[\![t]\!] = [\![u]\!]$ as probability kernels $[\![\Gamma]\!] \times \Sigma([\![\Delta]\!]) \to [0, 1]$.*

Proof notes. One must check that the axioms are sound under the interpretation. Each of the axioms are elementary facts about probability. For instance, commutativity (C2) amounts to Fubini's theorem, and the conjugacy axiom (Conj) is the well-known conjugate-prior relationship of Beta- and Bernoulli distributions. ◀

Interpretation as functionals

We write $\mathbb{R}^{I^m}$ for the vector space of continuous functions $I^m \to \mathbb{R}$, endowed with the supremum norm. Given a probability kernel $\kappa : I^\ell \times \Sigma\big(\sum_{j=1}^{k} I^{m_j}\big) \to [0, 1]$ and $\vec{p} \in I^\ell$, we define a linear map $\phi_{\vec{p}} : \mathbb{R}^{I^{m_1}} \times \dots \times \mathbb{R}^{I^{m_k}} \to \mathbb{R}$, by considering κ as an integration operator:

$$\phi_{\vec{p}}(f_1 \dots f_k) = \int f_j(r_1 \dots r_{m_j})\ \kappa(\vec{p}, \mathrm{d}(j, r_1 \dots r_{m_j}))$$

Here $\phi_{\vec{p}}$ are unital ($\phi(\vec{1}) = 1$) and positive ($\vec{f} \geq 0 \implies \phi(\vec{f}) \geq 0$).

When $\kappa = [\![t]\!]$, this $\phi_{\vec{p}}(\vec{f})$ is moreover continuous in $\vec{p}$, and hence a unital positive linear map $\phi : \mathbb{R}^{I^{m_1}} \times \dots \times \mathbb{R}^{I^{m_k}} \to \mathbb{R}^{I^\ell}$ [6, Thm. 5.1]. It is informative to spell out the interpretation of terms $p_1, \dots, p_\ell \,|\, x_1 : m_1, \dots, x_k : m_k \vdash t$ as maps $[\![t]\!] : \mathbb{R}^{I^{m_1}} \times \ldots \times \mathbb{R}^{I^{m_k}} \to \mathbb{R}^{I^\ell}$ since it fits the algebraic notation: we may think of the variables $x\colon m$ as ranging over functions $\mathbb{R}^{I^m}$.

▶ **Proposition 4.** *The functional interpretation is inductively given by*

$$[\![x_i(p_{j_1},\dots,p_{j_m})]\!](\vec{f})(\vec{p}) = f_i(p_{j_1},\dots,p_{j_m})$$
$$[\![u\ {}_i?_j\ v]\!](\vec{f})(\vec{p}) = \tfrac{1}{i+j}\Big(i([\![u]\!](\vec{f})(\vec{p})) + j([\![v]\!](\vec{f})(\vec{p}))\Big)$$
$$[\![u\ ?_{p_j}\ v]\!](\vec{f})(\vec{p}) = p_j([\![u]\!](\vec{f})(\vec{p})) + (1-p_j)([\![v]\!](\vec{f})(\vec{p}))$$
$$[\![\nu_{i,j}q.t]\!](\vec{f})(\vec{p}) = \int_0^1 [\![t]\!](\vec{f})(\vec{p},q)\,\beta_{i,j}(\mathrm{d}q)$$

For example, $[\![-\,|\,x,y\colon 0 \vdash x\ {}_1?_1\ y]\!] : \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ is the function $(x,y)\mapsto \frac{1}{2}(x+y)$, and $[\![-\,|\,x\colon 1 \vdash \nu_{1,1}p.x(p)]\!] : \mathbb{R}^I \to \mathbb{R}$ is the integration functional, $f \mapsto \int_0^1 f(p)\,\mathrm{d}p$.

(We use the same brackets $[\![-]\!]$ for both the measure-theoretic and the functional interpretations; the intended semantics will be clear from context.)

3.1 Technical background on Bernstein polynomials

▶ **Definition 5** (Bernstein polynomials)**.** For $i = 0,\dots,k$, we define the i-th basis Bernstein polynomial $b_{i,k}$ of degree k as $b_{i,k}(p) = \binom{k}{i}p^{k-i}(1-p)^i$. For a multi-index $I = (i_1,\dots,i_\ell)$ with $0 \le i_j \le k$, we let $b_{I,k}(\vec{p}) = b_{i_1,k}(p_1)\cdots b_{i_\ell,k}(p_\ell)$. A Bernstein polynomial is a linear combination of Bernstein basis polynomials.

The family $\{b_{i,k} : i = 0,\dots,k\}$ is indeed a basis of the polynomials of maximum degree k and also a partition of unity, i.e. $\sum_{i=0}^k b_{i,k} = 1$. Every Bernstein basis polynomial of degree k can be expressed as a nonnegative rational linear combination of degree $k+1$ basis polynomials.

The density function of the distribution $\beta_{i,j}$ on $[0,1]$ for $i,j > 0$ is proportional to a Bernstein basis polynomial of degree $i+j-2$. We can conclude that the measures $\{\beta_{i,j} : i,j > 0, i+j = n\}$ are linearly independent for every n. In higher dimensions, the polynomials $\{b_{I,k}\}$ are linearly independent for every k. Moreover, products of beta distributions β_{i_r,j_r} are linearly independent as long as $i_r + j_r = n$ holds for some common n. This will be a key idea for normalizing Beta-Bernoulli terms.

3.2 Normal forms and completeness

For the completeness proof of the measure-theoretic model, we proceed as follows: To decide $\Gamma\,|\,\Delta \vdash t = u$ for two terms t, u, we transform them into a common normal form whose interpretations can be given explicitly. We then use a series of linear independence results to show that if the interpretations agree, the normal forms are already syntactically equal. Normalization happens in three stages.

- If we think of a term as a syntax tree of binary choices and ν-binders, we use the conjugacy axiom to push all occurrences of ν towards the leaves of the tree.
- We use commutativity and discardability to stratify the use of free parameters $?_p$.
- The leaves of the tree will now consist of chains of ν-binders, variables and ratio choices ${}_i?_j$. Those can be collected into a canonical form.

We will describe these normalization stages in reverse order because of their increasing complexity.

3.2.1 Stone's normal forms for rational convex sets

Normal forms for the theory of rational convex sets have been described by Stone [29]. We note that if $-\,|\,x_1\dots x_k : 0 \vdash t$ is a term in the theory of rational convex sets (Def. 1) then

$[\![t]\!] : \mathbb{R}^k \to \mathbb{R}$ is a unital positive linear map that takes rationals to rationals. From the perspective of measures, this corresponds to a categorical distribution with k categories.

▶ **Proposition 6** (Stone). *The interpretation exhibits a bijective correspondence between terms $- \,|\, x_1 \dots x_k : 0 \vdash t$ built from ${}_i?_j$, modulo equations, and unital positive linear maps $\mathbb{R}^k \to \mathbb{R}$ that take rationals to rationals.*

For instance, the map $\phi(x, y, z) = \frac{1}{10}(2x + 3y + 5z)$ is unital positive linear, and arises from the term $t \stackrel{\text{def}}{=} x\, {}_2?_8 (y\, {}_3?_5\, z)$. This is the only term that gives rise to the ϕ, modulo equations. In brief, one can recover t from ϕ by looking at $\phi(1, 0, 0) = \frac{2}{10}$, then $\phi(0, 1, 0) = \frac{3}{10}$, then $\phi(0, 0, 1) = \frac{5}{10}$. We will write $\begin{pmatrix} ? & x_1 & \dots & x_k \\ & w_1 & \dots & w_k \end{pmatrix}$ for the term corresponding to the linear map $(x_1 \dots x_k) \mapsto \frac{1}{\sum_{i=1}^k w_k}(w_1 x_1 + \dots + w_k x_k)$. These are normal forms for the theory of rational convex sets.

3.2.2 Characterization and completeness for ν-free terms

This section concerns the normalization of terms using free parameters but no ν. Consider a single parameter p. If we think of a term t as a syntactic tree, commutativity and discardability can be used to move all occurrences of $?_p$ to the root of the tree, making it a *tree diagram* of some depth k. Let us label the 2^k leaves with $t_{a_1 \cdots a_k}$, $a_i \in \{0, 1\}$. As a programming language expression, this corresponds to successive bindings

let a_1=M.get(p) **in** ... **let** a_k=M.get(p) **in** $t_{a_1 \cdots a_k}$

Permutations $\sigma \in S_k$ of the k first levels in the tree act on tree diagrams by permuting the leaves via $t_{a_1 \cdots a_k} \mapsto t_{a_{\sigma(1)} \cdots a_{\sigma(k)}}$. By commutativity (C1), those permuted diagrams are still equal to t, so we can replace t by the average over all permuted diagrams, since rational choice is discardable. The average commutes down to the leaves (C5), so we obtain a tree diagram with leaves $m_{a_1 \cdots a_k} = \frac{1}{k!} \sum_\sigma t_{a_{\sigma(1)} \cdots a_{\sigma(k)}}$, where the average is to be read as a rational choice with all weights 1. This new tree diagram is now by construction invariant under permutation of levels in the tree, in particular $m_{a_1 \cdots a_k}$ only depends on the sum $a_1 + \dots + a_k$. That is to say, the counts are a sufficient statistic.

This leads to the following normalization procedure for terms $p_1 \dots p_\ell \,|\, x_1 \dots x_n : 0 \vdash t$: Write $C_k^{p_j}(t_0, \dots, t_k)$ for the permutation invariant tree diagram of p_j-choices and depth k with leaves $t_{a_1 \cdots a_k} = t_{a_1 + \cdots + a_k}$. Then we can rewrite t as $C_k^{p_1}(t_0, \dots, t_k)$ where each t_i is p_1-free. Recursively normalize each t_i in the same way, collecting the next parameter. By discardability, we can pick the height of all these tree diagrams to be a single constant k, such that the resulting term is a nested structure of tree-diagrams $C_k^{p_j}$. We will use multi-indices $I = (i_1, \dots, i_\ell)$ to write the whole stratified term as $C_k((t_I))$ where each leaf t_I only contains rational choices. The interpretation of such a term can be given explicitly by Bernstein polynomials

$$[\![C_k((t_I))]\!](\vec{x})(\vec{p}) = \textstyle\sum_I b_{I,k}(\vec{p}) \cdot [\![t_I]\!](\vec{x})(\vec{p}).$$

For example, normalizing $(v\, ?_p\, x) ?_p (y\, ?_p\, v)$ gives $(v\, ?_p (x\, {}_1?_1\, y)) ?_p ((x\, {}_1?_1\, y)\, ?_p\, v) = C_2(v, x\, {}_1?_1 y, v)$.

From this we obtain the following completeness result:

▶ **Proposition 7.** *There is a bijective correspondence between equivalence classes of terms $p_1 \dots p_\ell \,|\, x_1 \dots x_n : 0 \vdash t$ and linear unital maps $\phi : \mathbb{R}^n \to \mathbb{R}^{I^\ell}$ such that for every standard basis vector e_j of $\mathbb{R}^n$, $\phi(e_j)$ is a Bernstein polynomial with nonnegative rational coefficients.*

Proof. We can assume all basis polynomials to have the same degree k. If $\phi(e_j) = \sum_I w_{Ij} b_{I,k}$, then the unitality condition $\phi(1,\ldots,1) = 1$ means $\sum_I \left(\sum_j w_{Ij}\right) b_{I,k} = 1$, and hence by linear independence and partition of unity, $\sum_j w_{Ij} = 1$ for every I. If we thus let t_I be the rational convex combination of the x_j with weights w_{Ij}, then $[\![C_k((t_I))]\!] = \phi$. Again by linear independence, the weights w_{IJ} are uniquely defined by ϕ. ◀

Geometric characterizations for the assumption of this theorem exist in [20, 3]. For example, a univariate polynomial is a Bernstein polynomial with nonnegative coefficients if and only if it is positive on $(0,1)$. More care is required in the multivariate case.

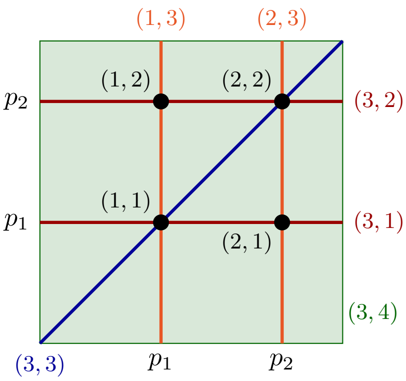
3.2.3 Normalization of Beta-Bernoulli

For arbitrary terms $p_1 \ldots p_\ell \,|\, x_1 : m_1, \ldots, x_s : m_s \vdash t$, we employ the following normalization procedure. Using conjugacy and the commutativity axioms (C2–C4), we can push all uses of ν towards the leaves of the tree, until we end up with a tree of ratios and free parameter choices only. Next, by conjugacy and discardability, we expand every instance of $\nu_{i,j}$ until they satisfy $i + j = n$ for some fixed, sufficiently large n. We then stratify the free parameters into permutation invariant tree diagrams. That is, we find a number k such that t can be written as $C_k((t_I))$ where the leaves t_I consist of ν and rational choices only.

In each t_I, commuting all the choices up to the root, we are left with a convex combination of chains of ν's of the form $\nu_{i_1,j_1} p_{\ell+1}.\ldots.\nu_{i_d,j_d} p_{\ell+d}.x_j(p_{\tau(1)}, \ldots, p_{\tau(m)})$ for some $\tau : m \to \ell+d$. By discardability, we can assume that there are no unused bound parameters. We consider two chains equal if they are α-convertible into each other. Now if $c_1, \ldots, c_m$ is a list of the distinct chains that occur in any of the leaves, we can give the leaves t_I the uniform shape $t_I = \left(? \begin{matrix} c_1 & \ldots & c_m \\ w_{I1} & \ldots & w_{Im} \end{matrix}\right)$ for appropriate weights $w_{Ij} \in \mathbb{N}$. We will show that this representation is a unique normal form.

3.2.4 Proof of completeness

Consider a *chain* $c = \nu_{i_1,j_1} p_{\ell+1}.\ldots.\nu_{i_d,j_d} p_{\ell+d}.\, x(p_{\tau(1)}, \ldots, p_{\tau(m)})$. Its measure-theoretic interpretation $[\![c]\!](p_1, \ldots, p_\ell)$ is a pushforward of a product of d beta distributions, supported on a hyperplane segment that is parameterized by the map $h_\tau : I^d \to I^m, h_\tau(p_{\ell+1}, \ldots, p_{\ell+d}) = (p_{\tau(1)}, \ldots, p_{\tau(m)})$. Note that the position of the hyperplane may vary with the free parameters. To capture this geometric information, we call τ the *subspace type* of the chain and d its *dimension*. Because of α-invariance of chains, we identify subspace types that differ by a permutation of $\{\ell+1, \ldots, \ell+d\}$.



For example, each chain with two free parameters p_1, p_2 and a variable $x : 2$ gives rise to a parameterized distribution on the unit square. On the right, we illustrate the ten possible supports that such distributions can have, as subspaces of the square. In the graphic we write (i,j) for $\nu p_3.\nu p_4.x(p_i, p_j)$, momentarily omitting the subscripts of ν because they do not affect the support. For instance, the upper horizontal line corresponds to $\nu p_3.x(p_3, p_2)$; the bottom-right dot corresponds to $x(p_2, p_1)$; the diagonal corresponds to $\nu p_3.x(p_3, p_3)$; and the entire square corresponds to

$\nu p_3.\nu p_4.x(p_3, p_4)$. All told there are four subspaces of dimension $d = 0$, five with $d = 1$, and one with $d = 2$. Notice that the subspaces are all distinct as long as $p_1 \neq p_2$.

▶ **Proposition 8.** *If $c_1, \ldots, c_s$ are distinct chains with $i_1 + j_1 = \cdots = i_d + j_d = n$, then the family of functionals $\{[\![c_i]\!](-)(\vec{p}) : \mathbb{R}^{I^{m_1}} \times \cdots \times \mathbb{R}^{I^{m_s}} \to \mathbb{R}\}_{i=1,\ldots,s}$ is linearly independent whenever all parameters p_i are distinct.*

Proof. Fix $\vec{p}$. Chains on different variables are clearly independent, so we can restrict ourselves to a single variable $x : m$. We reason measure-theoretically. The interpretation of a chain c_i of subspace type τ_i is a pushforward measure $h_{i*}(\mu_i)$ where μ_i is a product of d beta distributions, and h_i is the affine inclusion map $h_i(p_{\ell+1}, \ldots, p_{\ell+d}) = (p_{\tau_i(1)}, \ldots, p_{\tau_i(m)})$. Let $\sum a_i h_{i*}(\mu_i) = 0$ as a signed measure. We show by induction over the dimension of the chains that all a_i vanish. Assume that $a_i = 0$ whenever the dimension of c_i is less than d, and consider an arbitrary subspace τ_j of dimension d. We can define a signed Borel measure on I^d by restriction

$$\rho(A) \stackrel{\text{def}}{=} \sum_i a_i h_{i*}(\mu_i)(h_j(A)) = \sum_i a_i \mu_i(h_i^{-1}(h_j(A)))$$

as h_j sends Borel sets to Borel sets (e.g. [10, §15A]). We claim that $\rho(A) = \sum_{c_i \text{ has type } \tau_j} a_i \mu_i(A)$, as the contributions of chains c_i of different type vanish.

- If c_i has dimension $< d$, $a_i = 0$ by the inductive hypothesis.
- If c_i has dimension $> d$, we note that $h_i^{-1}(h_j(A))$ only has at most dimension d. It is therefore a nullset for μ_i.
- If c_i has dimension d but a different type, and all $p_1, \ldots, p_\ell$ are assumed distinct, then the hyperplanes given by h_i and h_j are not identical. Therefore their intersection is at most $(d-1)$-dimensional and $h_i^{-1}(h_j(A))$ is a nullset for μ_i.

By assumption, ρ has to be the zero measure, but the μ_i are linearly independent. Therefore $a_i = 0$ for all c_i with subspace type τ_j. Repeat this for every subspace type of dimension d to conclude overall linear independence. ◀

▶ **Theorem 9** (Completeness). *If $\Gamma \,|\, \Delta \vdash t, t'$ and $[\![t]\!] = [\![t']\!]$, then $\Gamma \,|\, \Delta \vdash t = t'$.*

Proof. From the normalization procedure, we find numbers k, n, a list of distinct chains $c_1, \ldots, c_s$ with $i + j = n$ and weights $(w_{Ij}), (w'_{Ij})$ such that $\Gamma \,|\, \Delta \vdash t = C_k((t_I))$ and $\Gamma \,|\, \Delta \vdash t' = C_k((t'_I))$ where $t_I = \begin{pmatrix} ? & c_1 & \ldots & c_s \\ & w_{I1} & \ldots & w_{Is} \end{pmatrix}$ and $t'_I = \begin{pmatrix} ? & c_1 & \ldots & c_s \\ & w'_{I1} & \ldots & w'_{Is} \end{pmatrix}$. The interpretations of these normal forms are given explicitly by

$$[\![t]\!](\vec{f})(\vec{p}) = \sum_j \frac{w_{Ij}}{w_I} \cdot b_{I,k}(\vec{p}) \cdot [\![c_j]\!](\vec{f})(\vec{p}) \text{ where } w_I = \sum_j w_{Ij}$$

and analogously for t'. Then $[\![t]\!] = [\![t']\!]$ implies that for all $\vec{f}$

$$\sum_j \left(\sum_I \left(\frac{w_{Ij}}{w_I} - \frac{w'_{Ij}}{w'_I} \right) b_{I,k}(\vec{p}) \right) [\![c_j]\!](\vec{f})(\vec{p}) = 0.$$

By Proposition 8, this implies $\sum_I \left(\frac{w_{Ij}}{w_I} - \frac{w'_{Ij}}{w'_I} \right) b_{I,k}(\vec{p}) = 0$ for every j and whenever the parameters p_i are distinct. By continuity of the left hand side, the expression in fact has to vanish for *all* $\vec{p}$. By linear independence of the Bernstein polynomials, we obtain $w_{Ij}/w_I = w'_{Ij}/w'_I$ for all I, j. Thus, all weights agree up to rescaling and we can conclude $\Gamma \,|\, \Delta \vdash t = t'$. ◀

4 Extensionality and syntactical completeness

In this section we use the model completeness of the previous section to establish some syntactical results about the theory of Beta-Bernoulli. Although the model is helpful in informing the proofs, the statements of the results in this section are purely syntactical.

The ultimate result of this section is equational syntactical completeness (Cor. 13), which says that there can be no further equations in the theory without it becoming inconsistent with discrete probability. In other words, assuming that the axioms we have included are appropriate, they must be sufficient, regardless of any discussion about semantic models or intended meaning. This kind of result is sometimes called 'Post completeness' after Post proved a similar result for propositional logic.

The key steps towards this result are two extensionality results. These are related to the programming language idea of 'contextual equivalence'. Recall that in a programming language we often define a basic notion of equivalence on closed ground terms: these are programs with no free variables that return (say) booleans. This notion is often defined by some operational consideration using some notions of observation. From this we define contextual equivalence by saying that $t \approx u$ if, for all closed ground contexts $\mathcal{C}$, $\mathcal{C}[t] = \mathcal{C}[u]$.

Contextual equivalence has a canonical appearance, but an axiomatic theory of equality, such as the one in this paper, is more compositional and easier to work with. Our notion of equality induces in particular a basic notion of equivalence on closed ground terms. Our extensionality results say that, assuming one is content with this basic notion of equivalence, the equations that we axiomatize coincide with contextual equivalence.

4.1 Extensionality

▶ **Proposition 10** (Extensionality for closed terms). *Suppose* $\Gamma, q \,|\, \Delta \vdash t$ *and* $\Gamma, q \,|\, \Delta \vdash u$*. If* $\Gamma \,|\, \Delta \vdash \nu_{i,j} q.t = \nu_{i,j} q.u$ *for all* i, j*, then also* $\Gamma \,|\, \Delta \vdash t = u$*.*

Proof. We show the contrapositive. By the model completeness theorem (Thm. 9), we can reason in the model rather than syntactically. So we consider t and u such that $[\![t]\!] \neq [\![u]\!]$ as functions $\mathbb{R}^{I^{m_1}} \times \mathbb{R}^{I^{m_k}} \to \mathbb{R}^{I^{l+1}}$, and show that there are i, j such that $[\![\nu_{i,j} q.t]\!] \neq [\![\nu_{i,j} q.u]\!]$. By assumption there are $\vec{f}$ and $\vec{p}, q$ such that $[\![t]\!](\vec{f})(\vec{p}, q) \neq [\![u]\!](\vec{f})(\vec{p}, q)$ as real numbers.

Now we use the following general reasoning: For any real $q \in I$ we can pick monotone sequences $i_1 < \cdots < i_n < \ldots$ and $j_1 < \cdots < j_n < \ldots$ of natural numbers so that $\frac{i_n}{i_n + j_n} \to q$ as $n \to \infty$. Moreover, for any continuous $h : I \to \mathbb{R}$, the integral $\int h \; \mathrm{d}\beta_{i_n, j_n}$ converges to $h(q)$ as $n \to \infty$: one way to see this is to notice that the variance of β_{i_n, j_n} vanishes as $n \to \infty$, so by Chebyshev's inequality, $\lim_n \beta_{i_n, j_n}$ is a Dirac distribution at q. Thus, $\int \big([\![t]\!](\vec{f})(\vec{p}, r) - [\![u]\!](\vec{f})(\vec{p}, r)\big) \; \beta_{i_n, j_n}(\mathrm{d}r)$ is non-zero as $n \to \infty$. By continuity, for some n, $\int [\![t]\!](\vec{f})(\vec{p}, r) \; \beta_{i_n, j_n}(\mathrm{d}r) \neq \int [\![u]\!](\vec{f})(\vec{p}, r) \; \beta_{i_n, j_n}(\mathrm{d}r)$. So, $[\![\nu_{i_n, j_n} q.t]\!] \neq [\![\nu_{i_n, j_n} q.u]\!]$. ◀

▶ **Proposition 11** (Extensionality for ground terms). In brief: *If* $t[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}] = u[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}]$ *for all suitable ground* $v_1 \ldots v_k$*, then* $t = u$*.*

In detail: *Consider* t *and* u *with* $- \,|\, x_1 \colon m_1 \ldots x_k \colon m_k \vdash t, u$*. Suppose that whenever* $v_1 \ldots v_k$ *are terms with* $(p_1 \ldots p_{m_1} \,|\, y, z : 0 \vdash v_1)$, $\ldots$, $(p_1 \ldots p_{m_k} \,|\, y, z : 0 \vdash v_k)$*, then we have* $- \,|\, y, z : 0 \vdash t[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}] = u[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}]$*. Then we also have* $- \,|\, x_1 \colon m_1 \ldots x_k \colon m_k \vdash t = u$*.*

Proof. Again, we show the contrapositive. Let $\Delta = (x_1 \colon m_1 \ldots x_k \colon m_k)$. Suppose we have t and u such that $\neg(- \,|\, \Delta \vdash t = u)$. Then by the model completeness theorem (Thm. 9), we have $[\![t]\!] \neq [\![u]\!]$ as linear functions $\mathbb{R}^{I^{m_1}} \times \cdots \times \mathbb{R}^{I^{m_k}} \to \mathbb{R}$. Since the functions are linear, there is an index $i \leq k$ and a continuous function $f : I^{m_i} \to \mathbb{R}$ with $[\![t]\!](0 \ldots 0, f, 0 \ldots 0) \neq [\![u]\!](0 \ldots 0, f, 0 \ldots 0)$. By the Stone-Weierstrass theorem, every such f

is a limit of polynomials, and so since $[\![t]\!]$ and $[\![u]\!]$ are continuous and linear, there has to be a Bernstein basis polynomial $b_{I,k} : \mathbb{R}^{m_i} \to \mathbb{R}$ that already distinguishes them. This function is definable, i.e. there is a a term $p_1, \ldots, p_{m_i} \mid y, z : 0 \vdash w$ with $[\![w]\!](1, 0) = b_{I,k}$. Define terms $v_j = w$ for $i = j$ and $v_j = z$ for $i \neq j$. Then

$$[\![t[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}]]\!](1, 0) = [\![t]\!](0, \ldots, b_{I,k}, \ldots, 0) \neq [\![u]\!](0, \ldots, b_{I,k}, \ldots, 0) = [\![u[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}]]\!](1, 0).$$

The required $\neg\big(- \mid y, z : 0 \vdash t[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}] = u[^{v_1 \ldots v_k}/_{x_1 \ldots x_k}]\big)$ follows from the above disequality because of the model soundness property (Props. 3 and 4). ◀

From the programming perspective, a term $- \mid y, z : 0 \vdash t_0$ corresponds to a closed program of type bool, for it has two possible continuations, y and z, depending on whether the outcome is true or false. From this perspective, Proposition 11 says that for closed t, u, if $\mathcal{C}[t] = \mathcal{C}[u]$ for all boolean contexts $\mathcal{C}$, then $t = u$.

4.2 Relative syntactical completeness

▶ **Proposition 12** (Neumann, [17]). *If t, u are terms in the theory of rational convexity (Def. 1), then either $t = u$ is derivable or it implies $x\ {}_i?_j\ y = x\ {}_{i'}?_{j'}\ y$ for all nonzero i, i', j, j'.*

▶ **Corollary 13.** *The theory of Beta-Bernoulli is syntactically complete relative to the theory of rational convexity, in the following sense. For all terms t and u, either $t = u$ is derivable, or it implies $x\ {}_i?_j\ y = x\ {}_{i'}?_{j'}\ y$ for all nonzero i, i', j, j'.*

This is proved by combining Propositions 10, 11 and 12. As an example for extensionality and completeness, consider the equation $\nu_{1,1}p.x(p, p) = \nu_{1,1}p.(\nu_{1,1}q.x(p, q))$. It is not derivable, as can be witnessed by the substitution $x(p, q) = (y\ ?_q\ z)\ ?_p\ z$. Normalizing yields $y\ {}_1?_2\ z = y\ {}_1?_3\ z$ which is incompatible with discrete probability (see the full paper [27]). In programming syntax, the candidate equation is written

LHS = **let** p = M.new(1,1) **in** (p,p)	RHS = (M.new(1,1) , M.new(1,1))

and the distinguishing context is C[−] = **let** (p,q)=(−) **in if** M.get(p) **then** M.get(q) **else** false. That is to say, the closed ground programs C[LHS] and C[RHS] necessarily have different observable statistics: this follows from the axioms.

4.3 Remark about stateful implementations

In the introduction we recalled the idea of using Pólya's urn to implement a Beta-Bernoulli process using local (hidden) state.

Our equational presentation gives a recipe for understanding the correctness of the stateful implementation. First, one would give an operational semantics, and then a basic notion of observational equivalence on closed ground terms in terms of the finite probabilities associated with reaching certain ground values. From this, an operational notion of contextual equivalence can be defined (e.g. [4, §6], [23, 32]). Then, one would show that the axioms of our theory hold up-to contextual equivalence. Finally one can deduce from the syntactical completeness result that the equations satisfied by this stateful implementation must be exactly the equations satisfied by the semantic model.

In fact, in this argument, it is not necessary to check that axioms (C1) and (D2) hold in the operationally defined contextual equivalence, because the axiomatized equality on closed ground terms is independent of these axioms. To see this, notice that our normalization procedure (§3.2.3) doesn't use (C1) or (D2) when the terms are closed and ground, since then we can take $n = k = 0$. This is helpful because the remaining axioms are fairly straightforward, e.g. (Conj) is the essence of the urn scheme and (D1) is garbage collection.

5 Conclusion

Exchangeable random processes are central to many Bayesian models. The general message of this paper is that the analysis of exchangeable random processes, based on basic concepts from programming language theory, depends on three crucial ingredients: commutativity, discardability, and abstract types. We have illustrated this message by showing that just adding the conjugacy law to these ingredients leads to a complete equational theory for the Beta-Bernoulli process (Thm. 9). Moreover, we have shown that this equational theory has a canonical syntactic and axiomatic status, regardless of the measure theoretic foundation (Cor. 13). Our results in this paper open up the following avenues of research.

Study of nonparametric Bayesian models: We contend that abstract types, commutativity and discardability are fundamental tools for studying nonparametric Bayesian models, especially hierarchical ones. For example, the Chinese Restaurant Franchise [30] can be implemented as a module with three abstract types, f (franchise), r (restaurant), t (table), and functions $\mathsf{newFranchise}{:}() \to \mathsf{f}$, $\mathsf{newRestaurant}{:}\mathsf{f} \to \mathsf{r}$, $\mathsf{getTable}{:}\,\mathsf{r} \to \mathsf{t}$, $\mathsf{sameDish}{:}\mathsf{t}{*}\mathsf{t} \to \mathsf{bool}$. Its various exchangeability properties correspond to commutativity/discardability in the presence of type abstraction. (For other examples, see [28].)

First steps in synthetic probability theory: As is well known, the theory of rational convex sets corresponds to the monad D of rational discrete probability distributions. Commutativity of the theory amounts to commutativity of the monad D [15, 12].

As any parameterized algebraic theory, the theory of Beta-Bernoulli (§2) can be understood as a monad P on the functor category $[\mathbf{FinSet}, \mathbf{Set}]$, with the property that to give a natural transformation $\mathbf{FinSet}(\ell, -) \to P(\coprod_{j=1}^{k} \mathbf{FinSet}(m_k, -))$ is to give a term $(p_1 \dots p_\ell \,|\, x_1 \colon m_1 \dots x_k \colon m_k \vdash t)$, and monadic bind is substitution ([25, Cor. 1], [26, §VIIA]). This can be thought of as an intuitionistic set theory with an interesting notion of probability. As such this is a 'commutative effectus' [7], a synthetic probability theory (see also [13]). Like D, the global elements $1 \to P(2)$ are the rationals in $[0, 1]$ (by Prop. 7) but unlike D, the global elements $1 \to P(P(2))$ include the beta distribution.

Practical ideas for nonparametric Bayesian models in probabilistic programming: A more practical motivation for our work is to inform the design of module systems for probabilistic programming languages. For example, Anglican, Church, Hansei and Venture already support nonparametric Bayesian primitives [11, 33, 16]. We contend that abstract types are a crucial concept from the perspective of exchangeability.

References

1 Nathanael L. Ackerman, Cameron E. Freer, and Daniel M. Roy. Exchangeable random primitives. Workshop on Probabilistic Programming Semantics (PPS 2016), 2016. URL: `http://pps2016.soic.indiana.edu/files/2015/12/xrp.pdf`.

2 Danel Ahman and Sam Staton. Normalization by evaluation and algebraic effects. In *Proc. MFPS 2013*, volume 298 of *Electron. Notes Theor. Comput. Sci*, pages 51–69, 2013.

3 M. Alves Diniz, L. E. Salasar, and R. Bassi Stern. Positive polynomials on closed boxes. *arXiv e-print 1610.01437*, 2016.

4 Ales Bizjak and Lars Birkedal. Step-indexed logical relations for probability. In *Proc. FOSSACS 2015*, pages 279–294, 2015.

5 Carsten Führmann. Varieties of effects. In *Proc. FOSSACS 2002*, pages 144–159, 2002.

6 Robert Furber and Bart Jacobs. From Kleisli categories to commutative C*-algebras: Probabilistic Gelfand duality. *Log. Methods Comput. Sci.*, 11(2), 2015. doi:10.2168/LMCS-11(2:5)2015.

7 Bart Jacobs. From probability monads to commutative effectuses. *J. Log. Algebr. Methods Program.*, 94:200–237, 2018.

8 Patricia Johann, Alex Simpson, and Janis Voigtländer. A generic operational metatheory for algebraic effects. In *Proc. LICS 2010*, pages 209–218, 2010.

9 Ohad Kammar and Gordon D. Plotkin. Algebraic foundations for effect-dependent optimisations. In *Proc. POPL 2012*, pages 349–360, 2012.

10 Alexander Kechris. *Classical Descriptive Set Theory.* Springer, 1995.

11 Oleg Kiselyov and Chung-Chieh Shan. Probabilistic programming using first-class stores and first-class continuations. In *Proc. 2010 ACM SIGPLAN Workshop on ML*, 2010.

12 Anders Kock. Monads on symmetric monoidal closed categories. *Arch. Math.*, 21:1–10, 1970.

13 Anders Kock. Commutative monads as a theory of distributions. *Theory Appl. Categ.*, 26(4):97–131, 2012.

14 Dexter Kozen. Semantics of probabilistic programs. *J. Comput. System Sci.*, 22(3):328–350, 1981.

15 F. E. J. Linton. Autonomous equational categories. *J. Math. Mech.*, 15:637–642, 1966.

16 Vikash Mansinghka, Daniel Selsam, and Yura Perov. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv e-print 1404.0099*, 2014.

17 Walter D. Neumann. On the quasivariety of convex subsets of affine space. *Arch. Math.*, 21:11–16, 1970.

18 Gordon Plotkin and John Power. Algebraic operations and generic effects. *Appl. Categ. Structures*, 11(1):69–94, 2003.

19 David Pollard. *A user's guide to measure theoretic probability.* Cambridge University Press, 2001.

20 Victoria Powers and Bruce Reznick. Polynomials that are positive on an interval. *Trans. Amer. Math. Soc.*, 352(10):4677–4692, 2000.

21 Matija Pretnar. *The Logic and Handling of Algebraic Effects.* PhD thesis, Univ. Edinburgh, 2010.

22 Anna B. Romanowska and Jonathan D. H. Smith. *Modes.* World Scientific, 2002.

23 Davide Sangiorgi and Valeria Vignudelli. Environmental bisimulations for probabilistic higher-order languages. In *Proc. POPL 2016*, 2016.

24 Mark J. Schervish. *Theory of statistics.* Springer, 1995.

25 Sam Staton. An algebraic presentation of predicate logic. In *Proc. FOSSACS 2013*, pages 401–417, 2013.

26 Sam Staton. Instances of computational effects: An algebraic perspective. In *Proc. LICS 2013*, 2013.

27 Sam Staton, Dario Stein, Hongseok Yang, Nathanael L. Ackerman, Cameron E. Freer, and Daniel M. Roy. The Beta-Bernoulli process and algebraic effects. *arXiv e-print 1802.09598*, 2018.

28 Sam Staton, Hongseok Yang, Nathanael Ackerman, Cameron Freer, and Daniel M. Roy. Exchangeable random processes and data abstraction. Workshop on Probabilistic Programming Semantics (PPS 2017), 2017. URL: https://pps2017.soic.indiana.edu/files/2017/01/staton-yang-ackerman-freer-roy.pdf.

29 M. H. Stone. Postulates for the barycentric calculus. *Ann. Mat. Pura Appl. (4)*, 29:25–30, 1949.

30 Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101(476):1566–1581, 2006.

31 John von Neumann. Various techniques used in connection with random digits. *Nat. Bur. Stand. Appl. Math. Series*, 12:36–38, 1951.

32 Mitchell Wand, Theophilos Giannakopoulos, Andrew Cobb, and Ryan Culpepper. Contextual equivalence for a probabilistic language with continuous random variables and recursion. Workshop on Probabilistic Programming Semantics (PPS 2018), 2018. URL: https://pps2018.soic.indiana.edu/files/2018/01/paper.pdf.

33 Jeff Wu. Reduced traces and JITing in Church. Master's thesis, Mass. Inst. of Tech., 2013.